

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI**

**DOĞRUSAL VE DOĞRUSAL OLMAYAN SİSTEMLERİN
YAPAY ARI KOLONİSİ ALGORİTMASI KULLANILARAK
MODELLENMESİ**

**Hazırlayan
İbrahim BARAN**

**Danışman
Prof. Dr. Şaban ÖZER**

Yüksek Lisans Tezi

**Ağustos 2014
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI**

**DOĞRUSAL VE DOĞRUSAL OLMAYAN SİSTEMLERİN
YAPAY ARI KOLONİSİ ALGORİTMASI KULLANILARAK
MODELLENMESİ
(Yüksek Lisans Tezi)**

**Hazırlayan
İbrahim BARAN**

**Danışman
Prof. Dr. Şaban ÖZER**

**Ağustos 2014
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.



Ibrahim BARAN

YÖNERGEYE UYGUNLUK

Doğrusal ve Doğrusal Olmayan Sistemlerin Yapay Arı Kolonisi Algoritması Kullanılarak Modellenmesi adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.



Tezi Hazırlayan
İbrahim BARAN



Danışman
Prof. Dr. Şaban ÖZER



Elektrik-Elektronik Mühendisliği ABD Başkanı
Prof. Dr. Necmi TAŞPINAR

Prof. Dr. Şaban ÖZER danışmanlığında İbrahim BARAN tarafından hazırlanan “Doğrusal ve Doğrusal Olmayan Sistemlerin Yapay Arı Kolonisi Algoritması Kullanılarak Modellenmesi” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

22 /09/ 2014

JÜRİ:

Danışman : Prof. Dr. Şaban ÖZER
Üye : Prof. Dr. Derviş KARABOĞA
Üye : Doç. Dr. Aytekin BAĞIŞ

Şaban ÖZER
Derviş Karaboğa
Doç. Dr. Aytekin Bağış

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 23/09/2014 tarih ve 2014/64-02 sayılı kararı ile onaylanmıştır.

Kâzım Keşlioğlu
23. / 09. / 2014

Prof. Dr. Kâzım KEŞLİOĞLU
Enstitü Müdürü

ÖNSÖZ / TEŞEKKÜR

Çalışmalarım bilimsel katkılarıyla beni aydınlatan, yakın ilgi ve yardımlarını esirgemeyen sayın hocam Prof. Dr. Şaban ÖZER'e teşekkürü bir borç bilirim.

Çalışmalarım sırasında karşılaştığım zorlukları aşmamda yardımlarından dolayı Yrd. Doç. Dr. Hasan ZORLU, Arş. Gör. Hüseyin ACAR, Öğr. Gör. Mehmet YILDIRIM, gidiş gelişlerim sırasında izinler konusunda her zaman yardımcı olan çok değerli Çüngüş Mehmet Adıgüzel Meslek Yüksek Okulu Müdür Vekili Öğr. Gör. Reşat ÇELİKEL'e teşekkür ederim.

Ayrıca; çalışmalarım süresince sabır göstererek beni daima destekleyen aileme en içten teşekkürlerimi sunarım.

İbrahim BARAN
Kayseri, Ağustos 2014

DOĞRUSAL VE DOĞRUSAL OLMAYAN SİSTEMLERİN YAPAY ARI KOLONİSİ ALGORİTMASI KULLANILARAK MODELLENMESİ

İbrahim BARAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Ağustos 2014

Danışman: Prof. Dr. Şaban ÖZER

ÖZET

Bu tez çalışmasında doğrusal parametrik, doğrusal olmayan parametrik ve doğrusal olmayan gerçek sistemlerin modellenmesi amacıyla yapay arı kolonisi (YAK) algoritması, genetik algoritma (GA) ve klonal seçme algoritması (KSA) kullanılmıştır. Bu amaçla doğrusal ve doğrusal olmayan parametrik sistemler ile doğrusal olmayan gerçek sistemlerin parametre tahminleri, önerilen modeller doğrultusunda yapılmış, sonuçlar birbiri ile karşılaştırılmış ve algoritmaların performansları incelenmiştir.

Giriş bölümünde tezin konusu, amacı işlenmiş, birinci bölümde ise modelleme ile ilgili genel bilgiler ve literatür çalışmasına yer verilmiştir. İkinci bölümde çalışmada kullanılan yöntem ve doğrusal olan sistemler ile doğrusal olmayan sistemler için modelleme kavramı üzerinde durulmuş, sistemlerin modellendirilmesinde kullanılan modelleme yapıları incelenmiş, YAK algoritması, GA ve KSA hakkında bilgiler verilmiştir. Üçüncü bölümde, örnek uygulamalara yer verilmiş, sonuçlar tablolar ve şekiller halinde sunulmuştur. Son bölümde de elde edilen sonuçlar değerlendirilip yapılacak olan çalışmalar için önerilerde bulunulmuştur.

Anahtar Kelimeler: Doğrusal parametrik sistemler; doğrusal olmayan parametrik sistemler; doğrusal olmayan gerçek sistemler; Yapay Arı Kolonisi (YAK) Algoritması; Genetik Algoritma (GA); Klonal Seçme Algoritması (KSA).

MODELLING OF LINEAR AND NON-LINEAR SYSTEMS BY USING ARTIFICIAL BEE COLONY ALGORITHM

İbrahim BARAN

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, August 2014

Supervisor: Prof. Dr. Şaban ÖZER

ABSTRACT

In this thesis, linear parametric systems, non-linear parametric systems and non-linear real systems were aimed to model by using Artificial Bee Colony (ABC) Algorithm, Genetic Algorithm (GA) and Clonal Selection Algorithm (CSA). For this reason, linear and non-linear parametric systems as well as non-linear real system parameter estimation were performed based on proposed models and the results were compared with one another.

The purpose of the thesis was introduced in the introduction while general information about modelling and related literature review was presented in the first chapter. The second chapter focused on the method used in the study and the construct of modelling for the linear and non-linear parametric systems. It also examined modelling structures used in the modelling systems as well as providing information about ABC algorithm, GA and CSA. In the third chapter, the results were presented in the tables and figures. Also, some applied examples was provided in this chapter. In the last chapter, the obtained results have been evaluated and recommendations have been provided for the further studies.

Keywords: Linear parametric systems; non-linear parametric systems; non-linear real systems; Artificial Bee Colony (ABC) Algorithm; Genetic Algorithm (GA); Clonal Selection Algorithm (CSA).

İÇİNDEKİLER

DOĞRUSAL VE DOĞRUSAL OLMAYAN SİSTEMLERİN YAPAY ARI KOLONİSİ ALGORİTMASI KULLANILARAK MODELLENMESİ

	<u>Sayfa</u>
BİLİMSEL ETİĞE UYGUNLUK SAYFASI.....	ii
YÖNERGEYE UYGUNLUK SAYFASI.....	iii
KABUL VE ONAY SAYFASI.....	iv
ÖNSÖZ.....	v
ÖZET.....	vi
ABSTRACT.....	vii
İÇİNDEKİLER.....	viii
KISALTMALAR VE SİMGELER.....	xi
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ.....	xiv

_Toc398674036

GİRİŞ	1
--------------------	----------

1. BÖLÜM

GENEL BİLGİLER VE LİTERATÜR ÇALIŞMASI

1.1. Modelleme Kavramı	4
1.1.1. Sistem Modelleme Adımları.....	5
1.2. Literatür Çalışması.....	7

2. BÖLÜM

YÖNTEM

2.1. Yöntem	10
2.2. Modelleme Çeşitleri	10
2.2.1. Doğrusal Sistemlerin Modellenmesi	10
2.2.1.1. Autoregressive (AR) Modelleme.....	10
2.2.1.2. Moving Average (MA) Modelleme	11
2.2.1.3. Autoregressive Moving Average (ARMA) Modelleme.....	11
2.2.2. Doğrusal Olmayan Sistemlerin Modellenmesi	12
2.2.2.1. Volterra Modelleme	12

2.2.2.2. Bilinear Modelleme	13
2.2.2.3. PAR (Polynomial Autoregressive) Modelleme	13
2.2.3. Adaptif Sistem Modelleme	14
2.2.4. Doğrusal Olan ve Olmayan Sistemlerin Modellenmesinde Kullanılan Adaptif Algoritmalar	15
2.2.4.1. En Küçük Ortalama Kareler (Least Mean Squares) Algoritması	16
2.2.4.2. İrsel En Küçük Kareler (Recursive Least Square) Algoritması.....	17
2.2.5. Myriad Filtre Yapısı	19
2.2.5.1. Ağırlıklandırılmış Myriad Filtre Yapısı	19
2.2.5.2. Adaptif Ağırlıklandırılmış Myriad Filtre Algoritmaları.....	20
2.3. Sistem Modellemede Yapay Arı Kolonisi, Genetik ve Klonal Seçme Algoritmalarının Kullanımı.....	22
2.3.1. Yapay Arı Kolonisi (YAK) Algoritması.....	22
2.3.1.1. Gerçek Arı Davranışları.....	22
2.3.1.2. Algoritma Adımları.....	24
2.3.1.3. Yapay Arı Kolonisi (YAK) Algoritmasının Özellikleri	30
2.3.2. Genetik Algoritma (GA)	31
2.3.2.1. GA Yapısı.....	31
2.3.2.2. GA İşlem Adımları.....	33
2.3.2.3. GA'da Kullanılan Operatörler	33
2.3.2.4. GA'ların Özellikleri	36
2.3.3. Klonal Seçme Algoritması (KSA)	37
2.3.3.1. KSA'daki Temel İşlemler	39
2.3.3.2. KSA'nın Kontrol Parametreleri	40
3. BÖLÜM	
BULGULAR	
3.1. Sistem Modelleme Uygulamaları.....	43
3.2. Bulgular.....	43
3.2.1. Uygulamalarda Kullanılan Bilgisayar Özellikleri	43
3.2.2. Doğrusal Parametrik Sistem Modellemesi Uygulamaları	43
3.2.2.1. Örnek Sistem 1	43
3.2.2.2. Örnek Sistem 2	47
3.2.2.3. Örnek Sistem 3	52

3.2.2.4. Örnek Sistem 4	56
3.2.3. Doğrusal Olmayan Parametrik Sistem Modellemesi Uygulamaları	60
3.2.3.1. Örnek Sistem 1	60
3.2.3.2. Örnek Sistem 2	65
3.2.3.3. Örnek Sistem 3	69
3.2.3.4. Örnek Sistem 4	75
3.2.4. Doğrusal Olmayan Gerçek Sistem Modellenmesi Uygulamaları	80
3.2.4.1. Örnek Sistem 1	80
3.2.4.2. Örnek Sistem 2	87
3.2.4.3. Örnek Sistem 3	95

4. BÖLÜM TARTIŞMA SONUÇ VE ÖNERİLER

4.1. Tartışma	103
4.2. Sonuç ve Öneriler	103
KAYNAKLAR	105
ÖZGEÇMİŞ	112

KISALTMALAR VE SİMGELER LİSTESİ

AR	Autoregressive
MA	Moving Average
ARMA	Autoregressive Moving Average
LS	Least Square
RLS	Recursive Least Square
LMS	Least Mean Square
GA	Genetic Algorithm
CSA	Clonal Selection Algorithm
KSA	Klonal Seçme Algoritması
ABC	Artificial Bee Colony
YAK	Yapay Arı Kolonisi
PAR	Polynomial Autoregressive
MSE	Mean Squared Error
MAE	Mean Absolute Error
FGA	Farksal Gelişim Algoritması
DEA	Differential Evolution Algorithm

TABLOLAR LİSTESİ

Tablo 3. 1. Denklem 3.2’de önerilen modele ait kullanılan kontrol parametreleri	44
Tablo 3. 2. Denklem 3.2’de önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	44
Tablo 3. 3. Denklem 3.2’de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi	46
Tablo 3. 4. Denklem 3.4’te önerilen modele ait kullanılan kontrol parametreleri.....	48
Tablo 3. 5. Denklem 3.4’te önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	48
Tablo 3. 6. Denklem 3.4’te önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi	50
Tablo 3. 7. Denklem 3.6’te önerilen modele ait kullanılan kontrol parametreleri.....	52
Tablo 3. 8. Denklem 3.6’da önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	53
Tablo 3. 9. Denklem 3.6’da önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi	54
Tablo 3. 10. Denklem 3.8’de önerilen modele ait kullanılan kontrol parametreleri.....	56
Tablo 3. 11. Denklem 3.8’de önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri	57
Tablo 3. 12. Denklem 3.8’de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	58
Tablo 3. 13. Denklem 3.10’da önerilen modele ait kullanılan kontrol parametreleri.....	61
Tablo 3. 14. Denklem 3.10’da önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri	61
Tablo 3. 15. Denklem 3.10’da önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi	63
Tablo 3. 16. Denklem 3.11’de önerilen model için YAK Algoritmasının referans çalışma ile karşılaştırılması	64
Tablo 3. 17. Denklem 3.13’te önerilen modele ait kullanılan kontrol parametreleri.....	66
Tablo 3. 18. Denklem 3.13’te önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri	66
Tablo 3. 19. Denklem 3.13’te önerilen model için koloni ve popülasyon sayılarının	

ortalama süre ve hatalara etkisi.....	68
Tablo 3. 20. Denklem 3.15'te önerilen modele ait kullanılan kontrol parametreleri.....	70
Tablo 3. 21. Denklem 3.15'te önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	70
Tablo 3. 22. Denklem 3.15'te önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	72
Tablo 3. 23. Denklem 3.16 için YAK Algoritmasının referans çalışma ile karşılaştırılması.....	74
Tablo 3. 24. Denklem 3.18'de önerilen modele ait kullanılan kontrol parametreleri.....	75
Tablo 3. 25. Denklem 3.18'de önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	76
Tablo 3. 26. Denklem 3.18'de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	77
Tablo 3. 27. Denklem 3.18 için YAK Algoritmasının referans çalışmalar ile karşılaştırılması.....	79
Tablo 3. 28. Denklem 3.19'da önerilen modele ait kullanılan kontrol parametreleri.....	80
Tablo 3. 29. Denklem 3.19'da önerilen modele ait algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	81
Tablo 3. 30. Denklem 3.19'da önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	81
Tablo 3. 31. Denklem 3.19'da önerilen modele ait ikinci yöntemde algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	84
Tablo 3. 32. Denklem 3.19'da önerilen model için ikinci yöntemde koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	85
Tablo 3. 33. Denklem 3.20 önerilen modele ait kullanılan kontrol parametreleri.....	88
Tablo 3. 34. Denklem 3.20'de önerilen model için algoritmaların bulunduğu Parametreler, MSE değerleri ve parametre bulma süreleri.....	88
Tablo 3. 35. Denklem 3.20'de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	89
Tablo 3. 36. Denklem 3.20'de önerilen modele ait ikinci yöntemde algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	91
Tablo 3. 37. Denklem 3.20'de önerilen model için ikinci yöntemde koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi.....	92

Tablo 3. 38. Denklem 3.21’de önerilen modele ait kullanılan kontrol parametreleri.....	95
Tablo 3. 39. Denklem 3.21’de önerilen model için algoritmaların bulunduğu Parametreler, MSE değerleri ve parametre bulma süreleri.....	96
Tablo 3. 40. Denklem 3.21’de önerilen model için koloni ve popülasyon ortalama süre ve hatalara etkisi.....	96
Tablo 3. 41. Denklem 3.21’de önerilen modele ait ikinci yöntemde algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri.....	99
Tablo 3. 42. Denklem 3.21’de önerilen model için ikinci yöntemde koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi	100

ŞEKİLLER LİSTESİ

Şekil 1. 1. Bir sistem şeması	4
Şekil 1. 2. Sistem Modelleme Yapısı	5
Şekil 1. 3. Modelleme işleminin akış diyagramı.....	6
Şekil 2. 1. Adaptif Sistem Modelleme.....	15
Şekil 2. 2. Adaptif İşaret Kestirimi	15
Şekil 2. 3. Bal arılarının yem arama davranışları.....	23
Şekil 2. 4. YAK Algoritmasının akış diyagramı	30
Şekil 2. 5. Basit bir GA akış diyagramı	32
Şekil 2. 6. Rulet tekerleği seçme operatörü.....	34
Şekil 2. 7. Tek noktalı çaprazlama	35
Şekil 2. 8. Kromozomdaki mutasyon.....	36
Şekil 2. 9. KSA'nın temel blok şeması	38
Şekil 3. 1. Denklem 3.1'deki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa.....	45
Şekil 3. 2. Denklem 3.1'deki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa.....	47
Şekil 3. 3. Denklem 3.4'deki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa.....	49
Şekil 3. 4. Denklem 3.4'teki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa.....	51
Şekil 3. 5. Denklem 3.6'daki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa.....	54
Şekil 3. 6. Denklem 3.6'daki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa.....	55
Şekil 3. 7. Denklem 3.8'deki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa.....	58
Şekil 3. 8. Denklem 3.8'deki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa.....	59
Şekil 3. 9. Denklem 3.10'daki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa.....	62
Şekil 3. 10. Denklem 3.10'daki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa.....	64

Şekil 3. 11. Denklem 3.9’da verilen sistem çıkışı ile yak algoritmasının hesapladığı Denklem 3.11’de önerilen model çıkışı	65
Şekil 3. 12. Denklem 3.13’teki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa	67
Şekil 3. 13. Denklem 3.13’teki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa	69
Şekil 3. 14. Denklem 3.15’teki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa	71
Şekil 3. 15. Denklem 3.15’teki model için hata yaklaşım eğrileri: (a)YAK algoritması, (b) GA, (c) KSA.....	73
Şekil 3. 16. Denklem 3.15’te verilen sistem çıkışı ile yak algoritmasının hesapladığı Denklem 3.16’da önerilen model çıkışı	74
Şekil 3. 17. Denklem 3.18’deki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa	77
Şekil 3. 18. Denklem 3.18 için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa	78
Şekil 3. 19. Gerçek sistem çıkışı ile Denklem 3.19’da önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa	82
Şekil 3. 20. Gerçek sistem çıkışı ile Denklem 3.19’da önerilen model için test çıkışları: (a) yak algoritması, (b) ga , (c) ksa	83
Şekil 3. 21. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.19’da önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa	86
Şekil 3. 22. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.19’da önerilen model için test çıkışları: (a) yak algoritması, (b) ga , (c) ksa.....	87
Şekil 3. 23. Gerçek sistem çıkışı ile Denklem 3.20’de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa	90
Şekil 3. 24. Gerçek sistem çıkışı ile Denklem 3.20’de önerilen model için test çıkışları: (a) yak algoritması, (b) ga , (c) ksa	91
Şekil 3. 25. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.20’de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa	93
Şekil 3. 26. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.20’de önerilen model için test çıkışları: (a) yak algoritması, (b) ga , (c) ksa.....	94
Şekil 3. 27. Gerçek sistem çıkışı ile Denklem 3.21’de önerilen modelin çıkışları:	

(a) yak algoritması, (b) ga , (c) ksa	97
Şekil 3. 28. Gerçek sistem çıkışı ile Denklem 3.21’de önerilen model için test çıkışları: (a) yak algoritması, (b) ga , (c) ksa	98
Şekil 3. 29. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.21’de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa	101
Şekil 3. 30. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.21’de önerilen model için test çıkışları: (a) yak algoritması, (b) ga , (c) ksa.....	102

GİRİŞ

Sistem tanım olarak, dış dünya ve kendi iç elemanları arasındaki ilişkilerin belirlenebildiği yapılardır. Sistemin basitleştirilmiş, genellikle matematiksel veya hesaplanabilir yapıdaki gösterimine model adı verilir. Model, herhangi bir olayı zaman ve mekan kısıtlaması olmaksızın incelemek için kurulan bir düzendir. Fiziksel olayların modelini kurmak ise gerçekte, bu olayları, bir takım matematiksel ifadelere dönüştürmek demektir. Sistem modelleme, deneysel yolla elde edilmiş verilerden faydalanarak sistemlerin modelinin elde edilmesidir [1,2]. Modellemede amaç bilinmeyen bir sistemin girişi ile çıkışı arasındaki ilişkiyi belirlemektir. Diğer bir deyişle sistemi karakterize eden transfer fonksiyonu parametrelerini bulmaktır. Bu sebeple kullanılan en etkili yöntem denklem hata formülasyonu olarak adlandırılan bir model yaklaşımıdır [3]. Modelleme karmaşık parametrelerin belirlenmesi için iyi tanımlamalara dayanmalıdır. Çünkü karmaşık olaylar ancak bu şekilde matematik ifadeler şeklinde getirilebilir. Bu ise iyi bir matematik bilgi ve tecrübeyi gerektirir. Elbette ki doğru seçilmiş ilişkilerle kurulan bir model, çözümünü daha hassas ve daha iyi sonuçlara götürecektir. Büyük oranda sistemler için doğrusal modeller zayıf kalmaktadır. Çünkü gerçek dünya karmaşık ve doğrusal olmayan yapıdadır. Bu tür sistemlerin modellendirilmesinde doğrusal modelleme yöntemleri yetersiz kalmakta ve doğrusal olmayan modelleme yöntemlerinin kullanılması gerekmektedir [4-15].

Modelleme Uygulamaları

Sistemlerin matematiksel modellerinin önemi bugün mühendislikteki tüm tasarımlarda hızla artmaktadır .

Modelleme işlemi, kağıt, demir, cam veya kimyasal birleşim üretimi gibi, endüstriyel ve kontrol alanlarında, tahmin, veri haberleşmesi, ses işaretleri işleme, radar, solar,

elektrokardiyogram analizi, kanal denkleştirme, yankı bastırımı ve adaptif gürültü bastırımı gibi, işaret işleme ve haberleşme alanlarında kullanılmakta olup biyoloji, ekoloji ve ekonomi gibi alanlarda da kullanılmaktadır [16].

Ayrıca aşağıdaki bir çok alanda modelleme işlemi kullanılmaktadır.

Ekoloji Çevre Bilimi: Ömür uzunluğu yönetimi ve kontrol, göl ve akarsulardaki bitkilerin kimyasal etkilerinin kontrolü, hava kirliliğinin önlenmesi, su dağıtımı ve kontrolü, orman büyüme yönetimi.

Medikal-Tıbbi: Görüntüleme ve kontrol için tıbbi araçlar, zeki organlar (suni kalp, suni böbrek).

Ev Araçları: Ev ısıtma, havalandırma ve ısıtma kontrol, elbise temizlemede elektronik kontrol, nem kontrolediciler, fırın sıcaklık kontrol.

Güç/Enerji: Güç sistemleri kontrol ve planlaması; petrol geri kazanma, yel değirmeni optimal kontrolü, denizaltı yer yüzü izleme optimal kontrolü, optimal güç dağıtımı ve güç üretim kontrol.

Ulaştırma-Taşımacılık: Sensör (elektrik algılayıcı) kullanılarak otoyolda araç trafik akışı kontrolü, otomobillerde otomatik hız kontrolü, raylı ulaşım sistemlerinde sevk kontrol; asansör ve yürüyen merdiven inşası.

İmalat: Kesme, delme, döküm, kaynak, paketlenme, montajda donanımlı robotların kullanımı, kimyasal süreç kontrolü, tekstil imalatında gerginlik kontrolü, sıcak çelik taşıyıcılarda optik hız kontrolü.

Uzay ve Askeri Araştırmalar: Füze yönetim ve kontrol; otomatik pilot; uzay aracı kontrol; izleme sistemleri; nükleer denizaltıların seyri ve kontrolü; ateş kontrol sistemleri

Tezin Amacı

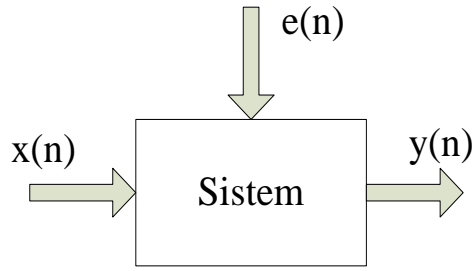
Bu tez çalışmasında doğrusal parametrik, doğrusal olmayan parametrik sistemler ile doğrusal olmayan gerçek sistemlerin modellenebilmesi için farklı matematiksel model yapıları ortaya konularak bu yapıların parametre değerleri elde edilmeye çalışılmıştır. Bunun için 4 adet doğrusal parametrik sistem, 4 adet doğrusal olmayan parametrik sistem ve 3 adet doğrusal olmayan gerçek sistem verileri kullanılmıştır. Parametrelerin belirlenmesi amacıyla Yapay Arı Kolonisi (YAK) Algoritması, Genetik Algoritma (GA) ve Klonal Seçme Algoritması (KSA) kullanılarak elde edilen sonuçlar birbiri ile karşılaştırılmış ve sürü zeka yaklaşımına iyi bir örnek olan literatüre yeni girmiş sayılabilecek YAK Algoritmasının performansı diğer algoritmalarla kıyaslanmıştır.

BÖLÜM 1

GENEL BİLGİLER VE LİTERATÜR ÇALIŞMASI

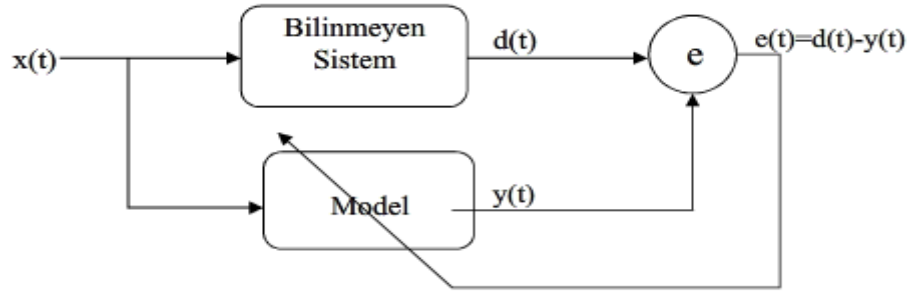
1.1. Modelleme Kavramı

Şekil 1.1’de de gösterildiği gibi sistem, $x(n)$ giriş işareti ve $e(n)$ bozucu işaretinden etkilenecek şekilde $y(n)$ çıkış işaretini üretir [17]. Sistemlerin uygulanabilir hale getirilmesi “modern kontrol teorisinin K.J. Aström ve I. Bohlin tarafından yayımlanan makaleleriyle 1960’lara rastlar [18].



Şekil 1. 1. Bir sistem şeması

Herhangi bir sistemin modelinin elde edilmesi işlemi $x(n)$ ve $e(n)$ işaretlerinden faydalanılarak, model çıkışı, $y(n)$ işaretinin elde edilmesi şeklinde tarif edilebilir. Genel sistem modelleme yapısı Şekil 1.2’de verilmiştir. Sistemlerin modelleri yalnızca sistemin giriş işareti kullanılarak, yalnızca sistemin çıkış işareti kullanılarak veya hem giriş hem de çıkış işaretleri kullanılarak elde edilebilir. Birçok endüstriyel alanda, örneğin kağıt, demir veya cam üretiminde, sistemin güvenli, verimli ve istenilen kalitede üretim yapabilmesi için kontrol düzeneğinin kurulması gerekir. Kontrol edilecek sistemin matematiksel modelinin bilinmesi zorunludur. Böylece sistemin modeli kullanılarak sistemin çıkışı, arzu edilen şekilde düzenlenebilir [19].



Şekil 1. 2. Sistem Modelleme Yapısı

1.1.1. Sistem Modelleme Adımları

Modelleme işleminin temel basamakları şu şekildedir.

- Deneysel planlama :

Sistemin girişine herhangi bir işaret (darbe, basamak, sinüs veya rastgele işaretler) uygulanıp, sistemin bu işarete cevabı çıkış işareti olarak kaydedilir.

- Model yapısının seçimi :

Sisteme uygun bir model yapısı tespit edilir. Bu model yapısı doğrusal olabileceği gibi, çoğu fiziksel sistemin davranışını gösteren doğrusal olmayan model yapısı da kullanılabilir.

- Parametre kestirimi :

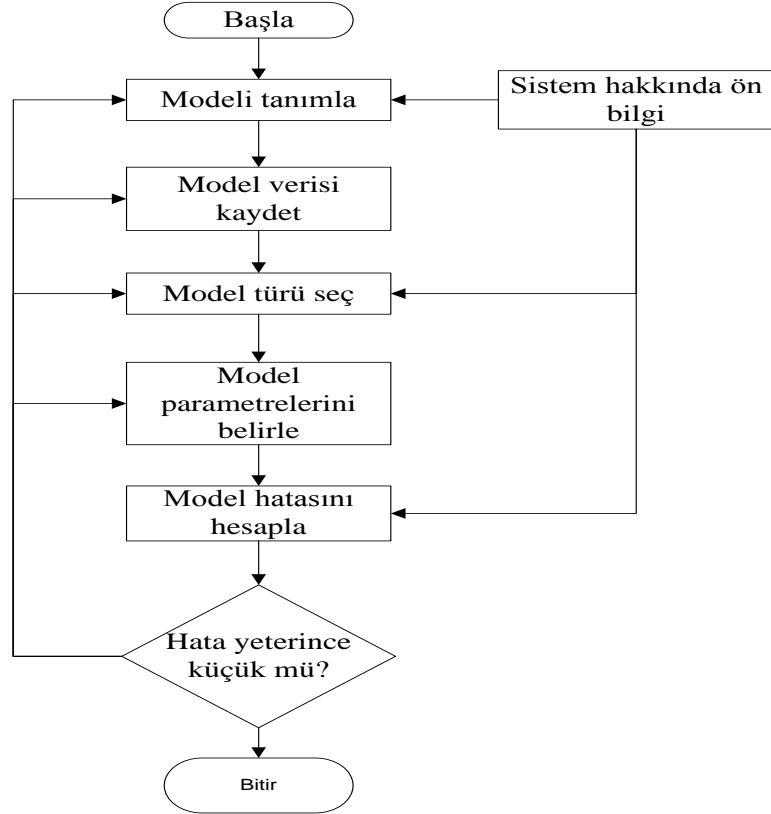
Elde edilen modelin parametreleri, bazı istatistiki veya tahmini yöntemlerle belirlenir. Modelleme işleminin en önemli aşaması, bu parametrelerin doğru şekilde belirlenmesidir.

- Kurulan modelin geçerliliğinin test edilmesi :

Parametreleri belirlenen modelin girişine, sisteme uygulanmış olan giriş işareti uygulanıp, modelden alınan çıkış işareti ile sistemin gerçek çıkışı arasındaki fark

bulunur. Eğer fark büyükse, başka bir model yapısı veya yeni bir parametre tespit yöntemi belirlenmesi için model yapısı seçimine geri dönülür. Eğer fark çok küçükse bu model sistemi tanımlamak ve kontrol etmek için kullanılabilir.

Sistem modelleme işlemi, sistemi uyaracak uygun giriş sinyalinin seçimi ile başlar. Sisteme uygun giriş sinyali ile bir model belirlenir ve belirlenen bir yöntemle parametreler kestirilir. Bu yolla elde edilen model geçerlilik testlerini geçemezse algoritmada yukarıdan aşağıya doğru hareket edilir. Önce parametre kestirim yöntemi değiştirilir. Eğer model çeşitli denemelere rağmen yine geçersizse model yapısı değiştirilir ve algoritmada aşağıya doğru hareket edilir. Model yapısını da değiştirmenin bir sonuca varmadığı durumlarda sistemin deneysel planlamasına geri dönülür. Sistemin doğrusal bir modelle temsil edilemediği veya tanılamının (hata ölçülerine göre) yeteri kadar başarılı bulunmadığı sonucuna varılırsa akıllı sistemlerin de içinde bulunduğu doğrusal olmayan modelleme yapılarına başvurulur [20]. Bu yöntemin akış diyagramı Şekil 1.3’de gösterilmiştir [21].



Şekil 1. 3. Modelleme işleminin akış diyagramı

1.2. Literatür Çalışması

Sistemlerin modellendirilmesi için doğrusal ve doğrusal olmayan modelleme yöntemleri kullanılmaktadır. Sistemin giriş-çıkış ilişkisinin doğrusal eşitliklerle ifade edildiği doğrusal modelleme, teorik altyapısının gelişmiş olmasından dolayı yaygın bir şekilde kullanılmaktadır. Gauss'un 1809'da başladığı bir süreç içinde devam eden araştırmalar sonucunda sistem modelleme kavramı ilk kez 1962'de Lotfi Zadeh tarafından kullanılmıştır. Zadeh bu kavramı "Bir sistemi giriş ve çıkış ilişkilerinden belirlemek" şeklinde tanımlamıştır [22-23]. Doğrusal sistemlerin modellendirilmesi için autoregressive (AR), moving average (MA) ve autoregressive moving average (ARMA) modeller kullanılmış ve üzerlerinde sıkça çalışılmıştır [3-24].

Halbuki gerçek hayatta karşılaşılan birçok sistem doğrusal olmayan davranışlara sahiptir. Bu tür sistemlerin modellendirilmesinde doğrusal modelleme yöntemleri yetersiz kalmakta ve doğrusal olmayan modelleme yöntemlerinin kullanılması gerekmektedir. Doğrusal olmayan modellemede, sistemin giriş-çıkış ilişkisi, diferansiyel denklemler, üstel ve logaritmik fonksiyonlar gibi doğrusal olmayan matematiksel ifadelerle sağlanır. AR, MA ve ARMA yöntemlerinin doğrusal olmayan sistemlerin modellendirilmesinde genel bir çözüm oluşturmadığı anlaşılmış ve doğrusal olmayan Volterra, bilinear ve polynomial autoregressive (PAR) modeller geliştirilmiştir [3-4,25].

Lennart Ljung, Tokunbu Gonfunmi ve Monson H.Hayes'e ait kaynaklar sistemlerin parametrik ve parametrik olmayan modellerle tanımlanması için gerekli prosedürleri oldukça ayrıntılı biçimde ele almaktadır [7-8,26-27].

Doğrusal olmayan sistemlerin modellendirilmesi için global ve lokal olmak üzere iki tür teknik kullanılmakta ve bunlar arasında gelişim stratejileri, GA, CSA, Gauss-Newton, tabu araştırmaları, Levenberg-Marquardt, Hooke-Jeeves, Volterra-Wiener Model gibi yöntem ve yaklaşımlarda bulunmaktadır [13-15,28-30].

Günümüzde karmaşık optimizasyon problemlerinin modellenmesi ve çözülmesinde doğal benzetimlerin kullanımına yönelik bir eğilim vardır. Bunun sebebi klasik optimizasyon algoritmalarının büyük boyutlu kombinatoryel, tamsayılı ve doğrusal olmayan matematiksel programların çözümünde yeterli olmamasıdır. Klasik

optimizasyon algoritmalarının çözüm stratejileri genellikle amaç fonksiyonu ve parametre türüne (doğrusal, doğrusal olmayan vb.) ve modellemede kullanılan değişkenlerin türüne (tamsayı, gerçek sayı vb.) bağlıdır. Bahsedilen algoritmaların verimliliği ise çözüm alanının boyutuna, modellemede kullanılan değişken ve parametre sayısına ve çözüm alanının yapısına bağlıdır. Diğer taraftan klasik optimizasyon algoritmaları farklı tipte değişkenler, amaç fonksiyonu ve parametreler içeren problem formülasyonlarına uygulanabilecek genel bir çözüm stratejisi sunmamaktadır.

Ancak birçok optimizasyon problemi aynı formülasyon içinde farklı tiplerde değişkenler, amaç fonksiyonları ve parametreler içermektedir. Dolayısıyla klasik optimizasyon teknikleri bu tür problemlerin çözümü için yeterli olmamaktadır. Araştırmacılar çeşitli optimizasyon problemlerini klasik optimizasyon yöntemlerine uyarlamak için oldukça çaba göstermişlerdir. Ancak gerçek hayattaki problemi belli bir çözüm yöntemine uyacak şekilde modellemek genellikle pek kolay değildir. İşte klasik optimizasyon tekniklerinin bu yetersizliklerini aşabilmek için problemden ve modelden bağımsız olan, doğadan esinlenmiş sezgisel optimizasyon algoritmaları önerilmektedir. Bu teknikler hem etkin hem de daha esnek olup belirli problem gereksinimlerine göre uyarlanabilmektedir [31].

Son yıllarda yapay zeka optimizasyon teknikleri, karmaşık optimizasyon problemlerinin çözümünde kullanılmakta ve olumlu sonuçlar verdiği görülmektedir. Uygulamalar klasik optimizasyon tekniklerinin kullanılmasının yetersiz olduğu durumlarda yeni arayışları beraberinde getirmiş ve araştırmacılar bu yönde çalışmalarını sürdürmüşlerdir. yapay zeka optimizasyon teknikleri ise bu çalışmaların bir neticesi olarak ortaya atılmıştır.

Genel olarak klasik optimizasyon tekniklerinden farklı olarak geliştirilen yapay zeka optimizasyon algoritmaları, uygulamalarda olumlu sonuçlar vermesine karşın, dezavantajları olduğu da görülmektedir. Araştırmacılar bu dezavantajları giderecek yeni çalışmalar yapmaktadırlar. Yapılan bu yeni araştırmalar yine yapay zeka tekniklerinde olduğu gibi modellemeler üzerinde geliştirilmiştir.

Doğadan esinlenen algoritmaların yeni bir dalı olan sürü zekâsı yaklaşımı, böceklerin içgüdüsel problem çözme becerilerini kullanan etkili metasezgisel yöntemler geliştirebilmek için böcek davranışlarının modellenmesine odaklanmıştır [32-36]. Karaboğa tarafından geliştirilen YAK algoritması da sürü zeka yaklaşımına bir örnektir [37-44]. Böcekler arasındaki etkileşimin bir sonucu olan kolektif zekânın en önemli parçalarından biri ise bireysel böcekler arasındaki bilgi paylaşımıdır. Bu tür etkileşimli davranışa örnek olarak bal arılarının buldukları yiyecek kaynağının kalitesi hakkındaki bilgiyi paylaştıkları, salınım dansı verilebilir. Bu dans aracılığıyla kaliteli bir yiyecek kaynağı bulan arılar, yiyecek kaynağı hakkındaki yön, uzaklık ve nektar miktarı bilgilerini diğer arılarla paylaşır. Bu başarılı mekanizma sayesinde koloni, kaliteli yiyecek kaynaklarının olduğu bölgelere yönlendirilebilmektedir [44].

BÖLÜM 2

YÖNTEM

2.1. Yöntem

Sistem modellemede amaç bilinmeyen bir sistemin girişi ile çıkışı arasındaki ilişkiyi belirlemektir. Diğer bir deyişle sistemi karakterize eden transfer fonksiyonu parametrelerini bulmaktır. Bunun için ise farklı çözüm yöntemleri geliştirilmiş ve bu yöntemler sıkça kullanılarak, modelleme uygulama alanlarının artmasında etkili olmuş ve geliştirmiştir.

2.2. Modelleme Çeşitleri

2.2.1. Doğrusal Sistemlerin Modellenmesi

Doğrusal sistem modelleme kontrol ve kestirim alanlarında doğrusal sistem uygulamaları için genel bir çatı niteliğindedir. Doğrusal sistem modellemede AR, MA ve ARMA modelleri kullanılır.

2.2.1.1. Autoregressive (AR) Modelleme

Pratik uygulamalarda karşılaşılan birçok ayrık zamanlı sistemlerde, veri olarak yalnızca çıkış değerlerinin yardımıyla sistemin modellenmesi gerekir. Bu tür sistemler AR modelleme yöntemi ile modellenebilir [26-27,45]. Kompleks frekans düzleminde sadece kutuplara sahip olan AR modelinin matematiksel denklemi

$$y(n) = -a_1y(n-1) - a_2y(n-2) - \dots - a_p y(n-p) + e(n) = -\sum_{k=1}^p a_k y(n-k) + e(n) \quad (2.1)$$

şeklinindedir. Eşitlikte $y(n)$ çıkış dizisini, $e(n)$ bozucu etki, a_k ise AR parametrelerini belirtir. AR modellerde, sistem parametrelerinin ayarlanması, sistemden elde edilen

çıkış işaretinin önceki değerlerini bazı katsayılarla ağırlıklandırarak çıkışın istenilen özellikte olması işlemidir.

2.2.1.2. Moving Average (MA) Modelleme

Bir diğer doğrusal sistem modelleme metodu da MA modelleme yapısıdır. Bu yapı kullanılarak yapılan modellemede veri olarak yalnızca giriş işareti değerleri kullanılır [8]. Kompleks frekans düzleminde sadece sıfırlara sahip olan bu yapının matematiksel denklemi

$$y(n) = b_0u(n) + b_1u(n-1) + \dots + b_qu(n-q) + e(n) = \sum_{k=0}^q b_k u(n-k) + e(n) \quad (2.2)$$

şeklinindedir. Burada $y(n)$ çıkış dizisini, $u(n)$ giriş dizisini ve b_k ise MA parametrelerini belirtir. MA modellerde, sistem parametrelerinin ayarlanması, sisteme uygulanan giriş işaretinin o andaki ve önceki değerlerini bazı katsayılarla ağırlıklandırarak çıkışın istenilen özellikte olması işlemidir.

2.2.1.3. Autoregressive Moving Average (ARMA) Modelleme

AR model yapısı sadece kutuplara sahip olan sistemlerin matematiksel modelinin oluşturulmasında, MA model yapısı ise sadece sıfırlara sahip olan sistemlerin matematiksel modelini oluşturmada kullanılmaktadır. Çoğu fiziksel sistemin hem kutuplara hem de sıfırlara sahip olduğu gerçeği göz önüne alınırsa bu tekniklerin yeterli olamayacağı açıktır. Bu yüzden kutup ve sıfırlara sahip sistemlerin matematiksel modellenmesi için ARMA modelleme yöntemleri geliştirilmiştir [27]. Genel olarak bu modellerde, giriş dizisi $u(n)$ ile, çıkış dizisi ise $y(n)$ ile ifade edilir. Bu diziler arasında

$$y(n) = - \sum_{k=1}^p a_k y(n-k) + \sum_{k=0}^q b_k u(n-k) \quad (2.3)$$

şeklinde doğrusal fark denklemi yazılabilir. p , AR model derecesini, q ise MA model derecesini belirtir. Bir ARMA modelin transfer fonksiyonu ele alındığında bütün sıfırları ve kutupları z düzleminde birim dairenin içinde olduğu varsayılır. Eğer sistemin bütün kutupları ve sıfırları birim dairenin içindeyse bu tür sistemlere kararlı ve minimum fazlı sistemler adı verilir. Hepsi dairenin dışında ise kararsız ve maksimum

fazlı sistemler olarak adlandırılır. Doğrusal modeller (AR, MA, ARMA), ses, radyo, radar ve sismoloji gibi sürekli işaretlerin, ayrık zamanlı olarak gösterilimi için kullanılırlar [24].

2.2.2. Doğrusal Olmayan Sistemlerin Modellenmesi

Gerçek hayattaki birçok sistem, doğrusal olmayan yapıya sahiptir. Fakat, analiz ve sentez tekniklerinin büyük kısmı, doğrusal matematiksel model yaklaşımını kullanır [4]. Halbuki, birçok durumda bu yaklaşım iyi sonuç vermemektedir. Doğrusal olmayan bir sistem, eğer dinamik değişimi küçükse ancak o zaman doğrusal olarak modellenabilir. Aksi takdirde doğrusal model, sistemin dinamik yapısını tam olarak belirleyememektedir. Genelde sistem değişkenleri çok fazla ise, sistemin doğrusal modellenbilme ihtimali düşüktür. Sistemler, doğrusal tasarlanırsa bile, sistemin iç yapısındaki elemanların doğrusal olmayan yapıya sahip olması çok rastlanılan durumlardandır [5]. Bundan dolayı gerçek hayat problemlerini çözmek için doğrusal olmayan modellere ihtiyaç duyulur.

Doğrusal modellerdeki AR, MA ve ARMA yapılarının aksine doğrusal olmayan modellerde temel bir yapı bulunmamaktadır. Güç serisine dayalı doğrusal olmayan modelleme yaklaşımı, birçok sistemin doğrusal olmayan davranışını modellemekte kullanılır. Volterra, bilinear ve PAR modelleri bunlardan sayılabilir.

2.2.2.1. Volterra Modelleme

Doğrusal olmayan sistemlerin modellenmesinde en çok kullanılan yöntem, Volterra yapısıdır [6]. Adaptif Volterra filtresi ise giriş ve gürültüdeki istatistiki değişmelere göre sistemi ayarlama özelliğine sahip olmasına rağmen, sadece sistemin Volterra serileri tarafından tanımlanabildiği durumlarda uygulanabilmektedir [46]. Volterra serileri

$$y_n = \sum_{i=0}^N h_i u_{n-i} + \sum_{i=0}^N \sum_{j=0}^N q_{i,j} u_{n-i} u_{n-j} + \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N q_{i,j,k} u_{n-i} u_{n-j} u_{n-k} + \dots \quad (2.4)$$

şeklinindedir. Burada y_n ; çıkış, u_n ; giriş dizisini, h_i ; doğrusal kısma ait, $q_{i,j}, \dots$ ise doğrusal olmayan kısma ait (kuadratik) parametreleri göstermektedir. Sadece h_i 'nin göz önüne alındığı sistemlerde (birinci dereceden Volterra modeli), Volterra yapısı MA yapısı

haline dönüşür. Genelde ikinci dereceden Volterra yapısı (sadece h_i ve $q_{i,j}$ değerlerinin hesaplandığı yapı) sistem modellemede kullanılır [47-48]. Çünkü bundan büyük yapılar çok fazla işlem gerektirir. Birçok araştırmacı Volterra modellerin blok ve adaptif uygulamaları üzerine çalışmıştır. Volterra modelleri, kanal denkleştirme, yankı bastırımı ve adaptif gürültü bastırımı gibi gerçek hayattaki problemlerini çözmede kullanılmaktadır [9-11].

2.2.2.2. Bilinear Modelleme

Volterra modelinde çıkış işareti, sadece giriş işaretinin ağırlıklandırılması sonucu elde edilir. Bu durum çıkış işaretinin önceki değerlerine bağlı olan ve doğrusal olmayan sistemlerin modellenmesinde yetersiz kalmaktadır. Bu durumda bilinear modeller kullanılır [12]. Bilinear modelde çıkış işareti

$$y_n = \sum_{i=1}^N h_i y_{n-i} + \sum_{i=1}^N \sum_{j=0}^N b_{i,j} y_{n-i} u_{n-j} + \sum_{j=0}^N c_j u_{n-j} \quad (2.5)$$

şeklinde tanımlanır. Model parameteleri h , b ve c ile ifade edilmiştir. Bu modelde çıkış işareti ile giriş işaretinin çarpımı modele doğrusal olmama özelliği kazandırmaktadır.

2.2.2.3. PAR (Polynomial Autoregressive) Modelleme

Yüksek dinamik yapıya sahip olan sistemlerin modellenmesinde hem Volterra hem de bilinear model yetersiz kalır. Bu tip sistemlere örnek olarak kaotik davranış gösteren sistemler verilebilir. Böyle sistemlerin modellenmesinde (PAR) modeli önerilmiştir [13]. PAR modelinde çıkış işareti

$$y_n = \sum_{i=1}^N a_i y_{n-i} + \sum_{i=1}^N \sum_{j=1}^N b_{i,j} y_{n-i} y_{n-j} + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N c_{i,j,k} y_{n-i} y_{n-j} y_{n-k} \quad (2.6)$$

denklemiyle elde edilir. a , b ve c model parametreleridir. Döngüsel yapısından dolayı, modelin kararsız olma eğilimi vardır [14]. Bu durum modelin kaotik davranış göstermesine sebep olur [15]. Bu modelde sadece a_i 'nin göz önüne alındığı sistemlerde AR yapısı haline dönüşür.

2.2.3. Adaptif Sistem Modelleme

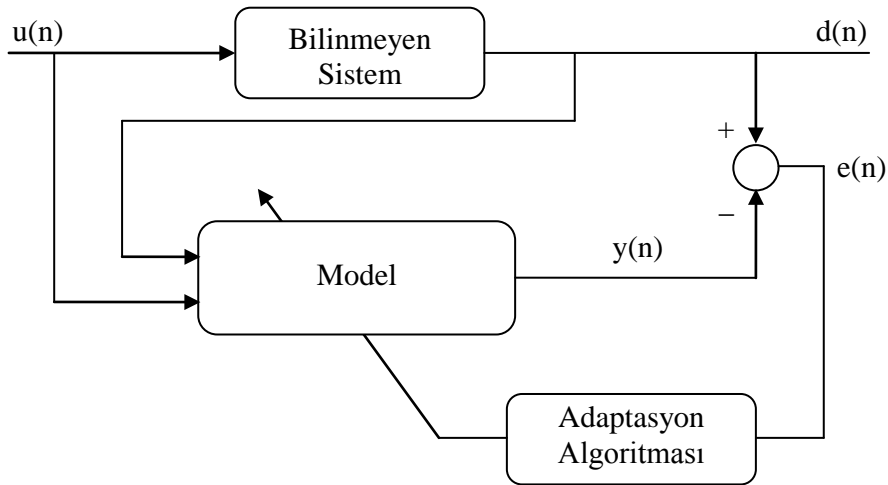
Adaptif sistemler çevre şartlarına göre kendisini en iyiye doğru yönlendiren sistemlerdir. Bu sistemler en çok kontrol ve işaret işleme sahasında uygulanmaktadır [49-53]. Adaptif sistemlerin en önemli özelliği zamanla değişen sistemlere rahatlıkla uygulanabilmesi ve yeni durumlara göre kendi kendini ayarlayabilmesidir. Doğrusal sistemlerde karşılaşılan, sadece belirli girişlere karşı çıkışın istenen şekilde olmasıdır. Diğer tür girişlerin uygulandığı durumlarda veya kontrol edilen sistemin zamanla, çevre şartlarından etkilenerek özelliklerinin değişmesi halinde ise sistemin kararsız davranması durumu adaptif sistemlerde daha az gözlenmektedir.

Adaptif sistemlerin ortak özellikleri aşağıda sıralanmıştır :

- Çevre şartlarının ve sistemden istenilen özelliklerin değişmesiyle, sistem kendisini otomatik olarak ayarlayabilir.
- Genellikle zamanla değişen sistemler olarak tanımlanır.
- Çoğu zaman karmaşık bir yapıya sahiptirler ve analiz edilmeleri adaptif olmayan sistemlere göre daha zordur. Fakat giriş işaretinin özellikleri bilinmediği veya zamanla değiştiği durumlarda yüksek performans elde edilmektedir.

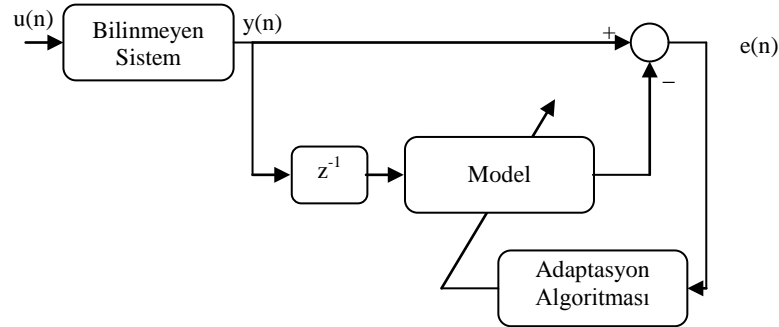
Adaptif sistemlerdeki temel yaklaşım, modellenecek sistem parametrelerinin elde edilmesi aşamasında, her bir iterasyon sonucunda oluşan hata değerinin minimize edilmesi için sistem parametrelerini belirli bir şekilde değiştirmektir. Hata değerinin minimuma indirilmesi için genellikle sistemin amaç fonksiyonunun (hatayı veren fonksiyonun) türevi kullanılır. Bu türev değerini her iterasyonda sıfır yapan parametre değerleri adaptif olarak bulunur. Sistemin çıkışının istenilen şekilde olması için gereken adaptif kontrol sistemi tasarımı Şekil 2.1’de gösterilmektedir.

Bu şekilde istenilen ile elde edilen çıkış arasındaki fark hata adını almakta ve adaptif sisteme giriş olarak verilmektedir. Bu farkın sıfır olması, sistemin arzu edilen şekilde çalışması anlamına gelir. Adaptif modelleme, sistem parametrelerini, hatayı sıfır yapacak şekilde ayarlamak için kullanılır. Şekil 2.1’de genel olarak bir adaptif sistem modelleme yapısı verilmiştir.



Şekil 2. 1. Adaptif Sistem Modelleme

y çıkış dizisinin n . elemanı olan $y(n)$ 'nin değerini, daha önceden elde edilen dizi değerlerini ($y(n-1)$, $y(n-2)$,...) kullanarak belirlenmesi yöntemine doğrusal tahmin (kestirim) yöntemi denir. Adaptif işaret kestiriminin blok şeması, Şekil 2.2'te gösterilmiştir.



Şekil 2. 2. Adaptif İşaret Kestirimi

2.2.4. Doğrusal Olan ve Olmayan Sistemlerin Modellenmesinde Kullanılan Adaptif Algoritmalar

Bu kısımda sistemlerin modellendirilmesi için kullanılan adaptif algoritmalar olarak adlandırılan LMS ve RLS algoritmaları ele alınıp doğrusal olmayan sistemlere uyarlanacaktır. Bu yöntemler, sistemin durumuna göre giriş ve çıkış işaretinin çarpımı ifadelerini doğrusalleştirme yaklaşımını kullanır.

2.2.4.1. En Küçük Ortalama Kareler (Least Mean Squares) Algoritması

Bu yöntemde parametreler her iterasyonda hatayı en aza indirecek şekilde değişmektedir [8]. Şekil 2.1’de verilen sistem modellendirme yapısı ele alındığında istenilen çıkış ile sistem çıkışı arasındaki fark

$$e(n) = d(n) - y(n) \quad (2.7)$$

şeklindedir. Burada $y(n)$ bilinmeyen sistem çıkışı, $d(n)$ ise arzu edilen çıkıştır. Sistem çıkışı $y(n)$ doğrusal MA model çıkışı olarak ele alınırsa sistem çıkışı Denklem (2.8)’deki gibi olur. Sistem çıkışının matris formunda gösterimi

$$y(n) = \begin{bmatrix} x(n) & x(n-1) & x(n-2) & \dots & x(n-N) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \mathbf{X}_1^T \mathbf{A} \quad (2.8)$$

şeklindedir. Hata fonksiyonunun karesi

$$[e(n)]^2 = [d(n) - \mathbf{X}(n)^T \mathbf{A}]^2 = [d(n)]^2 + \mathbf{A}^T \mathbf{X}(n) \mathbf{X}(n)^T \mathbf{A} - 2d(n) \mathbf{X}(n)^T \mathbf{A} \quad (2.9)$$

elde edilir. Hatanın ortalama değeri

$$E[e(n)]^2 = E[d(n)]^2 + \mathbf{A}^T E[\mathbf{X}(n) \mathbf{X}(n)^T] \mathbf{A} - 2E[d(n) \mathbf{X}(n)^T] \mathbf{A} \quad (2.10)$$

olur. Bu formülde $E[\mathbf{X}(n) \mathbf{X}(n)^T]$ değerine, çıkış özilişki matrisi adı verilir. $E[d(n) \mathbf{X}(n)]$ ise istenilen çıkış ile giriş arasındaki özilişkiyi belirtir ve bunlar

$$\mathbf{R} = E[\mathbf{X}_k \mathbf{X}_k^T] \quad \mathbf{P} = E[d_k \mathbf{X}_k^T] \quad (2.11)$$

ile gösterilir. Hatanın karesinin ortalamasını, ξ , (Mean Square Error, MSE) veren formül aşağıdaki gibi yazılabilir.

$$\text{MSE} = \xi = E[e_k^2] = E[d_k^2] + \mathbf{A}^T \mathbf{R} \mathbf{A} - 2\mathbf{P}^T \mathbf{A} \quad (2.12)$$

Hatayı en aza indirmek için MSE’nin \mathbf{A} ’ya göre türevini alıp sıfıra eşitlemek gereklidir.

$$\nabla = \frac{\partial \xi}{\partial \mathbf{A}} = 2\mathbf{R}\mathbf{A} - 2\mathbf{P} = 0 \quad (2.13)$$

$$\mathbf{A} = \mathbf{R}^{-1}\mathbf{P} \quad (2.14)$$

Böylece en uygun sistem parametreleri \mathbf{A} elde edilir. LMS algoritması için, Sistem adaptif modellendiği için k . iterasyonda ∇_k değeri, Denklem 2.3 kullanılarak

$$\begin{aligned} \nabla_k &= 2\mathbf{R}_k\mathbf{A}_k - 2\mathbf{P}_k \\ &= 2\mathbf{X}_k\mathbf{X}_k^T\mathbf{A}_k - 2d_k\mathbf{X}_k \\ &= -2\mathbf{X}_k(d_k - \mathbf{X}_k^T\theta_k) = -2\mathbf{e}_k\mathbf{X}_k \end{aligned} \quad (2.15)$$

şeklinde hesaplanabilir. Bu değer parametre vektörünün tahmininde belirli bir katsayı (adım büyüklüğü, μ) ile ağırlıklandırılarak yeni vektörün hesaplanmasına imkan verir.

$$\mathbf{A}_{k+1} = \mathbf{A}_k - \mu\nabla_k \quad (2.16)$$

$$\mathbf{A}_{k+1} = \mathbf{A}_k + 2\mu\mathbf{e}_k\mathbf{X}_k \quad (2.17)$$

$$\lambda = 2\mu \quad (2.18)$$

Denklem 2.17 ve Denklem 2.18 kullanılarak

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \lambda\mathbf{e}_k\mathbf{X}_k \quad (2.19)$$

elde edilir. (2.19) eşitliğindeki λ değerine unutma faktörü (forgetting factor) adı verilir. Eğer $\lambda < 1$ ise, $x(n)$ dizisinin n . elemana yakın değerlerinin ağırlığı daha öncekilere göre artar. Bu durumda yeni parametrelerin oluşmasında son çıkış değerlerinin etkili olduğu görülür. Böylece parametre vektörü, her iterasyon için bir önceki hata ve çıkış değerlerinden faydalanarak hatayı minimize edecek şekilde yeniden hesaplanır. Böylelikle adaptif LMS algoritması elde edilir.

2.2.4.2. İçsel En Küçük Kareler (Recursive Least Square) Algoritması

Bu yöntemde hata fonksiyonu LMS den farklı şekilde seçilmektedir.

$$\varepsilon(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (2.20)$$

Hata fonksiyonunu yine parametre değerine göre türevi alınırsa normal denklemi aşağıdaki şekilde elde edilir.

$$\Phi(n)\hat{w}(n) = \theta(n) \quad (2.21)$$

Burada, $\Phi(n)$ korelasyon matrisi ve $\theta(n)$ çapraz korelasyonu matrisidir.

$$\Phi(n) = \sum_{i=1}^n \lambda^n u(i)u^H(i) \quad (2.22)$$

$$\theta(n) = \sum_{i=1}^n \lambda^n u(i)d^*(i) \quad (2.23)$$

Denklem (2.21)'e dikkat edilecek olursa optimum parametre değerlerinin bulunması için korelasyon matrisinin tersine ihtiyaç duyulmaktadır. Bu matrisin tersini almak için, matrisin tersinin bulunması kuralı (Matris inversion lemma) kullanılır [53]. Sonuç olarak RLS algoritması

$n = 1, 2, 3, \dots$, için zamanın her anında hesaplanır.

$$k(n) = \frac{\lambda^{-1}P(n-1)u(n)}{1 + \lambda^{-1}u^H(n)P(n-1)u(n)} \quad (2.24)$$

$$\alpha(n) = d(n) - \hat{w}^H(n-1)u(n) \quad (2.25)$$

$$\hat{w}(n) = \hat{w}(n-1) + k(n)\alpha^*(n) \quad (2.26)$$

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)u^H(n)P(n-1) \quad (2.27)$$

şeklinde elde edilir ve başlangıç şartları

$$P(0) = \delta^{-1}I \quad \delta = \text{pozitif küçük bir sabit}$$

$$\hat{w}(0) = 0$$

şeklinindedir [54]. Bu yöntemde, adaptif kazanç, kovaryans matrisi P yardımıyla her iterasyon için ayarlanır. Çoğunlukla RLS yöntemi, LMS yöntemlerine göre daha hızlı yakınsar. Fakat başlangıç değerleri ve yuvarlatma hataları açısından LMS'den daha

hassastır [53]. LMS ve RLS algoritmasının doğrusal olmayan sistemler için kullanılabilmesi için sistem çıkışı Denklem (2.8)'dekine benzer şekilde doğrusallaştırılır.

2.2.5. Myriad Filtre Yapısı

Myriad filtre yapısı son yıllarda, dürtü gürültülü ortamlar (özellikle α -bağımlı gürültü) için güçlü bir doğrusal olmayan filtre yapısı olarak kullanılmaktadır [55]. Bu filtre yapısı haberleşme, işaret ve görüntü işleme alanlarına başarıyla uygulanmıştır [55-56]. α -bağımlı gürültü α 'nın $0 < \alpha \leq 2$ şeklindeki değişimine bağlıdır. Burada $\alpha=1$ durumunda oluşan gürültü Cauchy dağılımlı, $\alpha=2$ durumunda oluşan gürültü ise Gaussian dağılımlıdır [57]. α -bağımlı gürültü $\Phi(\omega) = e^{-\gamma|\omega|^\alpha}$ şeklinde bir karakteristik fonksiyona sahiptir. Burada γ dispersiyon parametresi ve α karakteristik değişkendir. Myriad filtre yapısı

$$\beta_K = \text{myriad}(K; x_1, x_2, \dots, x_N) = \arg \min \sum_{i=1}^N \log [K^2 + (x_i - \beta)^2] \quad (2.28)$$

şeklinde bir amaç (cost) fonksiyonuna sahiptir. Oluşacak olan myriad değeri, fonksiyonu minimum yapan β değeridir. Burada N , filtre uzunluğu, x_i giriş verilerinden alınan örnek veri değerleri, K ise doğrusallık faktörüdür. Burada K önemli bir faktördür, örneğin $\{3, 10, 1, -1, 6\}$ şeklindeki bir veri setinin $K=20, 2$ ve 0.2 değerleri için oluşacak myriad değerleri $\beta_{20} = 1.8, \beta_2 = 0.1$ ve $\beta_{0.2} = 1$ olmaktadır.

2.2.5.1. Ağırlıklandırılmış Myriad Filtre Yapısı

Ağırlıklandırılmış myriad filtre yapısı, bir önceki kısımda bahsedilen myriad filtre yapısında yer alan giriş örneklerinin pozitif ağırlıklandırılması ile daha genel hale getirilmiş halidir. Ağırlıklandırılmış myriad filtre yapısı

$$\beta_K = \text{myriad}(K; w_1 \circ x_1, w_2 \circ x_2, \dots, w_N \circ x_N) = \arg \min \sum_{i=1}^N \log [K^2 + w_i (x_i - \beta)^2] \quad (2.29)$$

şeklinde bir amaç fonksiyonuna sahiptir. Ağırlıklandırılmış myriad filtre yapısı daha çok adaptif yapıda kullanılır [55-58]. Yani ağırlık değerlerinin bulunabilmesi için adaptasyon algoritmalarına ihtiyaç duyar.

2.2.5.2. Adaptif Ağırlıklandırılmış Myriad Filtre Algoritmaları

Burada ağırlıklandırılmış myriad filtre (AMyF) yapısı için geliştirilmiş olan adaptif algoritmalarından bahsedilecektir. Filtre parametrelerinin optimizasyonu için daha önceki bölümlerde de bahsedildiği gibi sistem çıkışı ile istenilen çıkış arasındaki farkın minimize edilmesi gerekmektedir. Giriş örnekleri $x=[x_1, x_2, \dots, x_N]^T$, ağırlık vektörü $w=[w_1, w_2, \dots, w_N]^T$ ve doğrusallık faktörü K ile birlikte AMyF yapısının çıkışı $y=y_K(w,x)$ olarak gösterilmektedir. İstenilen çıkış ile filtre çıkışı arasındaki hata $e=y-d$ şeklindedir. Hata optimizasyonu için ortalama mutlak hata (Mean Absolute Error - MAE) ve MSE kriterleri kullanılabilir. MAE kriteri göz önüne alındığında $J_1(w,K)=E\{|e|\}=E\{|y_K(w,x)-d|\}$ ve MSE kriteri göz önüne alındığında $J_2(w,K)=E\{e^2\}=E\{|y_K(w,x)-d|^2\}$ şeklinde olur. Parametre optimizasyonu için bu hata fonksiyonlarının çıkışın ağırlığa göre türevi alınmalıdır. Bu durumda

$$\frac{dJ_1(w,K)}{dw_i} = \frac{d}{dw_i} E\{|y_K(w,x)-d|\} = E\left\{ \text{sgn}(y-d) \frac{dy}{dw_i} \right\} \quad (2.30)$$

ve

$$\frac{dJ_2(w,K)}{dw_i} = 2E\left\{ (y-d) \frac{dy}{dw_i} \right\} \quad (2.31)$$

şeklinde elde edilir.

Optimal filtre parametrelerinin bulunabilmesi için literatürde sıkça kullanılan en dik iniş (*steepest descent*) metodu [58]

$$w_i(n+1) = w_i(n) - \mu \frac{dJ}{dw_i}(n) \quad (2.32)$$

şeklindedir. Burada $dJ_1(w,K)/dw_i$ kullanılırsa

$$w_i(n+1) = w_i(n) - \mu \frac{dJ_1}{dw_i}(n) = P\{w_i(n) - \mu \text{sgn}[e(n)] \frac{dy}{dw_i}(n)\} \quad (2.33)$$

bulunur, burada

$$P[u] = \begin{cases} u, & u > 0 \\ 0, & u \leq 0 \end{cases} \quad (2.34)$$

şeklindedir. Denklem (2.33)'de görüldüğü gibi dy/dw_i (ağırlıklandırılmış myriad filtre çıkışının parametre değerine göre türevi) değerine ihtiyaç duyulmaktadır. Bu durumda

$$\frac{dy}{dw_i} = \frac{\left\{ \frac{-(y-x_i)}{\left[1 + \frac{w_i}{K^2}(y-x_i)^2\right]^2} \right\}}{K^2 \cdot \left\{ \sum_{j=1}^N \frac{w_j}{K^2} \cdot \frac{1 - \frac{w_j}{K^2}(y-x_j)^2}{\left[1 + \frac{w_j}{K^2}(y-x_j)^2\right]^2} \right\}} \quad (2.35)$$

şeklinde elde edilir. Böylelikle

$$w_i(n+1) = P \left(w_i(n) + \mu \operatorname{sgn}[e(n)] \frac{\left\{ \frac{(y-x_i)}{\left[1 + \frac{w_i}{K^2}(y-x_i)^2\right]^2} \right\}(n)}{K^2 \cdot \left\{ a + \sum_{j=1}^N \frac{w_j}{K^2} \cdot \frac{1 - \frac{w_j}{K^2}(y-x_j)^2}{\left[1 + \frac{w_j}{K^2}(y-x_j)^2\right]^2} \right\}(n)} \right) \quad (2.36)$$

olur. Burada a katsayısı kararlılık katsayısıdır. Algoritmanın payda kısmında bulunan fonksiyon çok küçük değerde olacağından, algoritma daha sade şekliyle

$$w_i(n+1) = P \left(w_i(n) + \mu \operatorname{sgn}[e(n)] \left\{ \frac{(y-x_i)}{\left[1 + \frac{w_i}{K^2}(y-x_i)^2\right]^2} \right\}(n) \right) \quad (2.37)$$

elde edilir [58].

2.3. Sistem Modellemede Yapay Arı Kolonisi, Genetik ve Klonal Seçme Algoritmalarının Kullanımı

2.3.1. Yapay Arı Kolonisi (YAK) Algoritması

Doğadan esinlenen algoritmaların yeni bir dalı olan sürü zekâsı yaklaşımı, böceklerin içgüdüsel problem çözme becerilerini kullanan etkili metasezgisel yöntemler geliştirebilmek için böcek davranışlarının modellenmesine odaklanmıştır [37-59].

Karaboğa tarafından geliştirilen YAK Algoritması (Artificial Bee Colony Algorithm, ABC) da sürü zeka yaklaşımına bir örnektir [37-44]. Böcekler arasındaki etkileşimin bir sonucu olan kollektif zekânın en önemli parçalarından biri ise bireysel böcekler arasındaki bilgi paylaşımıdır. Bu tür etkileşimli davranışa örnek olarak bal arılarının buldukları yiyecek kaynağının kalitesi hakkındaki bilgiyi paylaştıkları, salınım dansı verilebilir. Bu dans aracılığıyla kaliteli bir yiyecek kaynağı bulan arılar, yiyecek kaynağı hakkındaki yön, uzaklık ve nektar miktarı bilgilerini diğer arılarla paylaşır. Bu başarılı mekanizma sayesinde koloni, kaliteli yiyecek kaynaklarının olduğu bölgelere yönlendirilebilmektedir [44].

2.3.1.1. Gerçek Arı Davranışları

Bal arılarında amaç kovan etrafında bulunan nektar kaynaklarından toplanan balı kovana getirmektir. Bu işi yerine getirmek için arılar ikiye ayrılır çalışanlar ve çalışmayanlar.

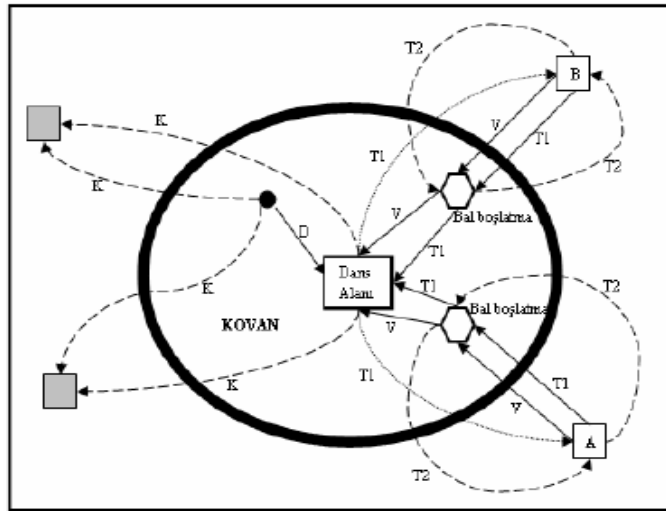
Çalışanlar: Bir nektar kaynağından kovana nektar getiren arıları,

Çalışmayanlar: Bir nektar kaynağını terk etmiş veya herhangi bir nektar kaynağına gitmemiş arıları ifade eder.

Yiyecek kaynakları, kovan etrafındaki nektar kaynaklarıdır. Bal arılarının yiyecek kaynakları; yuvaya yakınlık, zenginlik, enerji gibi birçok faktöre bağlıdır. Bir nektar kaynağının değerini nektar kaynağından getirilen nektarın harcanan enerjiye oranı belirler. Çalışan Toplayıcılar, belirli bir yiyecek kaynağından nektar getiren arılardır. Mevcut durumda nektar kaynağından faydalanmaya ve çalışmaya devam ederler. Bu

arılar kovana kaynağın değerini ve kovan-kaynak arasındaki uzaklık bilgisini taşırlar. Çalışmayan toplayıcılar, yararlanılacak yiyecek kaynağı ararlar. Kaşif ve seyirci olmak üzere iki tiptir. Kaşif arılar kovan etrafında yeni yiyecek kaynağı ararken kovanda bekleyen seyirci arılar çalışan arayıcılar tarafından taşınan bilgiyi beklerler. Bir kovandaki ortalama kaşif arıların sayısı uygun koşullarda %5-10 civarındadır [37]. Arılar arasındaki bilgi değişimi ortak bilginin oluşmasında en önemli olaydır. Genellikle kovanın bazı bölümlerini ayırt etmek mümkündür. Bir kovan için en önemli bölüm bilgi değişiminin gerçekleştiği dans edilen alandır.

Arılar arasında bilgi değişimi dansla olur ve bu dans alanında yapılır. Bal arıları bilgi değişimi sayesinde kaliteli yeni yiyecek kaynaklarını keşfeder. Bal arılarında kaynağın yeri ve değeri “sallanma dansı” denilen bir dans ile diğer arılarla paylaşılır [44]. Farz edelim keşfedilen iki yiyecek kaynağı A ve B olsun. Başlangıçta muhtemel bir toplayıcı işsiz bir toplayıcı gibi başlayacaktır ve bu toplayıcı yuva etrafındaki herhangi bir yiyecek kaynağı bilgisine sahip değildir [37-39]. Şekil 2.3’te bal arılarının yem arama davranışları gösterilmeye çalışılmıştır.



Şekil 2. 3. Bal arılarının yem arama davranışları

Bir arı için iki seçenek vardır:

- i) Bir kaşif olabilir ve içgüdüyle veya yüzeysel ipuçlarıyla yuva etrafındaki yiyecekleri kendiliğinden aramaya baslar. (Şekil 2.3’teki K)

ii) Boş olabilir ve sallanış dansını izler ve yiyecek kaynağını aramaya başlar. (Şekil-2.3'teki D)

Yiyecek kaynağı bulunduktan sonra arı kaynağın pozisyonunu hafızasına alır ve hemen nektar kaynağından yararlanmaya baslar. Bu yararlanmadan dolayı bu arı çalışan toplayıcı olur. Toplanan nektar kovana götürülür ve depolanır. Depolama işleminden sonra arının takip edebileceği üç yol vardır. Mevcut kaynağı terk edip bağımsız hale gelebilir (Şekil 2.'teki V). Aynı kaynağa dönmeden önce bilgi paylaşımında bulunabilir (Şekil 2.3'teki T1). Bilgi paylaşımında bulunmadan kaynaktan nektar getirmeye devam edebilir (Şekil 2.3'teki T2). Son seçenek bir arı için sosyal davranış olarak görünmese de aslında kaynaktan nektar getiren yeteri kadar arı varsa geçerli bir yoldur. Kendini organizasyonun temel özellikleri arılarda aşağıdaki adımlara dayanır:

- i) Pozitif Geri besleme (Positive Feedback): Yiyecek kaynaklarındaki nektar miktarı gözcü arıların onları ziyaret sayısı ile gelişir.
- ii) Negatif Geri besleme (Negative Feedback): Bir yiyecek kaynağının araştırma süreci terk edildiğinde arılar durdurulur.
- iii) Dalgalanmalar (Fluctuations): Kaşifler, yeni yiyecek kaynaklarının keşfi için rast gele bir arama sürecini tamamlar.
- iv) Çoklu Etkileşim (Multiple Interactions): Arılar dans alanında yiyecek kaynağı pozisyonları hakkındaki bilgilerini paylaşırlar [32,38].

2.3.1.2. Algoritma Adımları

Karaboğa'nın YAK algoritmasının temel aldığı model de bazı kabuller yapılmaktadır. Bunlardan birincisi her bir kaynağın nektarının sadece bir görevli arı tarafından alınmasıdır. Yani görevli sayısı toplam yiyecek kaynağı sayısına eşittir. Nektarı tükenmiş kaynağın görevli arısı artık kaşif arı haline dönüşmektedir. Yiyecek kaynaklarının yerleri optimizasyon problemine ait olası çözümlere ve kaynakların nektar miktarları ise o kaynaklarla ilgili çözümlerin kalitesine karşılık gelmektedir. Dolayısıyla YAK optimizasyon algoritması en fazla nektara sahip kaynağın yerini bulmaya çalışarak uzaydaki çözümlerden problemin minimumunu yada maksimumunu veren noktayı (çözümü) bulmaya çalışmaktadır.

Bu modele ait süreç adımları aşağıdaki şekilde verilebilir.

- Yiyecek arama sürecinin başlangıcında, kaşif arılar çevrede rastgele arama yaparak yiyecek aramaya başlarlar.
- Yiyecek kaynakları bulduktan sonra, kaşif arılar artık görevli olurlar ve buldukları kaynaklardan kovana nektar taşımaya başlarlar. Her bir görevli arı kovana dönüp getirdiği nektarı boşaltır ve bu noktadan sonra ya bulduğu kaynağa geri döner yada kaynakla ilgili bilgiyi dans alanında sergilediği dans aracılığıyla kovanda bekleyen gözcü arılara iletir. Eğer faydalandığı kaynak tükenmiş ise görevli arı kaşif haline gelir ve yeni kaynak arayışına yönelir.
- Kovanda bekleyen gözcü arılar zengin kaynakları işaret eden dansları izler ve yiyeceğin kalitesi ile orantılı olan dans frekansına bağlı olarak bir kaynağı tercih ederler.

YAK algoritmasının temel adımları aşağıdaki şekilde verilmiştir.

1. İşçi arıları kaynaklara gönder ve nektar miktarlarını hesapla
2. Gözcü arıları kaynaklara gönder ve nektar miktarlarını hesapla
3. Rastgele yeni kaynaklar bulmaları için kaşif arıları gönder
4. O ana kadarki en iyi kaynağı hafızada tut
5. Durdurma kriteri sağlanana dek devam et

Yiyecek arayan arılarda görülen zeki davranış ile bu davranışı simüle eden YAK algoritmasının temel birimleri takip eden alt bölümlerde açıklanmaktadır.

Başlangıç Yiyecek Kaynağı Bölgelerinin Üretilmesi

Arama uzayını yiyecek kaynaklarını içeren kovan çevresi olarak düşünürsek, algoritma arama uzayındaki çözümlere karşılık gelen rastgele yiyecek kaynağı yerleri üretmek çalışmayı başlatmaktadır. Rastgele yer üretme süreci her bir parametrenin alt ve üst sınırları arasında rastgele değer üretmek gerçektir. Bu durum Denklem 2.38'de gösterilmiştir.

$$x_{i,j} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \quad (2.38)$$

Burada $i=1\dots SN$, $j=1\dots D$ ve SN yiyecek kaynağı sayısı ve D ise optimize edilecek parametre sayısıdır. x_j^{\min} , j . parametrenin alt sınırı, x_j^{\max} ise j . parametrenin üst sınırıdır. Aynı zamanda başlangıç aşamasında her kaynağın geliştirilememe sayısını ifade eden $failure_i$ (i . kaynağın geliştirilememe sayısı) sayaçları da sıfırlamaktadır.

Başlangıç aşamasından sonra yiyecek kaynaklarının görevli arı, gözcü arı ve kaşif arı süreçlerinden geçirilerek, daha iyisi bulunmaya çalışılır. YAK algoritması için durdurma kriteri olarak maksimum çevrim sayısı (MCN) ve kabul edilebilir bir hata değeri (ϵ) veya diğer optimizasyon algoritmaları için kullanılan standart bir durdurma kriteri tanımlanabilir.

İşçi Arıların Yiyecek Bölgelerine Gönderilmesi

Daha öncede belirtildiği gibi, her bir kaynağın bir görevli arısı vardır. Dolayısıyla yiyecek kaynaklarının sayısı görevli arıların sayısına eşittir. İşçi, çalıştığı yiyecek kaynağı komşuluğunda yeni bir yiyecek kaynağı belirler ve bunun kalitesini değerlendirir. Yeni kaynak daha iyi ise bu yeni kaynağı hafızasına alır. Yeni kaynağın mevcut kaynak komşuluğunda belirlenmesinin benzetimi Denklem 2.39 ile tanımlanmaktadır.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2.39)$$

x_i ile gösterilen her bir kaynak için bu kaynağın yani çözümün tek bir parametresi (rastgele seçilen parametresi, j) değiştirilerek x_i komşuluğunda v_i kaynağı bulunur. Denklem 2.93'da j , $[1, D]$ aralığında rastgele üretilen bir tamsayıdır. Rastgele seçilen j parametresi değiştirilirken, yine rastgele seçilen x_k komşu çözümünün ($k \in \{1,2,\dots,SN\}$) j . parametresi ile mevcut kaynağın j . parametresinin farkları alınıp $[-1,1]$ arasında rastgele değer alan Φ_{ij} sayısı ile ağırlıklandırıldıktan sonra mevcut kaynağın j . parametresine eklenmektedir.

Denklem 2.39'tan görüldüğü gibi x_{ij} ve x_{kj} arasındaki fark azaldıkça, yani çözümler birbirine benzedikçe, x_{ij} parametresindeki değişim miktarı azalacaktır. Böylece bölgesel optimal çözüme yaklaştıkça değişim miktarı da adaptif olarak azalmaktadır.

Bu işlem sonucunda üretilen v_{ij} 'nin daha önceden belli olan parametre sınırlarını aşması durumunda Denklem 2.40'da gösterildiği gibi j. parametreye ait olan alt veya üst sınır değerlerine ötelenmektedir.

$$v_{ij} = \begin{cases} x_j^{\min}, v_{ij} < x_j^{\min} \\ v_{ij}, x_j^{\min} < v_{ij} < x_j^{\max} \\ x_j^{\max}, v_{ij} > x_j^{\max} \end{cases} \quad (2.40)$$

Sınırlar dahilinde üretilen v_i parametre vektörü yeni bir kaynağı temsil etmekte ve bunun kalitesi hesaplanarak bir uygunluk değeri atanmaktadır.

$$fitness_i = \begin{cases} 1/(1 + f_i), f_i \geq 0 \\ 1 + abs(f_i), f_i < 0 \end{cases} \quad (2.41)$$

Burada f_i , v_i kaynağının yani çözümünün maliyet değeridir. x_i ile v_i arasında nektar miktarlarına yani uygunluk değerlerine göre bir aç gözlü (greedy) seçme uygulanır. Yeni bulunan v_i çözümü daha iyi ise görevli arı hafızasından eski kaynağın yerini silerek v_i kaynağının yerini hafızaya alır. Aksi takdirde görevli arı x_i kaynağına gitmeye devam eder ve x_i çözümü geliştirmedeği için x_i kaynağı ile ilgili geliştirememeye sayacı ($failure_i$) bir artar, geliştirdiği durumda ise sayaç sıfırlanır.

Gözcü Arıların Seleksiyonda Kullanacakları Olasılık Değerlerinin Hesaplanması (Dans Benzetimi)

Tüm görevli arılar bir çevrimde araştırmalarını tamamladıktan sonra kovana dönüp buldukları kaynakların nektar miktarları ile ilgili gözcü arılara bilgi aktarırlar. Bir gözcü

arı dans aracılığıyla paylaşılan bilgidен faydalanarak yiyecek kaynaklarının nektar miktarları ile orantılı bir olasılıkla bir bölge (kaynak) seçer. Bu YAK' ın çoklu etkileşim sergilediğinin bir örneğidir. Olasılıksal seçme işlemi, algoritmada nektar miktarlarına karşılık gelen uygunluk değerleri kullanılarak yapılmaktadır. Uygunluk değerine bağlı seçme işlemi rulet tekerleği, sıralamaya dayalı, stokastik örnekleme, turnuva yöntemi, ya da diğer seleksiyon şemalarından herhangi biri ile gerçekleştirilebilir. Temel YAK algoritmasında bu seleksiyon işlemi rulet tekerleği kullanılarak yapılmıştır. Tekerlekteki her bir dilimin açısı uygunluk değeri ile orantılıdır. Yani bir kaynağın uygunluk değerinin tüm kaynakların uygunluk değeri toplamına oranı o kaynağın diğer kaynaklara göre nisbi seçilme olasılığı olduğunu vermektedir . Bu durum Denklem 2.42'de verilmiştir.

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (2.42)$$

Burada $fitness_i$, i. kaynağın kalitesini, SN görevli arı sayısını göstermektedir. Bu olasılık hesaplama işlemine göre bir kaynağın nektar miktarı arttıkça (uygunluk değeri arttıkça) bu kaynak bölgesini seçecek gözcü arı sayısı da artacaktır. Bu özellik YAK' ın pozitif geri besleme özelliğine karşılık gelmektedir.

Gözcü Arıların Yiyecek Kaynağı Bölgesi Seçmeleri

Algoritmada olasılık değerleri hesaplandıktan sonra bu değerler kullanılarak rulet tekerleğine göre seçim işleminde her bir kaynak için [0,1] aralığında rastgele sayı üretilir ve p_i değeri bu üretilen sayıdan büyükse görevli arılar gibi gözcü arı da Denklem 2.39'u kullanarak bu kaynak bölgesinden yeni bir çözüm üretir. Yeni çözüm değerlendirilir ve kalitesi hesaplanır. Sonra, yeni çözümlerle eski çözümün uygulandıklarının karşılaştırıldığı ve iyi olanın seçildiği aç gözlü seleksiyon işlemine tabi tutulur. Yeni çözüm daha iyi ise eski çözüm yerine bu çözüm alınır ve çözüm geliştirememeye sayacı ($failure_i$) sıfırlanır. Eski çözümün uygunluğu daha iyi ise bu çözüm muhafaza edilir ve geliştirememeye sayacı ($failure_i$) bir artırılır. Bu süreç tüm gözcü arılar yiyecek kaynağı bölgelerine dağılına kadar devam eder.

Kaynağı Bırakma Kriteri: Limit ve Kaşif Arı Üretimi

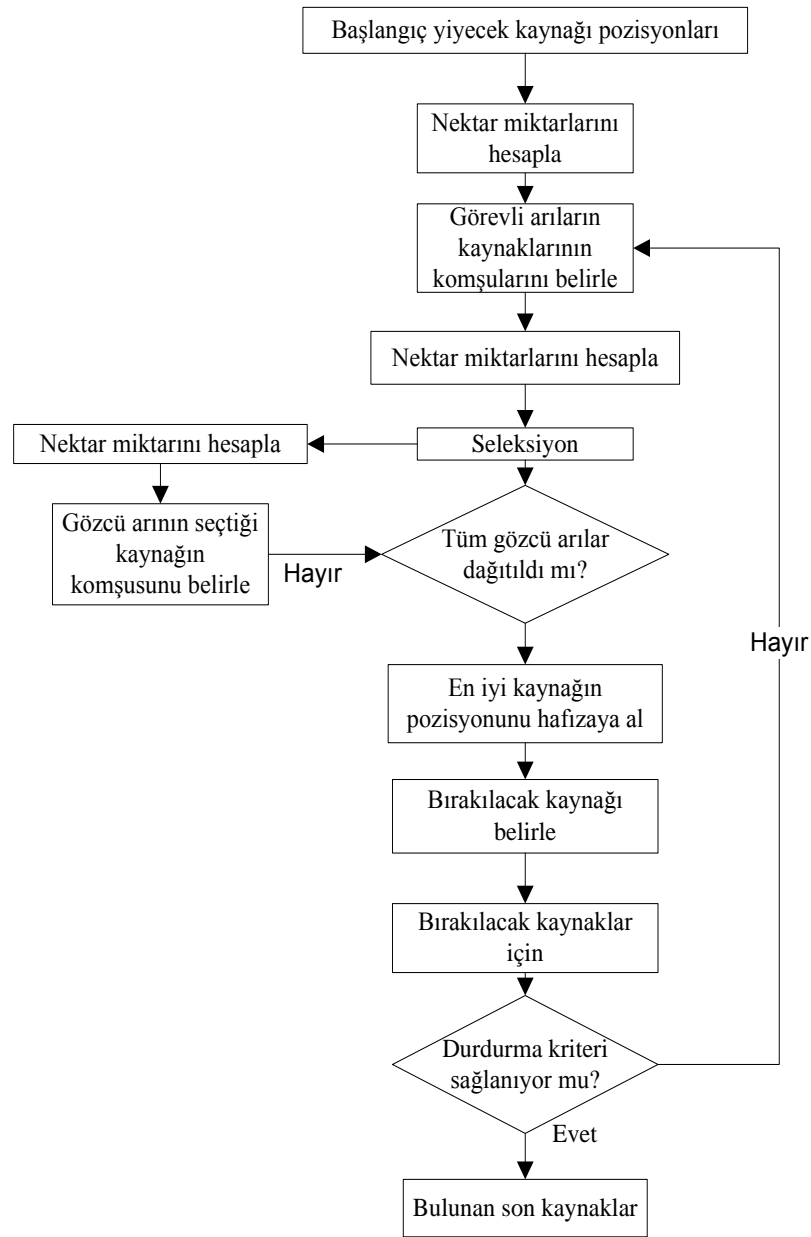
Bir çevrim sonunda tüm görevli ve gözcü arılar arama süreçlerini tamamladıktan sonra çözüm geliştirememeye sayaçları (failure;) kontrol edilir. Bir arının bir kaynaktan faydalanıp faydalanamadığı, yani gidip geldiği kaynağın nektarının tükenip tükenmediği çözüm geliştirememeye sayaçları aracılığıyla bilinir. Bir kaynak için çözüm geliştirememeye sayacı belli bir eşik değerinin üzerindeyse, artık bu kaynağın görevli arısının tükenmiş olan o çözümü bırakıp kendisi için başka bir çözüm araması gerekmektedir. Bu da biten kaynakla ilişkili olan görevli arının kaşif arı olması anlamına gelmektedir. Kaşif arı haline geldikten sonra, bu arı için rastgele çözüm arama süreci başlar (Denklem 2.38). Kaynağın tükendiğinin belirlenmesi için kullanılan eşik değeri, YAK algoritmasının önemli bir kontrol parametresidir ve "limit" olarak adlandırılmaktadır. Temel YAK algoritmasında her çevrimde bir kaşif arının çıkmasına izin verilir.

Tüm bu birikimler arasındaki ilişki ve döngü Şekil 2.4' teki gibi bir akış diyagramı ile şematize edilebilir [44].

Seleksiyon Mekanizmaları

YAK algoritması 4 farklı seleksiyon işlemi kullanmaktadır. Bunlar;

- Potansiyel iyi kaynakların belirlenmesine yönelik Denklem 2.42 ile olasılık değerlerinin hesaplandığı global olasılık temelli seleksiyon süreci
- Görevli ve gözcü arıların renk, şekil, koku gibi nektar kaynağının türünü belirlenmesini sağlayan görsel bilgiyi kullanarak bir bölgede kaynağın bulunmasına vesile olan bölgesel olasılık tabanlı seleksiyon işlemi (Denklem 2.39)
- İşçi ve gözcü arıların daha iyi olan kaynağı belirlemek amacıyla kullandıkları aç gözlü seleksiyon.
- Kaşif arılar tarafından Denklem 2.38 aracılığıyla gerçekleştirilen rastgele seleksiyon. Bütün bu seleksiyon metotlarının bir arada kullanılmasıyla YAK algoritması hem iyi bir global araştırma hem de bölgesel araştırma yapabilmektedir [44].



Şekil 2. 4. YAK Algoritmasının akış diyagramı

2.3.1.3. Yapay Arı Kolonisi (YAK) Algoritmasının Özellikleri

YAK algoritmasının özellikleri şu şekilde sıralanabilir;

- Basit ve esnekler,
- Gerçek arı davranışlarına oldukça yakın şekilde simüle eder,
- Sürü zeka optimizasyonuna dayanan bir algoritmadır,
- Az kontrol parametrelerine sahiptir.

2.3.2. Genetik Algoritma (GA)

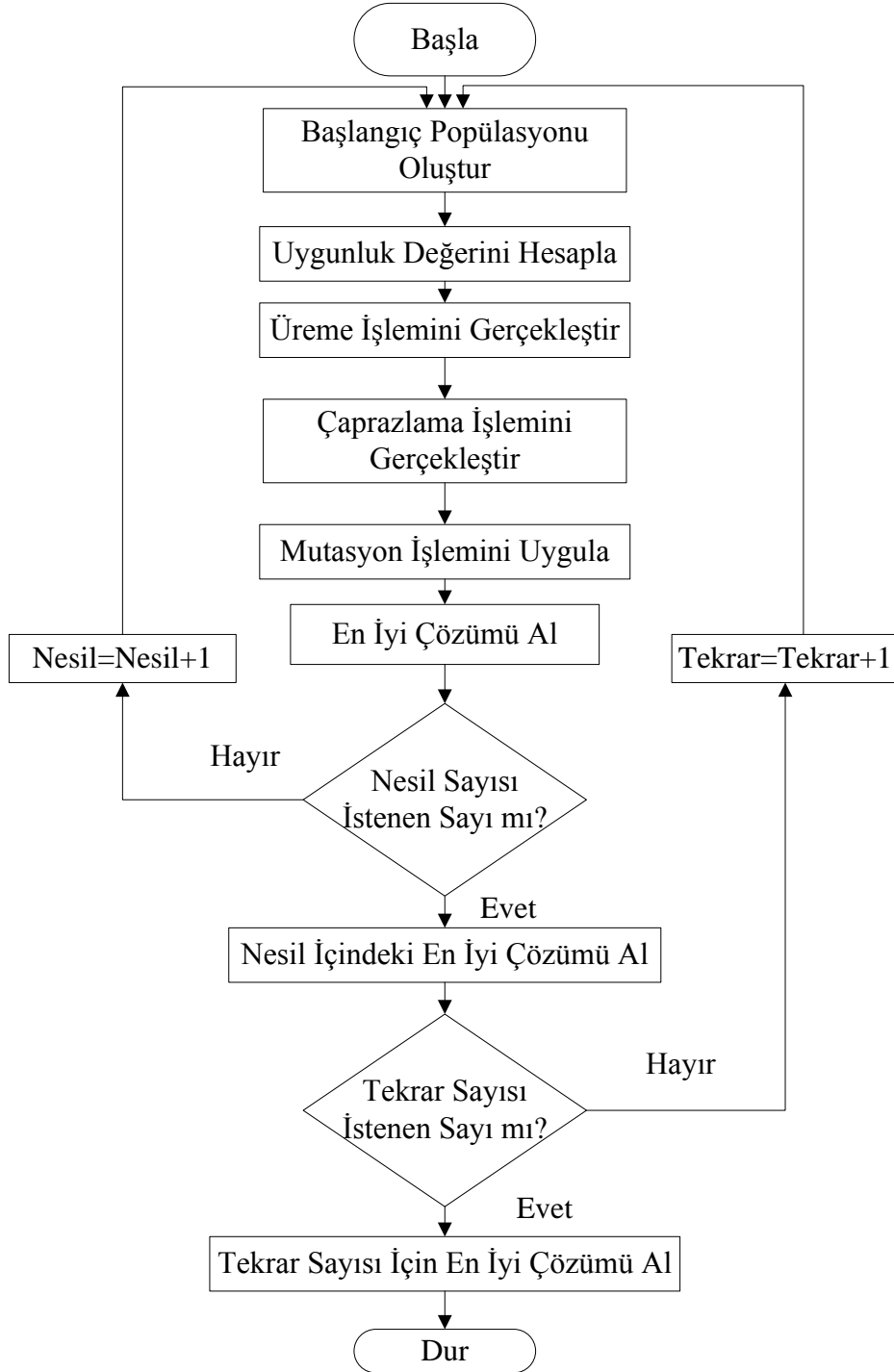
GA fikri J. Holland'a aittir. Holland, arkadaşları ve öğrencileri ile birlikte bu algoritmayı geliştirmiş ve ilk çalışmalarının sonucunda "Adaptation in Natural and Artificial System" isimli kitabını çıkartmıştır [60-62]. GA, sezgisel bir metot olduğundan dolayı verilen bir problem için optimum sonucu bulamayabilir, ancak bilinen metotlarla çözülemeyen yada çözüm zamanı problem büyüklüğü ile üstel artan problemlerde optimale çok yakın çözümler vermektedir. Başlangıçta sürekli doğrusal olmayan optimizasyon problemlerine uygulanan GA, sonraları gezgin satıcı, karesel atama, yerleşim, atölye çizelgeleme, ders/sınav programı hazırlanması gibi kombinatoriyal optimizasyon problemlerinde de başarıyla uygulanmıştır [63-67].

2.3.2.1. GA Yapısı

GA yönlendirilmiş rastgele araştırma algoritmalarının bir türüdür. Bu algoritma canlılarda bulunan genetik gelişimi simüle etmektedir. Algoritma, araştırma uzayında mevcut olan çözümlerin oluşturduğu bir başlangıç yoğunlunu (popülasyon) kullanmaktadır. Bu başlangıç yoğunluğu, her bir kuşakta (generation) tabii seçme (natural selection) ve tekrar üreme (reproduction) işlemleri vasıtası ile ard arda geliştirilir. En son kuşağın en uygun yani en kaliteli bireyi, problem için optimal bir çözümdür. Bu her zaman optimum olmaya bilir ama kesinlikle optimuma en yakın olan optimaldir.

Öncelikle başlangıç yoğunlu oluşturulur ve sonra genetik operatörler tekrar üreme, çaprazlama ve mutasyon bir sonraki kuşaktaki çözümleri üretmek için kullanılır. Kalite veya uygunluk değerlendirilme işlemi, tekrar üreme olayı boyunca uygulanan seçme işlemini gerçekleştirile bilmek için her bir bireye uygulanır. Birbirini takip eden kuşakların gelişmesi ve değerlendirilmesi çevrimi, optimal bir çözüm bulununcaya kadar devam eder [68]. Burada her çözüme birey yada kromozom adı verilir. Her kromozom genlerden oluşur ve bir gen, bir kromozom oluşturulurken kodlanabilen en küçük bilgiye verilen isimdir. Kısacası problemin özünü tasvir eden kriter, uygunluk fonksiyonu veya amaç fonksiyonudur. Uygunluk değeri bütün kromozomlar için hesaplanır ve popülasyon içerisinde birden fazla kromozom olduğundan GA'ların bir

diğer önemli özelliđi olarak bir grup çözümle uğraşmasıdır. Bu sayede çok sayıda çözümün içinden iyileri seçilip kötüleri elenebilir [69]. Bir GA'nın akış diyagramı Şekil 2.5'te verilmiştir [69].



Şekil 2. 5. Basit bir GA akış diyagramı

2.3.2.2. GA İşlem Adımları

- Çizelgelenecek bütün işler kromozom olarak kodlanır. Bir kromozomu oluşturan her bir elemana gen denir. Her bir kromozom, bu arama uzayında belirli bir çözüme tekabül eder.
- Rastgele olarak kromozom seçilir ve başlangıç popülasyonu olarak kabul edilir.
- Her bir kromozom için bir uygunluk değeri, tüm işler için tamamlanma zamanı hesaplanır, bulunan uygunluk değerleri kromozomların çözüm kalitesini gösterir.
- Bir grup kromozom belirli bir olasılık değerine göre rastgele seçilip üreme işlemi gerçekleştirilir.
- Üreme işleminde çeşitli genetik operatörler kullanılabilir.
- Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.
- Önceden belirlenen nesil sayısı boyunca 4 ve 6. adımlar devam ettirilir.
- İterasyon nesil sayısına ulaşıncaya kadar işlem bitirilir. Uygunluk değeri en yüksek olan kromozom en iyi çözüm olarak seçilir [70].

2.3.2.3. GA'da Kullanılan Operatörler

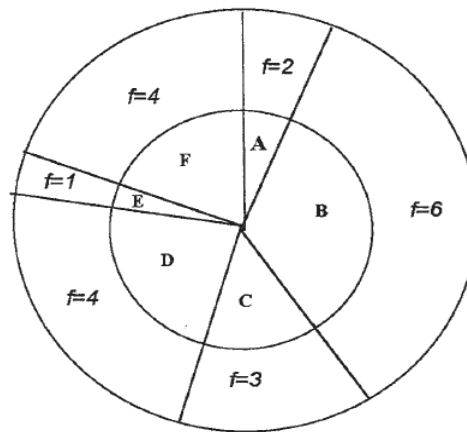
Kromozomlar üzerinde işlem yapmaya yarayan bu operatörler tekrar üreme, çaprazlama ve mutasyon operatörlerini ihtiva eder. Bu operatörlerin uygun şekilde seçimi GA performansı için çok önemlidir.

Tekrar Üreme Operatörü

Tekrar üreme operatörü, bir popülasyonda çaprazlamaya uğrayan dizilerin seçim görevini yerine getirir. Tekrar üreme operatörü, tabii seçme işlemi olarak adlandırılan ve kaliteli olanların hayatta kalmaları prensibine göre çalışan bir elemandır. Tabiatta bu seçme işlemi çevre tarafından kontrol edilirken yapay sistemlerde ise değerlendirme fonksiyonu ile diğer kalite yani uygunluk değerlendirme işlemleri tarafından kontrol edilir. Bireysel yapılar, bir kuşaktan diğer kuşağa geçerken uygunluk değerlerine göre kopya edilerek çoğaltılırlar. Uygunluğu daha yüksek olan çözümlerin diğerlerine göre daha fazla kopya edilme şansları vardır. Bu yüzden daha kaliteli çözümlerin yoğunluk

olarak baskın olmaları, yeterli uygunluğu taşımayanların (kalitesiz olanların) ise yavaş yavaş elenmeleri yani ölmeleri sağlanır. Bu operatörün uygulanması için kullanılan en basit metod kararlı rulet tekerleği tekniğidir [71].

Rulet tekerleği ebeveyn seçme operatörü, iyi bilinen ebeveyn belirleme metodlarından biridir. Bir sonraki jenerasyon için ebeveynlerin A kromozomlarından F'ye belirlendiği bir örnek Şekil 2.6'da verilmiştir. Seçme operatörü birkaç kademede çalışır. İlk olarak her bir kromozomun uygunlukları toplanarak popülasyonun toplam uygunluğu bulunur. Seçilen her bir ebeveyn için bu prosedür kullanılır. Sonra her bir kromozom uygunluğunun toplam uygunluğun yüzde kaçını temsil ettiği bulunur. Böylece herbiri bir çözümü ifade eden kromozomların uygunlukları oranında bunlara tekerlek üzerinde bir parça verilir. Şekil 2.5'da tekerlek toplam uygunluk değeri olan 20 eşit parçaya ayrılmış ve her kromozom bu tekerlek üzerinde ağırlıkları (uygunlukları) oranında yerleştirilmiştir. Tekrar üreme için rulet tekerleğinin döndürülmesi gerekir. Bunun için sıfırla toplam uygunluk değeri arasında rastgele bir sayı üretilerek bu sayının tekerleğin hangi parçasına karşılık geldiğine ve dolayısıyla hangi kromozomun seçildiğine bakılır. Artık tekerlek bir kere dönmüştür ve ilk kromozom belirlenmiştir. Benzer şekilde diğer kromozomların da belirlenmesinden sonra uygunluk değerleri en başarılı olan adaylar eşleştirme havuzu (mating pool) olarak anılan bir havuzda toplanarak, diğer genetik operatörlerin uygulanması için hazırlanırlar [71].



Şekil 2. 6. Rulet tekerleği seçme operatörü

Şekil 2.6'ya göre A-F kromozomları için uygunluğu f ve toplam uygunluk ise 20'dir. Yani şekildeki daire 20 eşit parçaya ayrılmış ve $f=2$ uygunluğuna sahip olan A kromozomu bu parçalardan ikisini almıştır. Benzer olarak B kromozomu bu parçalardan altısına, C kromozomu üçüne, d kromozomu dördüne, E kromozomu birine ve F kromozomu ise yine dördüne sahiptir. Şekle göre A'nın 1 ve 2, B'nin 3-8, C'nin 9-11, D'nin 12-15, E'nin 16 ve F'nin 17-20 nolu parçalara sahip olduğu düşünülürse, 8, 17, 3, 12, 9 ve 19 değerlerine sahip rastgele sayıların üretilmesi durumunda seçilen ebeveynler sırayla B, F, B, D, C ve F olacaktır. Kolaylıkla görülebileceği gibi $f=6$ ile en yüksek uygunluk değerine sahip olan B kromozomunun diğerlerine göre seçilme şansı daha fazla olacaktır.

Çaprazlama Operatörü

Çaprazlama operatörü GA'nın çok önemli bir elemanıdır. Bu eleman değişik kromozom konfigürasyonlarının test edilmesine ve hızlı şekilde tekrar yapılmasına olanak sağlar. Genelde tek noktalı çaprazlama kullanılır. Şekil 2.7'de böyle bir tek noktalı çaprazlama işlemi gösterilmiştir.

$$\begin{array}{l} 1. \text{ Ebeveyn } 101 | 01010 \implies 1. \text{ Çocuk } 101 | 10000 \\ 2. \text{ Ebeveyn } 111 | 10000 \implies 2. \text{ Çocuk } 111 | 01010 \end{array}$$

Şekil 2. 7. Tek noktalı çaprazlama

Bunun için eşleştirme havuzunda bulunan yapıların birer çifti rasgele olarak seçilir ve çaprazlama operatörü bu iki yapıdan yeni iki yapı meydana getirmek için kullanılır. Çaprazlamaya tabi tutulacak iki kromozom üzerinde yine rastgele şekilde bir çaprazlama noktası belirlenir. Bu çaprazlama bölgesinde yer alan tüm genetik materyaller karşılıklı yer değiştirerek iki yeni çocuk kromozomu elde edilir. Oluşturulan çocuklar ebeveynlerinden tamamen farklı olabilir. Bu yüzden her bir jenerasyonda oldukça farklı çözümler incelenebilir [71].

- Sınırlayıcılarla ilgili bilgiler araştırma sürecinden bağımsızdır.
- GA sınırlı sayıda tasarım değişkenleriyle çalışır.

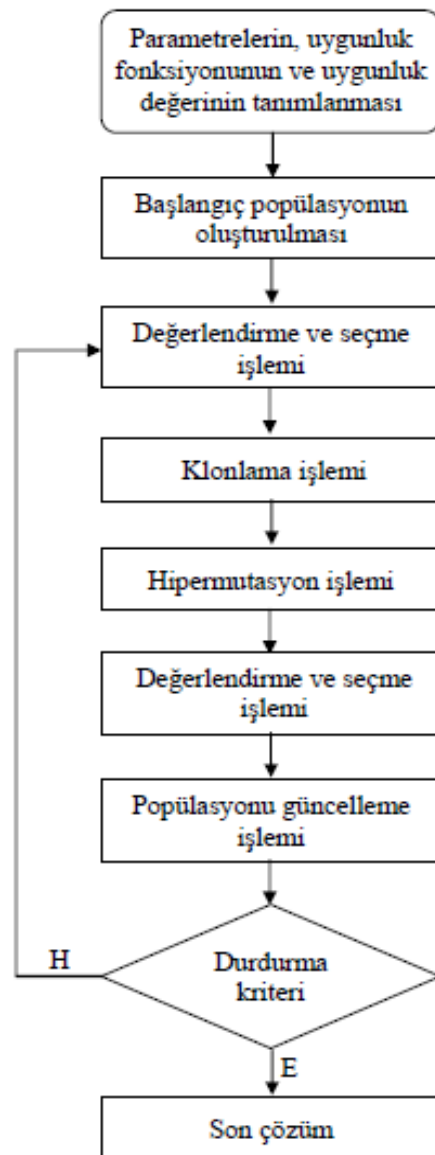
2.3.3. Klonal Seçme Algoritması (KSA)

KSA, insan bağışıklık sistemindeki klonal seçme prensibinden esinlenerek geliştirilmiş nispeten yeni bir optimizasyon algoritmasıdır. KSA, yerel minimumdan kurtulabilme, belirli kurallar yerine rastgele geçiş kuralları kullanma, çok parametrelili optimizasyon problemleri ile rahatlıkla çalışabilme yeteneklerine sahiptir. Ayrıca uygulanması basit ve anlaşılması kolaydır [73].

Doğal bağışıklık sistemi temellerine göre çalışan yapay bağışıklık sistemlerinde farklı tip problemlerde çözüme ulaşmak için üç farklı algoritma kullanılmaktadır [74]. Bunlar Negatif Seçme, Klonal Seçme ve Pozitif Seçme algoritmalarıdır.

Klonal seçme prensibi [74-76], temel olarak antijen uyarılarına karşı bağışıklık cevabının temel özelliklerini açıklamak için kullanılır. Bu prensibe göre sadece antijenleri tanıyan hücreler çoğalmak üzere seçilir. Seçilen bu hücreler, seçici antijenlere olan benzerliklerini artırmak için benzerlik olgunlaşması işlemine (affinity maturation process) tabi tutulur. Vücutta bir antijen tespit edildiği zaman bu antijenler kemik iliğinden üretilen antikörlerle karşılanırlar. Antijen, antikörlere bağlanır ve T yardımcı hücrelerinden aldığı sinyalle B hücrelerini, bölünmeleri ve plazma hücreleri denilen antikör salgılayan hücrelere dönüşmesi için uyarır. Hücre bölünmesi süreci bir klon oluşturur, tek bir hücreden bir hücre veya hücreler kümesi oluşturulabilir. Klonal seçim mekanizması, negatif seçimin rolünü tamamlayıcı olarak, bir "nonself" hücre, bir "B" hücresi tarafından tanındığı zaman nasıl bir bağışıklık tepkisi verileceğini açıklamak için kullanılmaktadır. Algoritmanın (CLONALG) işleyişi şu şekilde gerçekleşmektedir. Öncelikle rasgele üretilen hücrelerden bir başlangıç popülasyonu oluşturulmaktadır. Her bir hücre muhtemel bir çözümü antijen de problemi temsil etmektedir. Hücreler ikili bit dizilerinden oluşurlar ve bit dizilerinin uzunluğu istenen hassasiyete göre seçilmektedir. Ardından popülasyondaki her hücreye ait hata fonksiyonu değeri (uygunluk) hesaplanır ve hücreler uygunluk değerlerine göre sıralanır. En düşük uygunluk değerine sahip n tane hücre seçilir ve sonra uygunluk değerleri ile orantılı olarak klonlanır. Oluşturulan klonlar, uygunluk değerleri ile ters

orantılı olarak mutasyona uğrarlar. Daha sonra mutasyona uğratılmış (olgunlaşmış) klonların uygunluk değerleri hesaplanır. Seçilen her bir hücre ve klonları bir alt popülasyon oluşturur. Her bir alt popülasyonun en düşük uygunluk değerine sahip hücresi seçilir ve yaşamasına izin verilir. Popülasyonun en yüksek uygunluk değerlerine sahip d tane hücresi rastgele üretilen hücrelerle değiştirilir. Böylelikle popülasyondaki farklılaşma sağlanmış olur. Bu işlemler durdurma kriteri sağlanana kadar devam eder [74-76]. KSA'nın temel blok şeması Şekil 2.9'da verilmiştir [77].



Şekil 2. 9. KSA'nın temel blok şeması

KSA, ilk olarak rasgele üretilen hücrelerden verilen sınırlar dâhilinde bir başlangıç popülasyonu (N) oluşturulur. Her bir hücre muhtemel bir çözümü, antijen ise problemi temsil etmektedir. Hücreler ikili bit dizilerinden oluşur. Bit dizilerinin uzunluğu kullanıcı tarafından problem için istenen kriterlere göre seçilir. Daha sonra popülasyondaki her hücreye ait uygunluk fonksiyonu değeri hesaplanır ve hücreler uygunluk değerlerine göre sıralanır. En yüksek uygunluk değerine sahip n tane hücre seçilir ve sonra uygunluk değerleri ile orantılı olarak klonlanır. Oluşturulan klonlar, uygunluk değerleri ile ters orantılı olarak mutasyona uğrarlar. Daha sonra mutasyona uğratılmış klonların uygunluk değerleri hesaplanır. Seçilen her bir hücre ve klonları bir alt popülasyon oluşturur. Her bir alt popülasyonun en yüksek uygunluk değerine sahip hücresi seçilir ve yaşamasına izin verilir. Popülasyonun en düşük uygunluk değerlerine sahip d tane hücresi rastgele üretilen hücrelerle değiştirilir. Böylelikle popülasyondaki çeşitlilik sağlanmış olur. Bu işlemler durdurma ölçütü sağlanana kadar devam eder.

2.3.3.1. KSA'daki Temel İşlemler

Başlangıç Popülasyonunun Oluşturulması

KSA'da, ilk olarak rastgele üretilen hücrelerden verilen sınırlar dâhilinde bir antikor havuzu olarak tanımlanan N boyutlu bir başlangıç popülasyonu oluşturulur.

Değerlendirme ve Seçme İşlemi

Optimize edilecek bir amaç fonksiyonu bulunmaktadır. Bir antikorun benzerlik değeri, verilen antikor için hesaplanan amaç fonksiyonunun değerine karşılık gelir: Her bir Ab antikoru, girdi kümesinin (Ab) bir elemanıdır. Her bir Ab için benzerlik değeri (amaç fonksiyon değeri) hesaplanır. En yüksek benzerliği (en yüksek uygunluğu) gösteren n tane antikor Ab kümesinden seçilir ve yeni bir $\{n\}$ Ab kümesi oluşturulur.

Klonlama İşlemi

Seçilen n tane antikor bağımsız olarak ve antijenik benzerlikleriyle orantılı olarak klonlanır, klonlar bir C repertuarı oluşturur. Seçilen n antikorun her biri için oluşturulan

klon sayısı benzerlik değeri ile orantılıdır. Daha yüksek antijenik benzerlik (daha yüksek uygunluk değeri), daha fazla sayıda klon demektir. Seçilen n tane hücrenin her biri için oluşturulan klon sayısı aşağıdaki denklem ile verilir.

$$N_c = \sum_{i=1}^n \text{round}\left(\frac{\beta N}{i}\right) \quad (2.43)$$

Burada, β klonlama faktörüdür, n seçilen antikor sayısıdır ve round (.) argümanını en yakın tam sayıya tamamlayan bir operatördür.

Hipermutasyon İşlemi

C repertuarı, antijenik benzerlik ile ters orantılı olarak hipermutasyon (olgunlaştırma) işlemine uğratılır. Bu süreç sonunda olgunlaştırılmış kopyaların oluşturduğu bir C^* , hipermutasyona uğratılmış kopyalar popülasyonu oluşturulur. Kopyaların hipermutasyona uğratılma oranı benzerlik değerleri ile ters orantılıdır. Daha yüksek benzerlik (daha yüksek uygunluk değeri), daha az hipermutasyon oranı demektir.

Değerlendirme ve Seçme İşlemi (Hipermutasyon Sonrası)

Hipermutasyona uğratılmış C^* kopyalarının benzerlik (uygunluk) değerleri hesaplanır. En yüksek uygunluk değerine sahip n tane antikor yeniden seçilir ve Ab kümesine eklenir.

Popülasyonu güncelleme işlemi

Son olarak, Ab kümesinden en düşük benzerlik değerine sahip d tane antikor, yeni oluşturulmuş antikorlar ile değiştirilir.

2.3.3.2. KSA'nın Kontrol Parametreleri

KSA'nın temel kontrol parametreleri, popülasyon büyüklüğü, maksimum jenerasyon sayısı, klonal faktör ve hipermutasyon oranıdır. Bu kontrol parametrelerinin değerlerinin seçimi algoritmanın performansı üzerinde oldukça etkilidir

Popülasyon Büyüklüğü

Popülasyon büyüklüğü algoritma performansı açısından önemli bir kriterdir. Seçilen değerin aşırı küçülmesi, yetersiz örneklemeye sebep olacağından yakınsamayı sağlamak zorlaşacak ve araştırma belirli bir alt en iyi noktaya doğru sürüklenecektir. Tersine, aşırı yüksek değer seçildiğinde bir jenerasyonluk gelişim oldukça uzun zaman gerektirecektir.

Klonlama Faktörü

Seçilen antikorlar için oluşturulacak klon sayısını belirleyen bir kontrol parametresidir. Bu parametrenin uygun seçimi algoritmanın performansı açısından önemlidir. Bu kontrol parametresinin değeri genellikle $[0,1]$ aralığında seçilir. Değeri küçük seçilirse KSA, yerel bölgede daha fazla araştırma yapar.

Hipermutasyon Oranı

Hipermutasyon oranı da algoritma performansını etkileyen diğer bir parametredir. Hipermutasyon işlemi, araştırma sahasına yeni antikorların girmesini sağlar. Yüksek hipermutasyon oranı, araştırmaya aşırı bir rastgelelik kazandıracak ve araştırmayı çok hızlı olarak ıraksatacaktır. Bu durumun tersine çok düşük hipermutasyon oranı ıraksamayı aşırı düşürecek ve araştırma uzayının tamamen araştırılmasını engelleyecektir [77].

BÖLÜM 3

BULGULAR

3.1. Sistem Modelleme Uygulamaları

Bu bölümde, doğrusal ve doğrusal olmayan parametrik sistemler ile doğrusal olmayan gerçek YAK Algoritması, GA ve KSA ile modellenmesi verilmiş ve YAK Algoritmasının performansı ortaya konmuştur.

3.2. Bulgular

3.2.1. Uygulamalarda Kullanılan Bilgisayar Özellikleri

Uygulamalar yapılırken kullanılan bilgisayara ait özellikler aşağıdaki gibidir. Farklı özellikteki bilgisayarların kullanılması süre olarak sonuçları etkileyebilir.

İşletim sistemi: Windows 7 Ultimate 32 bit
İşlemci hızı: i5-2410M CPU @ 2.30 Ghz
Bellek: 4096 MB Ram

3.2.2. Doğrusal Parametrik Sistem Modellemesi Uygulamaları

3.2.2.1. Örnek Sistem 1

İlk olarak Denklem 3.1'de doğrusal olarak bilinen bir sistem ele alınmış ve algoritmaların bu sisteme ait parametreleri bulma sonuçları karşılaştırılmıştır [78-79]. Bu sistem için Denklem 3.2'deki model yapısı önerilmiştir. Model parametrelerinin tahmini amacıyla parametre kestirim çalışmalarında giriş verisi $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi $n=100$ adet veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Tablo 3.1'de algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların hesapladığı parametreler Tablo 3.2'de

verilmiştir. Tablo 3.2'deki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.1'de verilmiştir.

$$y(n) = 0.0798 u(n) + 0.0791 u(n-1) + 0.0791 u(n-2) + 0.0798 u(n-3) + 1.5558y(n-1) - 1.270 y(n-2) + 0.3983 y(n-3) \quad (3.1)$$

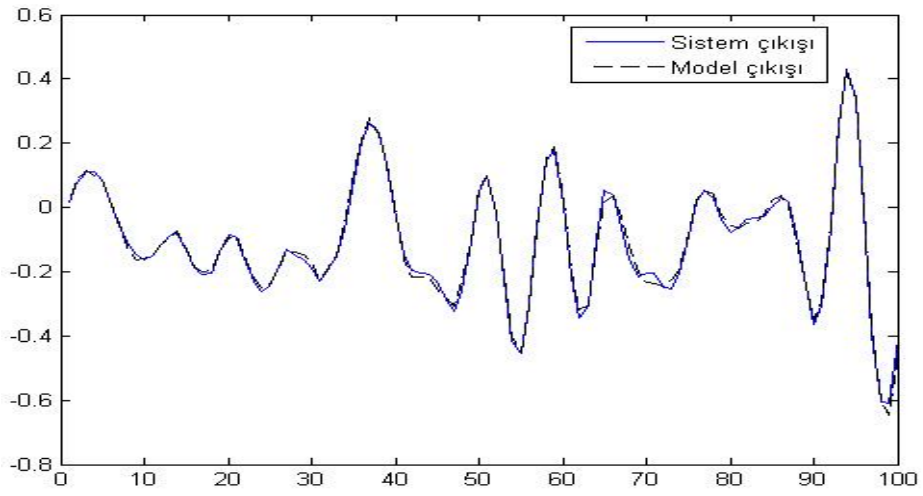
$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) + a_3 u(n-3) + b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3) \quad (3.2)$$

Tablo 3. 1. Denklem 3.2'de önerilen modele ait kullanılan kontrol parametreleri

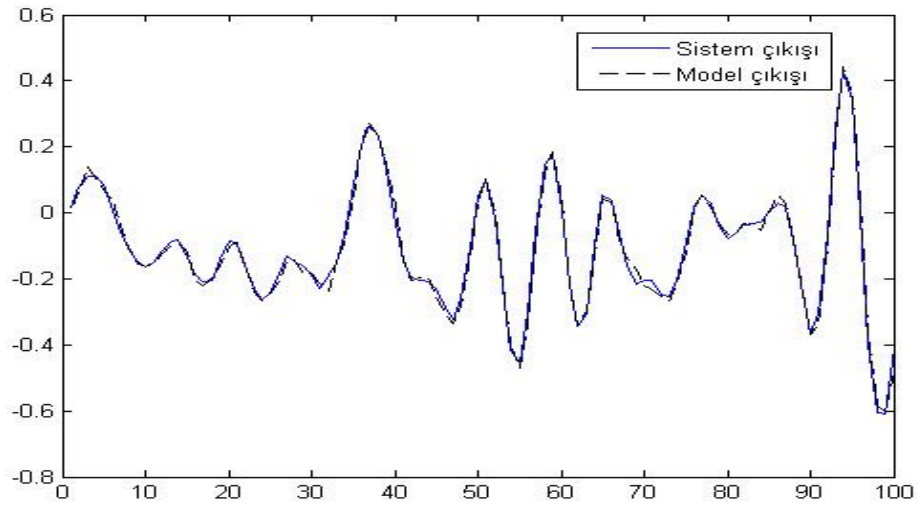
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-2 +2	-2 +2	-2 +2
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	100	100	100

Tablo 3. 2. Denklem 3.2'de önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

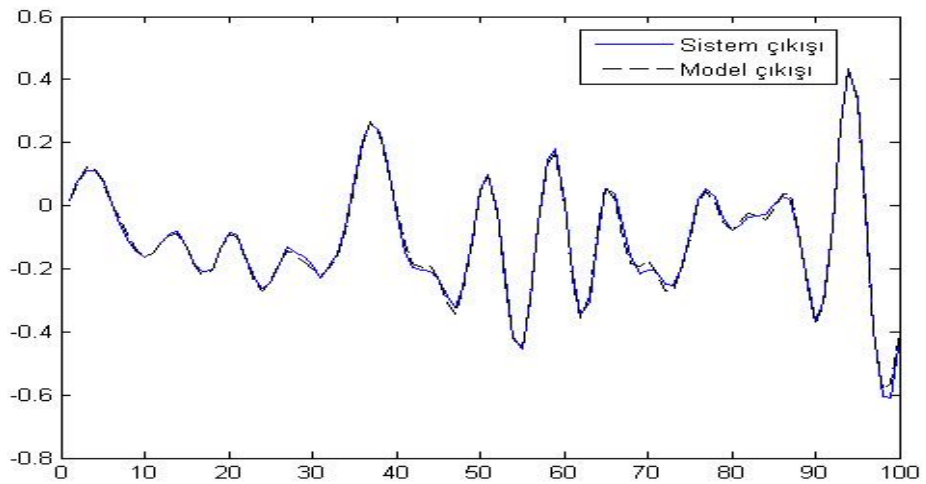
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a₀	0.0798	0.0878	0.0390	0.0608
a₁	0.0791	0.1349	0.0941	0.0611
a₂	0.0791	0.1154	0.1159	0.0957
a₃	0.0798	0.0928	0.0779	0.1025
b₁	1.5558	1.1002	1.4406	1.6832
b₂	-1.270	-0.5940	-1.0797	-1.5893
b₃	0.3983	0.0343	0.3581	0.6267
Ortalama süre (sn)		3.7384	3.6515	7.1880
MSE		2.070x10 ⁻⁴	4.347x10 ⁻⁴	1.95x10 ⁻⁴
Standart sapma		3.040x10 ⁻⁴	1.60x10 ⁻³	3.8x10 ⁻³



(a)



(b)



(c)

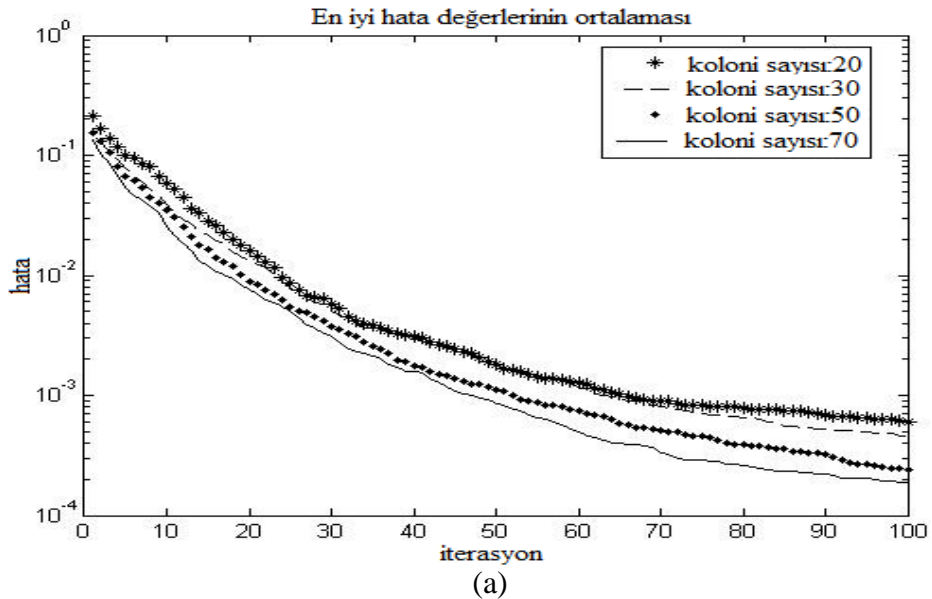
Şekil 3. 1. Denklem 3.1'deki model için sistem ve model çıktıları: (a) yak algoritması, (b) ga, (c) ksa

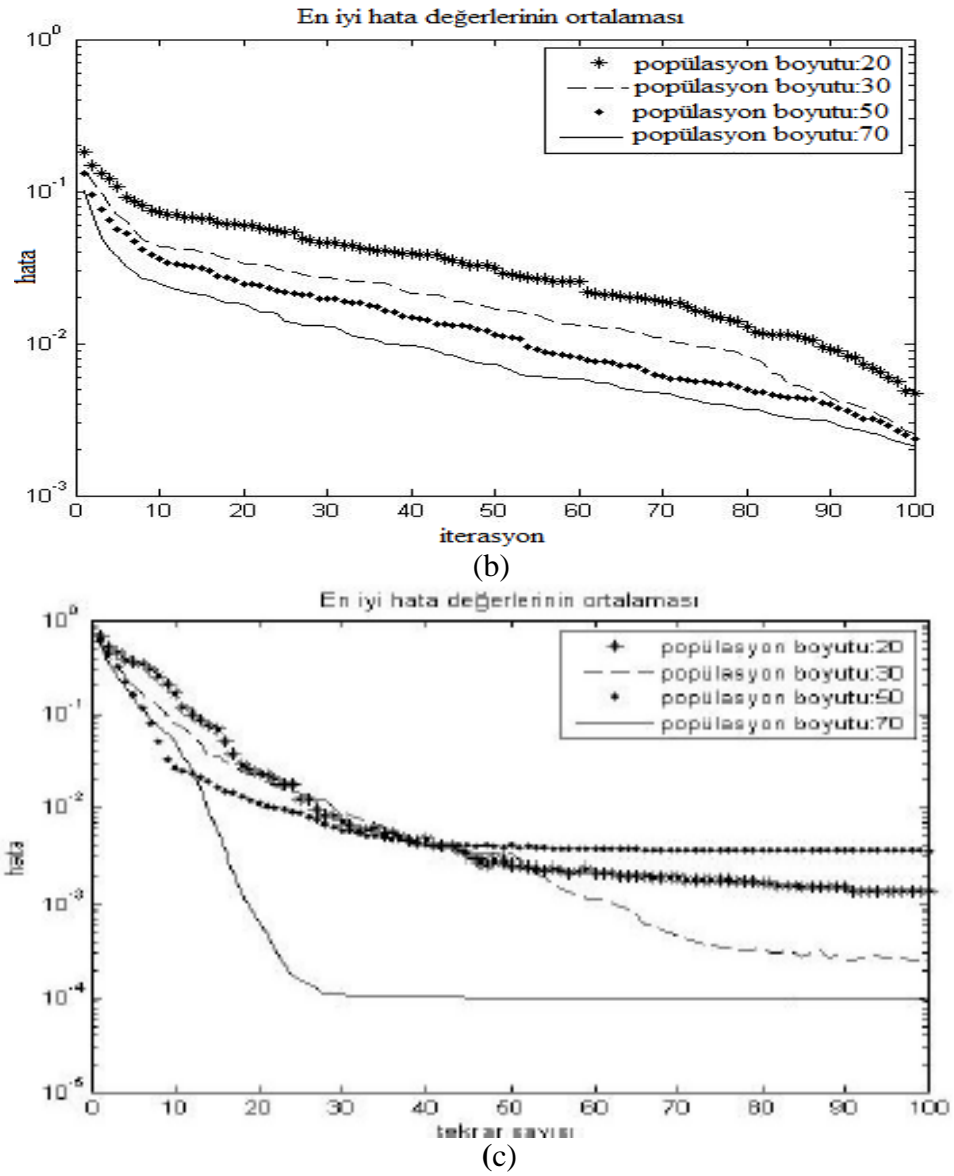
Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.3'te belirtilmiştir.

Tablo 3. 3. Denklem 3.2'de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	3.2720	2.4825	3.86
MSE	8.897×10^{-4}	3.80×10^{-3}	1.2×10^{-3}
Standart sapma	5.772×10^{-4}	3.60×10^{-3}	1×10^{-1}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	3.7384	3.6515	7.1880
MSE	2.070×10^{-4}	4.347×10^{-4}	1.95×10^{-4}
Standart sapma	3.040×10^{-4}	1.60×10^{-3}	3.8×10^{-3}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	6.0673	5.7881	16.9484
MSE	7.905×10^{-5}	1.10×10^{-3}	3×10^{-3}
Standart sapma	2.164×10^{-4}	1.40×10^{-3}	2.6×10^{-3}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	8.451	8.1558	31.0797
MSE	1.064×10^{-4}	7.843×10^{-4}	7.04×10^{-5}
Standart sapma	1.835×10^{-4}	1.60×10^{-3}	1.9×10^{-3}

Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.2'de verilmiştir.





Şekil 3. 2. Denklem 3.1'deki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

3.2.2.2. Örnek Sistem 2

İkinci örnek için doğrusal olarak bilinen Denklem 3.3'teki sistem ele alınmış ve algoritmaların bu sisteme ait parametreleri bulma sonuçları karşılaştırılmıştır [80]. Bu sistem içinde Denklem 3.4'teki model yapısı önerilmiştir. Model parametrelerinin tahmini amacıyla parametre kestirim çalışmalarında giriş verisi $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi $n=100$ adet veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Tablo 3.4'te algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşurulmuş ve algoritmaların bulduğu parametreler Tablo 3.5'te

verilmiştir. Tablo 3.5'teki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.3'te verilmiştir.

$$y(n) = 0.6282 u(n) - 1.2 u(n-1) + 0.6282 u(n-2) + 1.2 u(n-1) - 0.2563 y(n-2) \quad (3.3)$$

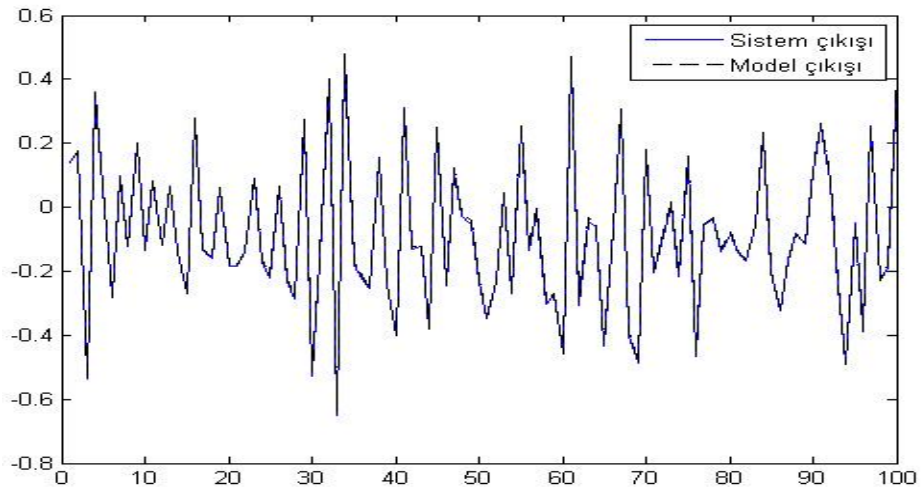
$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) + b_1 y(n-1) + b_2 y(n-2) \quad (3.4)$$

Tablo 3. 4. Denklem 3.4'te önerilen modele ait kullanılan kontrol parametreleri

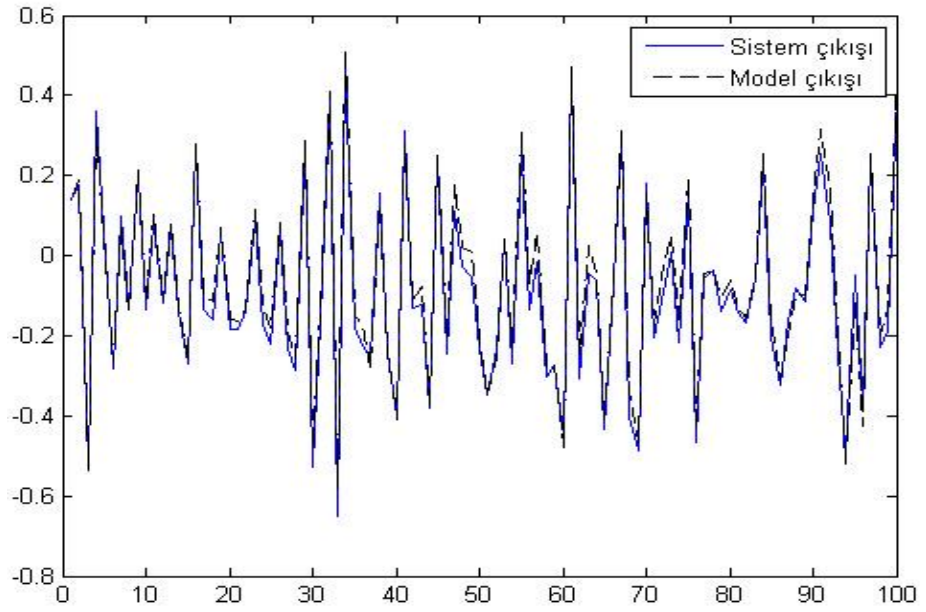
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-2 +2	-2 +2	-2 +2
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	200	200	200

Tablo 3. 5. Denklem 3.4'te önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

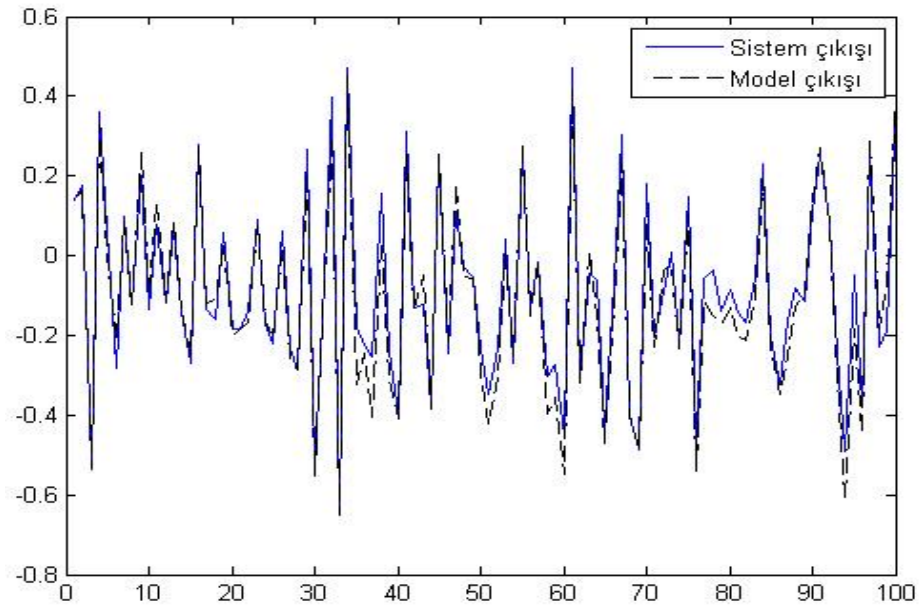
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a_0	0.6282	0.6309	0.6438	0.6089
a_1	-1.2	-1.1387	-0.9028	-1.3993
a_2	0.6282	0.5622	0.2631	0.5938
b_1	1.2	1.1110	0.7775	1.5000
b_2	-0.2563	-0.2168	-0.0070	4.76×10^{-7}
Ortalama süre (sn)		7.9877	7.0237	15.6216
MSE		4.206×10^{-5}	1.3×10^{-3}	3.1×10^{-3}
Standart sapma		2.249×10^{-4}	3.2×10^{-3}	5.4×10^{-3}



(a)



(b)



(c)

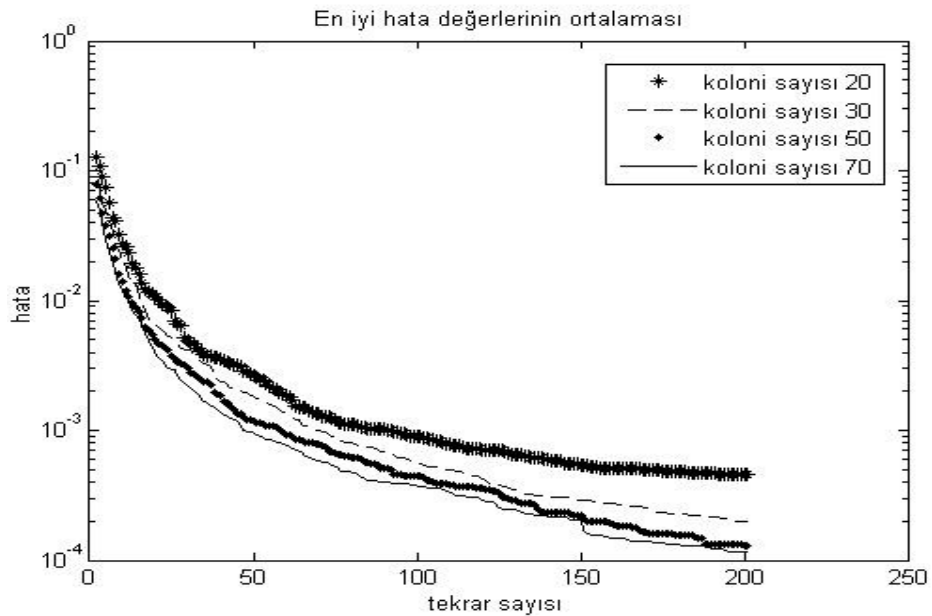
Şekil 3. 3. Denklem 3.4'deki model için sistem ve model çıktıları:
(a) yak algoritması, (b) ga, (c) ksa

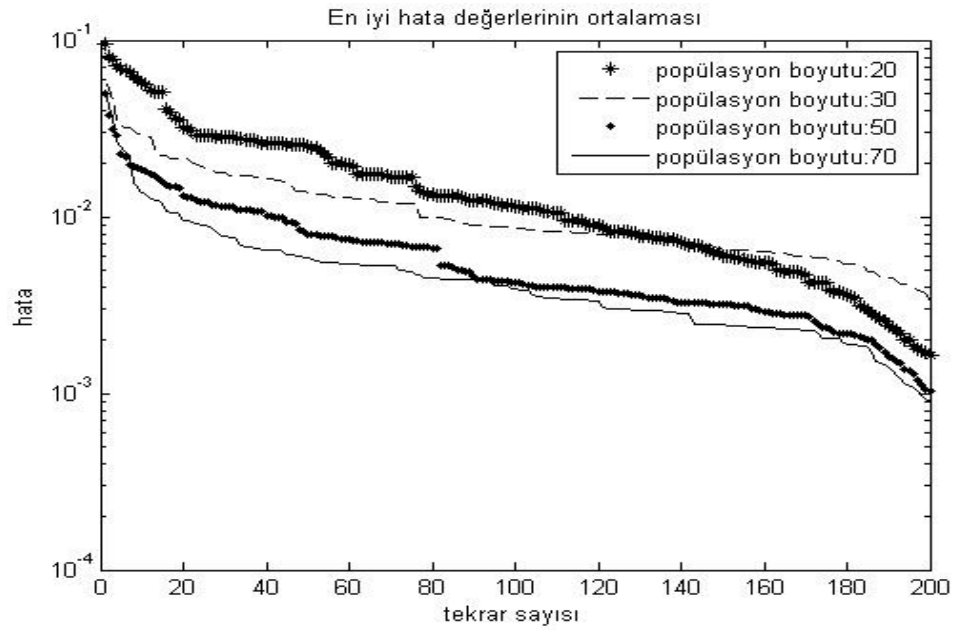
Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.6'da belirtilmiştir.

Tablo 3. 6. Denklem 3.4'te önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

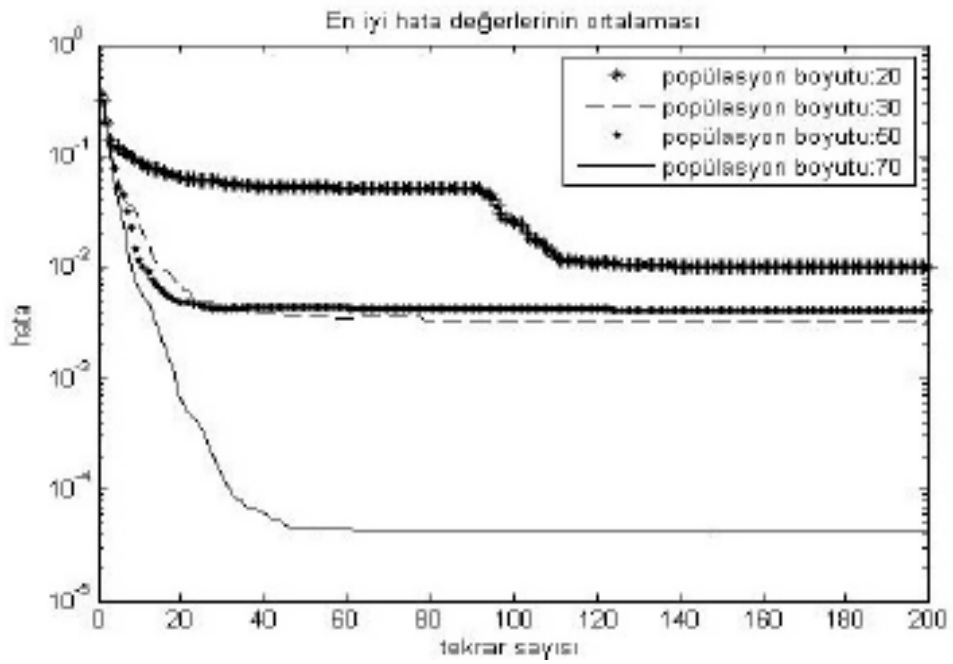
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	5.5362	4.8944	8.4847
MSE	6.790×10^{-5}	2.6×10^{-3}	9.7×10^{-3}
Standart sapma	6.483×10^{-4}	5.3×10^{-3}	1.01×10^{-2}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	7.9877	7.0237	15.6216
MSE	4.206×10^{-5}	1.3×10^{-3}	3.1×10^{-3}
Standart sapma	2.249×10^{-4}	3.2×10^{-3}	5.4×10^{-3}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	13.3755	11.7149	39.5002
MSE	1.747×10^{-4}	4.2×10^{-3}	3.7×10^{-3}
Standart sapma	1.293×10^{-4}	1.6×10^{-3}	2.1×10^{-3}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	8.451	15.8044	70.7394
MSE	3.652×10^{-5}	1.7×10^{-3}	3.25×10^{-5}
Standart sapma	1.563×10^{-4}	9.888×10^{-4}	2.8×10^{-3}

Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmaların hata yaklaşım eğrileri Şekil 3.4'te verilmiştir.





(b)



(c)

Şekil 3. 4. Denklem 3.4'teki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

Bu örnekte sisteme ait parametre sayısının 5 adet olmasına rağmen ilk örnekte 7 parametreye sahip sistem modellenmesi ile karşılaştırıldığında algoritmaların parametre bulma süreleri, sistemlerin parametre sayısı ile doğru orantılı olmadığı görülmüştür. Yani daha az parametreye sahip sistemler modellenirken, algoritmaların daha erken sürede parametre bulacağı düşünülmemelidir.

3.2.2.3. Örnek Sistem 3

Üçüncül olarak Denklem 3.5'teki başka bir doğrusal sistem ele alınmış ve algoritmaların bu sisteme ait parametreleri bulma sonuçları karşılaştırılmıştır [80]. Sistem için Denklem 3.6'daki model önerilmiştir. Model parametrelerinin tahmini amacıyla parametre kestirim çalışmalarında giriş verisi $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi $n=100$ adet veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Tablo 3.7'de algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulduğu parametreler Tablo 3.8'de verilmiştir. Tablo 3.8'deki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.5'te verilmiştir.

$$y(n) = u(n) - 1.051u(n-1) + 0.07818 u(n-2) + 0.05164 u(n-3) + 0.5322 u(n-4) - 0.5735 u(n-5) + 0.8713y(n-1) - 1.539y(n-2) - 1.3713y(n-3) - 0.645y(n-4) + 0.5827y(n-5) \quad (3.5)$$

$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) + a_3 u(n-3) + a_4 u(n-4) + a_5 u(n-5) + b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3) + b_4 y(n-4) + b_5 y(n-5) \quad (3.6)$$

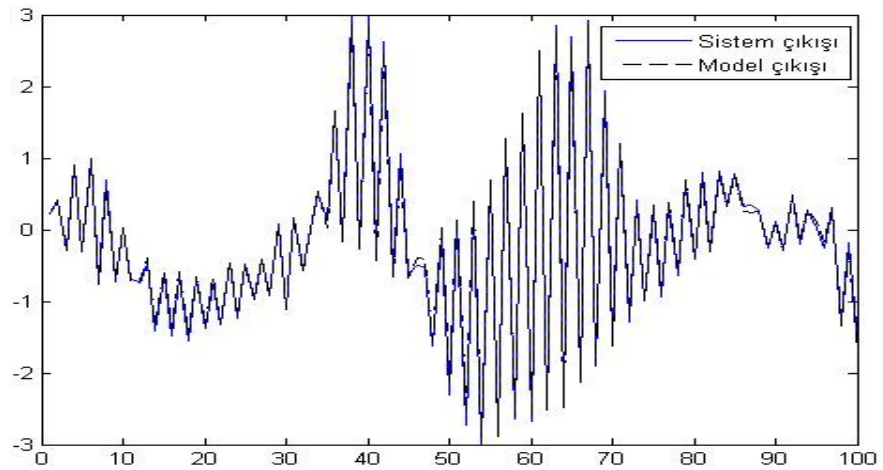
Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.9'da belirtilmiştir.

Tablo 3. 7. Denklem 3.6'te önerilen modele ait kullanılan kontrol parametreleri

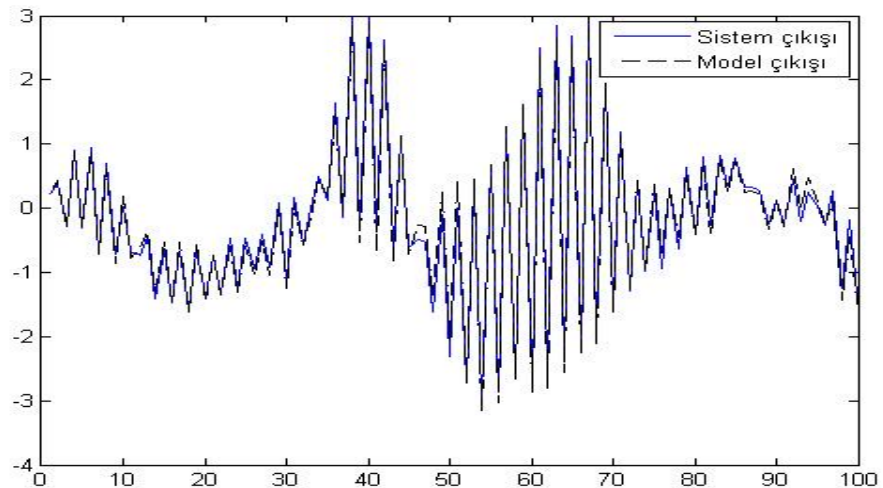
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-2 +2	-2 +2	-2 +2
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	200	200	200

Tablo 3. 8. Denklem 3.6'da önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

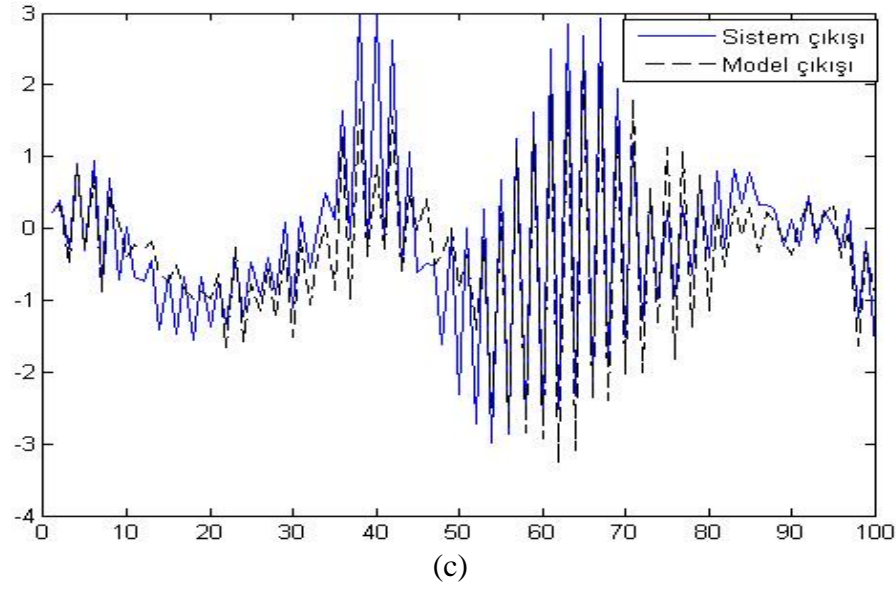
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a_0	0,8713	1.1887	1.2206	-0.5000
a_1	1,539	1.3941	1.3927	-2.38×10^{-6}
a_2	-1,371	-2.0000	-2.2637	0.7891
a_3	-0,645	-0.5279	-0.4750	0.6484
a_4	0,5827	0.9096	1.1467	-0.4844
a_5	1	1.0279	1.0528	1.0000
b_1	-1,051	-1.3536	-1.3370	-2.01×10^{-4}
b_2	0,0718	0.2515	0.2813	1.0482
b_3	0,05164	0.1158	0.4255	-0.4645
b_4	0,5322	0.6299	0.4258	1.2329
b_5	-0,5735	-0.6442	-0.6798	0.2048
Ortalama süre (sn)		8.6201	8.9011	16.4096
MSE		6×10^{-3}	2.14×10^{-2}	3.322×10^{-1}
Standart sapma		1.86×10^{-2}	3.44×10^{-2}	1.130×10^{-1}



(a)



(b)

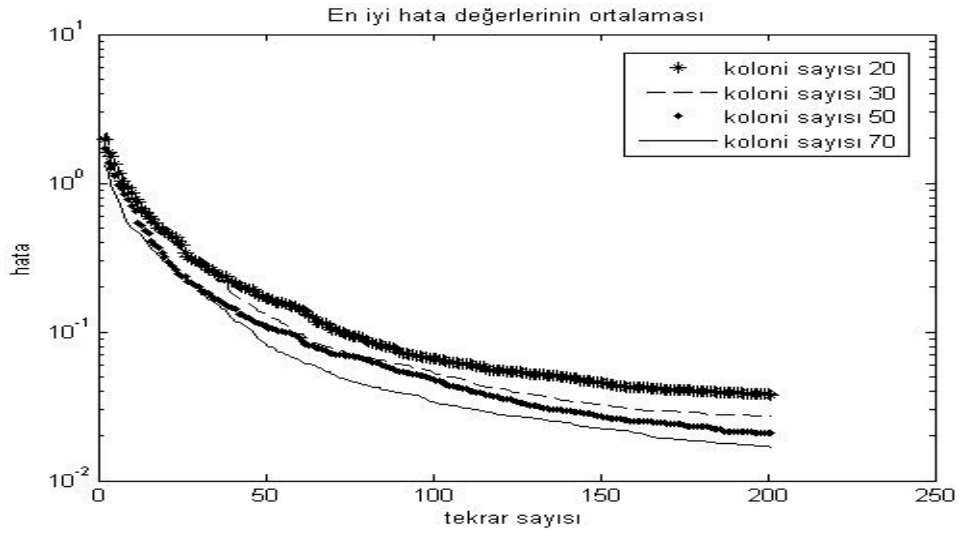


Şekil 3. 5. Denklem 3.6'daki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa

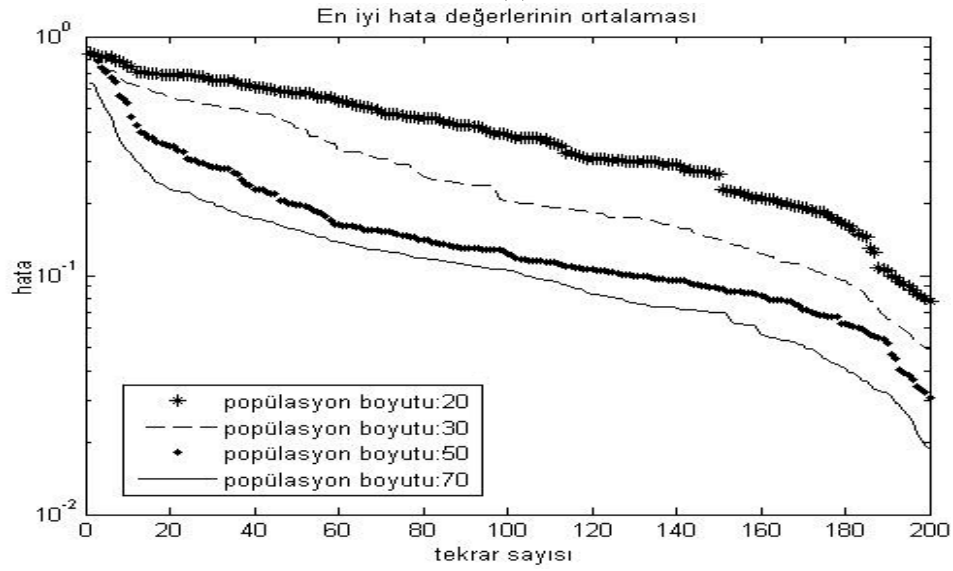
Tablo 3. 9. Denklem 3.6'da önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	6.6887	5.7720	10.2240
MSE	7.49×10^{-2}	1.009×10^{-1}	3.20×10^{-2}
Standart sapma	2.55×10^{-2}	6.73×10^{-2}	1.632×10^{-1}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	8.6201	8.9011	16.4096
MSE	6×10^{-3}	2.14×10^{-2}	3.322×10^{-1}
Standart sapma	1.86×10^{-2}	3.44×10^{-2}	1.130×10^{-1}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	17.2975	15.3995	40.5230
MSE	2.48×10^{-2}	2.87×10^{-2}	5.99×10^{-2}
Standart sapma	1.06×10^{-2}	1.84×10^{-2}	4.61×10^{-2}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	21.0065	25.4497	73.4358
MSE	2.58×10^{-2}	7.36×10^{-2}	1.73×10^{-2}
Standart sapma	1.16×10^{-2}	1.59×10^{-2}	3.73×10^{-2}

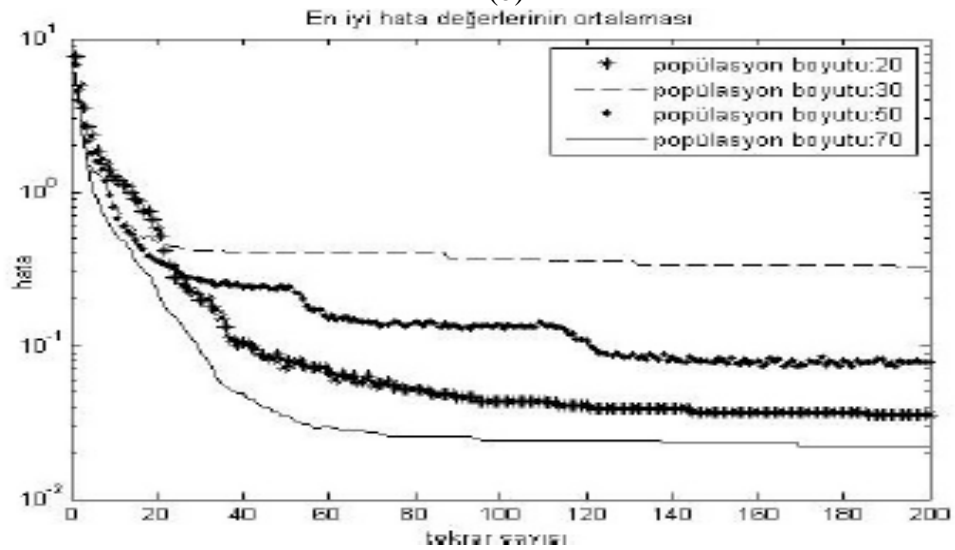
Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.6'da verilmiştir.



(a)



(b)



(c)

Şekil 3. 6. Denklem 3.6'daki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

Bu örnekte de 11 adet parametreye sahip sistem modellenmiş ve algoritmaların parametreleri bulma süreleri incelendiğinde ilk iki örnekteki gibi bu sürelerin parametre sayısı ile doğru orantılı olmadığı görülmüştür. Algoritmaların parametre bulma süreleri sistemin özelliklerine göre değişmektedir.

3.2.2.4. Örnek Sistem 4

Son olarak doğrusal olarak bilinen Denklem 3.7'deki sistem ele alınmış ve algoritmaların bu sisteme ait parametreleri bulma sonuçları karşılaştırılmıştır [21]. Bu sistem içinde Denklem 3.8'deki model yapısı önerilmiş ve önceki örneklerde olduğu gibi bu örnekte de model parametrelerinin tahmini amacıyla parametre kestirim çalışmalarında giriş verisi $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi $n=100$ adet veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Tablo 3.10'da algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulunduğu parametreler Tablo 3.11'de verilmiştir. Tablo 3.11'deki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.7'de verilmiştir.

$$y(n) = u(n) - 0.002 u(n-1) + 0.002 u(n-2) + 0.08 u(n-3) + 0.5 y(n-1) + 0.29 y(n-2) - 0.103 y(n-3) \quad (3.7)$$

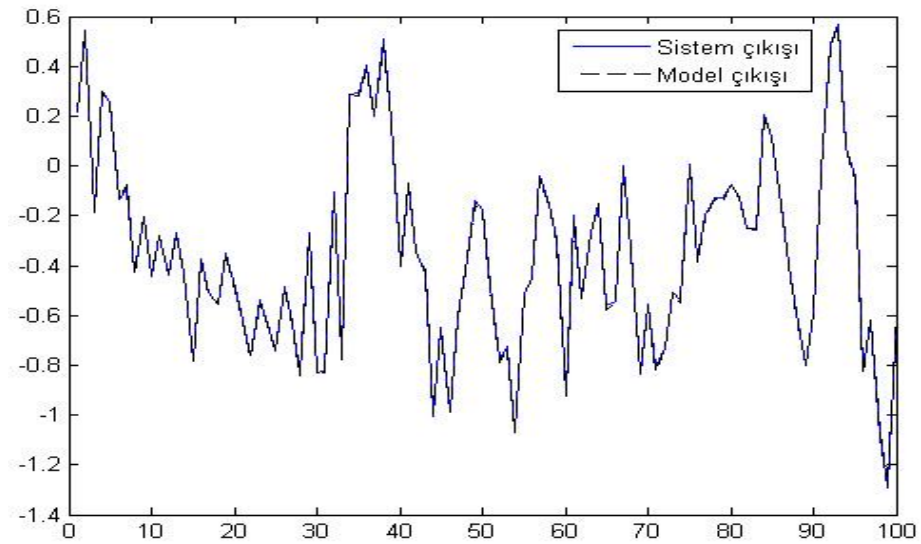
$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) + a_3 u(n-3) + b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3) \quad (3.8)$$

Tablo 3. 10. Denklem 3.8'de önerilen modele ait kullanılan kontrol parametreleri

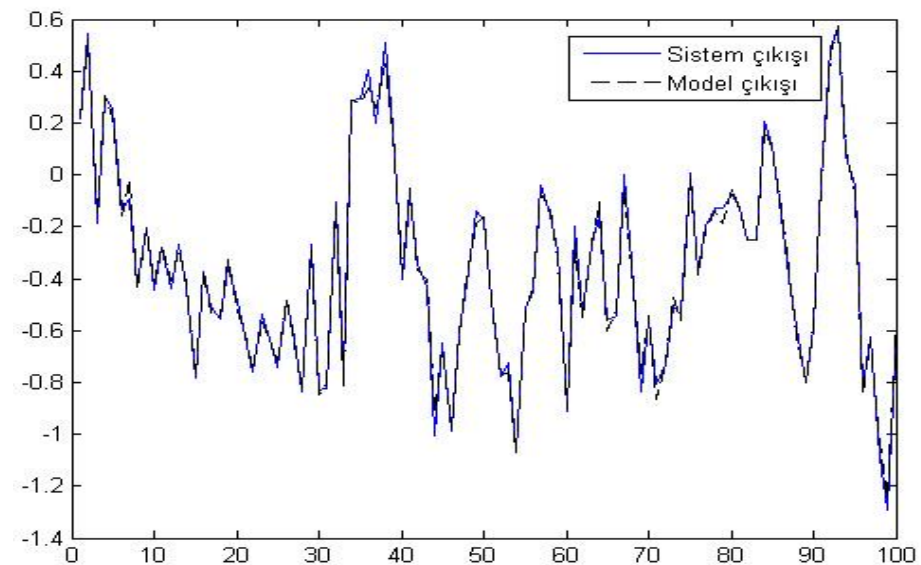
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-1 +1	-1 +1	-1 +1
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	200	200	200

Tablo 3. 11. Denklem 3.8’de önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

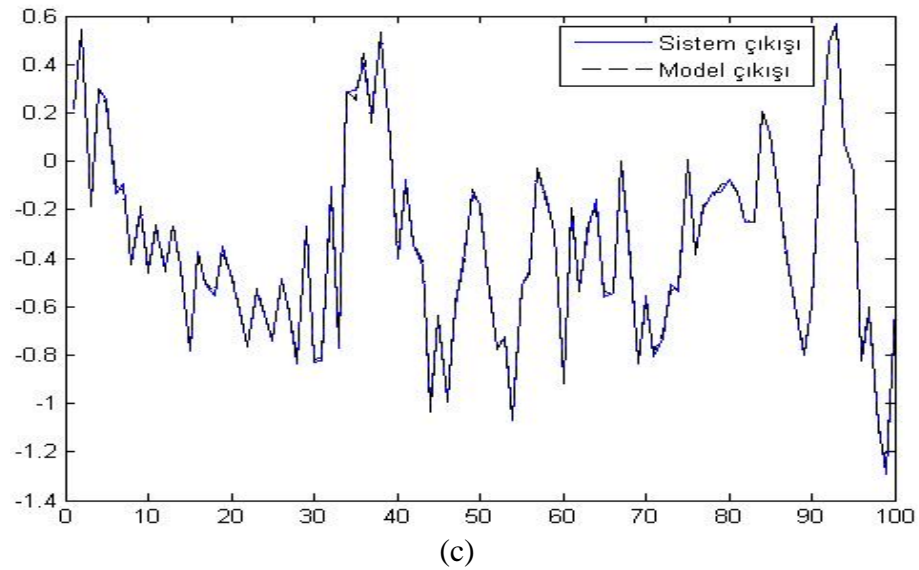
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a_0	1	0.9937	0.9506	1
a_1	0.002	0.0928	0.3246	0.1250
a_2	-0.002	0.1493	0.5963	-0.2656
a_3	0.08	0.0290	-0.0492	0.1250
b_1	0.5	0.4122	0.1739	0.3743
b_2	0.29	0.1858	-0.1213	0.6309
b_3	-0.103	0.0477	0.4294	-0.2949
Ortalama süre (sn)		7.8073	7.1729	16.0735
MSE		7.117×10^{-5}	9.218×10^{-4}	3.223×10^{-4}
Standart sapma		4.170×10^{-5}	4.562×10^{-4}	2.6×10^{-3}



(a)



(b)

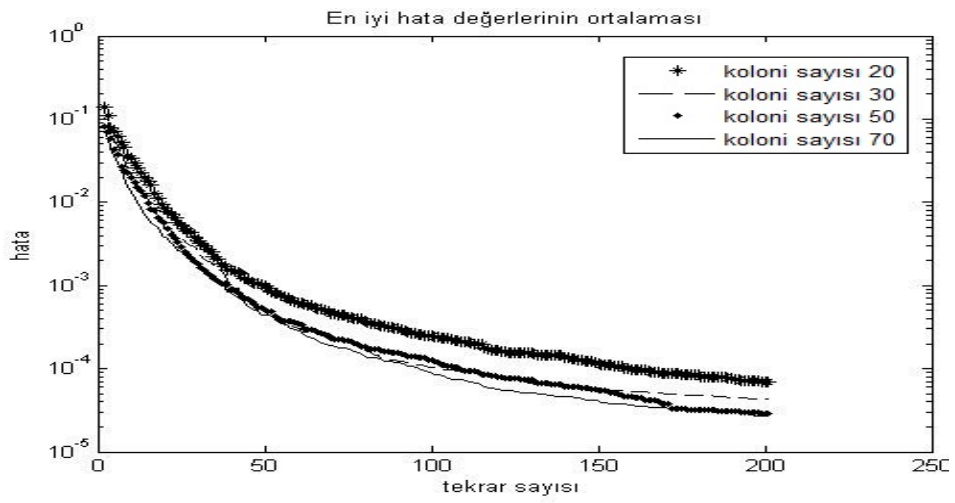


Şekil 3. 7. Denklem 3.8'deki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa

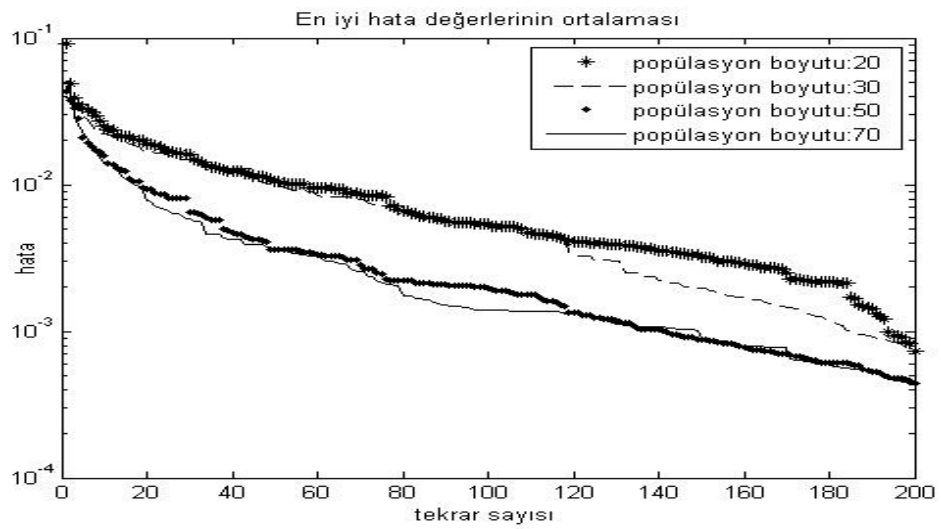
Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.12'de belirtilmiştir. Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.8'de verilmiştir.

Tablo 3. 12. Denklem 3.8'de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

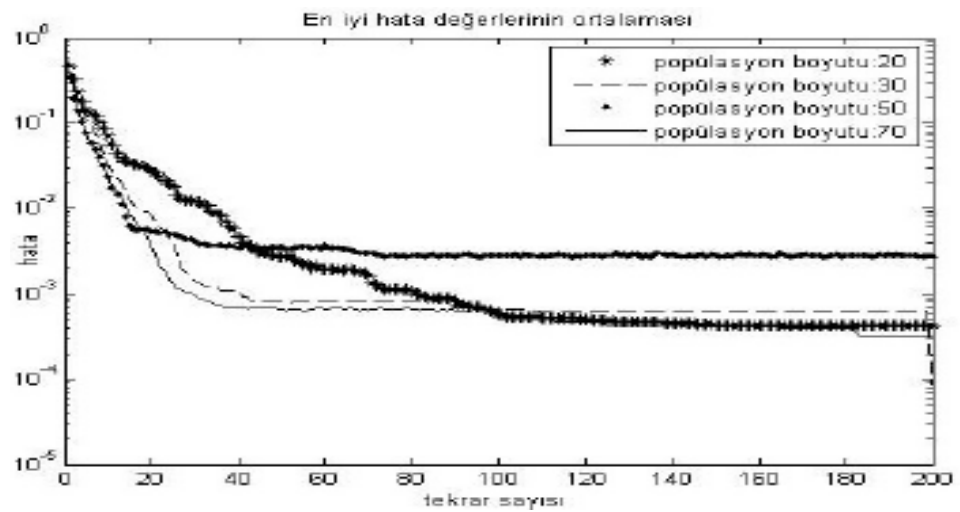
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	5.5567	4.8979	8.6749
MSE	2.827×10^{-5}	2.649×10^{-4}	3.903×10^{-4}
Standart sapma	6.066×10^{-5}	5.487×10^{-4}	3×10^{-3}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	7.8073	7.1729	16.0735
MSE	7.117×10^{-5}	9.218×10^{-4}	3.223×10^{-4}
Standart sapma	4.170×10^{-5}	4.562×10^{-4}	2.6×10^{-3}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	13.6074	11.7619	38.6363
MSE	2.340×10^{-5}	9.357×10^{-5}	2×10^{-3}
Standart sapma	2.232×10^{-5}	5.900×10^{-4}	7.177×10^{-4}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	19.2789	16.1230	63.5264
MSE	3.329×10^{-5}	2.531×10^{-4}	4.680×10^{-4}
Standart sapma	2.070×10^{-5}	4.870×10^{-4}	5.572×10^{-4}



(a)



(b)



(c)

Şekil 3. 8. Denklem 3.8'deki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

Kullanılan doğrusal parametrik sistemlerin parametre kestirim değerleri incelendiğinde, YAK algoritmasının GA'ya oldukça yakın bir sürede ve daha düşük bir hata ile çok başarılı bir şekilde parametre kestirimi yaptığı görülmüştür. Bu kestirimi yaparken de GA'dan daha düşük bir hata ile yapmasında YAK algoritmasının doğrusal parametrik sistemlerde parametre kestiriminde başarılı bir algoritma olduğunun göstergesidir. Aynı zamanda hata yaklaşım eğrilerinden de anlaşılacağı üzere hatayı minimize edip optimum çözüme kararlı bir şekilde yaklaşan algoritma YAK algoritmasıdır. Farklı koloni ve popülasyon sayılarında ise hem GA'dan hemde KSA'dan daha düşük bir MSE ile parametre kestirimi yaptığı görülmüştür. Parametre sayılarının ise algoritmaların parametreleri bulma sürelerini doğru orantılı olarak etkilemediği görülmüştür.

3.2.3. Doğrusal Olmayan Parametrik Sistem Modellemesi Uygulamaları

3.2.3.1. Örnek Sistem 1: Parametrik Doğrusal Olmayan Volterra Sistem

Bu ilk örnekte doğrusal olmayan Volterra sistemin modellendirilmesi amacıyla matematiksel ifadesi Denklem 3.9' da verilen sistem ele alınmıştır [81]. Bu sistem için Denklem 3.10' da verilen model yapısı önerilmiştir.

$$y(n) = -0.64 u(n) + 0.89 u(n-1) - 0.95 u(n-2) - 1.4 u^2(n-1) + 1.06 u(n-1)u(n-2) - 0.5 u^3(n) + u^3(n-1) - 1.2 u^3(n-2) \quad (3.9)$$

$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) + b_{2,2} u^2(n-1) + b_{1,2} u(n-1)u(n-2) + b_{0,0,0} u^3(n) + b_{1,1,1} u^3(n-1) + b_{2,2,2} u^3(n-2) \quad (3.10)$$

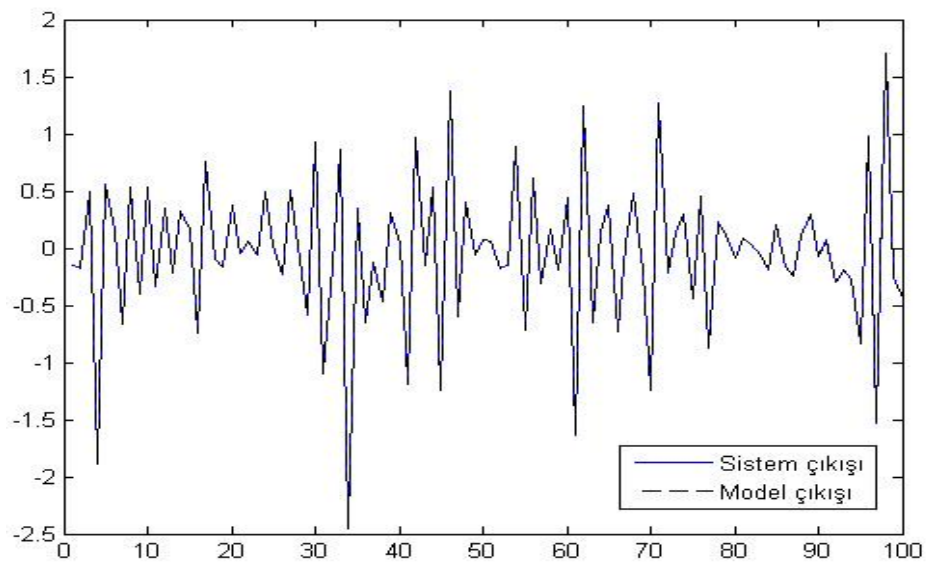
Model parametrelerinin tahmini amacıyla yapılan benzetim çalışmalarında giriş verisi, $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi 100 veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Tablo 3.13'te algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulduğu parametreler Tablo 3.14'te verilmiştir. Tablo 3.14'teki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.9'da verilmiştir.

Tablo 3. 13. Denklem 3.10’da önerilen modele ait kullanılan kontrol parametreleri

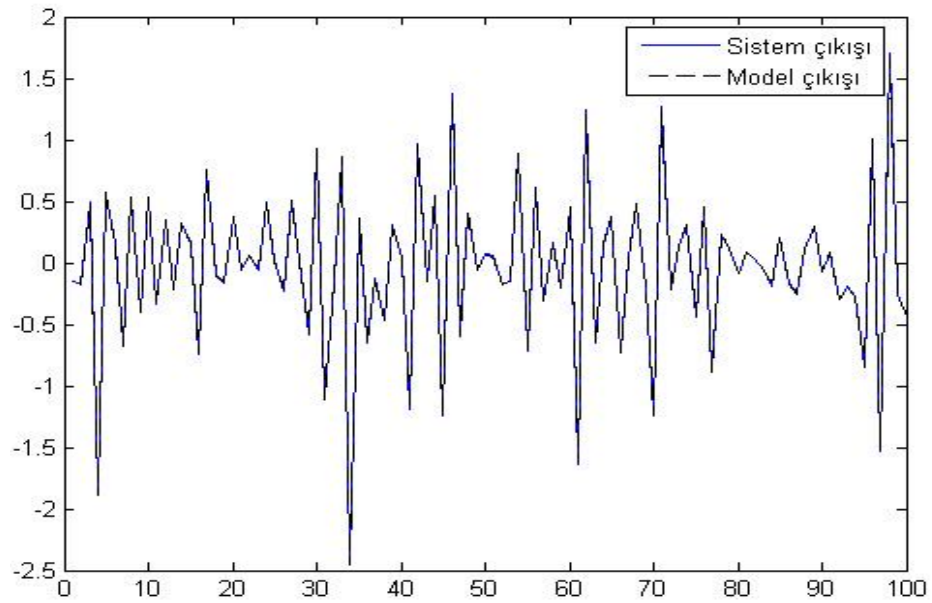
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-2 +2	-2 +2	-2 +2
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	500	500	500

Tablo 3. 14. Denklem 3.10’da önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

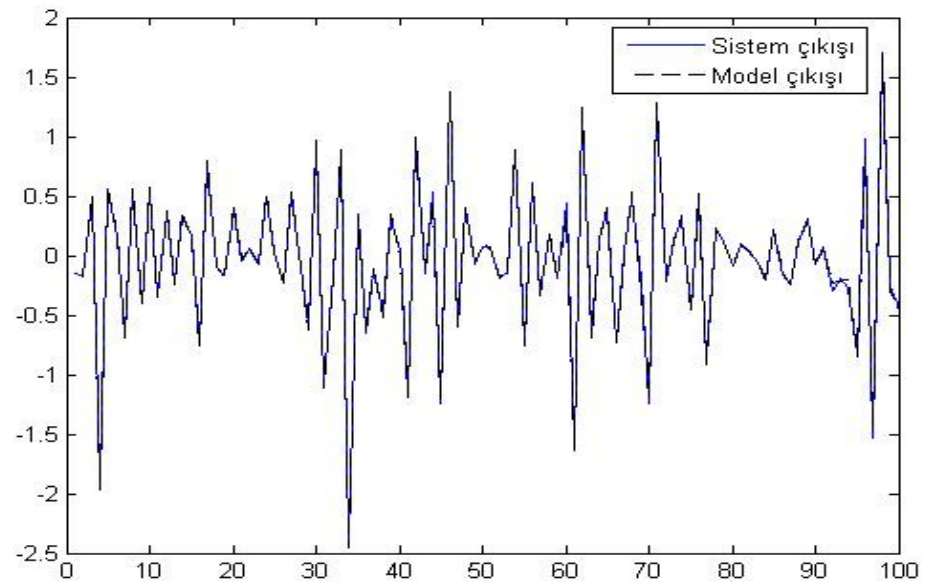
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a₀	-0.64	-0.6430	-0.6077	-0.7812
a₁	0.89	0.8910	0.8863	1.0000
a₂	-0.95	-0.9497	-0.9963	-1.0000
b_{2,2}	-1.4	-1.4004	-1.4024	-1.4066
b_{1,2}	1.06	1.0604	1.0401	1.0937
b_{0,0,0}	-0.5	-0.4915	-0.5827	4.76x10 ⁻⁷
b_{1,1,1}	1	0.9955	0.9988	0.7460
b_{2,2,2}	-1.2	-1.2006	-1.0363	-1.0657
Ortalama süre (sn)		27.7352	20.7191	44.6946
MSE		3.692x10 ⁻⁷	1.317x10 ⁻⁴	1.3x10 ⁻³
Standart sapma		5.463x10 ⁻⁷	1.368x10 ⁻⁴	3.6x10 ⁻³



(a)



(b)



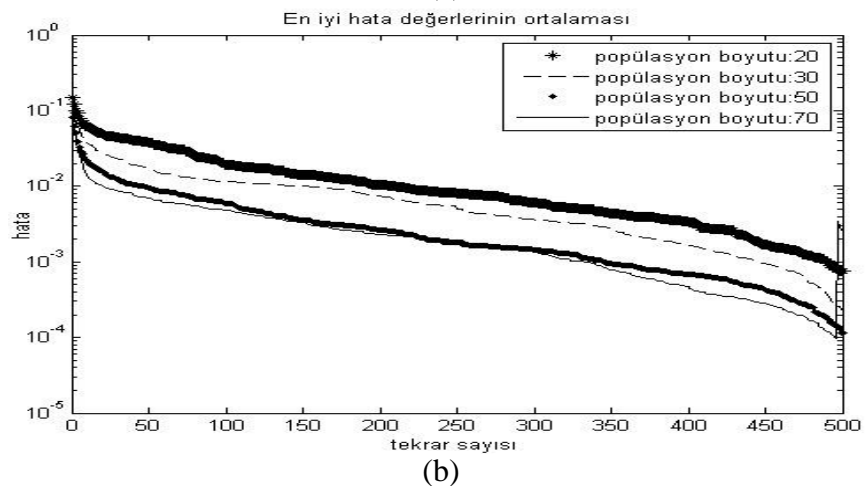
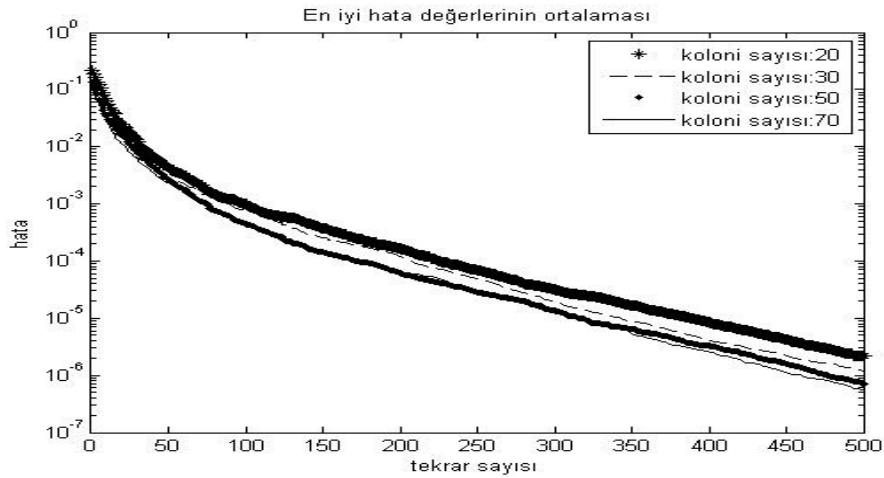
(c)

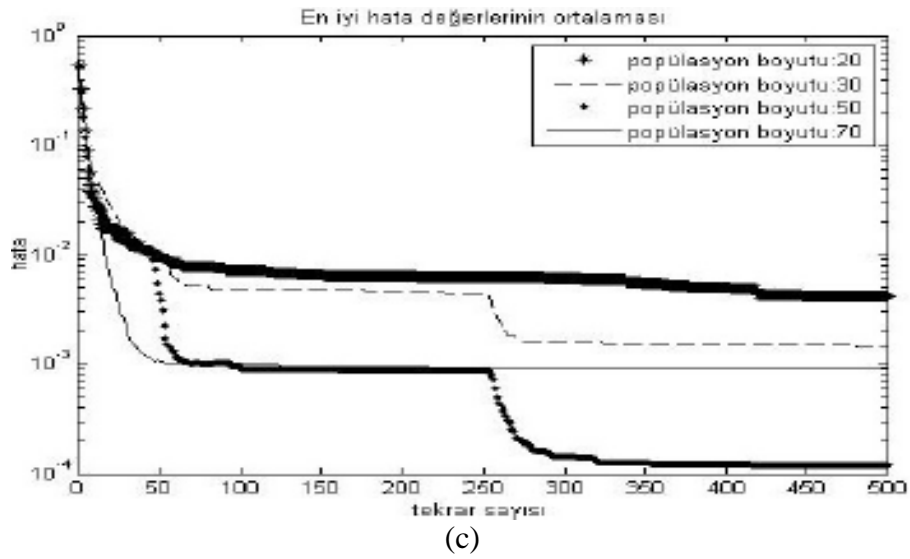
Şekil 3. 9. Denklem 3.10'daki model için sistem ve model çıktıları: (a) yak algoritması, (b) ga, (c) ksa

Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.15'te belirtilmiştir. Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.10'da verilmiştir.

Tablo 3. 15. Denklem 3.10'da önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	16.9846	14.4010	22.2874
MSE	2.949×10^{-6}	1.466×10^{-4}	6.1×10^{-3}
Standart sapma	1.976×10^{-6}	3.238×10^{-4}	5.3×10^{-3}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	27.7352	20.7191	44.6946
MSE	3.692×10^{-7}	1.317×10^{-4}	1.3×10^{-3}
Standart sapma	5.463×10^{-7}	1.368×10^{-4}	3.6×10^{-3}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	41.0308	37.3625	106.3142
MSE	4.196×10^{-7}	5.027×10^{-5}	9.244×10^{-5}
Standart sapma	5.531×10^{-7}	8.394×10^{-5}	3.8×10^{-3}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	49.5368	48.6560	179.5130
MSE	9.675×10^{-7}	1.185×10^{-5}	7.342×10^{-4}
Standart sapma	4.542×10^{-7}	3.809×10^{-5}	1.4×10^{-3}





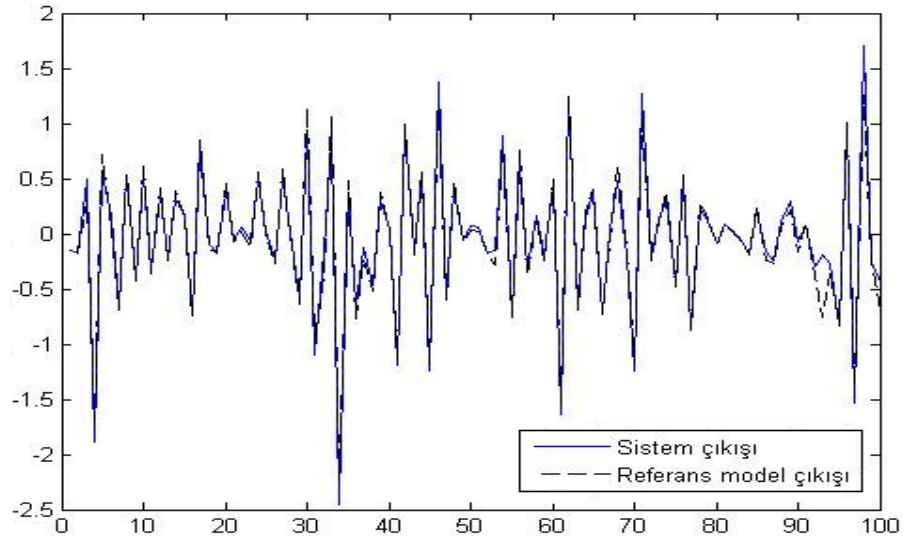
Şekil 3. 10. Denklem 3.10'daki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

Denklem 3.9 için önerilen Denklem 3.10'daki model yapısı, dikkate alınan sistemle aynı olması sebebiyle algoritmalar yardımıyla model parametrelerinin oldukça küçük bir hata ile elde edebildiği görülmektedir. Ancak farklı bir model yapısının seçilmesi durumunda MSE değerinin artması olasılığının bulunduğunu gözden kaçırmamak gerekir. Bu amaçla literatürde de uygulanmış olan, matematiksel ifadesi Denklem 3.9'da verilen sistem için Zorlu ve Özer'in önerdiği ve farklı algoritmalarla yapmış oldukları, Denklem 3.11'de verilen daha az parametreye sahip farklı bir model yapısı kullanılmış olup, YAK algoritmasının performansı incelenmiştir [77,81]. Algoritmaların buldukları MSE değerleri Tablo 3.16'da gösterilmiştir. Ayrıca referans çalışma için YAK algoritmasının hesapladığı parametrelere göre sistem çıkışı ile Denklem 3.11'de verilen model çıkışı Şekil 3.11'de verilmiştir.

$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2) + b_{1,1} u^2(n-1) + b_{0,0,0} u^3(n) \quad (3.11)$$

Tablo 3. 16. Denklem 3.11'de önerilen model için YAK Algoritmasının referans çalışma ile karşılaştırılması

Referans Çalışma	Algoritma	MSE
Zorlu ve Özer [77,81]	DGA	0.131
	GA	0.131
	KSA	0.132
	RLS	0.213
Bu çalışma	YAK	0.0131



Şekil 3. 11. Denklem 3.9’da verilen sistem çıkışı ile yak algoritmasının hesapladığı Denklem 3.11’de önerilen model çıkışı

Referans çalışma ile karşılaştırıldığında farklı model yapısının önerilmesi ile YAK Algoritmasının diğer algoritalardan daha düşük bir MSE değeri ile parametre kestirimi yaptığı görülmüştür.

3.2.3.2. Örnek Sistem 2: Parametrik Doğrusal Olmayan Volterra Sistem

Bu ilk örnekte doğrusal olmayan Volterra sistemin modellendirilmesi amacıyla matematiksel ifadesi Denklem 3.12’ de verilen sistem ele alınmıştır [82]. Bu sistem için Denklem 3.13’ te verilen model yapısı önerilmiştir.

$$y(n) = 2 u(n-6) + 5u^2(n-6) + 3.4 u(n-6) u(n-7) - 4 u^3(n-6) - 2.5u^2(n-6) u(n-7) - 3 u^4(n-6) + 3.5u^3(n-6) u(n-7) + u(n) \quad (3.12)$$

$$y(n) = a_6 u(n-6) + b_{6,6} u^2(n-6) + b_{6,7} u(n-6) u(n-7) - b_{6,6,6} u^3(n-6) - b_{6,6,7} u^2(n-6) u(n-7) - b_{6,6,6,6} u^4(n-6) + b_{6,6,6,7} u^3(n-6) u(n-7) + a_0 u(n) \quad (3.13)$$

Model parametrelerinin tahmini amacıyla yapılan benzetim çalışmalarında giriş verisi, $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi 100 veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Tablo 3.17’de algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol parametreleri

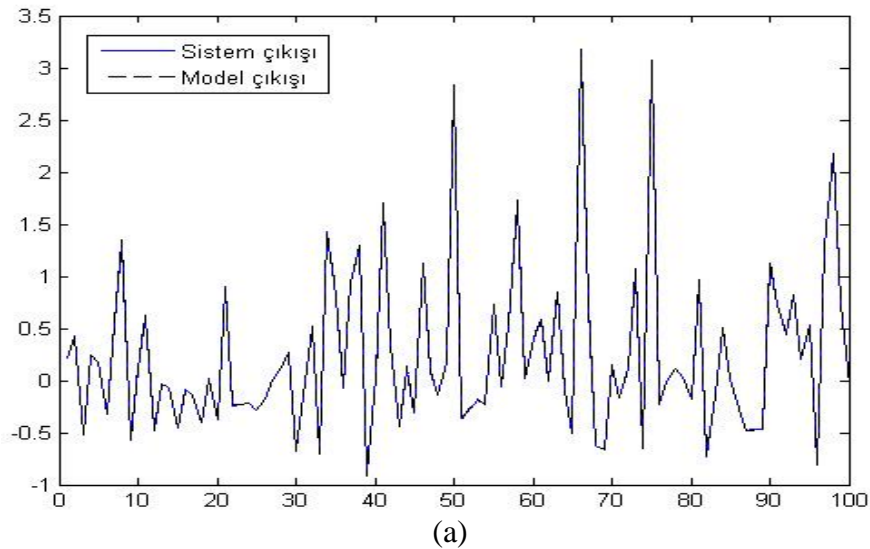
ile her bir algoritma 30 kez kořturulmuř ve algoritmaların bulduđu parametreler Tablo 3.18’de verilmiřtir. Tablo 3.18’deki hesaplanan parametrelere g6re sistem ve model ıkıřları Őekil 3.12’de verilmiřtir.

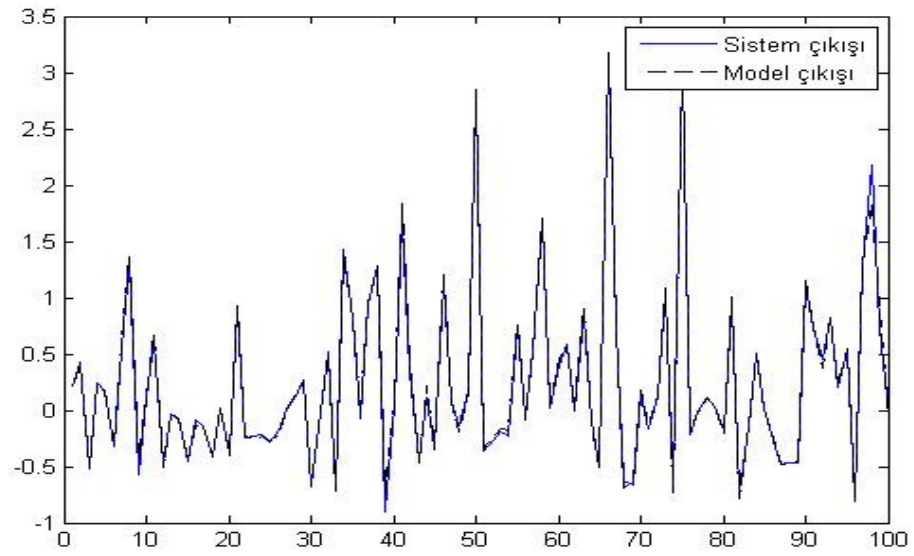
Tablo 3. 17. Denklem 3.13’te 6nerilen modele ait kullanılan kontrol parametreleri

Kontrol Parametreleri	YAK	GA	KSA
Parametre arařtırma aralıđı	-10+10	-10+10	-10+10
Pop6lasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
aprazlama oranı	-	0.8	-
İterasyon	500	500	500

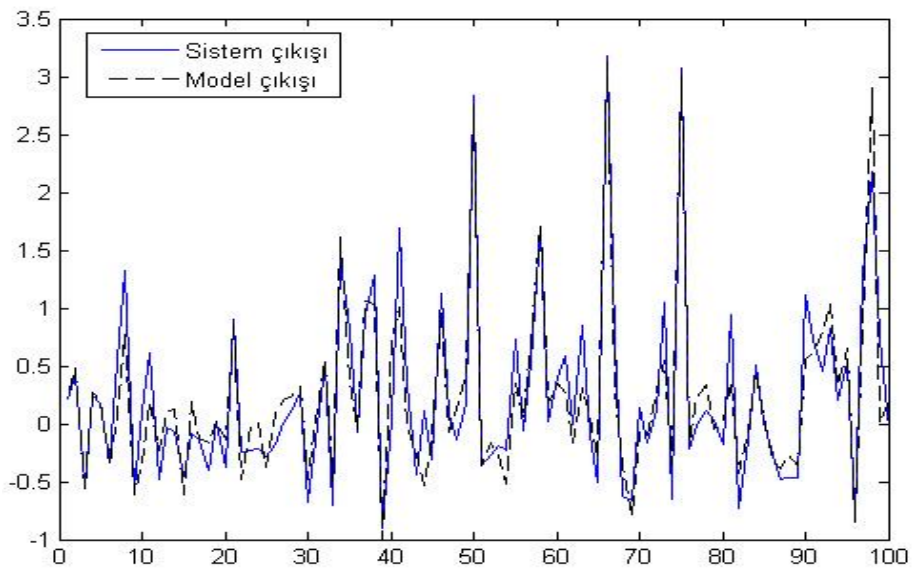
Tablo 3. 18. Denklem 3.13’te 6nerilen modele ait algoritmaların bulduđu parametreler, MSE deđerleri ve parametre bulma s6releri

Kat Sayı	Gerek Deđer	YAK	GA	KSA
a₆	2	2.0570	2.3122	-2.38x10 ⁻⁶
b_{6,6}	5	5.1736	5.3334	2.5000
b_{6,7}	3.4	3.3088	4.0552	2.5000
b_{6,6,6}	-4	-4.2063	-5.4764	2.8589
b_{6,6,7}	-2.5	-2.5420	-2.3698	-2.5702
b_{6,6,6,6}	-3	-3.5452	-4.0587	5.0000
b_{6,6,6,7}	3.5	3.8948	0.0160	7.3437
a₀	1	0.9953	0.9426	1.1182
Ortalama s6re (sn)		23.3714	23.9450	41.6232
MSE		1.421x10 ⁻⁴	3.5x10 ⁻³	7.58x10 ⁻²
Standart sapma		4.846x10 ⁻⁴	6.x10 ⁻³	5.23x10 ⁻²





(b)



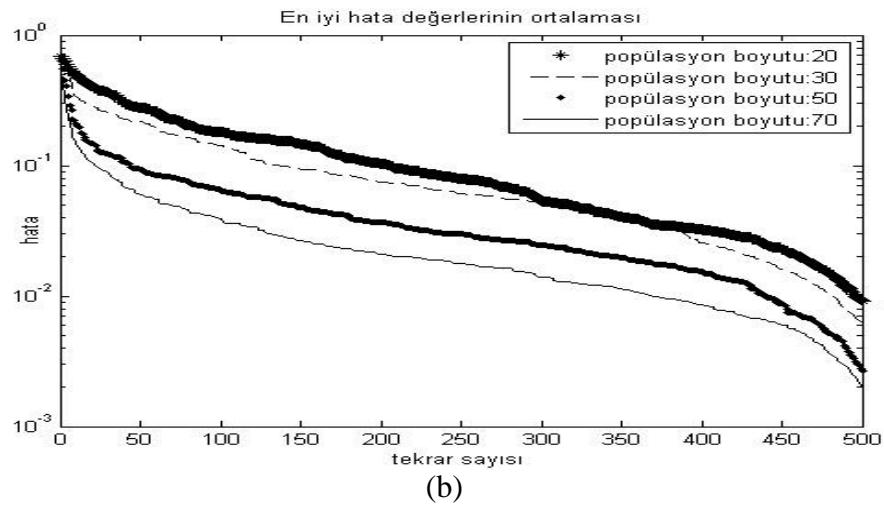
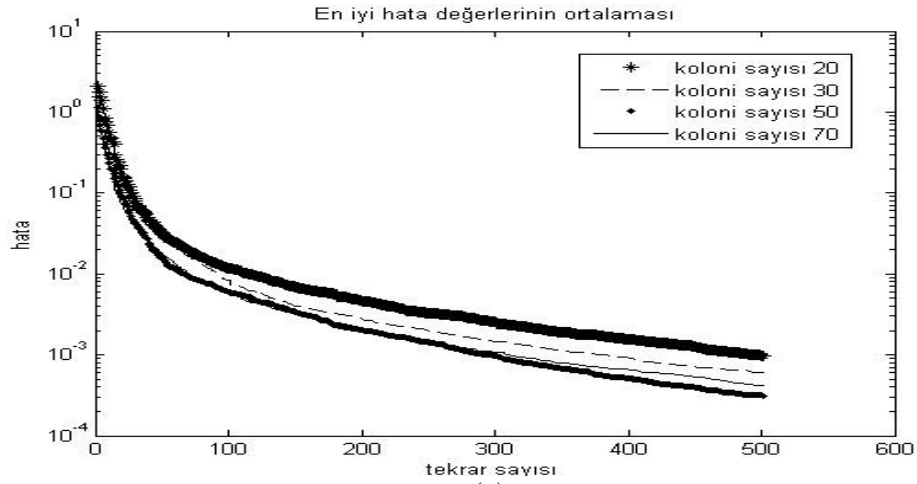
(c)

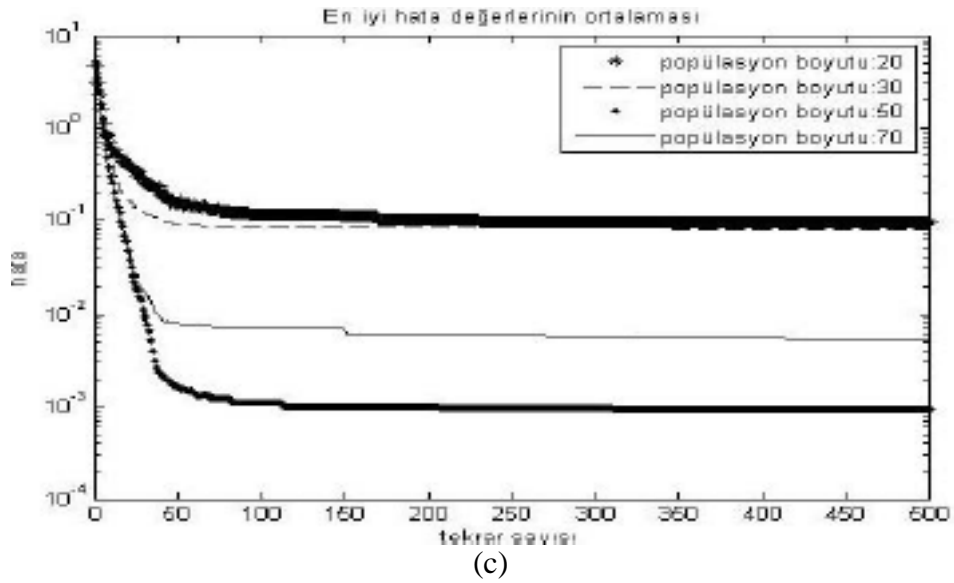
Şekil 3. 12. Denklem 3.13'teki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa

Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.19'da belirtilmiştir. Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.13'te verilmiştir.

Tablo 3. 19. Denklem 3.13'te önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	17.4868	16.7138	22.4555
MSE	1.6×10^{-3}	1.86×10^{-2}	1×10^{-1}
Standart sapma	7.380×10^{-4}	1×10^{-2}	4.82×10^{-2}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	23.3714	23.9450	41.6232
MSE	1.421×10^{-4}	3.5×10^{-3}	7.58×10^{-2}
Standart sapma	4.846×10^{-4}	$6. \times 10^{-3}$	5.23×10^{-2}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	42.0858	41.6254	102.1483
MSE	2.387×10^{-4}	4.3×10^{-3}	8.198×10^{-4}
Standart sapma	2.618×10^{-4}	2.4×10^{-3}	4.29×10^{-2}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	65.1456	54.2536	180.0238
MSE	2.102×10^{-4}	4.925×10^{-4}	4.6×10^{-3}
Standart sapma	2.804×10^{-4}	1.8×10^{-3}	3.14×10^{-2}





Şekil 3.13. Denklem 3.13'teki model için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

Bu çalışmada kullanılan Volterra sistemlerin parametre hesaplama performansları bakımından, YAK algoritmasının doğrusal olmayan sistemlerde çok başarılı bir şekilde parametre hesabı yaptığı ortadadır. Yine aynı şekilde, hata yaklaşım eğrilerinden de anlaşılacağı üzere hatayı minimize edip optimum çözüme kararlı bir şekilde yaklaşan YAK algoritmasıdır ve bu kestirimi yaparken de çok düşük bir MSE ile yapmasında YAK algoritmasının ne denli başarılı bir algoritma olduğunun göstergesidir. Parametre bulma süresi göz önüne alındığında ise doğrusal parametrik sistemde olduğu gibi YAK algoritması KSA'dan daha hızlı ve GA'ya çok yakın sürede parametre hesabı görülmüştür.

3.2.3.3. Örnek Sistem 3: Parametrik Doğrusal Olmayan Bilineer Sistem

Doğrusal olmayan diğer bir sistem örneği ise Zorlu-Özer tarafından da ele alınan ve Denklem 3.14'te verilen doğrusal olmayan parametrik bilinear sistem incelenmiştir. Bu problemde de temel amaç giriş ve çıkış bilgileri kullanılarak Denklem 3.15'te önerilen model parametrelerinin tahmin edilmesidir [83-84].

Bu model parametrelerinin tahmini amacıyla yapılan benzetim çalışmalarında giriş verisi, $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi 100 veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Modelleme çalışmasında bilinear model parametreleri, model çıkışı ile sistem çıkışı arasındaki hata minimize edilinceye

kadar YAK Algoritması , GA ve KSA tarafından optimize edilmiş ve gerçek sistem parametreleri ile karşılaştırılmıştır. Tablo 3.20’de algoritmaların parametre bulmaları için kullanılan kontrol parametreleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulduğu parametreler Tablo 3.21’de verilmiştir. Tablo 3.21’deki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.14’te verilmiştir.

$$y(n) = -0.1 y(n-1) + 0.02 y(n-2) + u(n) - 0.4 u(n-1) + 0.1 y(n-1) u(n-1) - 0.1 y(n-4) u(n-1) - 0.05 y(n-2) u(n-2) + 0.05 y(n-4) u(n-2) + 0.3 y(n-3) u(n-3) + 0.1 y(n-4) u(n-4) \quad (3.14)$$

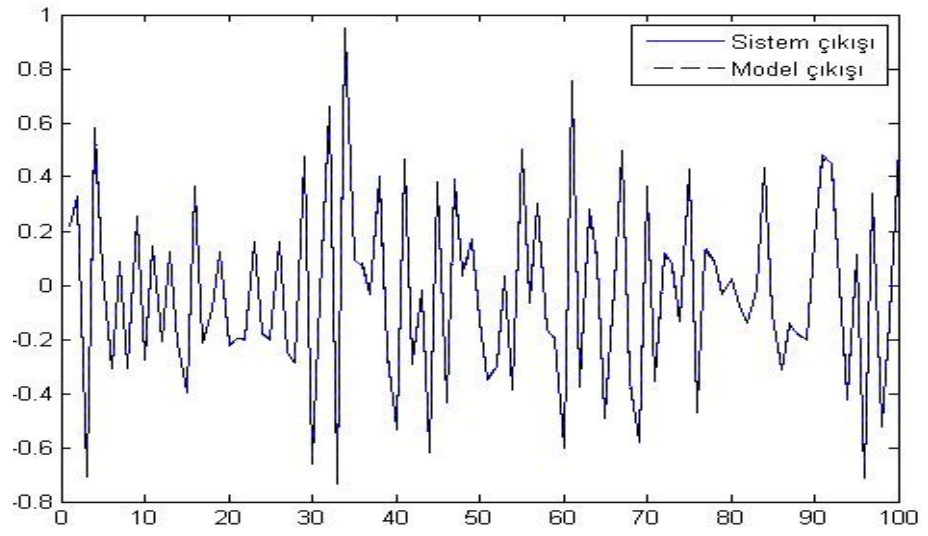
$$y(n) = -a_1 y(n-1) + a_2 y(n-2) + b_0 u(n) - b_1 u(n-1) + c_{1,1} y(n-1) u(n-1) - c_{4,1} y(n-4) u(n-1) - c_{2,2} y(n-2) u(n-2) + c_{4,2} y(n-4) u(n-2) + c_{3,3} y(n-3) u(n-3) + c_{4,4} y(n-4) u(n-4) \quad (3.15)$$

Tablo 3. 20. Denklem 3.15’te önerilen modele ait kullanılan kontrol parametreleri

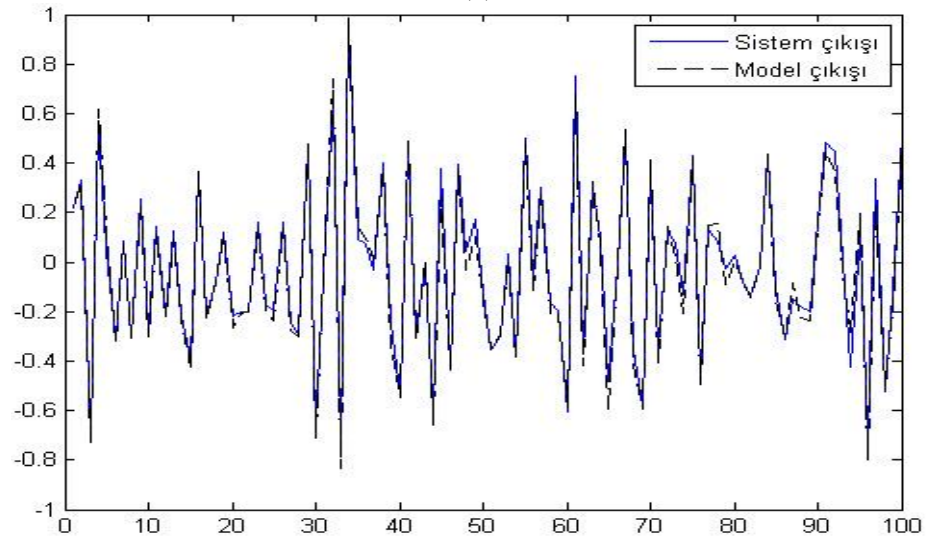
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-1+1	-1+1	-1+1
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	200	200	200

Tablo 3. 21. Denklem 3.15’te önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

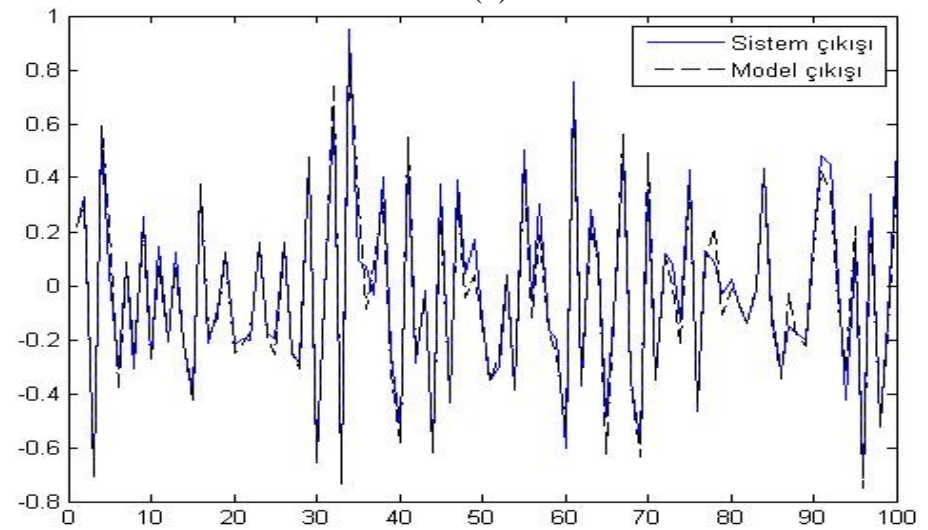
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a₁	-0.1	-0.1931	-0.8150	-1.0000
a₂	0.02	-0.0163	-0.2988	-0.5000
b₀	1	0.9953	1.0149	0.9687
b₁	-0.4	-0.3181	0.2784	0.5000
c_{1,1}	0.1	0.1014	-0.0178	0.0974
c_{4,1}	-0.1	-0.0983	0.0060	-5.48x10 ⁻⁶
c_{2,2}	-0.05	-0.0192	0.1476	0.0868
c_{4,2}	0.05	0.0571	0.0813	-5.48x10 ⁻⁶
c_{3,3}	0.3	0.2944	0.4721	0.5311
c_{4,4}	0.1	0.0874	0.2264	0.2037
Ortalama süre (sn)		8.3889	8.5825	18.8894
MSE		4.200x10 ⁻⁵	2.3x10 ⁻³	4.3x10 ⁻³
Standart sapma		3.266x10 ⁻⁵	5.348x10 ⁻⁴	1.4x10 ⁻³



(a)



(b)



(c)

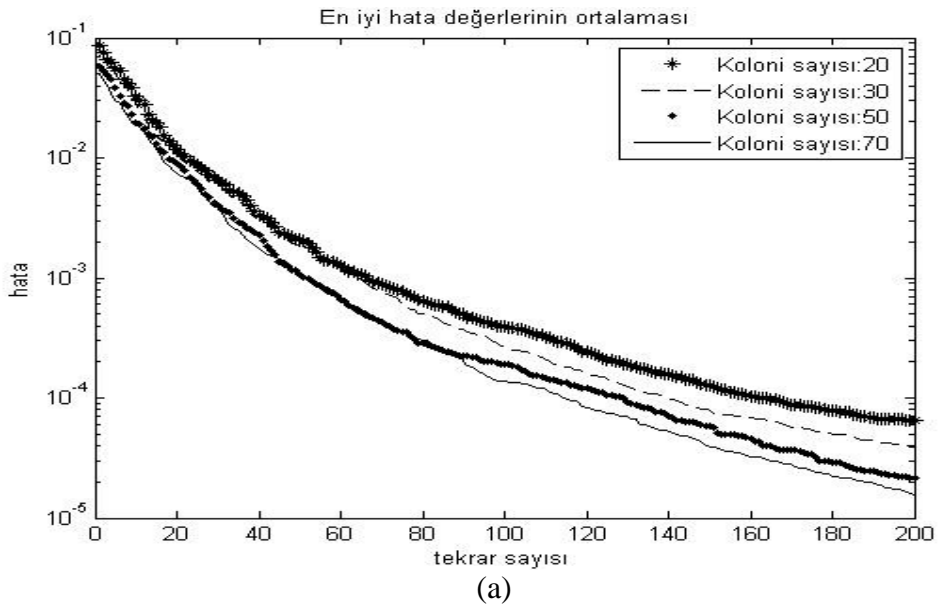
Şekil 3. 14. Denklem 3.15'teki model için sistem ve model çıktıları: (a) yak algoritması, (b) ga, (c) ksa

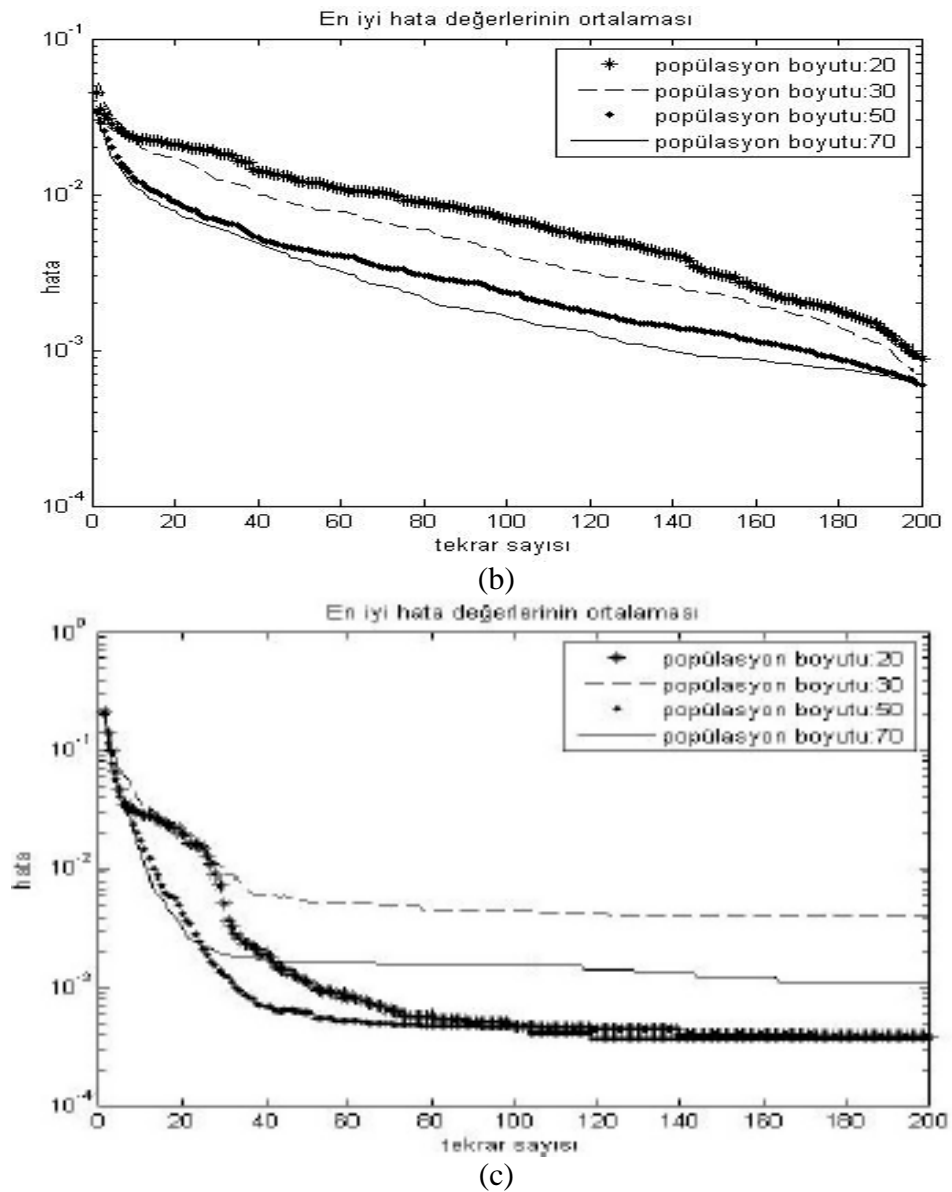
Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.22’de belirtilmiştir.

Tablo 3. 22. Denklem 3.15’te önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	5.7174	5.9789	9.7416
MSE	1.811×10^{-5}	3.173×10^{-4}	4.165×10^{-4}
Standart sapma	7.320×10^{-5}	8.708×10^{-4}	1.6×10^{-3}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	8.3889	8.5825	18.8894
MSE	4.200×10^{-5}	2.3×10^{-3}	4.3×10^{-3}
Standart sapma	3.266×10^{-5}	5.348×10^{-4}	1.4×10^{-3}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	13.7150	14.3037	102.1483
MSE	3.995×10^{-6}	8.027×10^{-4}	8.198×10^{-4}
Standart sapma	1.910×10^{-5}	4.958×10^{-4}	4.29×10^{-2}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	19.4460	19.4195	180.0238
MSE	2.996×10^{-5}	1×10^{-3}	4.6×10^{-3}
Standart sapma	1.124×10^{-5}	5.797×10^{-4}	3.14×10^{-2}

Aynı zamanda koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.15’te verilmiştir.





Şekil 3. 15. Denklem 3.15'teki model için hata yaklaşım eğrileri: (a)yak algoritması, (b) ga, (c) ksa

Denklem 3.14 için önerilen Denklem 3.15'teki model yapısı, dikkate alınan sistemle aynı olması sebebiyle algoritmalar yardımıyla model parametrelerinin oldukça küçük bir hata ile elde edebildiği görülmektedir. Ancak farklı bir model yapısının seçilmesi durumunda MSE değerinin artması olasılığının bulunduğunu gözden kaçırmamak gerekir.

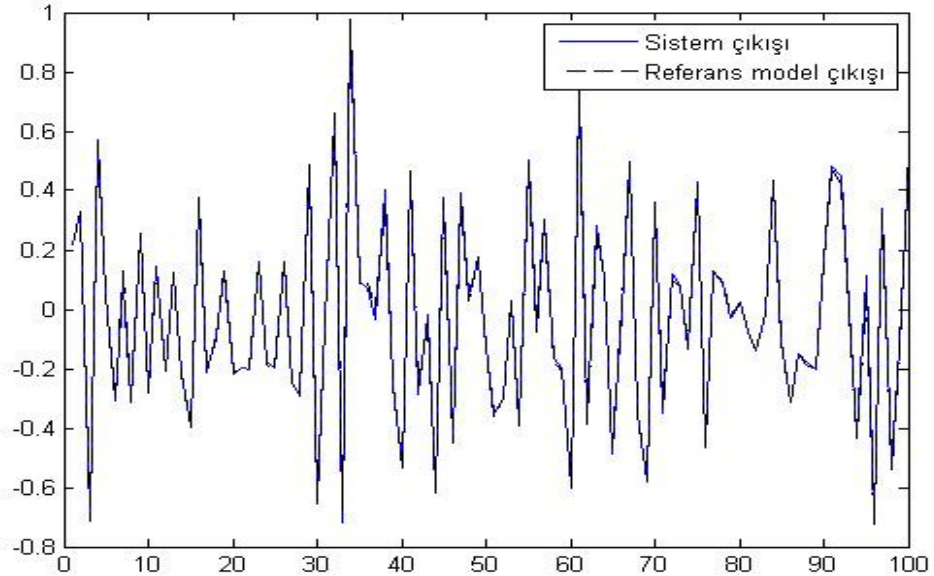
Bu amaçla literatürde de uygulanmış olan, matematiksel ifadesi Denklem 3.14'te verilen sistem için Zorlu ve Özer'in önerdiği ve farklı algoritmalarla yapmış oldukları, Denklem 3.16'da verilen daha az parametreye sahip farklı bir model yapısı

kullanılmış olup, YAK algoritmasının performansı incelenmiştir [77,83-84]. Algoritmaların buldukları MSE değerleri Tablo 3.23'te gösterilmiştir. Ayrıca referans çalışma için YAK algoritmasının hesapladığı parametrelere göre sistem çıkışı ile Denklem 3.16'de verilen model çıkışı Şekil 3.16'da verilmiştir.

$$y(n) = a_1 y(n-1) + b_0 u(n) + b_1 u(n-1) + a_{1,1} y(n-1) u(n-1) + b + c_{2,2} y(n-2) u(n-2) + c_{3,3} y(n-3) u(n-3) \quad (3.16)$$

Tablo 3. 23. Denklem 3.16 için YAK Algoritmasının referans çalışma ile karşılaştırılması

Referans Çalışma	Algoritma	MSE
Zorlu ve Özer [77,83]	DGA	0.0208
	GA	0.0209
	KSA	0.0209
	RLS	0.0233
Bu çalışma	YAK	2.192×10^{-4}



Şekil 3. 16. Denklem 3.15'te verilen sistem çıkışı ile yak algoritmasının hesapladığı Denklem 3.16'da önerilen model çıkışı

Referans çalışma ile karşılaştırıldığında farklı model yapısının önerilmesi ile YAK algoritmasının diğer algoritmalarından daha düşük bir MSE değeri ile parametre kestirimi yaptığı görülmüştür.

3.2.3.4. Örnek Sistem 4: Parametrik Doğrusal Olmayan Bilineer Sistem

Doğrusal olmayan diğer bir sistem örneği ise Zorlu-Özer ve Bağış tarafından da ele alınan ve Denklem 3.17’de verilen doğrusal olmayan parametrik bilineer sistem incelenmiştir. Bu problemde de temel amaç giriş ve çıkış bilgileri kullanılarak Denklem 3.18’de önerilen model parametrelerinin tahmin edilmesidir [83,85-87].

Bu model parametrelerinin tahmini amacıyla yapılan benzetim çalışmalarında giriş verisi, $u(n)$, hem sistem hem de model girişi olarak kullanılmıştır. Giriş dizisi 100 veriden oluşan Gaussian dağılımlı beyaz gürültüdür. Modelleme çalışmasında bilineer model parametreleri, model çıkışı ile sistem çıkışı arasındaki hata minimize edilinceye kadar YAK Algoritması, GA ve KSA tarafından optimize edilmiş ve gerçek sistem parametreleri ile karşılaştırılmıştır. Tablo 3.24’te algoritmaların parametre bulmaları için kullanılan kontrol parametreleri verilmiştir. Kullanılan kontrol parametreleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulduğu parametreler Tablo 3.25’te verilmiştir. Tablo 3.25’teki hesaplanan parametrelere göre sistem ve model çıkışları Şekil 3.17’de verilmiştir.

$$y(n) = 0.325 y(n-1) - 0.4 u(n-1) + 0.56 u(n-3) - u(n-4) - 0.65 y(n-2) u(n-2) + 0.065 y(n-3) u(n-5) + 0.065 y(n-5) u(n-2) \quad (3.17)$$

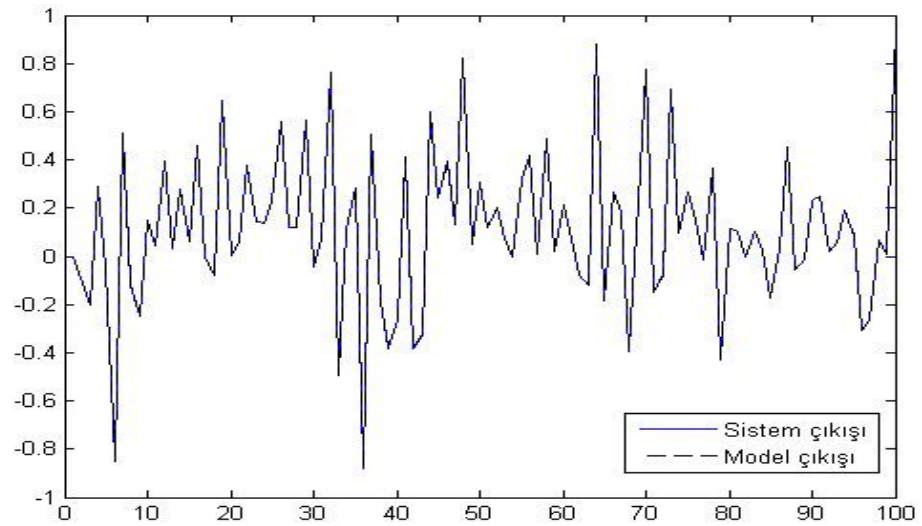
$$y(n) = a_1 y(n-1) + b_1 u(n-1) + b_3 u(n-3) + b_4 u(n-4) + c_{2,2} y(n-2) u(n-2) + c_{3,5} y(n-3) u(n-5) + c_{5,2} y(n-5) u(n-2) \quad (3.18)$$

Tablo 3. 24. Denklem 3.18’de önerilen modele ait kullanılan kontrol parametreleri

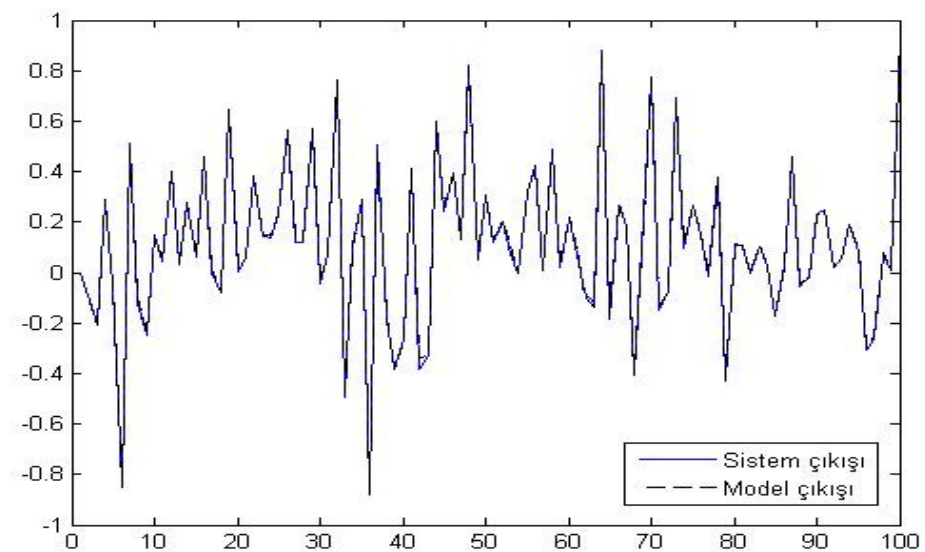
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-1+1	-1+1	-1+1
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	200	200	200

Tablo 3. 25. Denklem 3.18’de önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

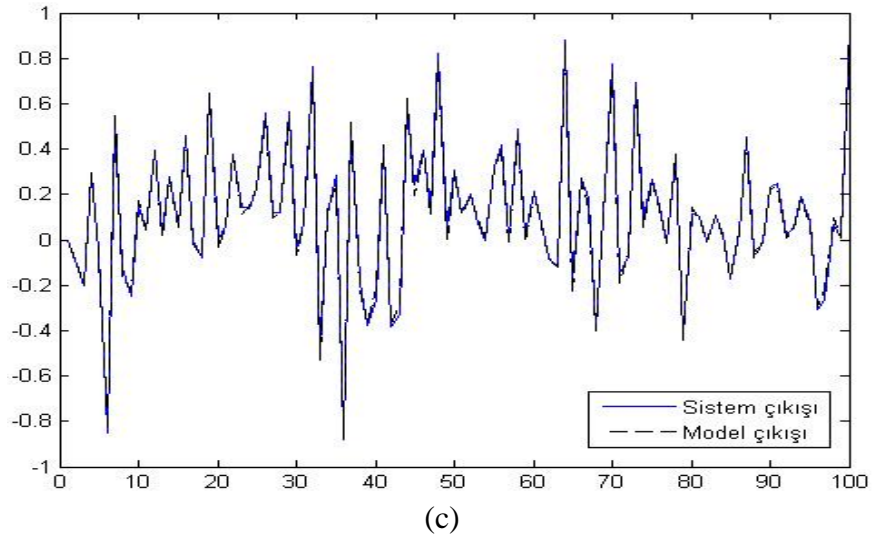
Kat Sayı	Gerçek Değer	YAK	GA	KSA
a_1	0.325	0.3251	0.3405	0.2500
b_1	-0.4	-0.4000	-0.4071	-0.3964
b_3	0.56	0.5601	0.5645	0.5389
b_4	-1	-1.0000	-0.9940	-0.9766
$c_{2,2}$	-0.65	-0.6446	-0.5989	-0.6250
$c_{3,5}$	0.065	0.0638	0.0468	-2.384×10^{-7}
$c_{5,2}$	0.065	0.0639	-0.0185	-0.0312
Ortalama süre (sn)		8.4978	9.1931	21.1482
MSE		2.291×10^{-7}	1.134×10^{-4}	5.099×10^{-4}
Standart sapma		4.401×10^{-7}	2.767×10^{-5}	2.6×10^{-3}



(a)



(b)

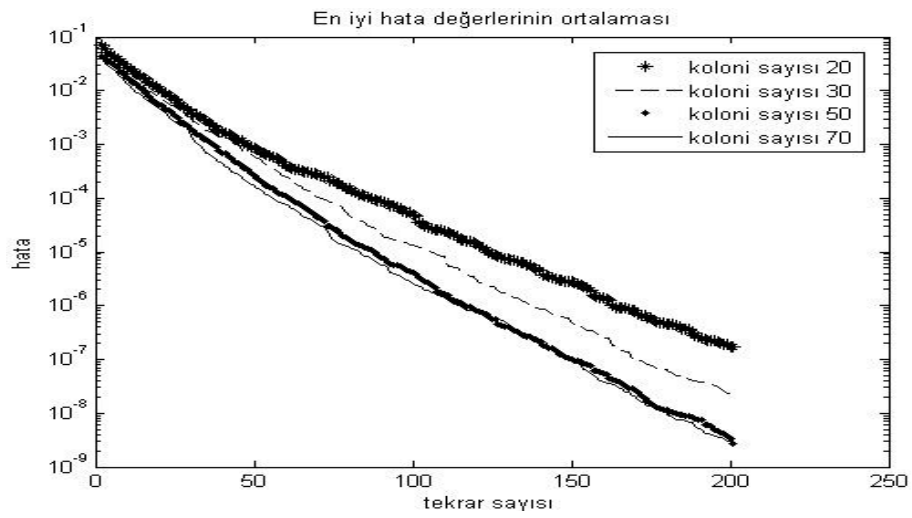


Şekil 3. 17. Denklem 3.18'deki model için sistem ve model çıkışları: (a) yak algoritması, (b) ga, (c) ksa

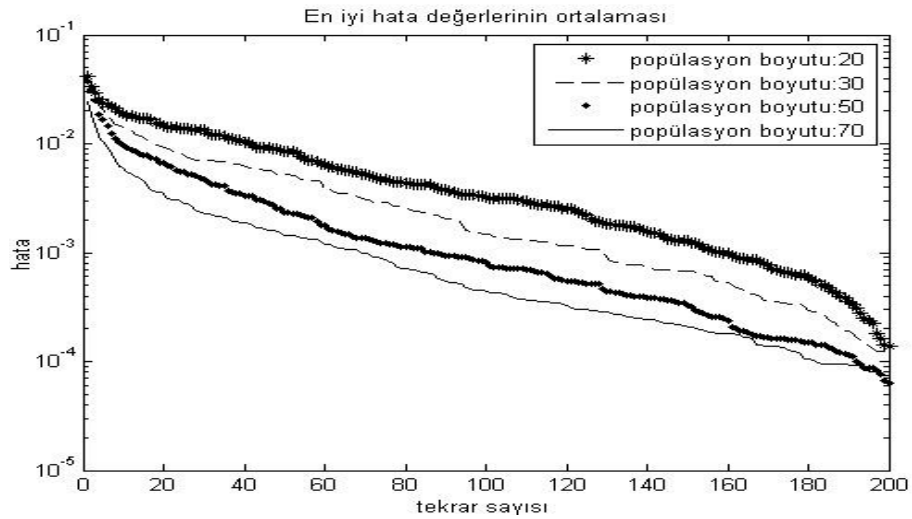
Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.26'da belirtilmiştir. Aynı zamanda koloni ve popülasyon sayılarının Tablo 3.24'te belirtilen değerlerde olması durumunda algoritmalara ait hata yaklaşım eğrileri Şekil 3.18'de verilmiştir

Tablo 3. 26. Denklem 3.18'de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

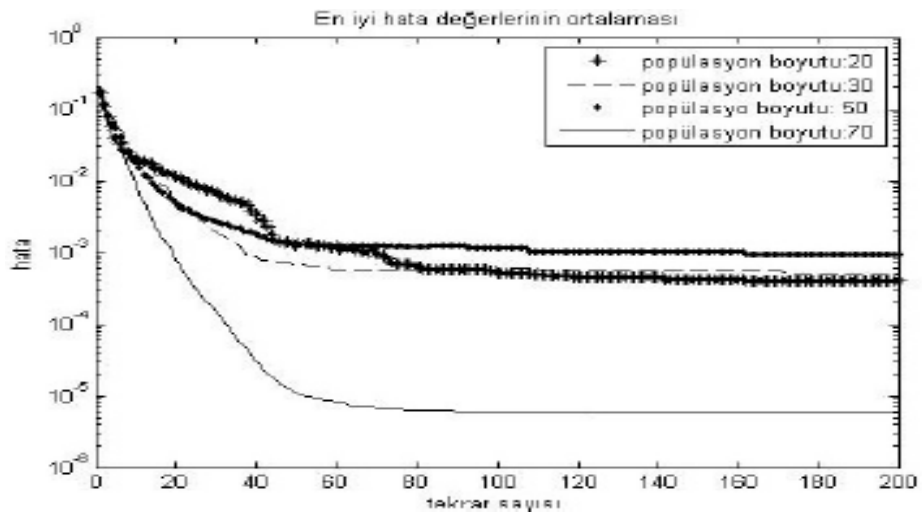
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	5.8474	6.1668	10.9044
MSE	7.955×10^{-9}	1.976×10^{-5}	3.980×10^{-4}
Standart sapma	4.257×10^{-7}	6.579×10^{-5}	8.5×10^{-3}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	8.4978	9.1931	21.1482
MSE	2.291×10^{-7}	1.134×10^{-4}	5.099×10^{-4}
Standart sapma	4.401×10^{-7}	2.767×10^{-5}	2.6×10^{-3}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	15.5823	13.9880	47.176
MSE	3.881×10^{-10}	2.035×10^{-5}	1×10^{-3}
Standart sapma	2.791×10^{-9}	1.661×10^{-5}	4.830×10^{-4}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	22.0117	20.1662	82.8476
MSE	1.237×10^{-8}	1.359×10^{-5}	4.169×10^{-6}
Standart sapma	4.731×10^{-9}	8.297×10^{-6}	3.626×10^{-4}



(a)



(b)



(c)

Şekil 3. 18. Denklem 3.18 için hata yaklaşım eğrileri: (a) yak algoritması, (b) ga, (c) ksa

Matematiksel ifadesi Denklem 3.17’de verilen sistem için Zorlu ve Özer’in, Denklem 3.18’de verilen model yapısı ile yapmış oldukları çalışma ile bu çalışmadaki algoritmaların buldukları MSE değerleri Tablo 3.27’de gösterilmiştir.

Tablo 3. 27. Denklem 3.18 için YAK Algoritmasının referans çalışmalar ile karşılaştırılması

Referans Çalışma	Algoritma	MSE
Zorlu ve Özer [85]	KSA	4.7×10^{-5}
	GA	5.6×10^{-5}
Zorlu ve Özer [86]	DEA	1.5×10^{-5}
	GA	5.6×10^{-5}
Bağış [87]	KSA	4.7×10^{-5}
	FGA	7.784×10^{-7}
Bu çalışma	YAK	1.237×10^{-8}

Referans çalışmalar ile karşılaştırıldığında aynı model yapısı veya farklı model yapısının önerilmesi ile YAK algoritmasının diğer algoritmalarından daha düşük bir MSE değeri ile parametre kestirimi yaptığı görülmüştür.

Bu çalışmada kullanılan Bilineer sistemlerin parametre hesaplama performansları bakımından, YAK algoritmasının doğrusal olmayan sistemlerde çok başarılı bir şekilde parametre hesabı yaptığı ortadadır. Yine aynı şekilde, hata yaklaşım eğrilerinden de anlaşılacağı üzere hatayı minimize edip optimum çözüme kararlı bir şekilde yaklaşan YAK algoritmasıdır ve bu kestirimi yaparken de çok düşük bir MSE ile yapmasında YAK algoritmasının Bilineer sistemlerin modellenmesinde ne denli başarılı bir algoritma olduğunun göstergesidir.

Parametre bulma süresi göz önüne alındığında ise önceki örneklerdekinin aksine belirli koloni sayılarında YAK algoritması hem GA’dan hemde KSA’dan daha hızlı olduğu görülmektedir. Parametre sayılarının ise doğrusal olmayan parametrik sistem modellemelerinde parametre bulma sürelerine doğru orantılı bir şekilde etki etdiğini söylemek mümkün değildir.

3.2.4. Doğrusal Olmayan Gerçek Sistem Modellenmesi Uygulamaları

3.2.4.1. Örnek Sistem 1

Burada ilk kullanılan gerçek sistem verileri, Katholieke Universiteit Leuven, Signals, Identification, System Theory and Automation's Identification Database'den alınmış, mekanik bir sistem olup saç kurutma makinesi gibi çalışan bir sistemdir. Bir tüpün içerisinde ısıtılan hava dışarı doğru üflenmiş ve sıcaklığı termokulp aracılığı ile ölçülmüştür. Sistem girişi ısıtıcı telleri direnci üzerinde ölçülen gerilimdir. Çıkış ise üflenen havanın sıcaklığıdır. İlk yöntem olarak bu sisteme ait 1000 adet datadan ilk 100 datası parametre hesabı için kullanılmıştır. Bulunan parametrelerin testi için ise sonraki 100 adetlik data kullanılmıştır [7-8,84,88].

Bu sistem için Denklem 3.19' da önerilen model yapısı uygulanmış ve parametreler ile sistem MSE' leri hesaplanmıştır. Tablo 3.28'de ise her üç algoritma için kontrol parametreleri sıralanmıştır. Aynı zamanda bu modellerin parametrelerinin ve parametre hesaplama süreleri ile hataları Tablo 3.29'da verilmiştir. Şekil 3.19'da ise koloni ve popülasyon sayılarının belirtilen değerlerde olması durumunda algoritmalara ait hata gerçek sistem çıkışı ile algoritmaların hesapladığı çıkışlar gösterilmiştir.

$$y(n) = a_1 y(n-1) + b_0 u(n) + c_{2,0} y(n-2) u(n) + c_{3,1} y(n-3) u(n-1) + c_{4,2} y(n-4) u(n-2) + c_{1,3} y(n-1) u(n-3) + c_{2,4} y(n-2) u(n-4) \quad (3.19)$$

Tablo 3. 28. Denklem 3.19'da önerilen modele ait kullanılan kontrol parametreleri

Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-1+1	-1+1	-1+1
Popülasyon boyutu	-	50	50
Koloni sayısı	50	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	2000	2000	2000

Tablo 3. 29. Denklem 3.19’da önerilen modele ait algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

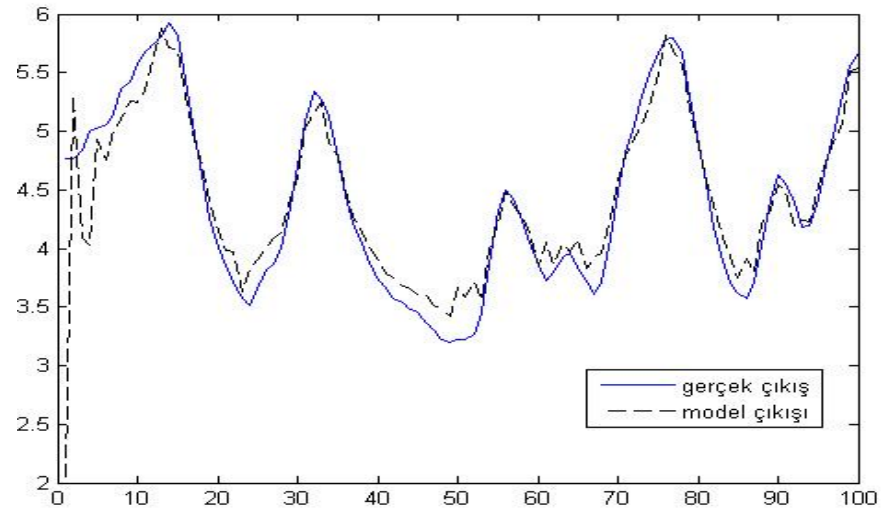
Kat Sayı	YAK	GA	KSA
a₁	0.7851	0.8449	0.8750
b₀	0.6008	0.3974	0.1582
c_{2,0}	-0.1360	-0.0845	2.384x10 ⁻⁷
c_{3,1}	-0.0256	-0.0052	-0.0299
c_{4,2}	0.0045	-0.0132	-0.0078
c_{1,3}	0.0356	0.0254	0.0293
c_{2,4}	0.0303	0.0209	2.384x10 ⁻⁷
Ortalama süre (sn)	186.4396	208.2069	781.623
MSE	0.1705	0.1004	0.1839
Standart sapma	0.0656	0.0029	0.1739

Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.30’da belirtilmiştir.

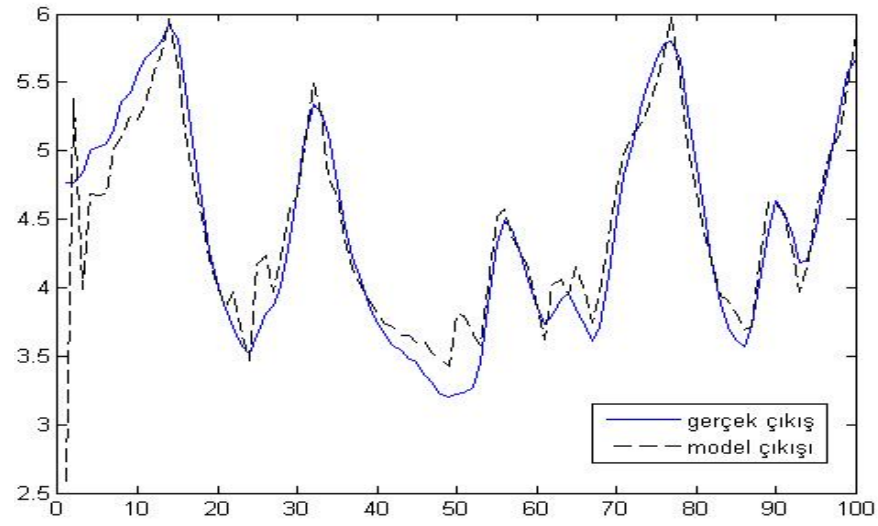
Tablo 3. 30. Denklem 3.19’da önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı = 20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	86.0584	77.5821	180.4166
MSE	0.7723	0.1318	0.4288
Standart sapma	0.3018	0.0174	2.6973
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	124.7085	101.9411	349.9589
MSE	0.8567	0.1115	0.1976
Standart sapma	0.1837	0.0186	2.7979
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	186.4396	208.2069	781.623
MSE	0.1705	0.1004	0.1839
Standart sapma	0.0656	0.0029	0.1739
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	254.7101	295.5211	254.7101
MSE	0.1627	0.1096	0.1627
Standart sapma	0.0567	0.0029	0.0567

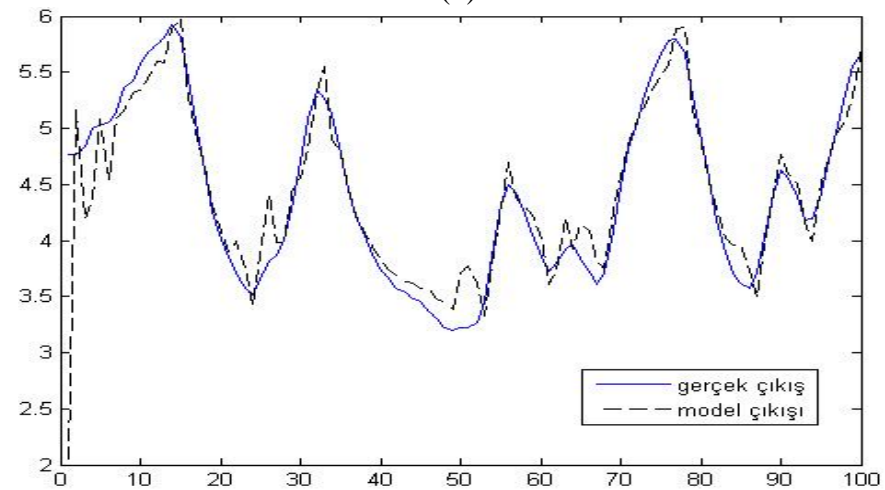
Tablo 3.29’da hesaplanmış olan parametreler için sistemin gerçek çıkışları ile algoritmaların hesapladığı model çıkışı Şekil 3.19’da ve test çıkışları ise Şekil 3.20’de verilmiştir.



(a)

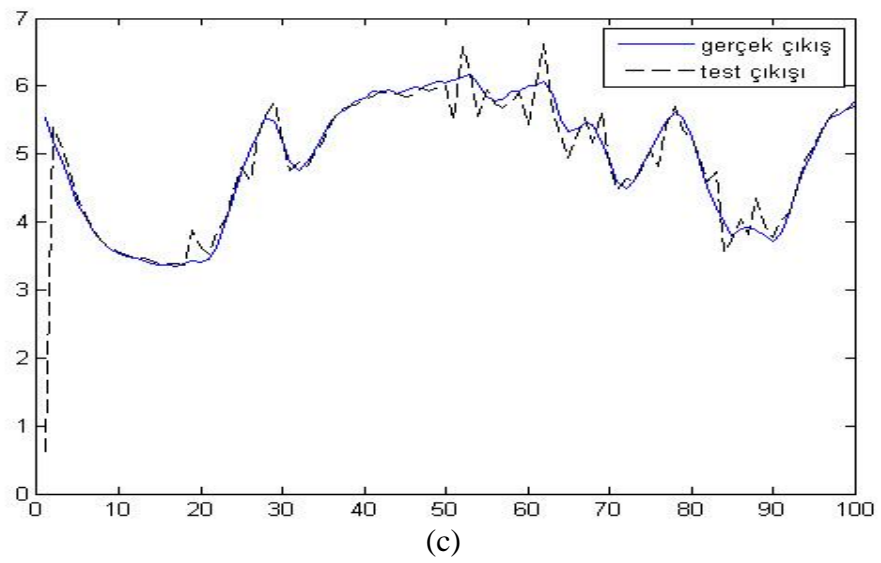
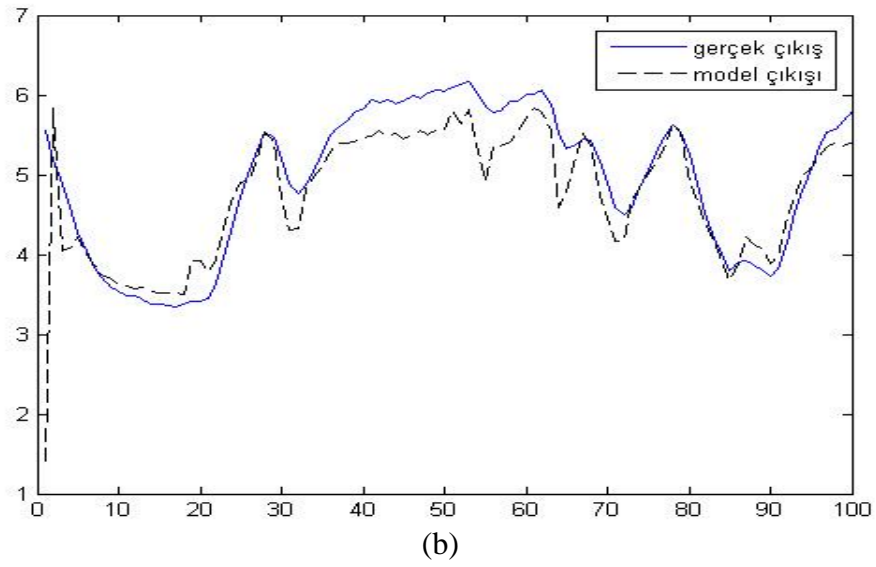
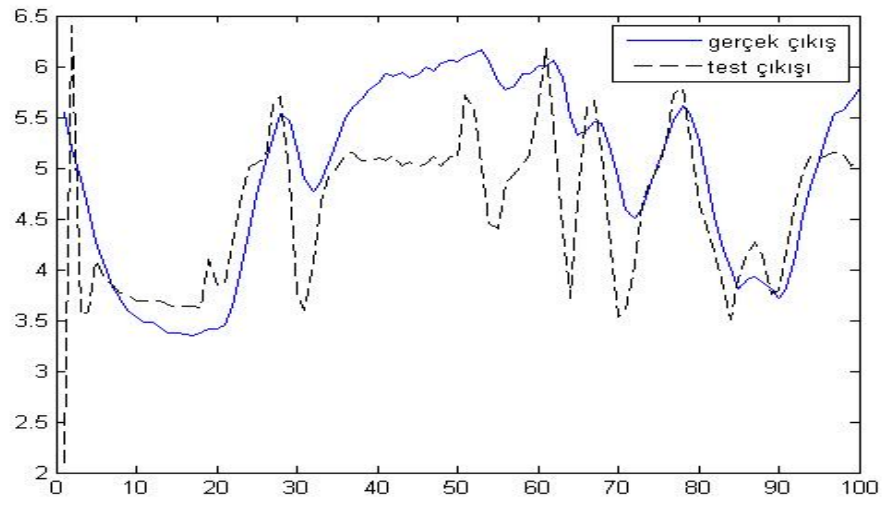


(b)



(c)

Şekil 3. 19. Gerçek sistem çıkışı ile Denklem 3.19'da önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa



Şekil 3. 20. Gerçek sistem çıkışı ile Denklem 3.19'da önerilen model için test çıktıları: (a) yak algoritması, (b) ga , (c) ksa

Bu ilk yöntemde sistemin geriye kalan 800 adet datasının kullanılmaması algoritmaların sistemi modellemesinde ve parametre bulmalarında sistem hakkında yeterince eğitilmemesi demektir. Bu ihtimali ortadan kaldırmak için ikinci yöntem olarakta 1000 adetlik datanın rastgele seçilmiş %80 adet datasını sistem parametrelerinin bulunması için sistemin eğiminde, geriye kalan %20 adetlik data ise bulunan modelin testi için kullanılmıştır.

Tablo 3.28'deki kontrol parametrelerinin kullanılmasıyla ikinci yöntemde alınan verilerle sistem modellenmiş ve algoritmaların bulunduğu parametreler Tablo 3.31'de belirtilmiştir. Aynı zamanda Tablo 3.31'de hesaplanmış olan parametreler için sistemin gerçek çıkışları ile algoritmaların hesapladığı model çıkışı Şekil 3.21'de ve test çıkışları ise Şekil 3.22'de verilmiştir.

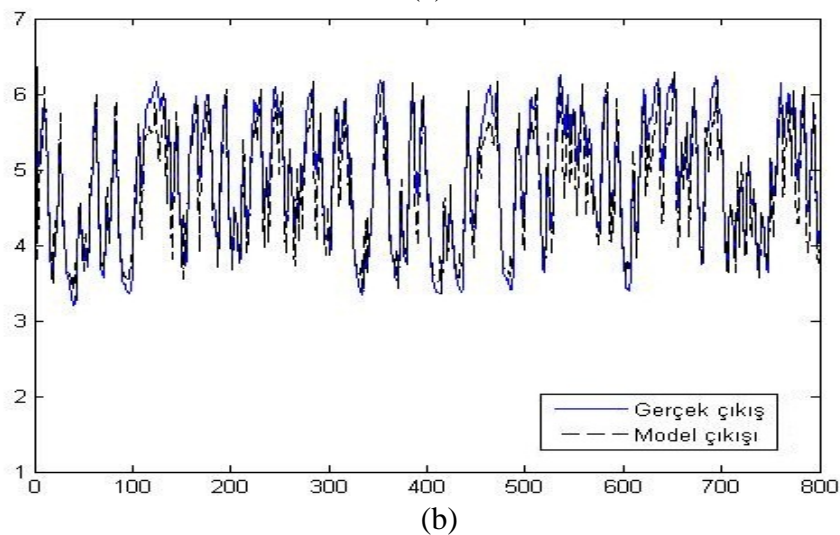
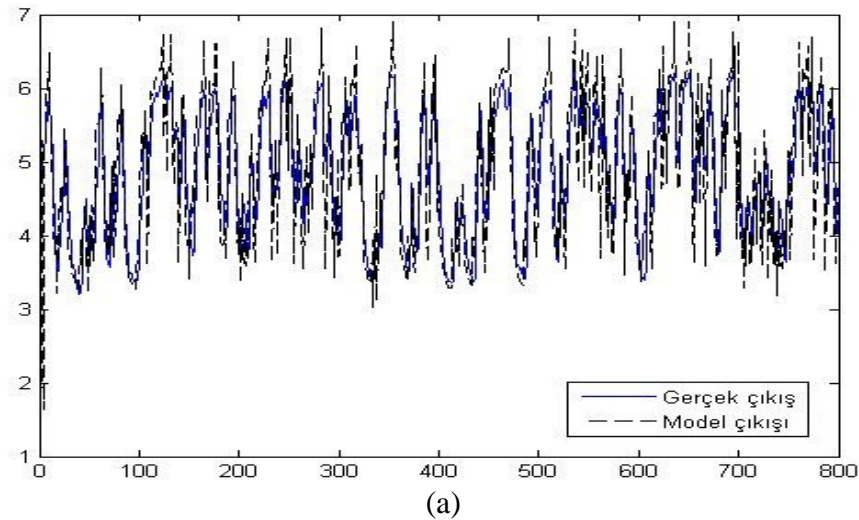
Ayrıca ikinci yöntemde YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.32'de belirtilmiştir.

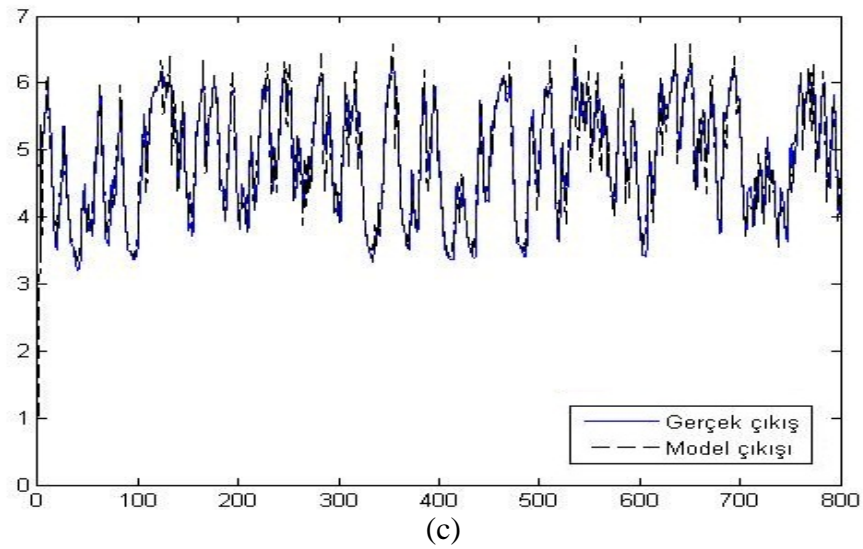
Tablo 3. 31. Denklem 3.19'da önerilen modele ait ikinci yöntemde algoritmaların bulunduğu parametreler, MSE değerleri ve parametre bulma süreleri

Kat Sayı	YAK	GA	KSA
a₁	0.9546	0.7841	0.7500
b₀	0.2225	0.1937	0.2812
c_{2,0}	-0.0391	-0.0387	-0.0625
c_{3,1}	-0.0418	0.0026	-0.0075
c_{4,2}	0.0402	0.0024	0.0313
c_{1,3}	9.95x10 ⁻⁴	0.0248	0.0313
c_{2,4}	0.0051	0.0107	-2.384x10 ⁻⁷
Ortalama süre (sn)	1264	1269	2030
MSE	0.1246	0.0419	0.0906
Standart sapma	0.0941	0.0122	0.2677

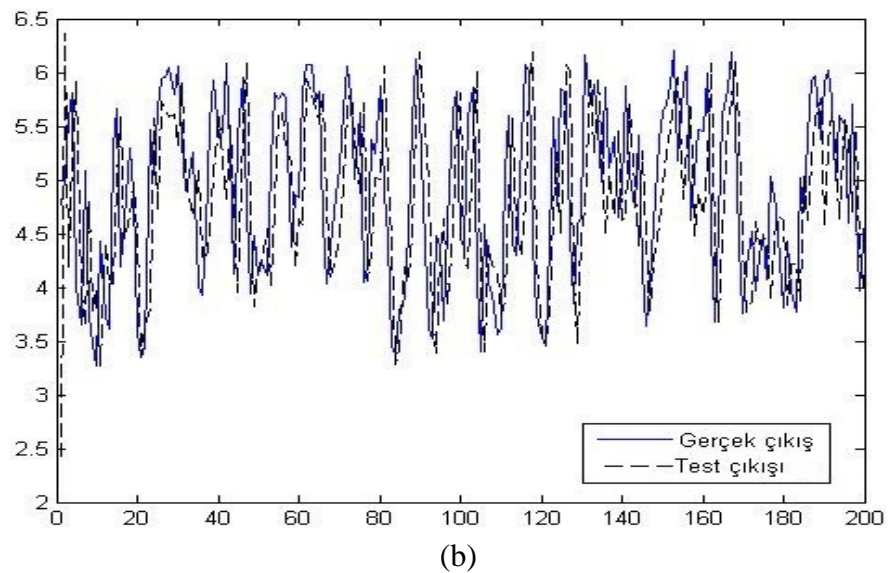
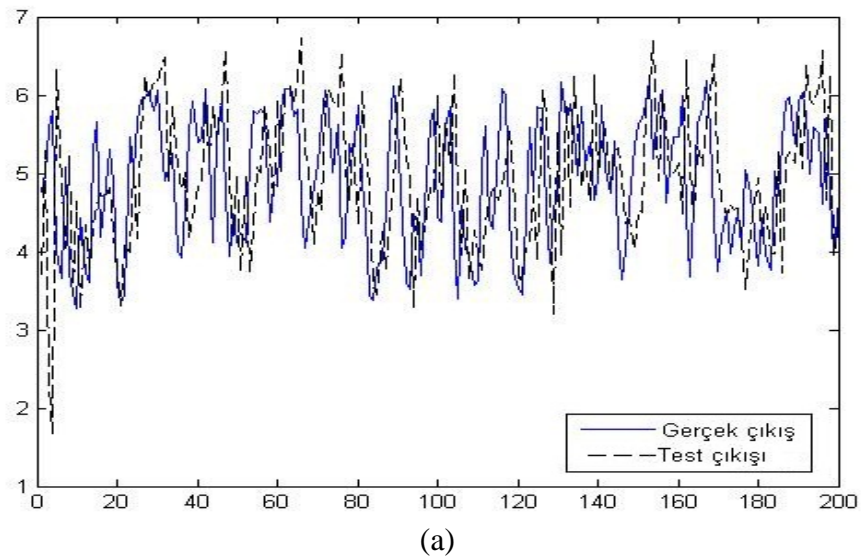
Tablo 3. 32. Denklem 3.19’da önerilen model için ikinci yöntemde koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

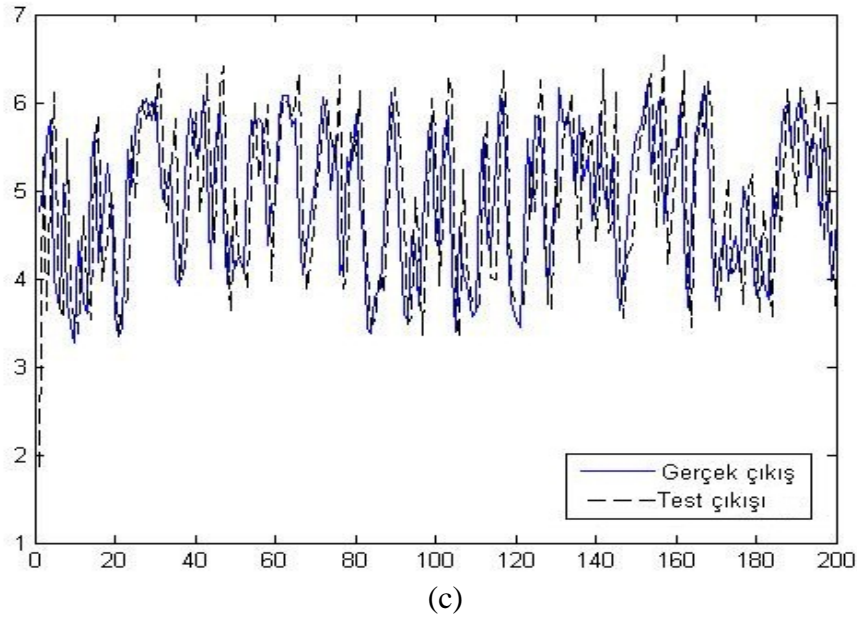
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	497.893	508.598	443.081
MSE	0.8251	0.0674	0.1485
Standart sapma	0.5519	0.0511	3.424
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	1291	774.352	817.582
MSE	0.2944	0.0436	0.7393
Standart sapma	0.1292	0.0146	4.734
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	1264	1269	2030
MSE	0.1246	0.0419	0.0906
Standart sapma	0.0941	0.0122	0.2677
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	1770	1771	3589
MSE	0.3205	0.0458	0.0470
Standart sapma	0.0625	0.0078	0.3824





Şekil 3. 21. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.19’da önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa





Şekil 3. 22. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.19’da önerilen model için test çıktıları: (a) yak algoritması, (b) ga , (c) ksa

Bu ilk örnekte sistemin modellenmesi için kullanılan her iki yöntem karşılaştırıldığında data sayısının artması ile algoritmaların parametre tahminlerindeki süresinin artmış olduğu görülmüştür. Ayrıca farklı koloni ve popülasyon sayılarında algoritmalar, farklı hata değerleri ile sistemi modellemiş ve parametre hesabı yapmıştır. Farklı model yapısının önerilmesi ile algoritmaların sistemi modelleme süresi ve hatası düşürülebilir.

3.2.4.2. Örnek Sistem 2

İkinci olarak kullanılan sistem verileri de gerçek sistem verileri olup, bu verilerde Katholieke Universiteit Leuven, Signals, Identification, System Theory and Automation’s Identification Database’den alınmış, NASA’nın F-18 modeli bir hava araştırma aracına ait kanat sarsıntularından alınan verilerdir (78,88-89).

Bu sistem için Denklem 3.20’ de önerilen model yapısı uygulanmış ve parametreler ile sistem MSE’ leri hesaplanmıştır. İlk örnekteki gibi ilk yöntem olarak önerilen model parametrelerinin hesaplanması için 1024 adet datadan ilk 300 data, bulunan parametrelerin testi için ise sonraki 100 adetlik data kullanılmıştır. Tablo 3.33’te algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol değerleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulduğu parametreler Tablo 3.34’te verilmiştir. Ayrıca YAK algoritması için koloni

sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.35'te belirtilmiştir.

Tablo 3.34'te hesaplanmış olan parametreler için sistemin gerçek çıkışları ile algoritmaların hesapladığı model çıkışı Şekil 3.23'te ve test çıkışları ise Şekil 3.24'te verilmiştir.

$$y(n) = c_{1,1} y(n-1)u(n-1) + a_2 y(n-2) + b_3 u(n-3) + b_4 u(n-4) + c_{3,3} y(n-3) u(n-3) + c_{4,2} y(n-4) u(n-2) + c_{1,2} y(n-1) u(n-2) \quad (3.20)$$

Tablo 3. 33. Denklem 3.20 önerilen modele ait kullanılan kontrol parametreleri

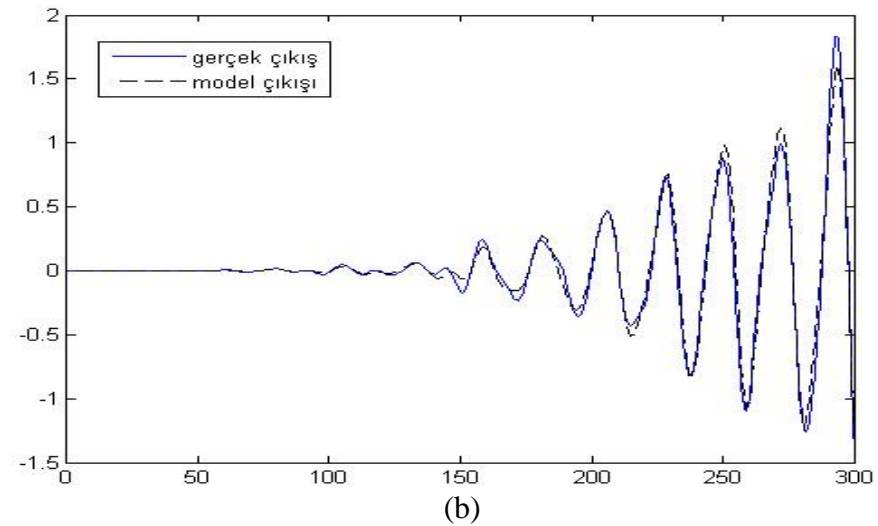
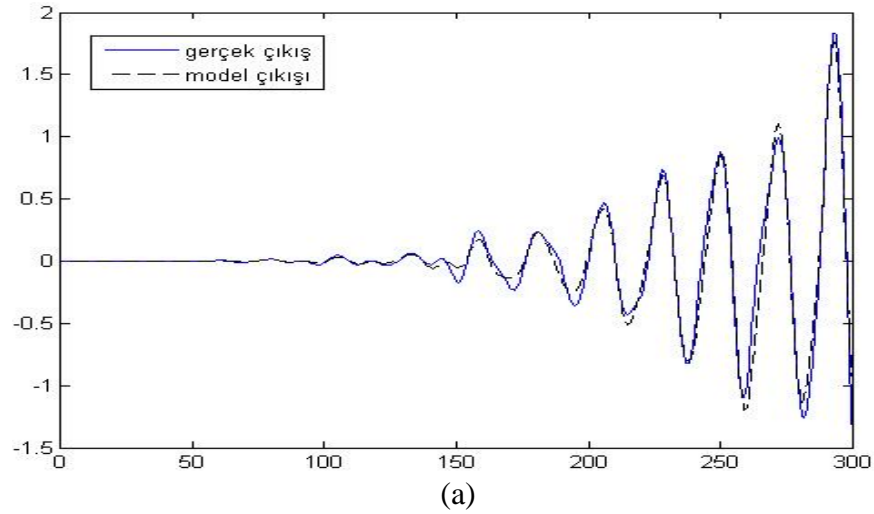
Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-1+1	-1 +1	-1 +1
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	500	500	500

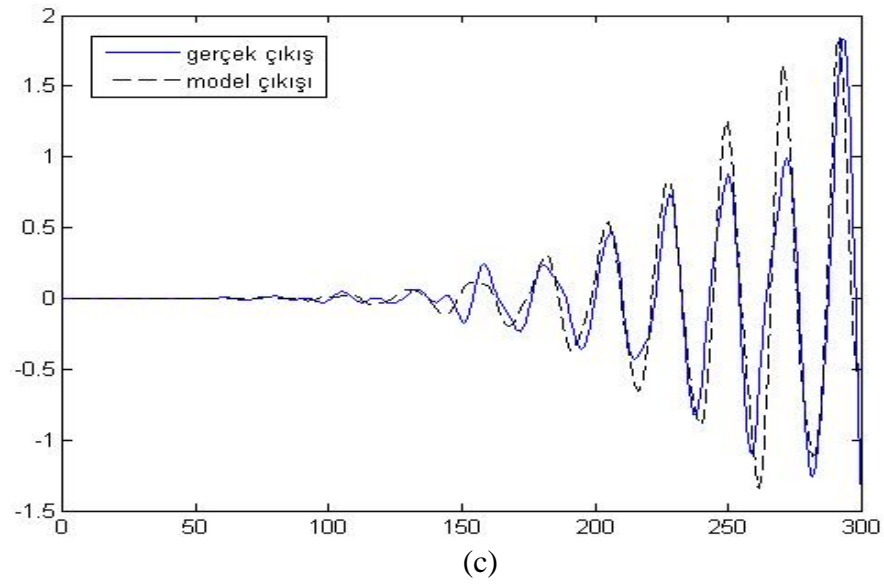
Tablo 3. 34. Denklem 3.20'de önerilen model için algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

Kat Sayı	YAK	GA	KSA
c_{1,1}	0.0107	0.1019	-0.3750
a₂	1.0000	0.9999	-0.7500
b₃	-0.5823	-0.5578	0.7500
b₄	-0.3258	-0.3269	-0.7500
c_{3,3}	-0.2574	-0.4420	0.6563
c_{4,2}	0.1128	0.2368	-0.4028
c_{1,2}	0.0986	0.0763	-3.57x10 ⁻⁷
Ortalama süre (sn)	27.49	59.92	128.87
MSE	0.0052	0.0057	0.0462
Standart sapma	9.878x10 ⁻⁴	0.0028	0.0132

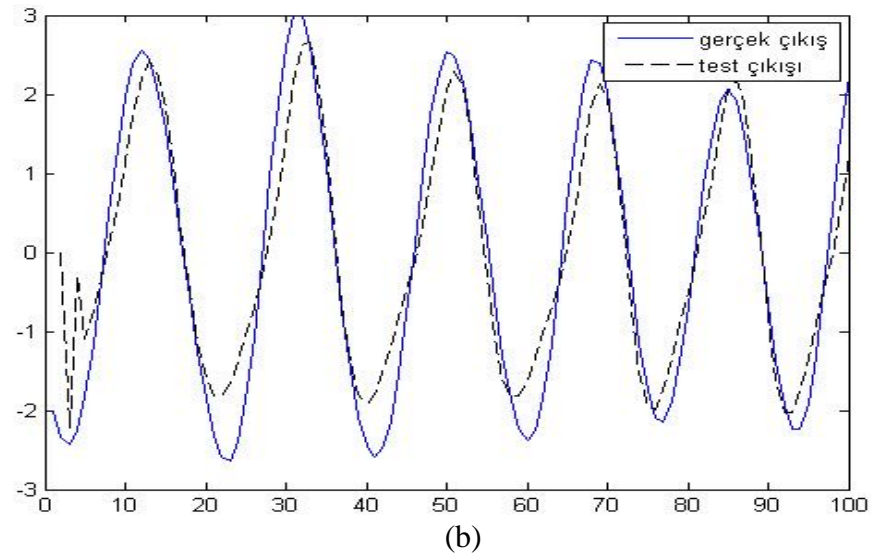
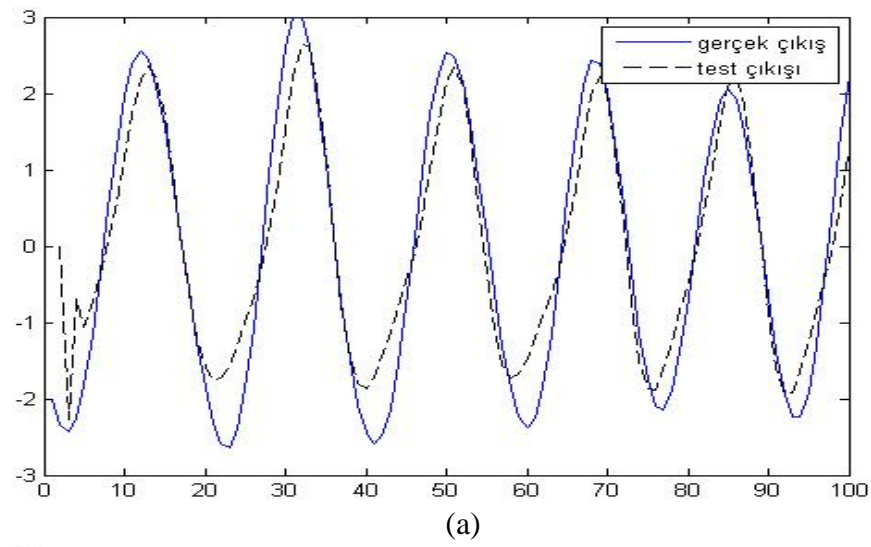
Tablo 3. 35. Denklem 3.20'de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

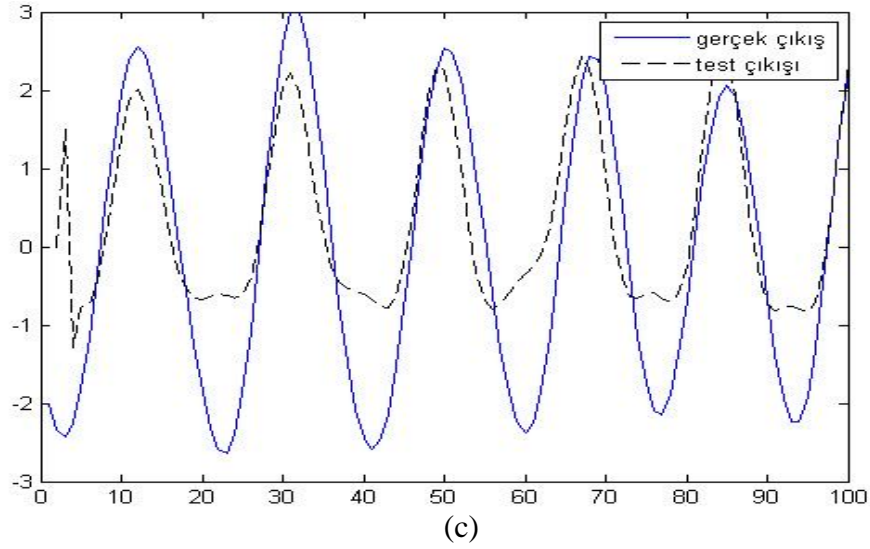
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	16.92	40.10	62.91
MSE	0.0061	0.0128	0.0125
Standart sapma	0.0011	0.0043	0.0194
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	27.49	59.92	128.87
MSE	0.0052	0.0057	0.0462
Standart sapma	9.878×10^{-4}	0.0028	0.0132
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	111.13	98.39	277.11
MSE	0.0049	0.0040	0.0306
Standart sapma	3.818×10^{-4}	0.0026	0.0084
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	156.311	136.46	381.56
MSE	0.0049	0.0017	0.0099
Standart sapma	1.778×10^{-4}	0.0020	0.0073





Şekil 3. 23. Gerçek sistem çıkışı ile Denklem 3.20’de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa





Şekil 3. 24. Gerçek sistem çıkışı ile Denklem 3.20’de önerilen model için test çıktıları: (a) yak algoritması, (b) ga , (c) ksa

Bu ilk yöntemde sistemin geriye kalan 624 adet datası kullanılmamıştır. İlk örnekte olduğu gibi bu örnekte de ikinci yöntem olarak 1024 adetlik datanın rastgele seçilmiş %80 adet datasını model parametrelerinin bulunması için sistemin eğiminde, geriye kalan %20 adetlik data ise bulunan modelin testi için kullanılmıştır. Tablo 3.32’deki kontrol parametrelerinin kullanılmasıyla ikinci yöntemde alınan verilerle sistem modellenmiş ve algoritmaların bulduğu parametreler Tablo 3.36’da belirtilmiştir. Aynı zamanda Tablo 3.37’de hesaplanmış olan parametreler için sistemin gerçek çıktıları ile algoritmaların hesapladığı model çıkışı Şekil 3.25’te ve test çıktıları ise Şekil 3.26’da verilmiştir.

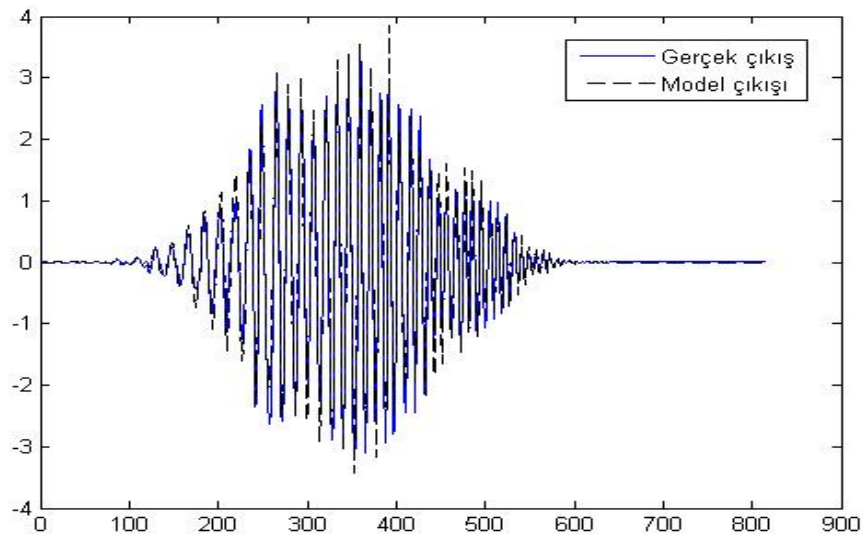
Tablo 3. 36. Denklem 3.20’de önerilen modele ait ikinci yöntemde algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

Kat Sayı	YAK	GA	KSA
a_1	0.0050	-0.0059	-0.0216
b_0	0.8108	0.6408	-0.1875
$c_{2,0}$	-0.7271	-0.5723	0.0469
$c_{3,1}$	-0.5783	-0.6784	-1.1250
$c_{4,2}$	0.0492	0.0842	0.1273
$c_{1,3}$	-0.0272	-0.0462	-0.0938
$c_{2,4}$	-0.0200	-0.0271	-0.0293
Ortalama süre (sn)	204.194	196.273	212.420
MSE	0.0982	0.1046	0.1903
Standart sapma	0.0016	0.0509	0.1001

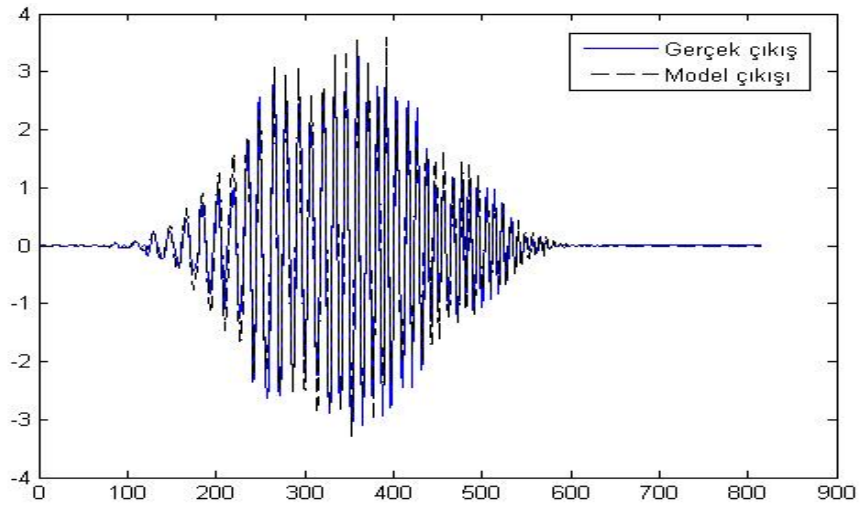
Ayrıca ikinci yöntemde YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.37’de belirtilmiştir.

Tablo 3. 37. Denklem 3.20’de önerilen model için ikinci yöntemde koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

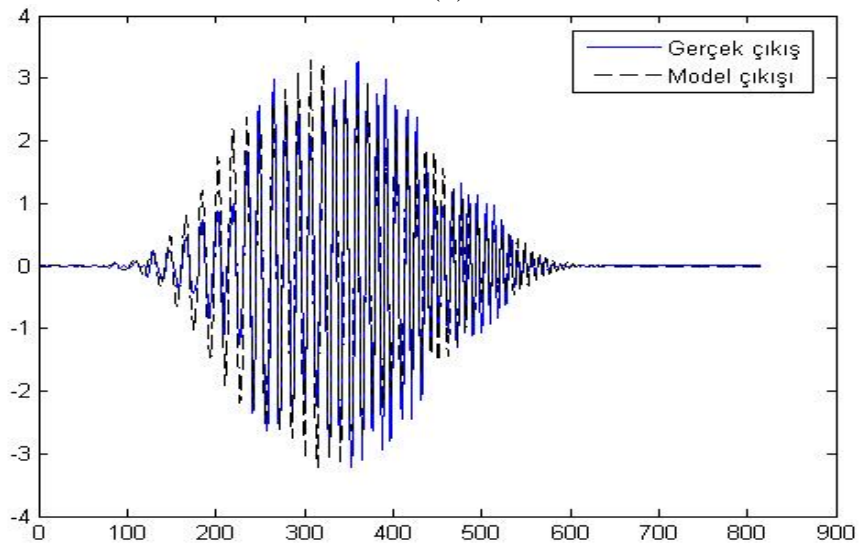
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	131.080	129.758	128.166
MSE	0.1010	0.1080	0.0993
Standart sapma	0.0019	0.0313	0.1500
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	204.194	196.273	212.420
MSE	0.0982	0.1046	0.1903
Standart sapma	0.0016	0.0509	0.1001
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	324.477	318.384	495.515
MSE	0.0982	0.1013	0.1632
Standart sapma	6.581×10^{-4}	0.0013	0.0285
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	449.221	459.015	914.067
MSE	0.0974	0.1038	0.1064
Standart sapma	9.281×10^{-4}	0.0025	0.0089



(a)

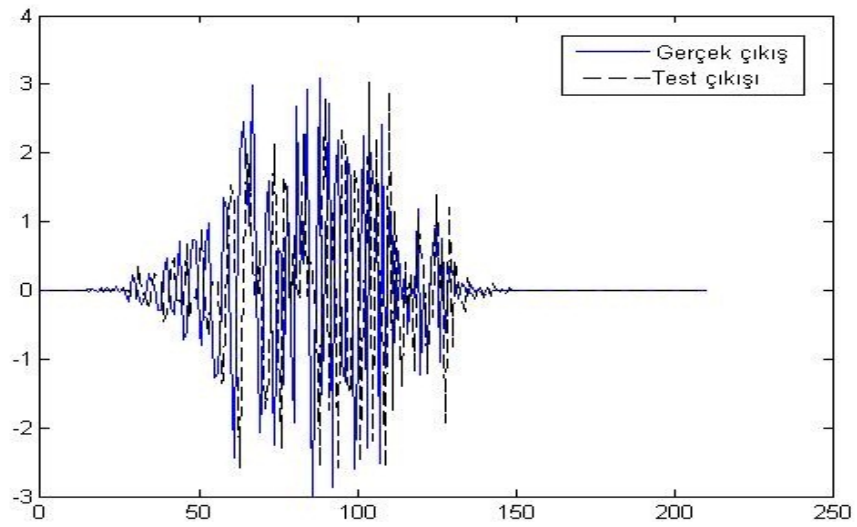


(b)

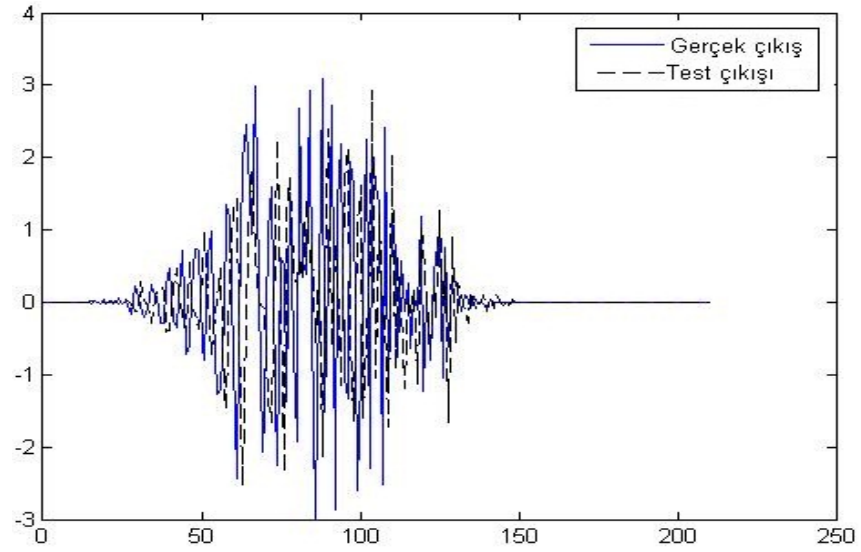


(c)

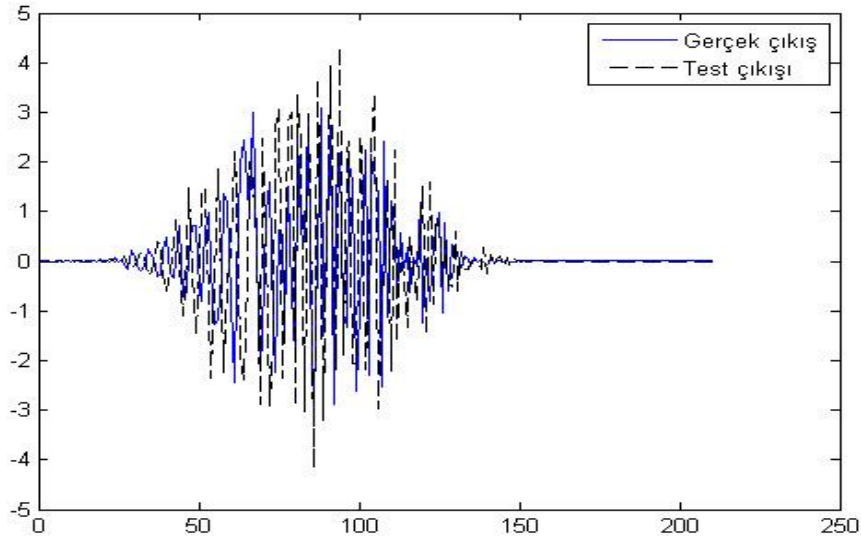
Şekil 3. 25. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.20'de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa



(a)



(b)



(c)

Şekil 3. 26. İkinci yöntemde gerçek sistem çıkışı ile Denklemler 3.20’de önerilen model için test çıktıları:
(a) yak algoritması, (b) ga , (c) ksa

Sistemin modellenmesi için kullanılan her iki yöntem karşılaştırıldığında data sayısının artması ile algoritmaların parametre tahminlerindeki süresinin bu örnek için de artmış olduğu görülmüştür. Ayrıca farklı koloni ve popülasyon sayılarında algoritmalar, farklı hata değerleri ile sistemi modellemiş ve parametre hesabı yapmıştır. YAK algoritmasının bu örnekte GA ve KSA’dan daha iyi performans gösterdiği de görülmüştür. Diğer örnekte olduğu gibi bu örnekte de farklı model yapısının önerilmesi ile algoritmaların sistemi modelleme süresi ve hatası düşürülebilir.

3.2.4.3. Örnek Sistem 3

Burada üçüncü ve son olarak olarak kullanılan sistem verileri, Katholieke Universiteit Leuven, Signals, Identification, System Theory and Automation's Identification Database'den alınmış, esnek bir robot kolunun verileridir. Kol elektrik motorlarından oluşmaktadır. Zemin üzerindeki esnek kolun torkunun etkisi ve tepki olarak robot kolunun ivmesi ölçülmüştür. Modellenen sistemin giriş verisi yapının torku, çıkışı ise esnek kolun ivmesidir. [88].

Bu sistem için Denklem 3.21' de önerilen model yapısı uygulanmış ve parametreler ile sistem MSE' leri hesaplanmıştır. İlk iki örnekte olduğu gibi ilk olarak önerilen model parametrelerinin hesaplanması için 1024 adet datadan ilk 100 data, bulunan parametrelerin testi için ise sonraki 100 adetlik data . Tablo 3.38'de algoritmaların parametre bulmaları için kullanılan kontrol değerleri verilmiştir. Kullanılan kontrol değerleri ile her bir algoritma 30 kez koşturulmuş ve algoritmaların bulduğu parametreler Tablo 2.39'da verilmiştir. Ayrıca YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.40'ta belirtilmiştir.

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + b_4 u(n-4) + b_3 u(n-3) + c_{3,3} y(n-3) u(n-3) + c_{2,4} y(n-2) u(n-4) + c_{1,4} y(n-1) u(n-1) \quad (3.21)$$

Tablo 3. 38. Denklem 3.21'de önerilen modele ait kullanılan kontrol parametreleri

Kontrol Parametreleri	YAK	GA	KSA
Parametre araştırma aralığı	-1+1	-1 +1	-1 +1
Popülasyon boyutu	-	30	30
Koloni sayısı	30	-	-
Klonlama oranı	-	-	0.1
Mutasyon oranı	-	0.3	-
Çaprazlama oranı	-	0.8	-
İterasyon	1000	1000	1000

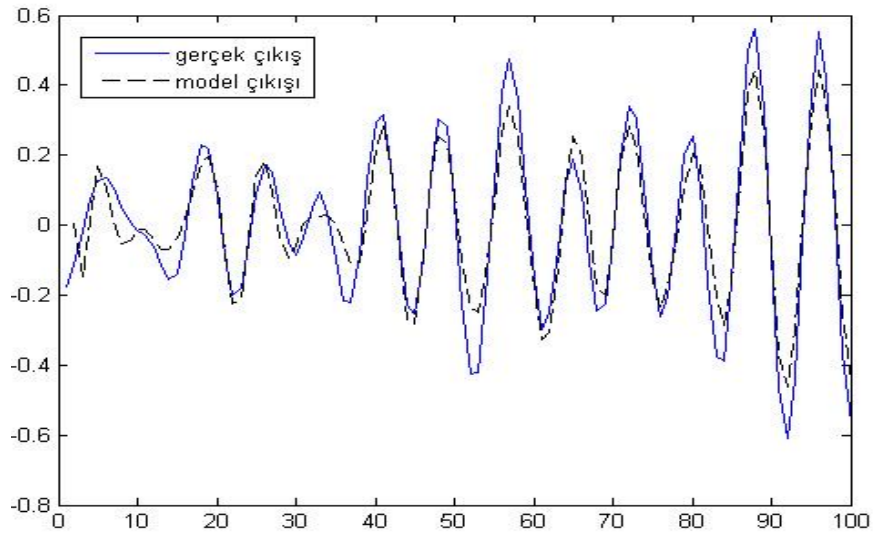
Tablo 3. 39. Denklem 3.21’de önerilen model için algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

Kat Sayı	YAK	GA	KSA
a₁	-1.0000	-1.4013	-1.4691
a₂	0.7414	1.0243	1.1250
b₄	0.3474	0.2046	0.1904
b₃	-0.3984	-0.3011	-0.2677
c_{3,3}	0.2945	0.0731	0.0335
c_{2,4}	-0.5081	-0.3447	-0.2520
c_{1,4}	-0.1797	-0.0019	0.0586
Ortalama süre (sn)	60.1391	54.7338	129.92
MSE	0.0083	0.0044	0.0046
Standart sapma	9.736×10^{-17}	1.03×10^{-6}	0.0011

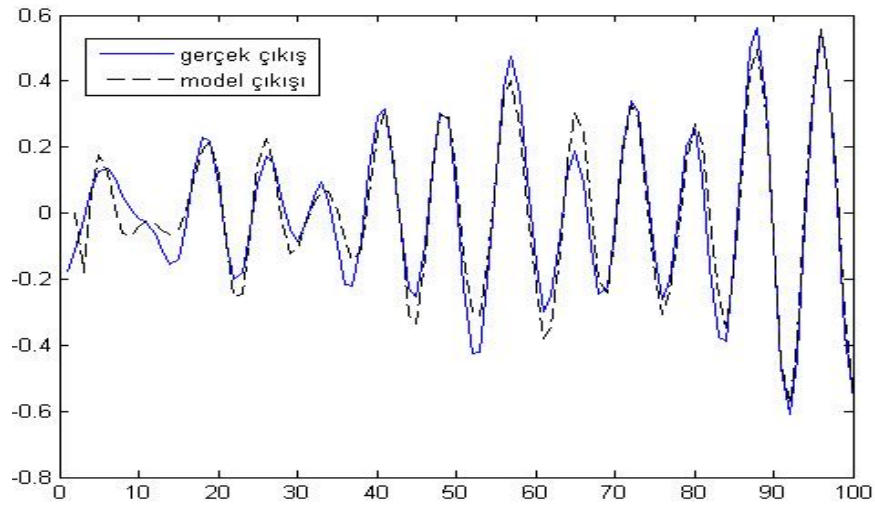
Tablo 3. 40. Denklem 3.21’de önerilen model için koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

Araştırılan değer	YAK	GA	KSA
Koloni sayısı = 20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	35.1463	38.4879	67.42
MSE	0.0083	0.0044	0.0044
Standart sapma	1.730×10^{-16}	1.732×10^{-6}	0.0012
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	60.1391	54.7338	129.92
MSE	0.0083	0.0044	0.0046
Standart sapma	9.736×10^{-17}	1.032×10^{-6}	0.0011
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	56.1655	104.76	267.53
MSE	0.0044	0.0044	0.0082
Standart sapma	1.071×10^{-16}	3.504×10^{-7}	8.52×10^{-4}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	97.5056	114.70	447.08
MSE	0.0044	0.0044	0.0044
Standart sapma	7.230×10^{-17}	2.058×10^{-7}	2.94×10^{-4}

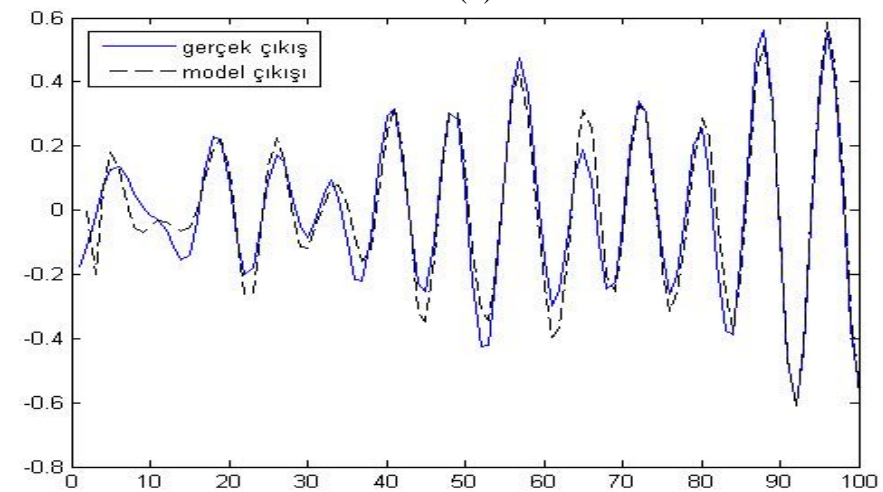
Tablo 3.39’da hesaplanmış olan parametreler için sistemin gerçek çıkışları ile algoritmaların hesapladığı model çıkışı Şekil 3.27’de ve test çıkışları ise Şekil 3.28’de verilmiştir.



(a)

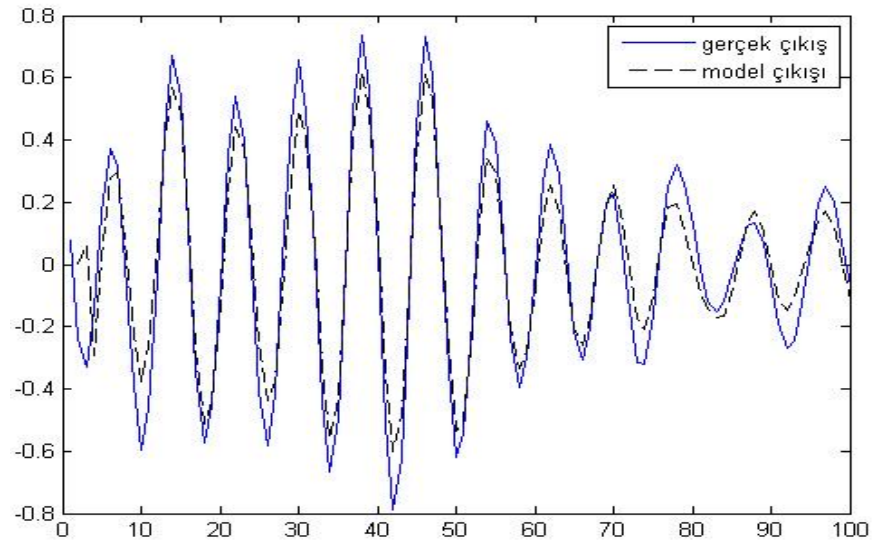


(b)

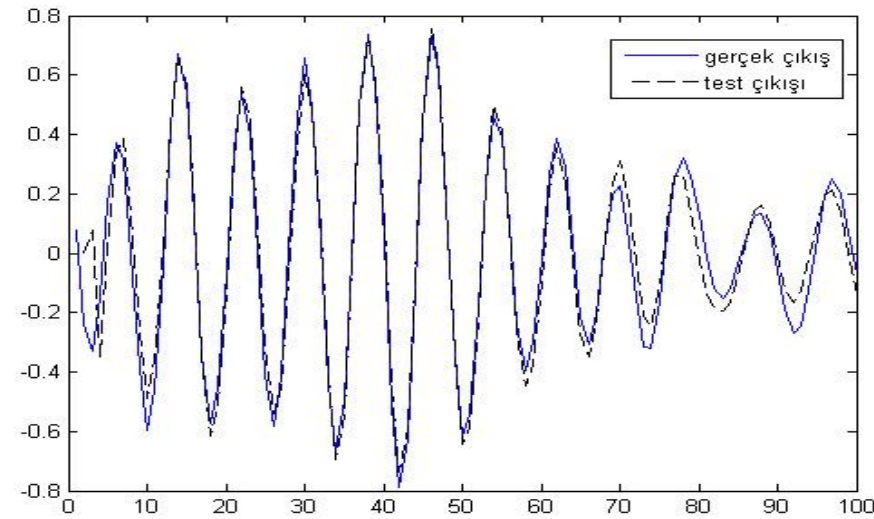


(c)

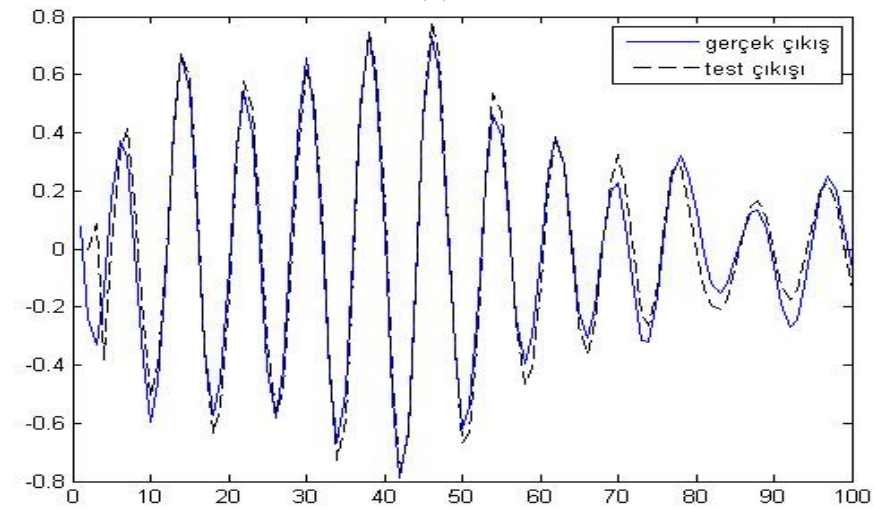
Şekil 3. 27. Gerçek sistem çıkışı ile Denklem 3.21'de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa



(a)



(b)



(c)

Şekil 3. 28. Gerçek sistem çıkışı ile Denklem 3.21’de önerilen model için test çıktıları: (a) yak algoritması, (b) ga , (c) ksa

İlk iki örnekte olduğu gibi bu örnekte de bu ilk yöntemde sistemin geriye kalan 824 adet datasının kullanılmamıştır. İkinci yöntem olarak 1024 adetlik datanın rastgele seçilmiş %80 adet datasını sistem parametrelerinin bulunması için sistemin eğiminde, geriye kalan %20 adetlik data ise bulunan modelin testi için kullanılmıştır.

Tablo 3.38'deki kontrol parametrelerinin kullanılmasıyla ikinci yöntemde alınan verilerle sistem modellenmiş ve algoritmaların bulduğu parametreler Tablo 3.41'de belirtilmiştir. Aynı zamanda Tablo 3.41'de hesaplanmış olan parametreler için sistemin gerçek çıkışları ile algoritmaların hesapladığı model çıkışı Şekil 3.29'da ve test çıkışları ise Şekil 3.30'da verilmiştir.

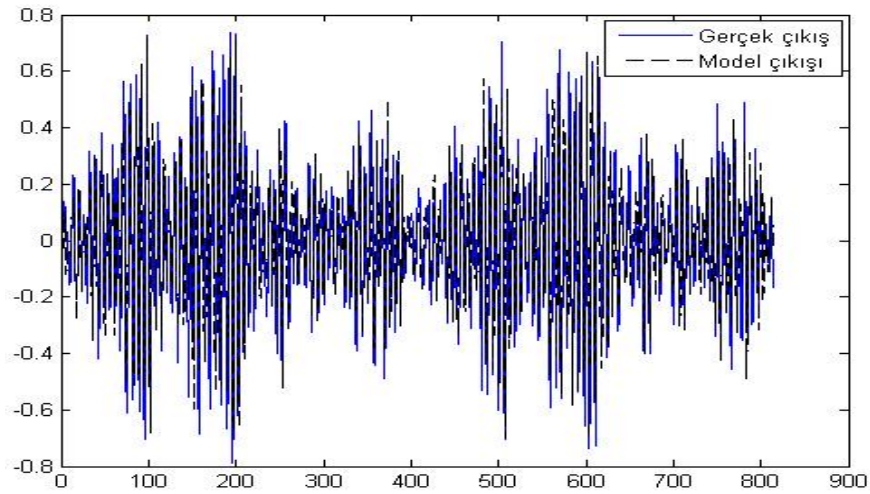
Tablo 3. 41. Denklem 3.21'de önerilen modele ait ikinci yöntemde algoritmaların bulduğu parametreler, MSE değerleri ve parametre bulma süreleri

Kat Sayı	YAK	GA	KSA
a₁	-0.9778	-0.9812	-0.9824
b₀	0.2414	0.2460	0.2500
c_{2,0}	0.2344	0.2333	0.2267
c_{3,1}	-0.2007	-0.1968	-0.1875
c_{4,2}	-0.0378	-0.0502	2.384x10 ⁻⁷
c_{1,3}	0.0300	0.0369	-2.384x10 ⁻⁷
c_{2,4}	-0.0214	-0.0253	2.384x10 ⁻⁷
Ortalama süre (sn)	392.415	389.470	213.432
MSE	0.0169	0.0169	0.0169
Standart sapma	5.003x10 ⁻¹⁷	4.794x10 ⁻⁷	9.688x10 ⁻⁴

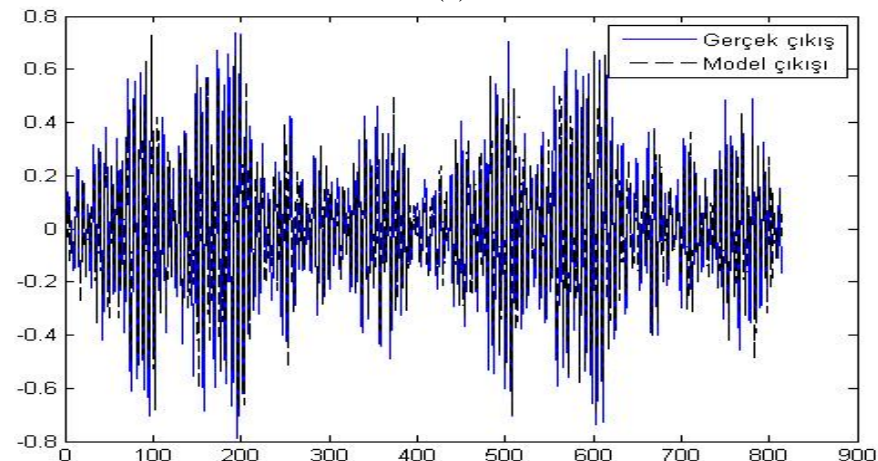
Ayrıca ikinci yöntemde YAK algoritması için koloni sayısı, GA ve KSA için popülasyon değerlerinin 20, 30, 50 ve 70 olması durumunda bu değerlerin algoritmaların parametre bulma sürelerine, en düşük MSE değerlerine ve MSE değerlerinin standart sapmasına etkisi incelenmiş olup, bu durum Tablo 3.42'de belirtilmiştir.

Tablo 3. 42. Denklem 3.21’de önerilen model için ikinci yöntemde koloni ve popülasyon sayılarının ortalama süre ve hatalara etkisi

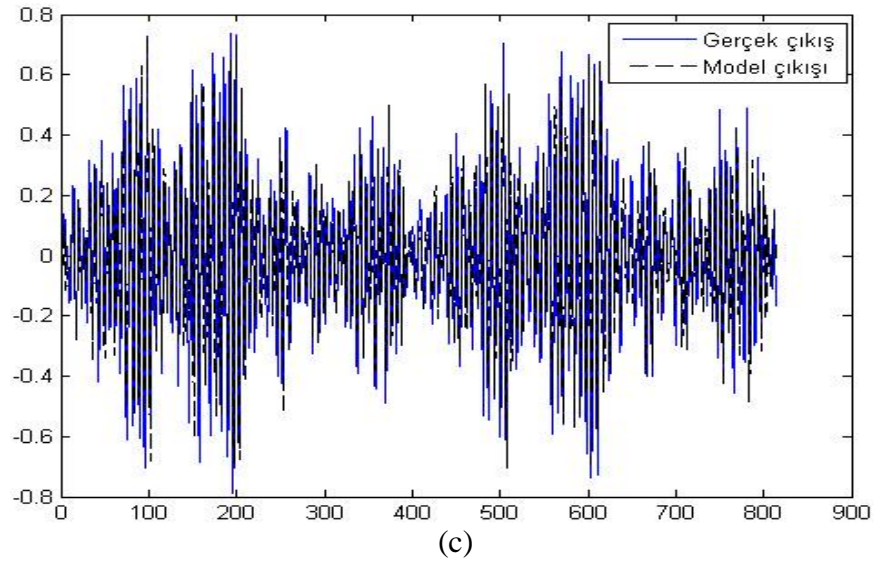
Araştırılan değer	YAK	GA	KSA
Koloni sayısı =20, Popülasyon sayısı = 20			
Ortalama Süre(sn)	131.720	135.193	124.304
MSE	0.0169	0.0169	0.0169
Standart sapma	3.213×10^{-16}	3.141×10^{-6}	3.204×10^{-4}
Koloni sayısı = 30, Popülasyon sayısı = 30			
Ortalama Süre(sn)	392.415	389.470	213.432
MSE	0.0169	0.0169	0.0169
Standart sapma	5.003×10^{-17}	4.794×10^{-7}	9.688×10^{-4}
Koloni sayısı = 50, Popülasyon sayısı = 50			
Ortalama Süre(sn)	328.432	321.816	508.687
MSE	0.0169	0.0169	0.0176
Standart sapma	8.760×10^{-17}	4.458×10^{-7}	4.988×10^{-4}
Koloni sayısı = 70, Popülasyon sayısı = 70			
Ortalama Süre(sn)	468.354	451.312	1040
MSE	0.0169	0.0169	0.0169
Standart sapma	4.246×10^{-17}	3.338×10^{-7}	2.230×10^{-5}



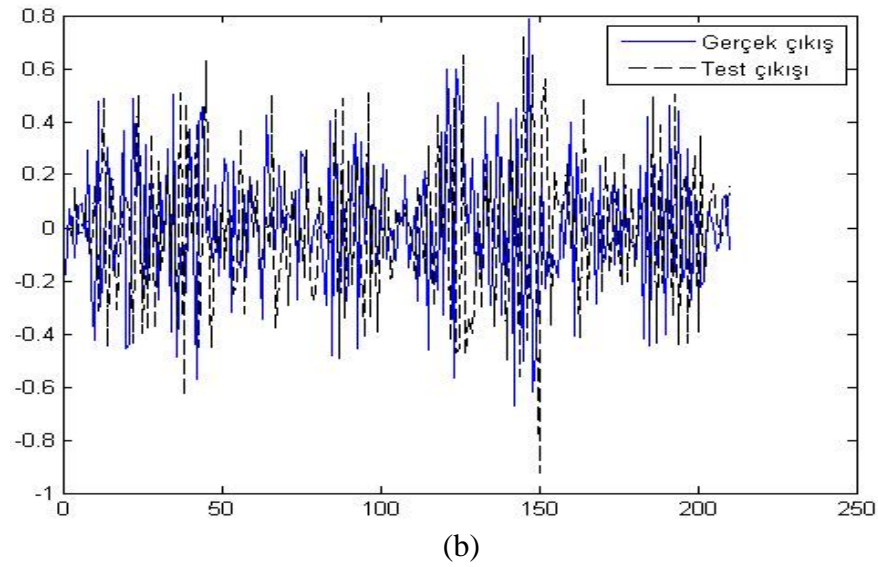
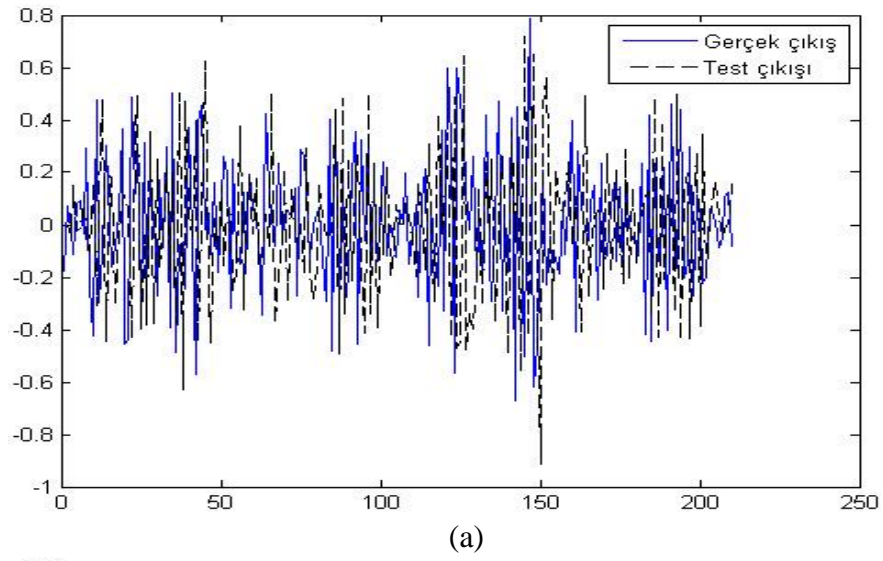
(a)

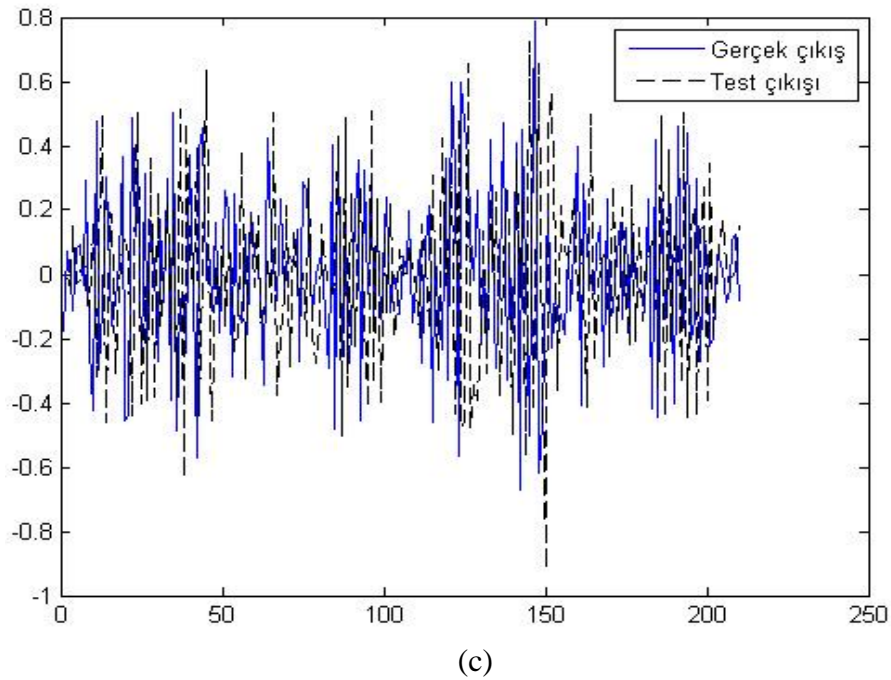


(b)



Şekil 3. 29. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.21'de önerilen modelin çıkışları: (a) yak algoritması, (b) ga , (c) ksa





(c)
Şekil 3. 30. İkinci yöntemde gerçek sistem çıkışı ile Denklem 3.21’de önerilen model için test çıktıları: (a) yak algoritması, (b) ga , (c) ksa

Son örnekte de sistemin modellenmesi için kullanılan her iki yöntem karşılaştırıldığında data sayısının artması ile algoritmaların parametre tahminlerindeki süresinin bu örnek için de artmış olduğu görülmüştür. Fakat ilk iki örnekten farklı olarak sistem datalarının yaklaşık %80’ninin kullanıldığı bu ikinci yöntemde farklı koloni ve popülasyon sayılarında da algoritmalar, aynı hata değerleri ile sistemi modellemiş ve parametre hesabı yapmıştır. Buda sistem veya önerilen model yapısı ile ilişkili olduğu düşünülmektedir. Ayrıca sistemin modellenmesi için eğitim ve test işlemleri için farklı sayıda örneklerin alınması da hataların değişiminde etkili olabilir. Bu sistem içinde farklı model yapısının önerilmesi ile algoritmaların sistemi modelleme süresi ve hatası düşürülebilir.

İlk üç örnekten de anlaşılacağı üzere doğrusal olmayan gerçek sistem modellemelerinde bazı popülasyon ve koloni sayılarında YAK algoritması hem GA hem de KSA’dan daha kısa sürede ve daha düşük hata ile sistemi modelleyebileceği görülmüştür.

BÖLÜM 4

TARTIŞMA, SONUÇ VE ÖNERİLER

4.1. Tartışma

Sürü zeka yaklaşımını temel alan algoritmalar gün geçtikçe çoğalmakta, geliştirilmekte ve sistem modelleme işlemlerinde çok daha fazla kullanılmaya başlamıştır. Bu tür algoritmaların her türlü sistemde iyi sonuç vermesi beklenmemelidir. Fakat bu tür algoritmaların geliştirilmesi ile daha iyi sonuçların elde edileceği de unutulmamalıdır.

4.2. Sonuç ve Öneriler

Örnek sistemlerin sonuçları incelendiğinde, YAK algoritmasının doğrusal parametrik sistemlerin parametre hesaplama değerleri bakımından, GA'ya oldukça yakın bir sürede hem GA'dan hem de KSA'dan daha düşük bir hata ile çok başarılı bir şekilde parametre hesabı yaptığı görülmüştür. Hata yaklaşım eğrilerinden de anlaşılacağı üzere hatayı minimize edip optimum çözüme kararlı bir şekilde yaklaşan algoritma YAK algoritmasıdır.

Doğrusal olmayan parametrik sistemlerin parametre hesaplama performansları bakımından, YAK algoritmasının çok başarılı bir şekilde parametre hesabı yaptığı görülmüştür. Parametre bulma süresi göz önüne alındığında ise doğrusal parametrik sistemde olduğu gibi YAK algoritması KSA'dan daha hızlı ve GA'ya çok yakın sürede parametre hesabı yaptığı görülmüştür. Doğrusal olmayan parametrik Bilineer sistemlerde ise YAK algoritması hem GA'dan hemde KSA'dan daha hızlı çözüme gittiği görülmektedir.

Doğrusal olmayan gerçek sistemlerde benzer durum söz konusu olup, popülasyon ve koloni sayılarının değişimine bağlı olarak hata oranları ve süre değişimle birlikte YAK

algoritması, GA'ya ve KSA'na göre her bir kořturmada bulunan hatalarda daha düşük bir standart sapma deęeri ile parametre hesabı yaptıęından, doęrusal olmayan gerçek sistem parametre hesaplamasında da YAK algoritmasının başarılı olduęu görölmüřtür. Standart sapmanın düşük olması her kořturmada bulunan hatanın bir birine çok yakın olması anlamına gelir ki bu da algoritmanın parametre hesaplamasındaki kararlılıęının göstergesidir.

Parametre bulma süresi açısından deęerlendirildięinde ise doęrusal olmayan gerçek sistemlerin modellenmesinde YAK algoritmasının GA'ya göre belirli koloni sayılarında daha geç sürede ve KSA'na göre de daha erken sürede parametre hesabı yaptıęı görölmüřtür.

Kullanılan sistemler için farklı model yapıları uygulanıp parametre kestiriminde, YAK algoritmasının daha erken sürede ve daha düşük MSE deęerlerinde sonuç vermesi beklenebilir. Aynı zamanda farklı özelliklere sahip bilgisayar ve farklı sürüm Matlab programlarının kullanılması, modellemede algoritmaların parametre hesaplamasındaki süreyi etkileyeceęi unutulmamalıdır.

KAYNAKLAR

1. Söderstrom T., 1989. System Identification, Prentice-Hall, 612 pp.
2. Zorlu H., 2004. Doğrusal Olmayan Sistemlerin Yapay Sinir Ağları Kullanılarak Kimliklendirilmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri, 64 s.
3. Özer Ş., 1988. Adaptif Lattice Arma Modeli ile Lineer Sistemlerin Modellenmesi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 60 s.
4. Widrow, B., Stearns, D., 1985. Adaptive Signal Processing, Prentice Hall, 474 pp.
5. Honig, H.L., Messerschmitt, D.G., 1984. Adaptive Filters Structures, Algorithms and applications, Kluwer Academic Publishers, 377 pp.
6. Isidori A., 1985. Nonlinear Control Systems : An Introduction, Lecture Notes in Control An Information Science, 72, Springer-Verlag, Berlin, 293 pp.
7. Ljunq, L., 2010. Systems Identification Toolbox User's Guide, The MathWorks, Inc., Mass., USA., 566 pp.
8. Ljunq, L., 1987. System Identification : Theory For The User, Englewood Cliffs, NJ, Prentice-Hall, 609 pp.
9. Özgünel, S., Kayran, A. H., Panayirci, E., 1991. Nonlinear channel equalization and identification, **Proc. Of Int. Conf. on Digital Signal Proc., Florence**, 260-265.
10. Özden, M. T., Kayran, A. H., Panayirci, E., 1996. Adaptive volterra filtering with complete lattice orthogonalization, **IEEE Trans. On Signal Proc.**, **44** (8): 2092-2098.
11. Griffith, D.W., Arce, G.R., 1997. Partially decoupled volterra filters: Formulation and LMS adaptation. **IEEE Trans. on Signal Processing**, **45** (6): 1485-1494.
12. Lee, J., Mathews, V.J., 1994. A stability condition for certain bilinear systems. **IEEE Trans. on Signal Processing**, **42** (7): 1871-1873.
13. Cowan, C.F, Grant, P.M., 1984. Nonlinear system modelling - concept and application. **Proceedings Of IEEE Int. Conference On Acoustic Speech and Signal Processing, San Diego, California**, **45** (6): 1-4.
14. Priestley, M. B., 1988. Nonlinear and Non-Stationary Time Series Analysis, Academic Press, 237 pp.

15. Rauf, F., 1993. Nonlinear Adaptive Filtering : A Unified Approach, Ph.D. Thesis, Boston University, Boston, 392 pp.
16. Coşkun, A., 2007. Yapay zeka optimizasyon teknikleri: literatür değerlendirmesi, **Doğu Anadolu Bölge Araştırmaları**, 5 (2):142-146.
17. Sage A.P., Melsa J.L., 1971. System Identification, Academic Press, INC, 221 pp.
18. Aström K.J., Bohlin I., 1965. Numerical identification of linear system from normal operating records, **IEEE Automatic Control Group, Institute of Electrical and Electronics Engineers. Automatic Control Group**
19. Kızılaslan K., 2004. Çimento Endüstrisinde Hammadde Harmanlama Prosesinin Klasik ve Akıllı Yöntemler Kullanarak Modellenmesi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 113 s.
20. Melsa S., System Identification, Mathematics in Science and Engineering Vol.80
21. Kaplan A., 1995. Sistem Parametrelerinin Modern Spektral Analiz Yöntemleri Kullanılarak Belirlenmesi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri, 82 s.
22. Gauss, C. F., 1963. Theory of the Motion of Heavenly Bodies, Dover, 410 pp.
23. Zadeh, L.A., 1962. From circuit theory to system theory, **Proc IRE**, 50 (1): 856-865.
24. Özer, Ş., Sağıroğlu, Ş., Kaplan, A., 2001. Performance analysis of algorithms on linear arma models. **Proc. Of the Int. Symposium Computer and Information Science XVI**, 445-451.
25. Özer Ş., Sağıroğlu, Ş., Kaplan, A., 2002 AR Sistem Modellemede Kullanılan Adaptif ve Yapay Zeka Metotlarının Karşılaştırılması, **Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi**, 18 (1-2):44-50.
26. H.Hayes, M. 1996. Statistical Digital Signal Processing and Modeling, Jhon-Wiley&Sons, Inc, America, 608 pp.
27. Ljung L., Söderström T., 1985. Theory and Practice Of Recursive Identification, Cambridge, MA, MIT Press, 529 pp.
28. Nelles, O., 2001. Nonlinear System Identification, Springer-Verlag Berlin Heidelber, Germany, 576 pp.
29. Schetzen, M., 2006. The Volterra&Wiener Theories of Nonlinear Systems, Krieger Publishing Company, Germany, 595 pp.

30. Hu, W., 2011. Electronic and Signal Prosesing, Springer-Werlag Berlin Heilberg, Germany, 1060 pp.
31. Tapkan, P., Özbakır L., Baykasoğlu, A., 2001. Arı algoritması ve genelleştirilmiş atama problemi: Farklı komşuluk yapılarının karşılaştırılması. **Endüstri Mühendisliği Dergisi**, **21** (2) : 2-13.
32. Kıran, M.S., Gündüz M., Şahman M.A., 2009. Arı kolonisi optimizasyon algoritması kullanarak en kısa yol bulma, **EMO 9. Öğrenci Kurultayı Bildirisi**
33. Yıldırım, M., 2008. Yapay Sinir Ağları ile Lineer Olmayan Mekanik ve Elektriksel Sistemlerin Modellenmesi. Dicle Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Diyarbakır, 147 s.
34. Aybars, U., Aydın, D., 2006. Ant system algoritmasının java ile görselleştirilmesi, **Akademik Bilişim Bildirisi, Bildiri No: 53**
35. Ülker, Ö. ve Özcan, E., 2003. Çizge boyama problemler için evrimsel tabu arama algoritması, **TBD Bilişim Kurultayı Bildirisi.**
36. Öztürk. A., Çobanlı, S., Duman, S., Tosun S., Döşoğlu, K., 2011. Yapay arı koloni algoritması ile elektrik güç sistemi optimal yakıt maliyetinin belirlenmesi. **6. International Advanced Tecnologies Symposium**, Elazığ.
37. Karaboğa, D., 2005. An Idea on honey bee swarm for numerical optimization, **Technical report-TR06.**
38. Karaboğa, D., 2010. Artificial bee colony algorithm, **Scholarpedia**, **5** (3): 6915-6915.
39. Karaboğa, D., Baştürk, B., 2008. On the performance of artificial bee colony (abc) algorithm, **Applied Soft Computing**, 687-697.
40. Karaboğa, D. 2011 Yapay Arı Koloni Algoritması, Yapay Zeka Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, İstanbul, 231 s.
41. Karaboğa, D. ve Akay, B., 2010. A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization, **Information Science, Available Online.**
42. Karaboğa D. ve Akay B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, **Journal of Global Optimization**, **39** (3) : 459-471.
43. Karaboğa D., Ozturk C., Karaboğa N. Görkemli B., 2012. Artificial Bee Colony Programming for Symbolic Regression, **Information Sciences**, **209** (1):1-15

44. Akay, B., 2009. Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi. Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, Kayseri, 301 s.
45. Ogunfunmi, T., 2007. Adaptive Nonlinear System Identification Springer Science+Business Media, 229 pp.
46. Griffith, D.W. ve Arce, G.R., 1997. Partially decoupled volterra filters: Formulation and LMS adaptation, **IEEE Trans. on Signal Processing**, s. 1485-1494.
47. Prochazka, A., Neumann, R., 1975. High frequency distortion analysis of a semiconductor diode for catv applications, **IEEE Trans. on Consumer Electronics, CE-21**, s: 120-129.
48. Nam, S.W., Powers, E.J., 1994. Application of higher order spectral analysis to cubically nonlinear system identification, **IEEE Trans. on Signal Processing**, **42**, 2124-2135.
49. Giannakis, G.B., Serpedin, E., 2001. A Bibliography on Nonlinear System Identification, **Signal Processing**, **81**, 533-580.
50. Khurram, M. U., 1994. Fast Learning Nonlinear Adaptive Filtering Structures, Ph.D. Thesis, University Of Boston, 310 pp.
51. Koh, T., Powers, E. J., 1985. Second order volterra filtering and its application to nonlinear system identification, **IEEE Trans. on ASSP**, **33**, 1445-1455.
52. Mathews, V.J., 1987. A fast recursive least squares adaptive nonlinear filter, proceedings of 21st annual conference on systems, **Signals and computers, Pacific Grove, California**.
53. Haykin, S., 2002. Adaptive Filter Theory, Prentice-Hall, 920 pp.
54. Zorlu, H., 2011. Esnek Hesaplama Teknikleri ile Doğrusal Olmayan sistemlerin Kimliklendirilmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kayseri, 110 s.
55. Kalluri, S., Arce, G.R., 1999. A general class of nonlinear adaptive filtering algorithms, **IEEE Trans. on Signal Processing**, **47**, 2262-2272.
56. Gonzalez, J.G., Griffith, D.W., Arce, G.R., 1998. Matched myriad filtering for robust communications, In Proc. Of The 1998 Conf. on Information Science and Systems, Princeton University.
57. Kalluri, S., Arce, G.R., 1999. A general class of nonlinear adaptive filtering algorithms, **IEEE Trans. on Signal Processing**, **47**, 2262-2272.

58. Kalluri, S., Arce, G.R., 1998. Adaptive weighted myriad filter algorithms for robust signal processing in α -stable noise environments, **IEEE Trans. on Signal Processing**, **46**, 322-334.
59. Dorigo, M., Maniezzo, V., Colorni, A., 1991. Positive Feedback as a Search Strategy, **Technical Report N. 91-016** , Politecnico di Milano.
60. Holland, J., 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.
61. Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, USA
62. Gen, M., Cheng, R., 2000. Genetic Algorithms and Engineering Optimization, John Wiley&Sons., Inc, USA
63. Emel, G.G., Taşkın, Ç., 2002. Genetik algoritmalar ve uygulama alanları, **İİBF Dergisi, Uludağ Üniversitesi**, **21** (1) : 129-152
64. İşçi, O., Korukoğlu, S. 2003. Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama, **İİBF Dergisi, Celal Bayar Üniversitesi**, **10** (2) : 191-208
65. Gülsün, B., Tuzkaya, G., Duman C., 2009. Genetik algoritmalar ile tesis yerleşimi tasarımı ve bir uygulama, **Doğuş Üniversitesi Dergisi**, **10** (1) : 73-87
66. Eroğlu, E., Keskintürk, T., 2006. Maksimum akış probleminin çözümü için genetik algoritma yaklaşımı, **6. Üretim Araştırmaları Sempozyumu, Bildiriler Kitabı**, s.129-135
67. Biroğul, S., Güvenç, U., 2007. Genetik algoritma ile çözümü gerçekleştirilen atölye çizelgeleme probleminde ürün sayısının etkisi, **Akademik Bilişim Konferansı Bildirileri**, s. 613-619
68. Karaboğa, N., 1994. Sayısal Filtre Katsayılarının Genetik Algoritma Kullanılarak Yuvarlanması, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kayseri, 109 s.
69. Kaya, S., 2006. Operasyonel Sabit İş Çizelgeleme Problemlerinin Genetik Algoritmalar ile Çözümü. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya, 76 s.
70. Engin, O., 2006, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Artırılmasında Parametre Optimizasyonu. İTÜ Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul 120 s

71. Bağış, A., 1996. Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri 76 s.
72. Dede, T., 2003. Değer Kodlaması Kullanılarak Kafes Sistemlerin Genetik Algoritma ile Minimum Ağırlıklı Boyutlandırılması. Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tez, Trabzon 72 s.
73. Babayiğit, B., Güney. K.,2006. Klonal seçme algoritması kullanılarak genlik ve faz kontrolü ile doğrusal anten dizi diyagramında sıfırların üretilmesi, **Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu Bildirisi**, s. 72-76.
74. Castro D., Von Zuben, F.J., 2000. The clonal selection algorithm with engineering application, In Workshop Proceedings of GECCO'00, **Workshop on Artificial Immune Systems and their Applications**, Las Vegas, 36-37.
75. Castro, D., Von Zuben, F.J., 2001. Learning and optimization using clonal selection principle. **IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems**, 239-251.
76. Aslantas, V., Ozer, Ş. Ozturk, S. 2007. A novel clonal selection algorithm based fragile watermarking method, **LNCS, 4628**, 358-369.
77. Zorlu, H., 2011. Esnek Hesaplama Teknikleri ile Doğrusal Olmayan Sistemlerin Kimliklendirilmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kayseri, 110 s.
78. Özer, Ş., Baran, İ., 2014. Doğrusal parametrik ve doğrusal olmayan gerçek sistemlerin yapay arı kolonisi algoritması kullanılarak modellenmesi, **Dicle Üniversitesi, Mühendislik Dergisi, 5 (2) :Basım Aşamasında**
79. Özer, Ş., Sağıroğlu, Ş., Zorlu, H., 2003. Arma sistem modellemede klasik ve yapay sinir ağları algoritmalarının karşılaştırılması, **Elektrik Elektronik -Bilgisayar Mühendisliği 10. Ulusal Kongresi**.
80. Baran, A., 2001. Maksimum Olabilirlik Yaklaşımı ile Frekans Domeninde Lineer Sistemlerin Kimliklendirilmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri, 47 s.
81. Zorlu, H., Özer, Ş., 2010. Doğrusal olmayan volterra sistemlerin diferansiyel gelişim algoritması kullanılarak kimliklendirilmesi, **ELECO-2010, Bildiriler CD'si**.

82. Yao, L., 1996. Genetic algorithm based identification of nonlinear systems by sparse volterra filters, *Emerging Technologies and Factory Automation*, s. 327-333.
83. Özer, Ş., Zorlu, H., 2011. Identification of bilinear systems using differential evolution algorithm, **Sadhana Indian Academy of Sciences**, **36** (3) : 281-297.
84. Özer. Ş., Baran, İ., 2012. Doğrusal olmayan bilineer ve doğrusal olmayan gerçek sistemlerin yapay arı kolonisi algoritması ile modellenmesi, **Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı Bildiriler Kitabı**, **2** (1) : 845-850
85. Zorlu, H., Özer. Ş., 2009. Doğrusal olmayan sistemlerin, klonal seçme algoritması kullanılarak kimliklendirilmesi, **IEEE 17. Sinyal İşleme ve İletişim Uygulamaları Kurultayı**.
86. Zorlu, H., Özer, Ş, 2004. Yapay sinir ağları ile bilinear sistemlerin kimliklendirilmesi (Bilinear system identification with artificial neural network), **ASYU-INISTA, Turkey** 179–182
87. Bağış, S., 2009. Yapay Zeka Algoritmaları Kullanılarak Sistem Modelleme. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri, 66 s.
88. De Moor B.L.R., DaISy: Database for the Identification of Systems, Department of Electrical Engineering, ESAT/SISTA, K.U.Leuven, Belgium, (Web sayfası: <http://homes.esat.kuleuven.be/~smc/daisy/>) (Erişim Tarihi: Mart 2014)
89. Feron E., Brenner, M., Paduano, Turevsky, J., 1998. Time-frequency analysis for transfer function estimation and application to flutter clearance, in *AIAA J. on Guidance, Control&Dynamics*, **21** (3) : 375-382

ÖZGEÇMİŞ

İbrahim BARAN 1985 yılında Diyarbakır'da doğdu. İlk ve orta öğrenimini Kazım Karabekir İ.Ö.O.'da tamamladıktan sonra liseyi Melik Ahmet Lisesi'nde okudu. 2003 yılında ise Dicle Üniversitesi Elektrik Elektronik Mühendisliği Bölümün'de okumaya hak kazandı ve aynı bölümden Elektrik-Elektronik Mühendisi ünvanıyla mezun oldu. Mezun olduktan sonra sırasıyla AnelTech A.Ş. ve Asur İletişimde Transmisyon Sorumlusu Yardımcılığı görevlerinde bulundu. 2009 yılı sonu itibariyle Dicle Üniversitesi Çüngüş Mehmet Adıgüzel MYO'da Öğretim Görevlisi olarak çalışmaya başlayan İbrahim BARAN, halen bu kadroda görevine devam etmektedir.

Telefon : +90 412 241 10 00 - 7827

Belgegeçer : +90 412 541 30 14

E-posta : ibaran@dicle.edu.tr

Yazışma Adresi : Dicle Üniversitesi, Çüngüş Mehmet Adıgüzel Meslek
Yüksekokulu Elektrik Bölümü, 21870 Çüngüş/Diyarbakır