

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

KAOTİK SİSTEMLERİN KLASİK VE ZEKİ YAKLAŞIMLAR İLE KONTROLÜ

DOKTORA TEZİ

Uğur Erkin KOCAMAZ

Enstitü Anabilim Dalı	:	ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
Enstitü Bilim Dalı	:	ELEKTRONİK
Tez Danışmanı	:	Doç. Dr. Yılmaz UYAROĞLU

Kasım 2018

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**KAOTİK SİSTEMLERİN KLASİK VE ZEKİ
YAKLAŞIMLAR İLE KONTROLÜ**

DOKTORA TEZİ

Uğur Erkin KOCAMAZ

**Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ**
Enstitü Bilim Dalı : ELEKTRONİK

Bu tez 07/11/2018 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.


**Doç. Dr.
Yılmaz UYAROĞLU**
Jüri Başkanı


**Prof. Dr.
Resul KARA**
Üye


**Prof. Dr.
Cemil ÖZ**
Üye


**Doç. Dr.
İrfan YAZICI**
Üye


**Prof. Dr.
Ali ÖZTÜRK**
Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Uğur Erkin KOCAMAZ

07.11.2018

TEŞEKKÜR

Bu çalışmanın ortaya çıkmasında bana yardımcı olan, ilgi ve desteğini hiç eksiltmeyen, yol gösterici olan, fikirlerime önem veren, engin bilgi ve tecrübesiyle beni yönlendiren değerli danışmanım Doç. Dr. Yılmaz UYAROĞLU'na teşekkür ederim.

Benim bu aşamaya gelmemde en çok emeği geçen, her zaman maddi ve manevi desteklerini arkamda hissettiğim annem babam Cemile ve Numan KOCAMAZ'a ve kardeşim Çiğdem ALTUN'a en içten saygı, sevgi ve şükranlarımı sunarım.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	v
TABLolar LİSTESİ	viii
ÖZET	ix
SUMMARY	x
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
MATERYAL VE YÖNTEM	5
2.1. Kaotik Sistemler	5
2.2. Kaotik Elektronik Sistemler	9
2.2.1. Chua devresi	9
2.2.2. Bonhoeffer–van der Pol devresi	12
2.2.3. Colpitts devresi	15
2.3. Kaotik Sistemlerin Kontrolü	18
2.4. Yapay Zekâ	21
2.4.1. Yapay sinir ağları	22
2.4.1.1. Yapay sinir ağlarının tarihçesi	24
2.4.1.2. Biyolojik sinir hücreleri ve yapay sinir ağları	26
2.4.1.3. Aktivasyon fonksiyonları	27
2.4.1.4. Yapay sinir ağları modelleri	29

2.4.1.5. Çok katmanlı yapay sinir ağılar	31
2.4.1.6. Çok katmanlı yapay sinir ağlarının eğitilmesi	33
2.4.2. Bulanık mantık	40
2.4.2.1. Üyelik fonksiyonu ve türleri	41
2.4.2.2. Bulanık mantık tabanlı sistemlerin temel yapısı	44
2.4.3. Sinirsel-bulanık ağlar (ANFIS)	50
2.5. Kaotik Sistemlerin Yapay Zekâ ile Kontrolü	53

BÖLÜM 3.

KAOTİK SİSTEMLERİN KONTROLÜ	57
3.1. Chua Kaotik Sisteminin Kontrolü	57
3.1.1. Kayma kipli kontrol	57
3.1.2. Yapay sinir ağılar ile kontrol	59
3.1.3. Kayma kipli yapay sinir ağılar ile kontrol	62
3.1.4. Simülasyon sonuçları	63
3.2. Bonhoeffer–Van Der Pol Kaotik Sisteminin Kontrolü	70
3.2.1. Pasif kontrol	70
3.2.2. Bulanık mantık ile kontrol	74
3.2.3. Pasif bulanık mantık ile kontrol	78
3.2.4. Simülasyon sonuçları	79
3.3. Colpitts Kaotik Sisteminin Kontrolü	85
3.3.1. Doğrusal geri-beslemeli kontrol	85
3.3.2. Sinirsel-bulanık ağlar ile kontrol	88
3.3.3. Doğrusal geri-beslemeli sinirsel-bulanık ağlar ile kontrol	89
3.3.4. Simülasyon sonuçları	91

BÖLÜM 4.

SONUÇ VE ÖNERİLER	99
KAYNAKLAR	102
ÖZGEÇMİŞ	112

SİMGELER VE KISALTMALAR LİSTESİ

ADALINE	: Adaptif lineer nöron
ANFIS	: Uyarlamalı sinirsel-bulanık çıkarım sistemi
ANFISK	: Uyarlamalı sinirsel-bulanık çıkarım sistemi tabanlı kontrolör
ART	: Adaptif rezonans teorisi
BK	: Bulanık kontrolör
DGBK	: Doğrusal geri-beslemeli kontrolör
GRNN	: Genel regresyon ağları
IEEE	: Institute of electrical and electronics engineering
KKK	: Kayma kipli kontrolör
LVQ	: Öğrenmeli vektör kuantalama
OGY	: Ott, Grebogi ve Yorke
PBK	: Pasif bulanık kontrolör
PK	: Pasif kontrolör
PNN	: Probabilistik ağlar
RBF	: Radyal tabanlı fonksiyonlar
SMIB	: Tek-makine sonsuz-bara
SOM	: Kendi kendine öğrenme nitelik haritaları
YSA	: Yapay sinir ağları
YSAK	: Yapay sinir ağları tabanlı kontrolör

ŞEKİLLER LİSTESİ

Şekil 1.1. Lorenz'in birbirinden uzaklaşan iki hava tahmini grafiği (a) x sinyali, (b) y sinyali, (c) z sinyali	3
Şekil 2.1. Kaotik Lorenz sisteminin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali	8
Şekil 2.2. Kaotik Lorenz sisteminin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi	8
Şekil 2.3. Kaotik Lorenz sisteminin üç boyutlu faz yüzeyi grafiği	9
Şekil 2.4. Chua devresi (a) devre diyagramı, (b) doğrusal olmayan direncin akım-gerilim karakteristiği (Hanbay ve ark., 2007)	9
Şekil 2.5. Kaotik Chua devresinin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali	10
Şekil 2.6. Kaotik Chua devresinin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi	11
Şekil 2.7. Kaotik Chua devresinin üç boyutlu faz yüzeyi grafiği	11
Şekil 2.8. Bonhoeffer-van der Pol devresi (Nishiuchi ve ark., 2006)	12
Şekil 2.9. Kaotik Bonhoeffer-van der Pol devresinin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali	14
Şekil 2.10. Kaotik Bonhoeffer-van der Pol devresinin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi	14
Şekil 2.11. Kaotik Bonhoeffer-van der Pol devresinin üç boyutlu faz yüzeyi grafiği	15
Şekil 2.12. Colpitts devresi (Cenys ve ark., 2003)	16
Şekil 2.13. Kaotik Colpitts devresinin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali	17
Şekil 2.14. Kaotik Colpitts devresinin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi	17

Şekil 2.15. Kaotik Colpitts devresinin üç boyutlu faz yüzeyi grafiği	18
Şekil 2.16. Yapay sinir hücresinin yapısı	23
Şekil 2.17. Biyolojik sinir hücrelerinin yapısı	26
Şekil 2.18. Yapay sinir ağı modeli	27
Şekil 2.19. Lineer fonksiyon	28
Şekil 2.20. Sigmoid fonksiyonu	28
Şekil 2.21. Tanjant-sigmoid fonksiyonu	28
Şekil 2.22. Üç katmanlı yapay sinir ağı yapısı	32
Şekil 2.23. Çizgi üyelik fonksiyonları (a) çizgisel artan fonksiyon, (b) çizgisel azalan fonksiyon	42
Şekil 2.24. Üçgen üyelik fonksiyonu	42
Şekil 2.25. Yamuk üyelik fonksiyonu	43
Şekil 2.26. Çan biçimli üyelik fonksiyonu	43
Şekil 2.27. Ortamın sıcaklığını gösteren üyelik fonksiyonları	44
Şekil 2.28. Bulanık mantık tabanlı bir sistemin genel yapısı	44
Şekil 2.29. Mamdani maksimum-minimum bulanık çıkarım yöntemi	47
Şekil 2.30. Maksimum-çarpım bulanık çıkarım yöntemi	47
Şekil 2.31. Sugeno bulanık çıkarım yöntemi	49
Şekil 2.32. Durulaştırma yöntemleri (a) ağırlık merkezi, (b) en büyüklerin ortası, (c) ağırlıklı ortalama, (d) maksimum üyelik	49
Şekil 2.33. ANFIS mimarisi	51
Şekil 3.1. Yapay sinir ağı ile kaotik sistemlerin kontrol modeli	61
Şekil 3.2. Kayma kipli kontrol ve YSA ile tasarlanan kaotik sistemlerin kontrol modeli	62
Şekil 3.3. Kontrolörler $t = 20$ anında aktive edildiğinde Chua kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	67
Şekil 3.4. Kontrolörler $t = 25$ anında aktive edildiğinde Chua kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	68
Şekil 3.5. Kontrolörler $t = 30$ anında aktive edildiğinde Chua kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	69
Şekil 3.6. Bulanık mantık denetleyici blok yapısı	75

Şekil 3.7. Bulanık kontrolör üyelik fonksiyonları (a) e_1 giriş değişkeni, (b) e_2 giriş değişkeni, (c) e_3 giriş değişkeni, (d) çıkış değişkeni	76
Şekil 3.8. Bulanık kontrolör kuralları	77
Şekil 3.9. Bulanık kontrolör yüzey diyagramları (a) e_1-e_2-u , (b) e_1-e_3-u , (c) e_2-e_3-u	78
Şekil 3.10. Kontrolörler $t = 40$ anında aktive edildiğinde Bonhoeffer–van der Pol kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	82
Şekil 3.11. Kontrolörler $t = 50$ anında aktive edildiğinde Bonhoeffer–van der Pol kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	83
Şekil 3.12. Kontrolörler $t = 60$ anında aktive edildiğinde Bonhoeffer–van der Pol kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	84
Şekil 3.13. ANFIS ile kaotik sistemlerin kontrol modeli	88
Şekil 3.14. Doğrusal geri-beslemeli kontrol ve ANFIS ile tasarlanan kaotik sistemlerin kontrol modeli	90
Şekil 3.15. Kontrolörler $t = 20$ anında aktive edildiğinde Colpitts kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	95
Şekil 3.16. Kontrolörler $t = 25$ anında aktive edildiğinde Colpitts kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	96
Şekil 3.17. Kontrolörler $t = 30$ anında aktive edildiğinde Colpitts kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali	97

TABLÖLAR LİSTESİ

Tablo 3.1. Bulanık kontrolör kuralları	76
--	----



ÖZET

Anahtar kelimeler: Kaotik devreler, kaos kontrolü, yapay sinir ağları, bulanık mantık, sinirsel-bulanık ağlar

Elektronik devreler için kaos istenmeyen bir davranıştır. Bu tezde, kaotik sistemlerin kontrolü için iyi bilinen bazı kaos kontrol yöntemleri ile yapay zekâ tekniklerinin birlikte kullanımı önerilmiştir. Chua devresinin kontrolü kayma kipli kontrol yöntemi, yapay sinir ağları ve ikisinin bir arada kullanımı ile, Bonhoeffer–van der Pol devresinin kontrolü pasif kontrol yöntemi, bulanık mantık ve ikisinin bir arada kullanımı ile, Colpitts devresinin kontrolü ise geri-beslemeli kontrol yöntemi, sinirsel-bulanık ağlar ve ikisinin bir arada kullanımı ile gerçekleştirilmiştir. Sonuçlar karşılaştırmalı grafikler ile sunulmuştur. Sonuçlar, önerilen yaklaşımın kaotik sistemlerin denge noktasına kontrolünü daha hızlı sağladığını göstermiştir.

CONTROL OF CHAOTIC SYSTEMS WITH CLASSICAL AND INTELLIGENT APPROACHES

SUMMARY

Keywords: Chaotic circuits, chaos control, artificial neural networks, fuzzy logic, neuro-fuzzy

Chaos is an undesired behaviour for electronic circuits. In this thesis, usage of some well-known chaos control methods with artificial intelligence techniques is proposed for the control of chaotic systems. Sliding mode control method, artificial neural networks and using both of them are applied for the control of Chua's circuit, the passive control method, fuzzy logic and using both of them are applied for the control of Bonhoeffer–van der Pol circuit, and the feedback control method, adaptive neuro-fuzzy inference system and using both of them are applied for the control of Colpitts circuit. The results are presented by comparative figures. They show that the proposed approach provides the control of chaotic systems to their equilibrium points more effectively.

BÖLÜM 1. GİRİŞ

Fiziksel olarak oluşturulan sistemlerin çoğu, sistemin belli bir bölgedeki değişimi için doğrusal davranışlar gösterir. Fakat gerçekleşen bazı olaylarda ilk başta rastlantıya dayalıymış gibi gözüken ve rastgele olarak isimlendirilecek tipte davranışlar görmek mümkündür. Bu tarz rastgele davranış türleri, ekonomide, borsada, biyolojik sistemlerde, kimyasal reaksiyonlarda, meteorolojide, mühendislik ile ilgili sistemlerde yani çok geniş alanlarda gözlemlenebilmektedir. Basit birkaç örnek vermek gerekirse, yoğun trafikte hangi şeritteki araçların daha hızlı ilerleyeceği, rüzgârda dalgalanan bayrağın aldığı şekiller, musluktan düzenli damlayan suyun düzeninin zamanla değişmesi bu tarz davranışlardır. İşte böyle doğrusal olmayan sistemlerde rastgele veya düzensiz gibi gözüken, zamanla değişen ve sonucu değiştiren, önceden kestirilemeyen davranışlara kaos denilmektedir (Baker ve Gollub, 1990). Kaos davranışının oluşturduğu sisteme de kaotik sistem adı verilmektedir.

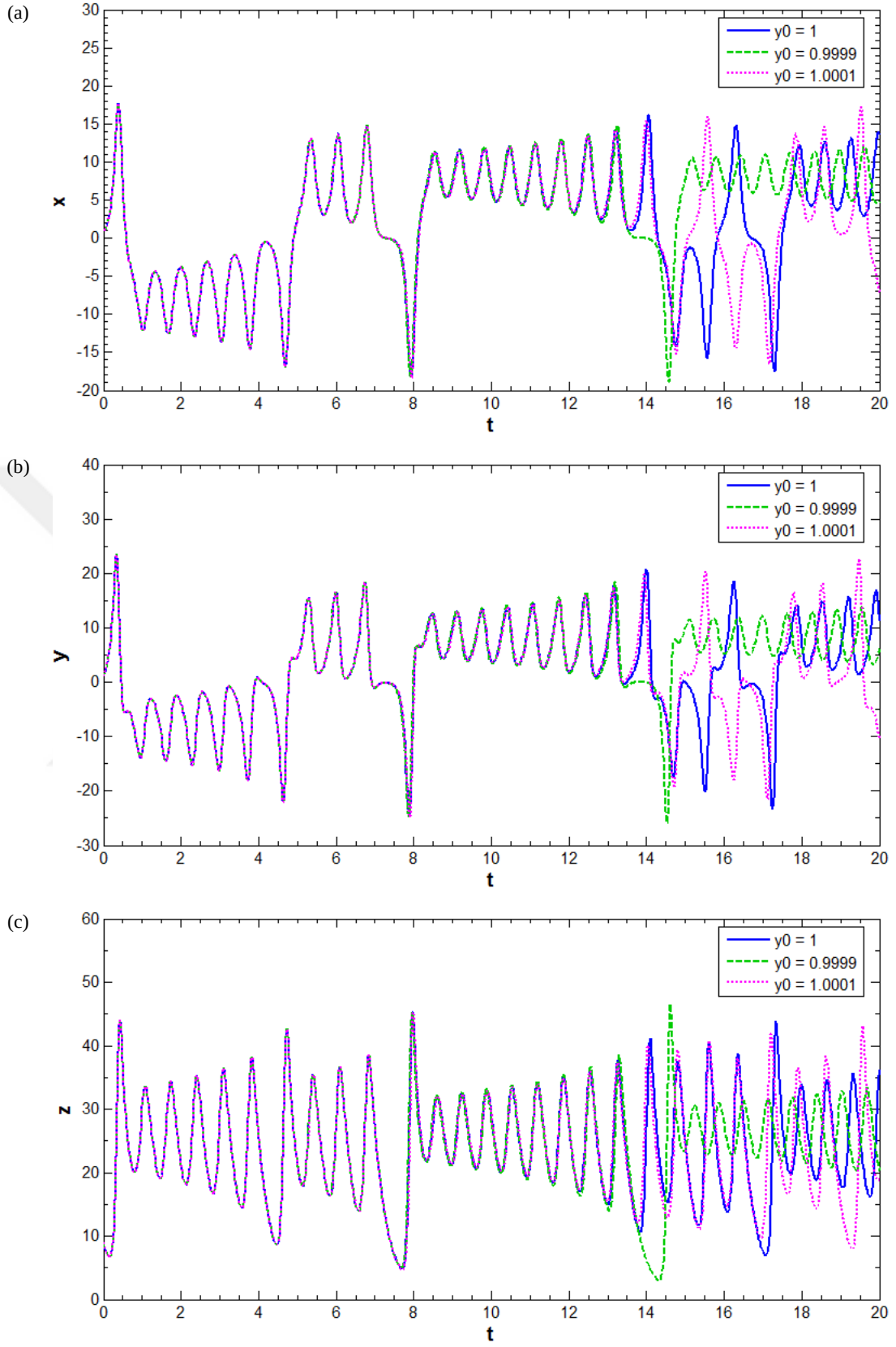
Edward Lorenz'in 1961'de o yılların ilgi çekici buluşlarından biri olan hava tahmin makinesi ile yaşadığı tecrübe, kaosun bilimsel bir disiplin dalı olarak ortaya çıkmasını sağlamıştır. Lorenz hava tahmini için meteorolojik sistem eşitliklerini bilgisayar ortamında modellemesi esnasında, kısa bir yol olarak programı tekrar başa alarak çalıştırmak yerine yazıcıdan daha önce çıkarttığı dizilerin son değerleri üzerinden programı devam ettirmiştir. Program arayüzünün bilgisayarın hafızasındaki gibi virgülden sonra 6 basamaklı olan ondalıklı sayılar yerine sadece 3 basamak girme imkânı tanınması sebebiyle, veri girişlerini böyle yapmıştır. Bir önceki yazıcı dökümü devamı ile çıktıların birebir örtüşmesini beklerken hava durumu tahmininin farklılaştığını ve birkaç aylık süre zarfında aralarındaki tüm benzerliğin ortadan kalktığını görmüştür. Başlangıçta bu 2 hava tahmininin grafiksel seyirlerindeki fark çok çok az iken, birebir aynı gibi devam ederken, belli bir

aşamadan sonra yavaş yavaş farklı noktalara yönelmeye başlamış ve bir süre sonra aralarında hiçbir benzerlik kalmamıştır (www.buzlu.org, 2016). Örneğin, $x_0 = 1$, $y_0 = 1$, $z_0 = 9$ başlangıç şartlarının $y_0 = 0,9999$ ve $y_0 = 1,0001$ olarak on binde bir değiştirildiğindeki durumlar Şekil 1.1.'de gösterilmiştir. Lorenz'in önce vakumlu tüplerden birinin bozulmasından kaynaklandığını düşündüğü, sonrasında fark ettiği bu başlangıç şartlarına hassas bağıllık kaos teorisinin temel yapıtaşlarından birini oluşturmuştur. Daha sonra Lorenz, böyle sistemlerde argümanların sadece belli değer aralıkları içinde rastgele gibi değişim gösterdiğini ve bu sınırlar içindeki hareketlerinin tahmin edilemez olduğunu görmüştür. Ayrıca, matematiksel denklemlerde hava sürtünmesi, atom-altı parçacıkların davranışları gibi hesaplaması nerdeyse imkânsız bazı küçük detaylar ihmal edilmektedir. Teorideki matematik ile gerçek dünya verileri çoğu zaman tam örtüşmemektedir. Sadece basit matematiksel denklemler baz alınsa bile sistem kaotik davranış sergileyebilmektedir. Sistemi kaotik yapan belirlenemezlikteki asıl etkenler, değişkenlerin çok sayıda olması ve çok ufak farklılıkların bile sonuçsal değişikliklere yol açmasıdır. Burada, özellikle dikkat edilmesi gereken unsur kaosun tesadüfi olmadığıdır. Nedensel bir durumdur. Kaos aslında oldukça karmaşık, anlaşılmaz bir düzendir. Kaotik davranışlar belli bir sayısal değerler kümesi biçiminde hareket etmek zorundadır (www.evrenindili.com, 2016).

Böylece, kaotik sistemlerin insanlar tarafından tam olarak belirlenemese de bir iç düzene uydukları, rastgele olmadıkları, oldukça karmaşık bir düzen oldukları kaos teorisinin ana yapıtaşlarını oluşturmuştur.

Kaotik sistemlerin önemli özelliklerinden bir diğeri de sonsuz sayıda tam periyodik olmayan kararsız yörünge barındırmalarıdır. Bu özellik kaotik sistemlerin hiçbir şekilde birbirini tekrara düşmeyen yörüngelerde salınım yaptığını ifade etmektedir (http://ozguraktekin.blogspot.com, 2016).

Günümüzde kaos teorisinin meteoroloji, fizik, kimya, biyoloji, mekanik, güvenli haberleşme, şifreleme, kontrol sistemleri ve finansal modelleme gibi pek çok farklı alanda başarılı uygulamaları vardır.



Şekil 1.1. Lorenz'in birbirinden uzaklaşan iki hava tahmini grafiği (a) x sinyali, (b) y sinyali, (c) z sinyali.

Kaotik sistemlerin sıra dıřı, hassas, öngörölemeyen yapısı bu sistemleri kontrol etme fikrinin ortaya çıkmasına sebep olmuřtur. Kaos kontrolüne dair yapılan alıřmaların çoęu için kontrol, sistemi denge noktalarından birinde kararlı hale getirebilmek, yani kaotik davranıřı sonlandırıp kararlı bir salınımı gerekleřtirmesini saęlamak olarak ifade edilebilir. Kaos kontrolü bir kaotik sistemin istenen kararlı bir duruma gemesini amalamaktadır. Kaos kontrolü için kontrol teorisinin genel yöntemlerinin kullanılmasının yanısıra kaotik sistemlere özgü yeni yöntemler de geliřtirilmiřtir (<http://ozguraktekin.blogspot.com>, 2016).

Yapay zekânın amacı, insan zekâsını bilgisayar aracılığı ile taklit ederek, bilgisayarlara belli bir ölçüde zeki davranma yeteneęi kazandırabilmektir. Yapay zekâ yöntemleri verilerden öęrenebilme, genelleme yapabilme, iliřkilendirme, sınıflandırma, tahmin edebilme, çok sayıda deęiřkenle alıřabilme gibi önemli özelliklere sahiptir. Özellikle 90'lı yıllardan sonra yapay zekâ alıřmaları bilgisayar kullanımının hızla yaygınlařması sonucunda artmıřtır. Günümüzde, kontrol de dâhil mühendislik, tıp, savunma sanayi, haberleřme, üretim, otomasyon gibi birçok alanda yapay zekâ tekniklerinden faydalanılmaktadır.

Bu doktora alıřmasında, tezin ikinci bölümünde kaotik sistemler ve yapay zekâ teknikleri arařtırılmıřtır. Elektronik tabanlı kaotik sistemler, yapay zekâ tekniklerinden yapay sinir aęları, bulanık mantık ve sinirsel-bulanık aęlar genel olarak ele alınmıř, kaotik sistemlerin klasik ve zeki yöntemler ile kontrolü üzerine literatür bilgileri verilmiřtir. Üüncü bölümde hem klasik kaos kontrol yöntemlerinin hem de zeki kontrol yöntemlerinin uygulanması, yeni oluřturulan ortak yaklařımlarla beraber gerekleřtirilmiřtir. Ayrıca, bu bölümde karřılařtırmalı grafiklerle performans deęerlendirmesine de yer verilmiřtir. Son bölümde ise, sonuçlar özetlenmiř, ileriki alıřmalara yönelik bazı önerilerde bulunulmuřtur.

BÖLÜM 2. MATERYAL VE YÖNTEM

2.1. Kaotik Sistemler

Kaotik sistemler, başlangıç şartlarına hassas duyarlılık gösteren karmaşık sistemler olarak tanımlanabilir. Kaotik bir sistemin başlangıç şartlarındaki önemsiz derecede çok ufak bir değişiklik bile, sistemin ilerleyen aşamalarında öngörülemeyen çok farklı değişikliklere neden olabilir (www.evrenindili.com, 2016).

Kaotik sistemler, matematiksel modellemelerine göre ayrık (discrete) ve sürekli (continuous) zamanlı sistemler olarak ikiye ayrılırlar (Demirkol, 2007). Uygun bir $f(x)$ fonksiyonun iterasyonu sonucu kaotik özellik oluşturan diziler, ayrık zamanlı kaotik sistemlerdir. Bu sistemlerin gösterimi genellikle $x_{n+1} = f(x_n)$ biçimindedir. Hénon, lojistik, Tent ve Tinkerbell haritaları iyi bilinen ayrık kaotik sistemler arasında sayılabilir. Sürekli zamanlı kaotik sistemler ise $d(x(t))/dt = f(x(t))$ biçiminde diferansiyel denklemler ile ifade edilirler. İkinci dereceden diferansiyel denklem takımından oluşan Duffing ve van der Pol, üçüncü dereceden diferansiyel denklem takımından oluşan Lorenz, Rössler, Chua, Chen ve Sprott önemli ve sık kullanılan sürekli zamanlı kaotik sistemlerdir.

Kaotik sistemlerin muazzam karmaşıklığını göstermek ve kaotik davranış sergilediği hassas aralığı yakalayabilmek için bilgisayar ortamında bazı özel teknikler ve çizgi grafik türleri geliştirilmiştir. Kaotik sistemlerdeki başlıca özellikler; otokorelasyon fonksiyonunun üstel olarak azalması, güç spektrumunun gürültü yapısına sahip olması, Poincaré kesit düzleminin bir bölümünün tamamen ve düzensiz bir şekilde dolması, en az bir Lyapunov üstelinin pozitif olmasıdır (Strogatz, 2001; Demirkol, 2007).

Bir sistemin kaosta olup olmadığı, olabildiği kadar uzun süre ile kaydedilmiş kayıtlarının değişik parametre ve başlangıç şartı değerlerine göre davranışı incelenerek anlaşılabilir. Sistemin zamanla değişebilen bu tarz davranışını gösteren verilere “zaman serileri” oluşturdukları grafiklere ise “zaman serisi grafikleri” denilmektedir. Sistem davranışının faz uzayındaki görünümünün elde edilmesi ve bakılması bir diğer kaotik sistem tespit biçimidir. Bu görünümlere “kaotik çekerler” (chaotic attractors), “garip çekerler” ya da “tuhaf çekerler” (strange attractors), kısaca “çekerler” (attractors) denir. Bunlardan grafik olarak en ünlüleri kelebek biçimli Lorenz çekeri ve çift sarmal biçimli Chua çekeridir.

Oluşturulan çekerlerin görsel olarak incelenmesi genellikle çok karmaşık yapılara sahip olduklarından bazen zor olabilmektedir. Bu zorluğu aşmadaki en önemli yardımcılardan birisi Poincaré kesitleri olarak bilinen yöntemdir. Bu yöntemde, faz uzayında çizilen sistemin uygun noktalarından kesitler alınır ve elde edilen bu görüntüler sistemin dinamiği hakkında fikir verirler. Poincaré kesitindeki noktaların dağılımı tek ve küçük bir bölgede sonlu sayıda ise sistemin hareketi periyodik, kapalı bir eğri ise sistemin hareketi yarı periyodik, belirli alanlarda yoğunlaşmış kümeler şeklinde ise sistemin hareketi kaotiktir (www.evrenindili.com, 2016).

Lyapunov üstelleri, sistemin hangi parametre değerlerinde kaotik olduğunu veya kaos içermediğini zaman serisi verileri üzerinden analizle anlaşılmasına yarayan matematiksel bir yöntemdir. Lyapunov üsteli, çeker üzerinde sistemin olası durumlarına göre, başlangıçta yakın olan rastgele iki noktanın birbirlerinden ayrılma derecesi gösteren sayısal ifadedir. Eğer bu komşu noktalar birbirlerinden hızla uzaklaşıyorsa, hesaplanan en büyük Lyapunov üsteli pozitif değere sahip olur. Lyapunov üstellerinin sayısı, sistemin faz uzayı boyut sayısı kadardır. 3-boyutlu bir faz uzayında karşılaşılabilecek Lyapunov üstelleri ($\lambda_1, \lambda_2, \lambda_3$) şöyledir; $(-, -, -)$: sabit nokta, $(0, -, -)$: limit döngü, $(0, 0, -)$: simit, ve $(+, 0, -)$: tuhaf çeker (kaos). En büyük Lyapunov üstelinin pozitif olması kaotik durumun bir göstergesidir (www.evrenindili.com, 2016).

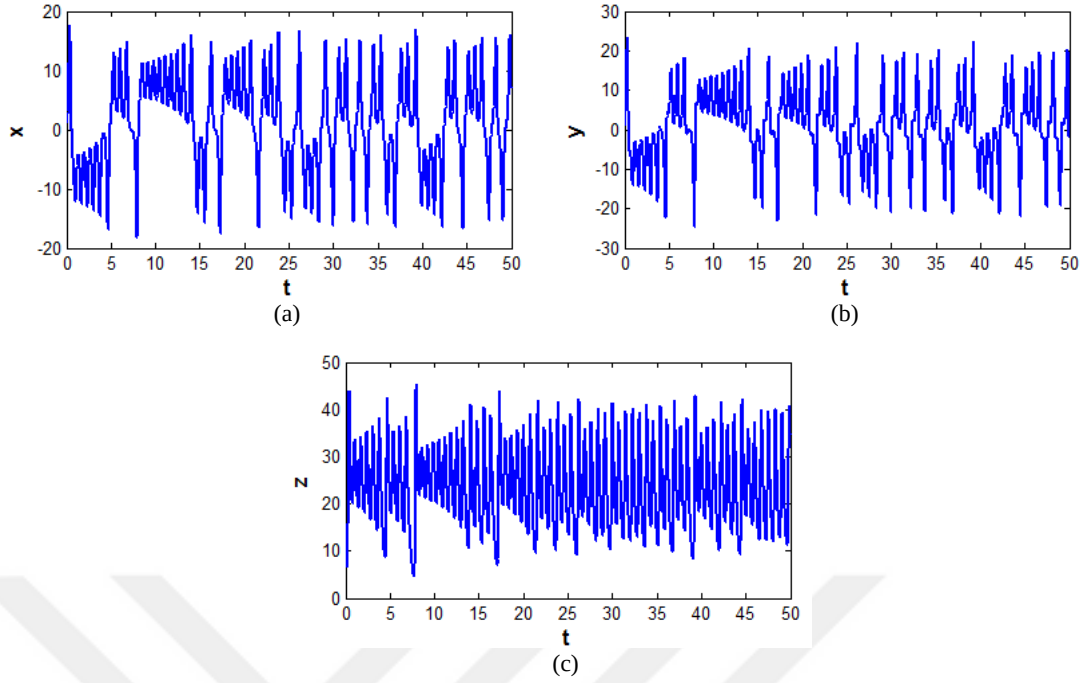
Kaotik analiz için kullanılan bir diğer teknik zaman serisinde izlenen sinyallerin doğrusal olmayışının tespitidir. Bu amaç için sıklıkla vekil veri analizi denilen yöntemle başvurulmaktadır. Bu analiz tipinde, doğrusal bir algoritma ile de mevcut sinyalin bir benzeri oluşturulur ve üretilen yapay sinyal ile gerçek sinyal serisi arasındaki ilişki incelenir. İlişkinin yokluğu, sinyalin doğrusal olmadığı sonucunu gösterir (Gleick, 1995).

Kaotik sistemlerin tespiti için yukarıda sayılan yöntemlerin yanısıra başka teknikler de önerilmiştir fakat bunların her durumda geçerli olmaması gibi bazı zayıf tarafları olabilmektedir.

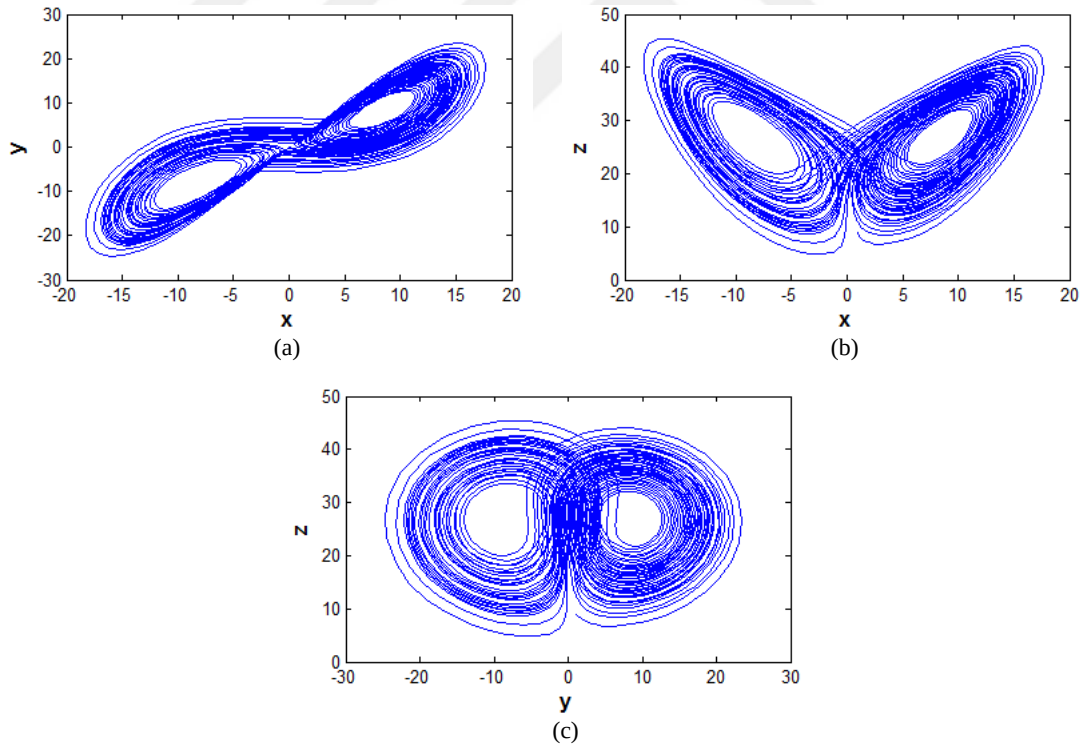
Kaotik sistemler denilince genellikle ilk akla gelen sistem kelebek biçimli Lorenz çekeridir. Bu çeker aşağıdaki Denklem 2.1'de verilen, Lorenz'in kaosu fark ettiği, atmosferdeki ısı aktarımının sadeleştirilmiş bir modeli olan diferansiyel denklemler kümesi neticesinde oluşmaktadır (Lorenz, 1963):

$$\begin{cases} \dot{x} = \alpha (y - x), \\ \dot{y} = \rho x - y - xz, \\ \dot{z} = -\beta z + xy. \end{cases} \quad (2.1)$$

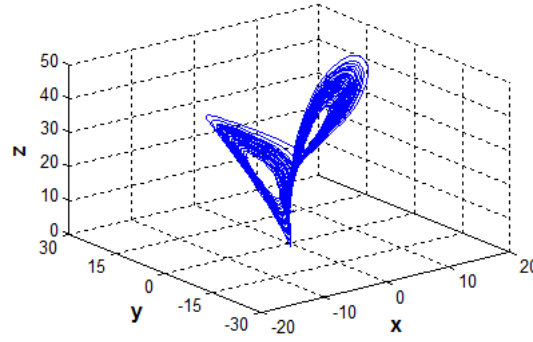
Bu denklemin fiziksel karşılığı, kapalı bir kap içerisinde alttan ısıtılan ya da ısınan, yukarıdan ise soğutulan ya da soğuyan bir akışkanın hareketini ifade etmektedir. Burada x , y ve z durum değişkenleridir, sırasıyla ısı aktarımını, yatay sıcaklık değişimini ve hücredeki normal sıcaklıktaki sapmayı temsil etmektedirler. Sistem parametreleri α , ρ ve β ise sırasıyla Prandtl sayısı, Rayleigh sayısı ve geometrik çarpandır. $\alpha = 10$, $\rho = 28$, $\beta = 8/3$ sistem parametreleri ve $x_0 = 1$, $y_0 = 1$, $z_0 = 9$ başlangıç şartları için Şekil 2.1., Şekil 2.2. ve Şekil 2.3.'te gösterilen kaotik çözümler elde edilir.



Şekil 2.1. Kaotik Lorenz sisteminin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 2.2. Kaotik Lorenz sisteminin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi.



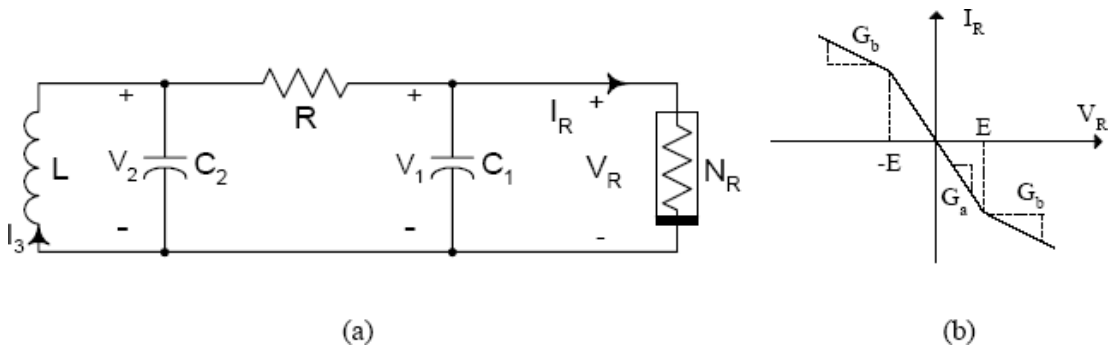
Şekil 2.3. Kaotik Lorenz sisteminin üç boyutlu faz yüzeyi grafiği.

2.2. Kaotik Elektronik Sistemler

Bu kısımda yaygın kullanılan kaotik elektronik sistemlerden Chua, Bonhoeffer–van der Pol ve Colpitts devreleri kısaca açıklanmış, denklemleri verilmiş ve kaotik grafikleri gösterilmiştir.

2.2.1. Chua devresi

Elektronik devre alanında kaos durumu ilk kez Leon O. Chua tarafından önerilen devre ile incelenmiştir (Chua ve ark., 1986). Chua devresi Şekil 2.4.’te gösterildiği gibi 2 adet doğrusal kondansatör (C_1 ve C_2), 1 adet doğrusal direnç (R), 1 adet doğrusal bobin (L) ve gerilim kontrollü doğrusal olmayan 1 direnç olan Chua diyotundan (N_R) oluşur (Hanbay ve ark., 2007). Chua üçüncü dereceden kaotik bir osilatör devresidir.



Şekil 2.4. Chua devresi (a) devre diyagramı, (b) doğrusal olmayan direncin akım-gerilim karakteristiği (Hanbay ve ark., 2007).

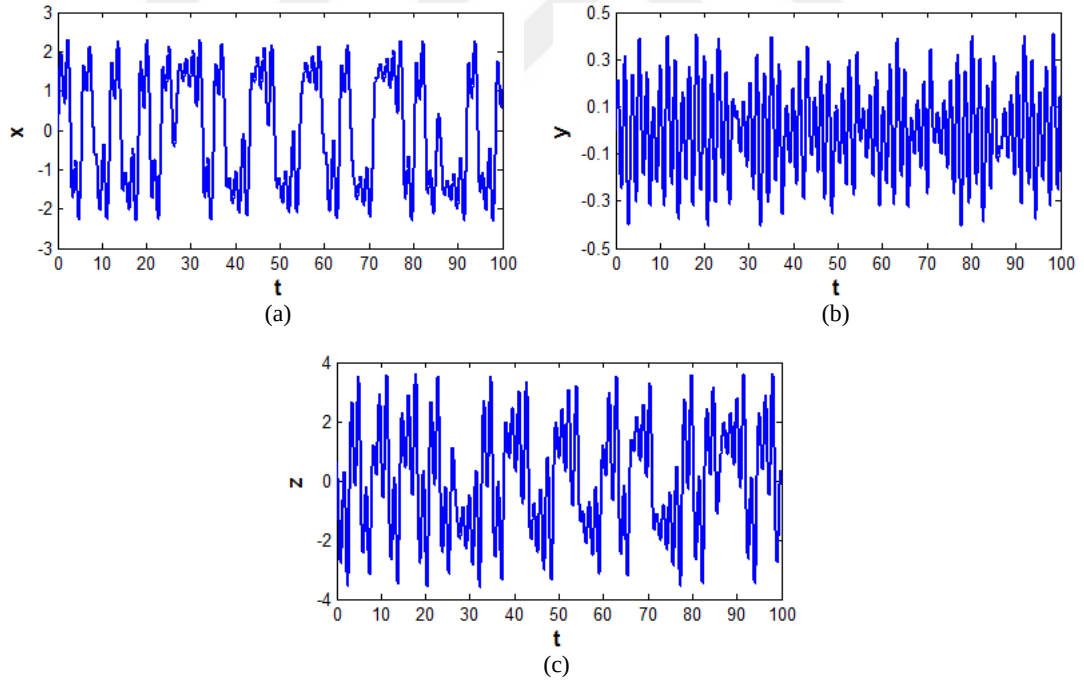
Chua devresinin kaotik diferansiyel denklemleri Denklem 2.2 ile tanımlanmıştır (Harb ve Smadi, 2004):

$$\begin{cases} \dot{x} = \alpha(y - x - f(x)), \\ \dot{y} = x - y + z, \\ \dot{z} = -\beta y, \end{cases} \quad (2.2)$$

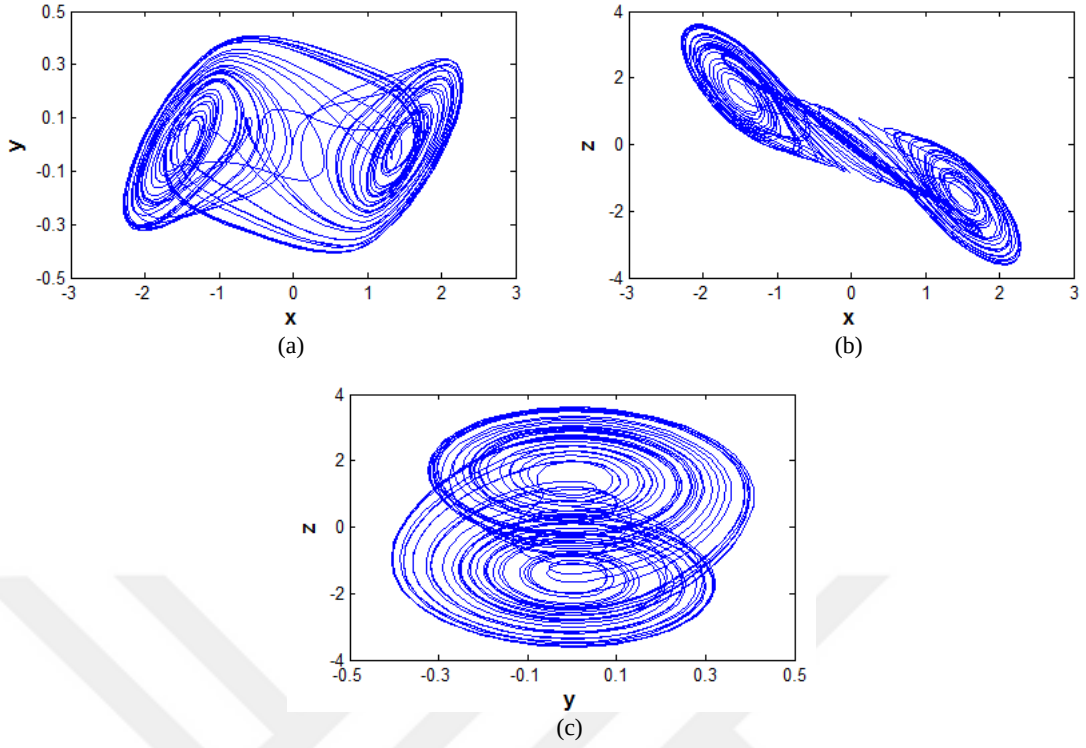
burada $f(x)$ ile gösterilen direncin fonksiyonu ise Denklem 2.3'te verilmiştir:

$$f(x) = bx + 1/2(a - b)(|x + 1| - |x - 1|). \quad (2.3)$$

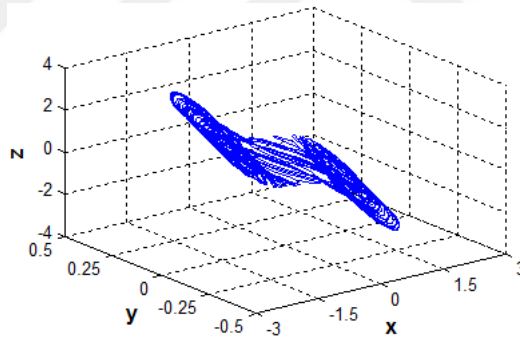
$\alpha = 15,6$, $\beta = 25,58$ sistem parametreleri $a = -8/7$, $b = -5/7$ fonksiyon değerleri ve $x_0 = 0$, $y_0 = 0$, $z_0 = 0,6$ başlangıç koşulu için Chua devresinin oluşturduğu kaotik yapı Şekil 2.5., Şekil 2.6. ve Şekil 2.7.'de gösterilmiştir.



Şekil 2.5. Kaotik Chua devresinin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 2.6. Kaotik Chua devresinin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi.



Şekil 2.7. Kaotik Chua devresinin üç boyutlu faz yüzeyi grafiği.

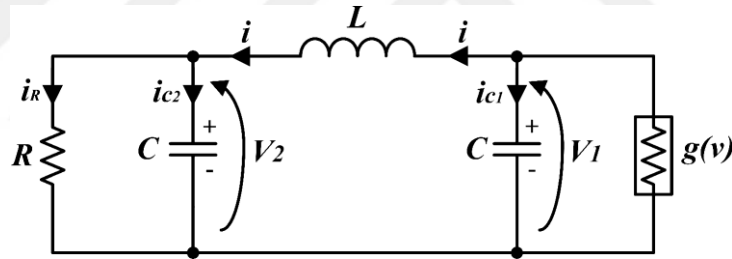
Chua kaotik sisteminin denge noktaları Denklem 2.2'de $\dot{x} = 0$, $\dot{y} = 0$ ve $\dot{z} = 0$ alınarak bulunabilir:

$$\begin{cases} \alpha(y - x - bx - 1/2(a - b)(|x + 1| - |x - 1|)) = 0, \\ x - y + z = 0, \\ -\beta y = 0. \end{cases} \quad (2.4)$$

Denklem 2.4 çözüldüğünde, Chua kaotik sisteminin 1 tane denge noktası olduğu görülmektedir: $E(0, 0, 0)$.

2.2.2. Bonhoeffer–van der Pol devresi

Bonhoeffer–van der Pol devresi, Chua ve Colpitts gibi elektronik devre tabanlı kaotik çekicilerden biridir. Kaotik analizi bazı makalelerde detaylı olarak incelenmiştir (Nishiuchi ve ark., 2006; Kyprianidis ve ark., 2012). Şekil 2.8.’de bu tezde kullanılan kaotik Bonhoeffer–van der Pol osilatörünün devre diyagramı verilmiştir (Nishiuchi ve ark., 2006). Bu devredeki R , L , C , $g(v)$, i_r , i , i_{C1} , i_{C2} , V_L , V_1 ve V_2 sırasıyla doğrusal direnç, endüktans bobini, kapasitör, doğrusal olmayan direnç, doğrusal direncin akımı, endüktans bobininin akımı, birinci ve ikinci kapasitörlerin akımları, endüktans bobininin voltajı, birinci ve ikinci kapasitörlerin voltajlarını ifade etmektedir.



Şekil 2.8. Bonhoeffer–van der Pol devresi (Nishiuchi ve ark., 2006).

Eğer bu devreye Kirchhoff kanunu uygulanırsa, aşağıdaki denklem takımı elde edilir:

$$\begin{cases} i + i_{C1} + g(v) = 0, \\ i = i_r + i_{C2}, \\ V_L = V_1 - V_2. \end{cases} \quad (2.5)$$

Denklem 2.5, uygun parametrelere göre yeniden yazılırsa,

$$\begin{cases} C \frac{dV_1}{dt} = -i - g(v), \\ C \frac{dV_2}{dt} = i - \frac{V_2}{R}, \\ L \frac{di}{dt} = V_1 - V_2, \end{cases} \quad (2.6)$$

elde edilir. Denklem 2.6'daki doğrusal olmayan $g(v)$ direnci aşağıdaki formül ile tanımlanır:

$$g(v) = -aV_1 - b \tanh(cV_1), \quad (2.7)$$

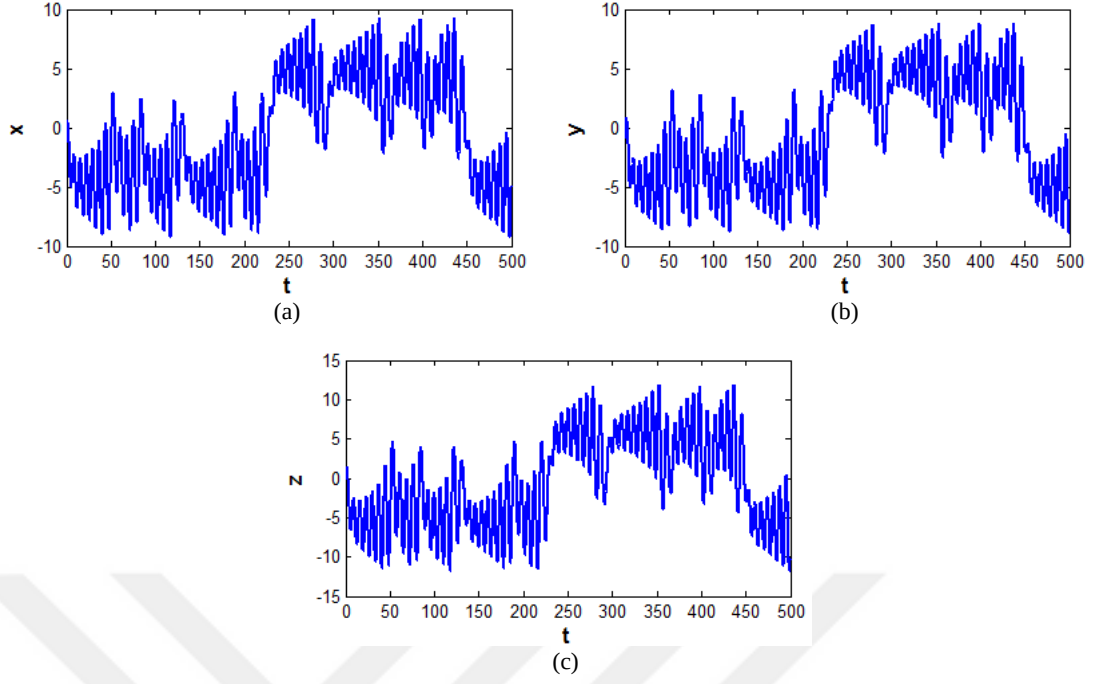
ki burada a , b ve c Bonhoeffer–van der Pol osilatörünün parametreleridir (Nishiuchi ve ark., 2006; Zribi ve Alshamali, 2012). Denklem 2.6'nın boyutsuz, ölçekleri olmayan halini elde etmek için, zaman değişkeni τ , sistem parametreleri s_1 , s_2 , s_3 ve durum değişkenleri x , y , z aşağıdaki biçimde alınabilir (Kyprianidis ve ark., 2012; Ojo ve ark., 2013):

$$\tau = \frac{1}{\sqrt{LC}} t, \quad s_1 = a \sqrt{\frac{L}{C}}, \quad s_2 = bc \sqrt{\frac{L}{C}}, \quad s_3 = \frac{1}{R} \sqrt{\frac{L}{C}}, \quad x = \frac{V_1}{b} \sqrt{\frac{C}{L}}, \quad y = \frac{V_2}{b} \sqrt{\frac{C}{L}}, \quad z = \frac{i}{b}. \quad (2.8)$$

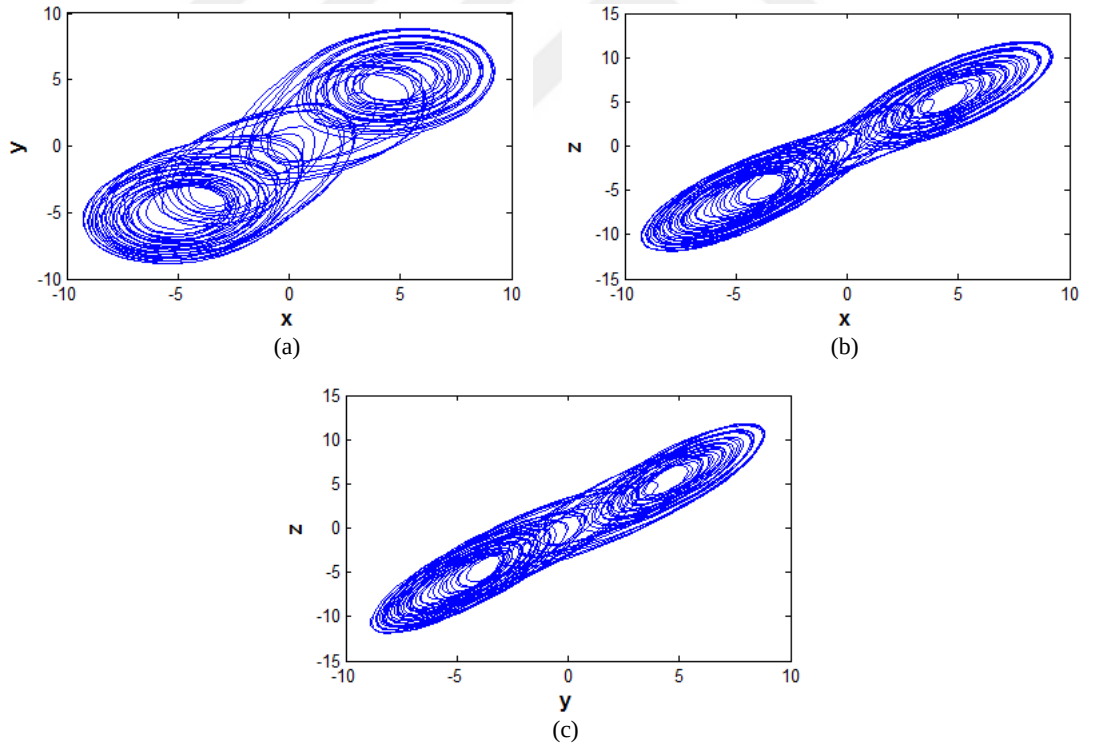
Böylece, Bonhoeffer–van der Pol osilatörünün alttaki diferansiyel denklemleri elde edilir:

$$\begin{cases} \dot{x} = -z + s_1 x + \tanh(s_2 x), \\ \dot{y} = z - s_3 y, \\ \dot{z} = x - y. \end{cases} \quad (2.9)$$

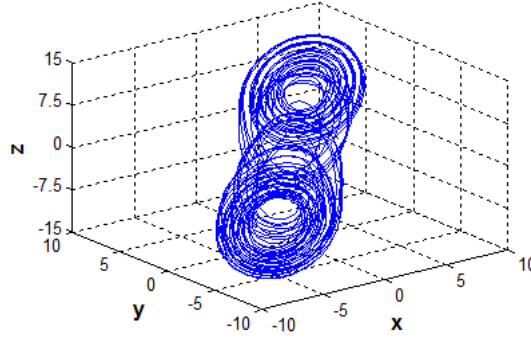
Bonhoeffer–van der Pol sistemi 2.9'un parametre değerleri $s_1 = 1$, $s_2 = 1$, $s_3 = 1,2$ alındığında (Njah ve Vincent, 2009; Zribi ve Alshamali, 2012) ve başlangıç şartı $x(0) = 0,5$, $y(0) = -1$, $z(0) = 0,5$ iken kaotik davranış sergiler. Bu değerlerdeki kaotik Bonhoeffer–van der Pol sisteminin zaman serisi grafikleri Şekil 2.9.'da, iki-boyutlu faz portreleri Şekil 2.10.'da, üç-boyutlu faz portresi ise Şekil 2.11.'de verilmiştir.



Şekil 2.9. Kaotik Bonhoeffer-van der Pol devresinin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 2.10. Kaotik Bonhoeffer-van der Pol devresinin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi.



Şekil 2.11. Kaotik Bonhoeffer–van der Pol devresinin üç boyutlu faz yüzeyi grafiği.

Bonhoeffer–van der Pol kaotik sisteminin denge noktalarını bulmak için Denklem 2.9’da $\dot{x} = 0$, $\dot{y} = 0$ ve $\dot{z} = 0$ alınmıştır:

$$\begin{cases} -z + s_1 x + \tanh(s_2 x) = 0, \\ z - s_3 y = 0, \\ x - y = 0. \end{cases} \quad (2.10)$$

Denklem 2.10 çözüldüğünde, $s_1 = 1$, $s_2 = 1$ ve $s_3 = 1,2$ değerleri için Bonhoeffer–van der Pol kaotik sisteminin 1 adet denge noktası olduğu görülmektedir: $E(0, 0, 0)$.

2.2.3. Colpitts devresi

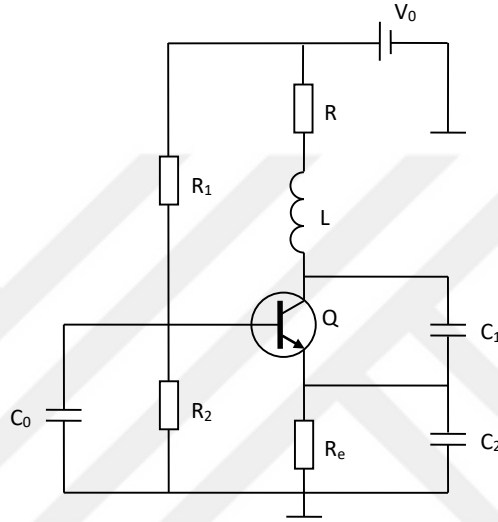
Edwin H. Colpitts (1918) tarafından bulunan Colpitts devresi basit kaotik elektronik devrelerden biridir. Şekil 2.12.’de gösterildiği gibi, tek çift-kutuplu bir jonksiyon transistör ile çıkış devresini besleyen bir LC devresinden oluşan bir transistörlü amplifikatördür (Setoudeh ve Dousti, 2014).

Colpitts devresinin dinamikleri aşağıdaki formda ifade edilir:

$$\begin{cases} \dot{x} = y - aF(z), \\ \dot{y} = c - x - by - z, \\ \dot{z} = (y - dz) / \varepsilon, \end{cases} \quad (2.11)$$

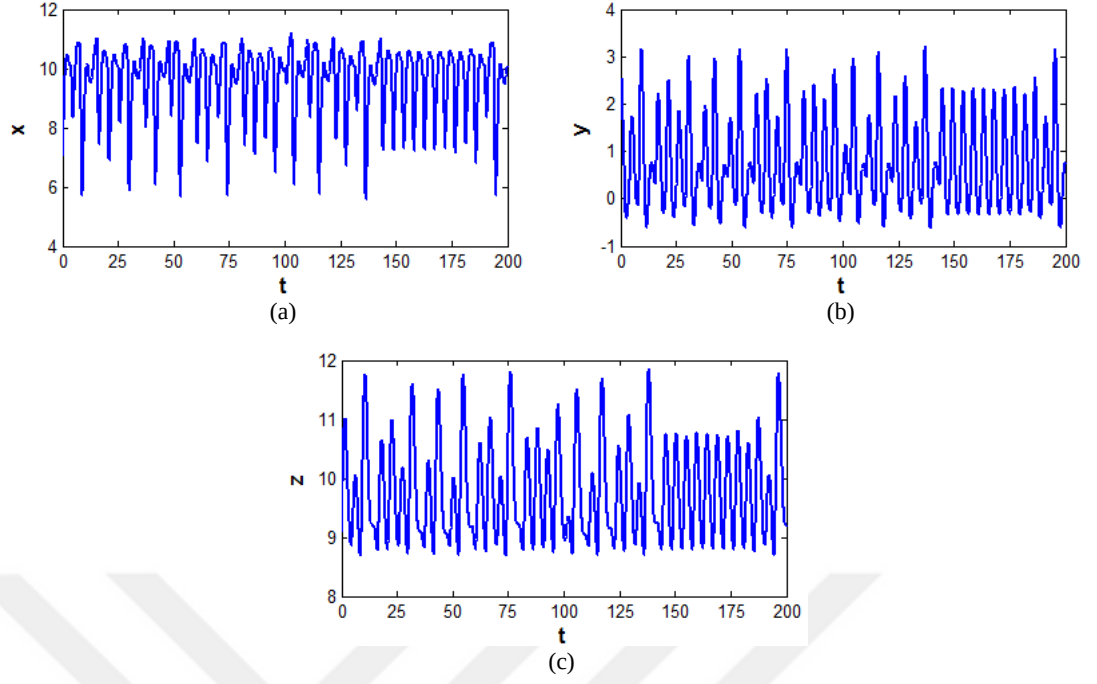
burada $x = V_{c1} / V^*$, $y = \rho I_L / V^*$, $z = V_{c2} / V^*$, $\rho = \sqrt{L/C_1}$, $\tau = \sqrt{LC_1}$, $\varepsilon = C_2 / C_1$, $a = \rho / r$, $b = R / \rho$, $c = V_0 / V^*$, $d = \rho / R_e$, $e = cR_2 / (R_1 + R_2)$ ve doğrusal olmayan $F(z)$ fonksiyonu:

$$F(z) = \begin{cases} e - 1 - z, & z < e - 1, \\ 0, & z \geq e - 1. \end{cases} \quad (2.12)$$

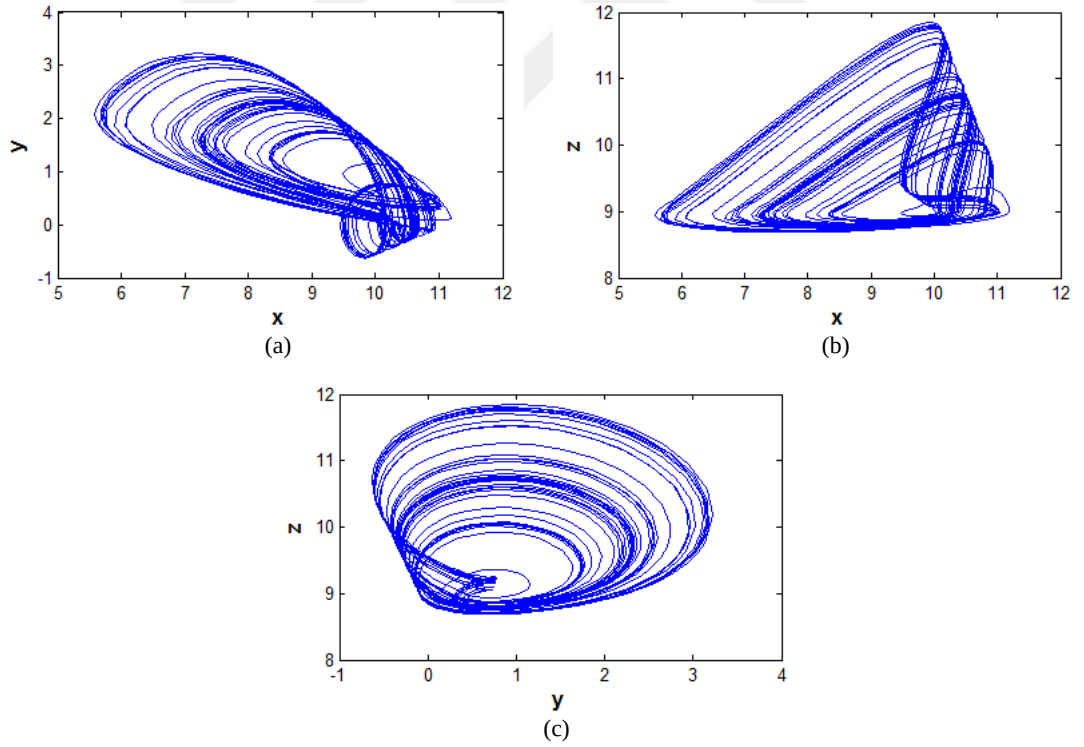


Şekil 2.12. Colpitts devresi (Cenys ve ark., 2003).

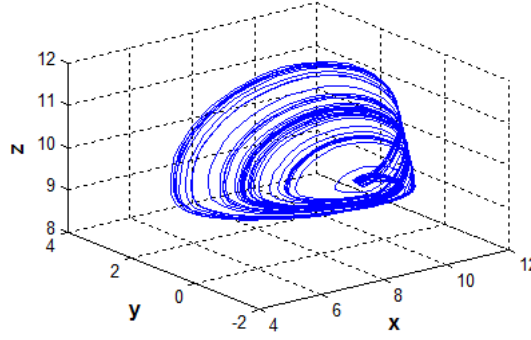
Devre parametreleri genellikle $Q = 2N3904$ tipi, $L = 850 \mu\text{H}$, $C_1 = C_2 = 470 \text{ nF}$, $C_0 = 47 \mu\text{F}$, $R = 36 \Omega$, $R_e = 510 \Omega$, $R_1 = R_2 = 3 \text{ k}\Omega$ ve $V_0 = 15 \text{ V}$ olarak alınmaktadır (Kennedy, 1994; Cenys ve ark., 2003; Li ve ark., 2007). Bu değerlere göre, Denklem 2.11 ve Denklem 2.12'deki Colpitts devresinin parametreleri $\varepsilon = 1$, $a = 30$, $b = 0,8$, $c = 20$, $d = 0,08$ ve $e = 10$ olmaktadır (Cenys ve ark., 2003). $x_0 = 7,02$, $y_0 = 2,06$ ve $z_0 = 9,09$ başlangıç noktasına göre, Colpitts devresinin zaman serisi grafikleri Şekil 2.13.'te, iki boyutlu faz portreleri Şekil 2.14.'te ve üç boyutlu faz yüzeyi Şekil 2.15.'te gösterilmiştir.



Şekil 2.13. Kaotik Colpitts devresinin zamana bağlı değişim grafikleri (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 2.14. Kaotik Colpitts devresinin iki boyutlu grafikleri (a) x - y faz portresi, (b) x - z faz portresi, (c) y - z faz portresi.



Şekil 2.15. Kaotik Colpitts devresinin üç boyutlu faz yüzeyi grafiği.

Colpitts sistemin denge noktaları ise aşağıdaki denklemin çözümü ile elde edilir:

$$\begin{cases} y - aF(z) = 0, \\ c - x - by - z = 0, \\ (y - dz) / \varepsilon = 0. \end{cases} \quad (2.13)$$

Bu sistem sadece 1 denge noktasına sahiptir: $E(c - (bd + 1)k, dk, k)$ ki burada $k = a(e - 1) / (a + d)$ olarak alınmıştır (Li ve ark., 2007). Yukarıda verilen parametre değerlerine göre denge noktası $E(10,449, 0,718, 8,976)$ olarak hesaplanır.

2.3. Kaotik Sistemlerin Kontrolü

Mühendislik ile ilgili çoğu sistemde kaotik davranış istenmez. Kaos ilk tespit edildiğinde istenmeyen bir davranış olarak nitelendirilmiştir. Başlangıçta, kaotik davranış gösteren doğrusal olmayan sistemlerinin parametreleri değiştirilerek kaotik davranış olma olasılığı tamamen yok edilmiş ve kontrol bu şekilde tasarlanmıştır. Sonrasında, bu kadar ilginç özellikleri olan kaotik işaretlerden olumlu yönde faydalanılan çalışmalar da yapılmıştır. Veri şifreleme ve güvenli haberleşme gibi önemli alanlarda kullanılmaktadır (Pehlivan ve ark., 2007). Bu bölümde tezin hedefi doğrultusunda kaotik sistemlerin kontrolü hakkında bilgi verilecektir.

Kaotik sistemlerin kontrol edilmesinden kasıt, sistemin belirli bir denge noktasında kararlı olmasını sağlamaktır. Bu amaç için OGY (Ott, Grebogi ve Yorke) kontrol, doğrusal geri-beslemeli (linear feedback) kontrol, doğrusal-olmayan geri-beslemeli

(nonlinear feedback) kontrol, zaman-gecikmeli geri-beslemeli kontrol (Pyragas), kayma kipli kontrol, adaptif kontrol, pasif kontrol gibi birçok yöntem geliştirilmiştir.

Kaotik sistemlerin başlangıç şartlarına hassas bağıllık gösteren doğrusal olmayan dinamik sistemler olduğu göz önüne alındığında, kontrolün mümkün olamayacağı düşünülmüştür. Ott, Grebogi ve Yorke (1990) bu düşüncüyü ortadan kaldıran önemli bir çalışma yapmışlardır. Kapalı döngü (closed-loop) metodu kullanarak OGY kısaltmalı kaos kontrol yöntemini geliştirmişlerdir. Bu yöntemde, her bir devirde arzulanan kararsız periyodik yörüngenin yakınında olmasını sağlamak için sisteme küçük, mantıklı seçilen seğirmeler uygulanır (https://en.wikipedia.org/wiki/control_of_chaos, 2016).

Günümüzde kaosun kontrolü denilince kolay uygulanabilirliği, basit yapısı ve başarısı nedeniyle akla ilk gelen yöntem doğrusal geri-beslemeli kontroldür. Lorenz (Gambino ve ark., 2006), Rössler (Liao ve Yu, 2006), Duffing (Jiang, 2002), Liu (Wang ve Li, 2010) gibi birçok kaotik sistemin kontrolünde kullanılmıştır. Bu yöntem her bir denklemin (x, y, z) kendi durum değerlerini bir katsayı çarpımı ile geri besleyerek kaotik sistemlerin kararsız periyodik salınımlarını sıfıra yakınsatmaktadır. Lyapunov kararlılık teoremini içermektedir ve kaotik sistemi farklı bir denge noktasında kararlı kılma özelliğine de sahiptir.

Doğrusal-olmayan geri-beslemeli kontrolde ise x, y, z ve sabit bir katsayı yerine lineer olmayan kontrol sinyalleri tercih edilerek durum değişkenlerine eklenir ve sistemin kontrolü sağlanır. Çok sayıda kaotik sistemin kontrolünde kullanılmıştır (Alvarez–Ramirez, 1994; Chen ve Han, 2003; Zhou ve ark., 2009, Hu ve ark., 2016).

Zaman-gecikmeli geri-besleme ile kontrol, Pyragas (1992) tarafından geliştirilmiştir. Kararsız yörüngelere sahip dinamik sistemlerin denge noktalarında kararlı hale getirilmesi için etkili ve basit bir yöntemdir. Bu yöntemde kontrol sinyali, sistemin mevcut değeri ile t zaman birimi geçmişteki değeri arasındaki fark kullanılarak bulunur (Küçükefe ve Kaypmaz, 2010). Birçok kaotik sistemin kontrolünde kullanılmaktadır (Mei ve ark., 2007; Saini ve Saini, 2014; Mahmoud ve ark., 2017).

Kayma kipli kontrol (sliding mode control) kavramı ilk olarak Emelyanov (1967) tarafından ortaya atılmıştır. Çalışmasında ikinci dereceden sistemlerin faz uzayında özel bir doğruyun tanımlanması ve herhangi bir başlangıç koşulu altında sistem durumlarının bu doğruya yaklaşacak şekilde yönlendirilmesi üzerinde durmuştur. İyi performans göstermesinden dolayı, çalışmalar kayma kipli kontrol üzerinde yoğunlaşmıştır. Bahsi geçen doğru, çok boyutlu bir hiper yüzey halini almış ve anahtarlama yüzeyi ya da kayma yüzeyi olarak adlandırılmıştır (Efe ve ark., 2000). Kayma kipli kontrol Lorenz (Yau ve Yan, 2004), Rössler (Chang ve ark., 2008a), Chua (Wang ve ark., 2009), Liu (Li ve Liu, 2010) ve diğer kaotik sistemlerin (Vaidyanathan ve Rhif, 2017; Alrifai ve Zribi, 2018) kontrolünde başarıyla uygulanmıştır.

Adaptif kontrol (adaptive control) tekniği kaotik sistem davranışını yok edecek şekilde sistemin parametrelerini değiştirmeye dayanır. Sistemin asimptotik davranışı üzerinde parametre değişiminin etkisinin analizi, değişim parametresinin hangi değerlerinde sistemin dallanmaya veya kaotik bölgeye girdiğini gösteren çatallaşma diyagramları kullanılarak yapılabilir. Bu çatallaşma diyagramlarından yararlanılarak kararlı sabitleştirilmiş noktalardan, periyot-1, periyot-2 ya da daha yüksek periyot yörüngelerinden herhangi birini elde etmek için değişim parametresinin uygun değerleri belirlenebilir (Kılıç, 1996). Kaotik sistemin kontrolünde yaygın olarak kullanılmıştır (Zeng ve Singh, 1997; Pan ve ark., 2010; Li ve Tong, 2013; Wang ve ark., 2017; Luo ve ark., 2018).

Pasif kontrol (passive control) yöntemi kaotik sistemlerdeki durum değişkenlerinin bir kısmının pasifleştirilmesi, etkin olan diğer durum değişken(ler)i üzerinden Lyapunov üstelleri yardımıyla sistemin kontrolünün sağlanması mantığına dayanmaktadır. Pasif kontrol kullanılarak Lorenz sisteminin kontrolü Yu (1999) tarafından önerilmiştir. Sonrasında Chen (Qi ve ark., 2004), Lü (Kemih ve ark., 2006) ve Rabinovic (Emiroğlu ve Uyaroğlu, 2010) kaotik sistemlerine uygulanmıştır. Hiperkaotik 4 boyutlu (Lai ve He, 2018) ve n -boyutlu (Mahmoud ve ark., 2013) sistemlerin kontrolünde de kullanılmıştır.

Kaotik sistemlerin kontrolünde, formülü net olarak bilinmeyen doğrusal olmayan sistemlere çözüm yöntemi olarak kullanılan yapay zekâ tekniklerinden de faydalanılmaktadır. Bu kontrol yöntemlerinin anlatılması için önce yapay zekâ ve metodolojileri hakkında bilgi verilecektir.

2.4. Yapay Zekâ

Zekâ; öğrenme, anlama, düşünme ve karar verme yetisidir. Bilgisayarların belli ölçülerde öğrenme, anlama, düşünme, değerlendirme, karar verme yeteneğine sahip olmasına ise yapay zekâ denir. Bu amaca ulaşmak için, yapay zekâ genellikle beynin çalışma mekanizmasını, doğanın biyolojik evrimini veya insanın düşünme yeteneğini modellemeye çalışan yöntemleri kullanmaktadır (<https://fikirjeneratoru.com>, 2017). Günümüzde kullanılan başlıca yapay zekâ teknikleri uzman sistemler, yapay sinir ağları, bulanık mantık, genetik algoritmalar ve zeki etmenlerdir.

Uzman sistemler (expert systems), kural tabanlı sistemler olarak tanımlanabilir. Çözümüne nitel yönden ulaşılamayan veya ulaşılması zor olan problemleri, uzmanlık ve deneyim ile çözer. Bu çözümleme işlemini; teknik bilgi işletimi, sezgisellik ve insan uzmanlığı süreçleri yardımıyla gerçekleştirir.

Yapay Sinir Ağları (YSA) (artificial neural networks), beynin öğrenme sistemindeki sinir hücreleri ve bağlantılarının çok basit bir modelinin bilgisayar ortamında uygulanması ile başlamış bir yapay zekâ tekniğidir. Makine öğrenmesi gerçekleştirir. Örnek verilerden elde ettikleri bilgileri kendi matematik sistemlerine adapte eder ve daha sonra yakın konularda benzer, uygun, mantıklı sonuçlar üretirler.

Bulanık mantık (fuzzy logic), belirsizlikleri algılayabilen ve onlar üzerinde işlemler yapabilen bir matematik düzeni olarak tanımlanabilir. Günlük hayatta yapılan çoğu dilsel nitelendirme kesin verilerden oluşmaz. Bu veriler bulanık kümeler halinde ifade edilir ve uzman sistemlerde olduğu gibi bir kural tabanı ile çözüme ulaşılır (Baykal ve Beyan, 2004).

ANFIS (Adaptive Neuro-Fuzzy Inference System – Uyarlamalı Sinirsel-Bulanık Çıkarım Sistemi) Sugeno bulanık çıkarım sisteminin sinirsel öğrenme kabiliyetine sahip bir ağ yapısı olarak uyarlanmış halidir. Sinirsel-bulanık ağlar olarak da adlandırılmaktadır. ANFIS, hem bulanık mantık çıkarım hem de YSA teknikleri yapılarının her ikisinden birden faydalanır. Bu da, daha iyi öğrenme açısından avantaj sağlayabilmektedir.

Genetik algoritmalar (genetic algorithms), çoğunlukla karmaşık optimizasyon problemlerine çözüm arayan zeki bir tekniktir. Yapısı kromozom, gen, çaprazlama, mutasyon gibi biyolojik evrimde kullanılan kavramları içerir (Öztemel, 2003). Temel mantığı, problemlerin çözümlerini sürekli değiştirmekle daha iyi çözümler aramaya ve üretmeye dayanmaktadır.

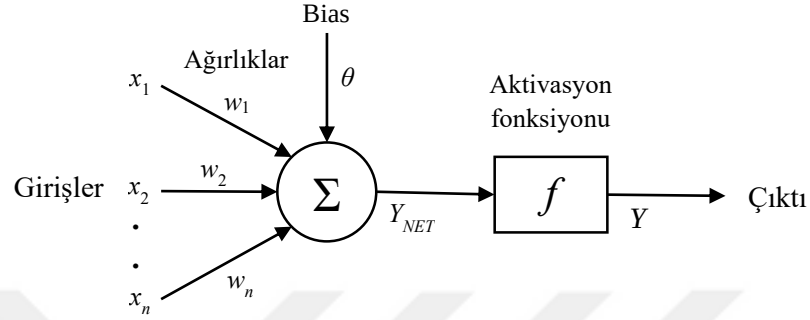
Zeki etmenler (intelligent agents), bağımsız karar verebilen ve değişik yapay zekâ tekniklerini kullanabilen esnek bilgisayar sistemleridir. Hem donanım hem de yazılım versiyonları mevcuttur. Öğrenme ve gerçek zamanlı çalışabilme özellikleri vardır. Genel olarak bir zeki etmenin yapısında algılama, kavrama/ıdrak ve eylem elemanları bulunmaktadır (Öztemel, 2003).

Kaotik sistemler diferansiyel denklemlerle ifade edilmektedir ve bunlar sayısal veriler oluşturmaktadır. Bundan dolayı, kaotik sistemlerin yapay zekâ teknikleri ile kontrolü için en yatkınları YSA ve sinirsel-bulanık ağlardır. Ayrıca kontrol parametrelerinin daha büyük, büyük, küçük, daha küçük vb. değerlere göre alınmasına da yapılan önemli çalışmalarda sıkça rastlanılmaktadır, burada kullanılan yapay zekâ metodolojisi ise bulanık mantıktır. Bu yüzden tezin devamında, YSA, bulanık mantık ve sinirsel-bulanık ağlar teknikleri detaylarıyla verilmiştir.

2.4.1. Yapay sinir ağları

İnsan beyni 10^{10} adet sinir hücresi ve bunlar arasında 6×10^{13} 'ten fazla sayıda bağlantıdan oluşan karmaşık bir sistemdir. Sinir hücreleri bu bağlantılar sayesinde birbirleriyle iletişim kurarlar. YSA algoritmaları, biyolojik sinir sistemi

mimarisinden esinlenilerek geliştirilmiştir. Günümüzde kullanılan çok sayıda farklı YSA modeli vardır. Bu modellerin geliştirilmesinde hem biyolojik sinir sistemi prensiplerinden hem de mühendislik bilimi imkânlarından faydalanılmıştır. Şekil 2.16.'da bir yapay sinir hücresi yapısı gösterilmiştir.



Şekil 2.16. Yapay sinir hücresinin yapısı.

YSA klasik teknikler kullanılarak modellenmesi imkânsız veya çok zor olan sistemlerin modellenmesinde oldukça etkilidirler. İnsan beyninin işlevsel özelliklerine benzer şekilde öğrenme, tahmin etme, tanıma, sınıflandırma, ilişkilendirme ve optimizasyon gibi konularda başarıyla uygulanmaktadır. Çalışma stili, klasik programlama yöntemlerine benzememektedir. YSA makine öğrenmesi gerçekleştirir. Önce eğitilmeleri gerekmektedir. Bunun için örnek veriler ağı tek tek gösterilerek işlenir. YSA'nda bilgi, ağı bağlantılarında bulunan ağırlık değerlerinde saklanır. YSA eğitimi, verilen giriş-çıkış çiftlerinden ağı parametrelerini ayarlama temeline dayanır. Eğitimden sonra YSA, eğitim esnasında görmediği örneklerle de kabul edilebilir doğrulukta bilgi, cevap üretebilirler.

YSA özellikle sisteme ait bilginin tam olmadığı veya hatalı olduğu lineer olmayan sistemlerde çözüme ulaşmak için uygundur. Bunu, mevcut verilerden öğrenme ile, ağırlık bağlantılarıyla kendi kendilerini biçimlendirerek gerçekleştirirler. Eksik bilgi ile çalışabilmektedirler ve hata toleransına sahiptirler. YSA'nın dezavantajı ise, bir uzman bilgisi olsa dahi bunu problem çözümüne aktarmadaki zorluktur, uzmanlığın sayısal örnek kümesi haline dönüştürülmesi bazen zordur. Görüntü ve ses tanıma, tahmin ve kestirim, kontrol, iş çizelgeleme, tıp, haberleşme, askeriye, trafik, üretim yönetimi başlıca kullanım alanları olarak sayılabilir (Öztemel, 2003; Haykin, 2005).

2.4.1.1. Yapay sinir ağlarının tarihçesi

İnsanoğlu beynin nasıl çalıştığını tarih boyunca merak etmiştir. İnsan beyninin yapısı ve fonksiyonları ile ilgili ilk eser 1890 yılında yayınlandı. 1910'lu yıllarda beynin bileşenleri, nöronlar ve çalışma düzeneği hakkında daha detaylı veriler elde edildi. YSA kavramı 1940'dan önce ortaya çıkmıştır ama mühendislik değeri olan asıl çalışmalar Hebb, McCulloch ve Pitts gibi bilim adamlarının 1940'lı yıllarda yapay sinir hücrelerine dayalı hesaplama teorisini öne sürmesi, ilk yapay sinir hücresi yapısını oluşturması ve hücrelerin birbirleri ile paralel çalışması gerektiğini belirlemesi ile gerçekleşti. Hebb 1949 yılında, yapay sinir hücrelerinin öğrenme prosedürünü gerçekleştirebilecek biçimde ağırlıklarını değiştirebilmesini sağladı. "Hebbian öğrenme" adı verilen bu kural, daha sonra birçok öğrenme kuralının temelini oluşturdu.

1950'li ve 1960'lı yıllarda YSA üzerine ilgi arttı. İlk nöro-bilgisayar 1951 yılında üretildi. Farley ve Clark 1954 yılında rassal ağlar (random networks) ile adaptif tepki üretme kavramını ortaya attı. Rosenblatt tarafından 1958 yılında geliştirilen tek katmanlı algılayıcı modeli (perceptron), daha sonra geliştirilecek olan çok katmanlı algılayıcıların temelini oluşturdu. 1960 yılında Widrow ve Hoff, ADALINE (ADaptive LInear NEuron – Adaptif Lineer Nöron) modelini geliştirdi. Rosenblatt'ın algılayıcı modelinin sadece öğrenme algoritmasının daha gelişmiş bir hali olan ADALINE, mühendislik uygulamalarında YSA kullanılmasını başlatan adımlardan biri oldu. Bu model, 1970'li yılların sonlarında Multiple-ADALINE (MADALINE) modeline yerini bıraktı.

Bu arada, 1956 yılında "yapay zekâ" kavramı gündeme geldi ve bilim adamlarının ilgisi yapay zekâya yöneldi. İlk yapay zekâ çalışmalarında YSA'na az değinilmesi nedeniyle YSA popüleritesini yitirdi. YSA çalışan bilim adamlarından Nilssons 1965 yılında "Öğrenen makineler" adında bir kitap yayınladı ve tekrardan ilgiyi YSA'na çekti.

1969 yılında Misnky ve Pappert tek katmanlı algılayıcıların doğrusal (lineer) olmayan problemleri çözme yeteneğinin olmadığını XOR (YA DA) örneği üzerinden gösterdi ve YSA çalışmaları durma noktasına geldi. XOR problemi çok katmanlı algılayıcılar ile çözülmeye kadar araştırmacıların ilgisi YSA'na pek yönelmedi.

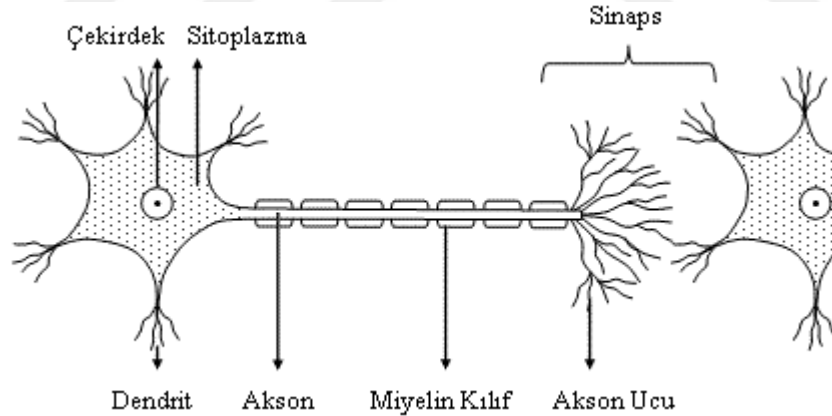
1970'lerin başlarında çok katmanlı algılayıcıların ilk çalışmaları olan geri yayılım modeli oluşturuldu. Grossberg ve Carpenter 1978 yılında adaptif rezonans teorisi (Adaptive Resonance Theory – ART) ile, Kohonen 1982 yılında kendi kendine öğrenme nitelik haritaları (Self Organizing feature Maps – SOM) ile öğretmensiz öğrenme yapabilen YSA'nı geliştirdi. Fukushima 1980 yılında örüntü tanıma amaçlı öğretmensiz öğrenme yapan bir NEOCOGNITRON modeli üretti. Bu modelini 1982 yılında öğretmenli öğrenme haline dönüştürdü. Ayrıca bu model, ara katmanlı öğrenme konusuna da değiniyordu.

Hopfield 1982 ve 1984 yıllarında yayınladığı çalışmalarında geri-beslemeli YSA modelini ortaya çıkardı ve çözülmesi zor gezgin satıcı problemi üzerinde başarıyla uyguladı. Rumelhart ve McClelland 1986 yılında paralel hesaplama konusundaki çalışmalarını sundu ve bunlar da çok katmanlı algılayıcıların gelişimi açısından önemli rol oynadı. Aynı yıllarda çok katmanlı algılayıcıların bulunması, genelleştirilmiş delta öğrenme kuralının geliştirilmesi, XOR probleminin çözülmesi, dikkatleri YSA üzerine tekrardan çekti.

Broomhead ve Lowe 1988 yılında çok katmanlı algılayıcılara alternatif olan, filtreleme problemlerinde iyi sonuçlar üreten radyal tabanlı fonksiyonlar (Radial Basis Functions – RBF) modelini oluşturdu. Specht bu ağların daha gelişmiş biçimleri olan probabilistik ağları (Probabilistic Neural Network – PNN) 1988 yılında ve genel regresyon ağlarını (General Regression Neural Network – GRNN) 1991 yılında geliştirdi. YSA konusunda 1991'den günümüze, bilgisayarların yaygınlaşması, kapasitelerinin artması ile birlikte sayısız çalışma ve uygulama konferanslarda sunulmuş, dergilerde yayınlanmış, yeni model ve öğrenme teknikleri ileri sürülmüştür (Öztemel, 2003).

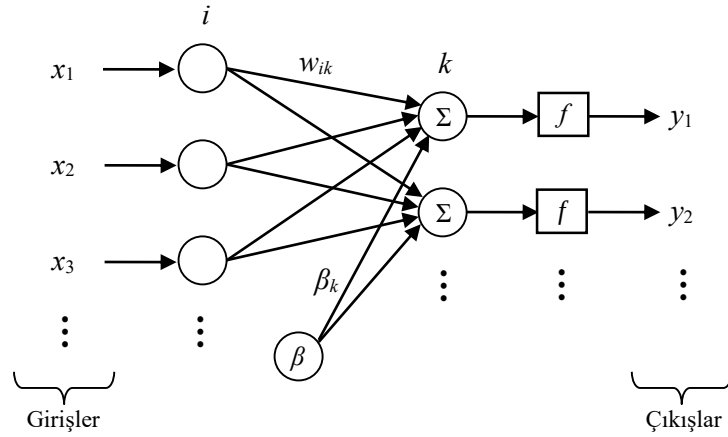
2.4.1.2. Biyolojik sinir hücreleri ve yapay sinir ağları

Sinir sistemi; beyin, omurilik, duyu organları ve vücuttaki diğer organlar arasındaki bağlantıyı sağlayan milyarlarca sinir hücresinden (nöronlardan) oluşmaktadır. Şekil 2.17.'de gösterildiği gibi sinir hücreleri; hücre gövdesi, bu gövdeden çıkan kısa uzantılar olan dendrit ve uzun tek bir uzantı olan akson olmak üzere 3 ana parçadan oluşur. Hücre gövdesinde çekirdek ve sitoplazma bulunur. Gelen uyarılar dendritler tarafından alınarak hücre gövdesine iletilir, gövdeden aksona aktırılır. Bazı sinir hücrelerinde akson üzerinde iletimin daha hızlı geçişini sağlayan miyelin kılıf da vardır. Akson kendisine gelen uyarıları bir sonraki hücreye sinapslar yoluyla aktarır. Sinapslar sinir hücrelerinden bir diğer sinir hücresine veya salgı bezleri, kas gibi farklı hücelere iletimin aktarılmasını sağlayan özelleşmiş bağlantı noktalarıdır. İç ve dış merkezlerden gelen uyarılar (ısı, ışık, ağrı, vb.) sinir hücreleri ile merkezi sinir sistemine taşınır, burada değerlendirilir, oluşan cevaplar kas ve salgı bezi gibi hücelere ileterek uygun cevapların verilmesi sağlanır. Biyolojik sinirlerin bu özelliklerinden esinlenilerek YSA geliştirilmiştir.



Şekil 2.17. Biyolojik sinir hücrelerinin yapısı.

Biyolojik sinir ağlarındaki sinir hücreleri yapay sinirlere yani işlem elemanlarına, sinapslar ağırlıklara, dendritler toplama fonksiyonuna, hücre gövdesi aktivasyon fonksiyonuna (nöronun aktivasyon düzeyine) ve aksonlar çıkış elemanlarına benzetilerek YSA oluşturulmuştur. YSA'nın temel bir modeli Şekil 2.18.'de gösterilmiştir.



Şekil 2.18. Yapay sinir ağı modeli.

Şekil 2.18.'de $x_1, x_2, \dots, x_n$ matematiksel sembolleri ile gösterilen değerler nöron girişleridir. Bu girdi değerlerinin her biri önce w_{ik} ağırlık değeri ile çarpılmaktadır. Sonra bu çarpımlar toplanır. Ayrıca, eşik değeri β_k eklenerek bu değer artırılabilir. Bu işlemlerin matematiksel gösterimi aşağıdaki gibidir:

$$Net_{y_k} = w_{1k}x_1 + w_{2k}x_2 + \dots + w_{nk}x_n + \beta_k = \sum_{i=1}^n w_{ik}x_i + \beta_k. \quad (2.14)$$

Girişler X vektörü ile ağırlıkları ise W vektörü ile gösterilirse, Denklem 2.14 vektör notasyonu olarak Denklem 2.15'deki gibi ifade edilebilir (Sağıroğlu ve ark., 2003):

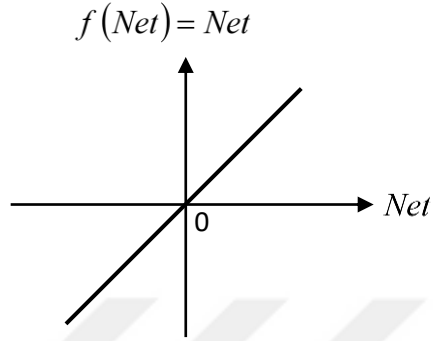
$$Net = WX + \beta. \quad (2.15)$$

Son olarak, toplam Net değerleri aktivasyon fonksiyonuna gönderilerek $y_1, y_2, \dots, y_m$ çıktı değerleri hesaplanır.

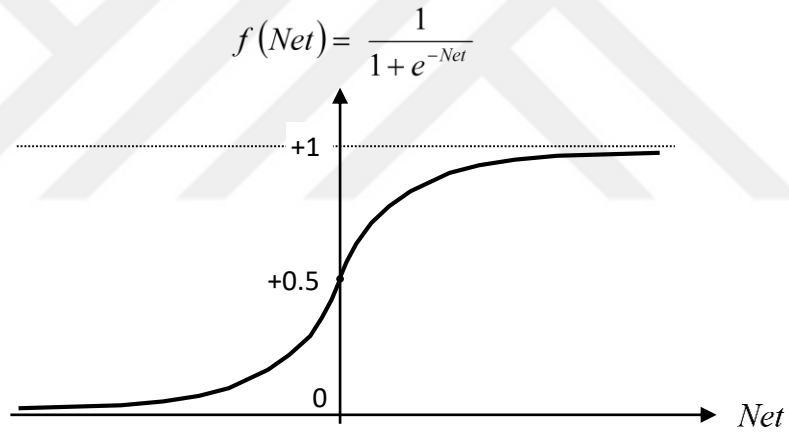
2.4.1.3. Aktivasyon fonksiyonları

Aktivasyon veya diğer adıyla transfer fonksiyonları, gelen Net girdi değerini işler ve bu girdiye karşılık gelen çıktı değerini belirlerler. Bu çıktı değerleri çoğunlukla $[0, 1]$ ya da $[-1, 1]$ aralığındadır. Aktivasyon fonksiyonu olarak değişik formüller

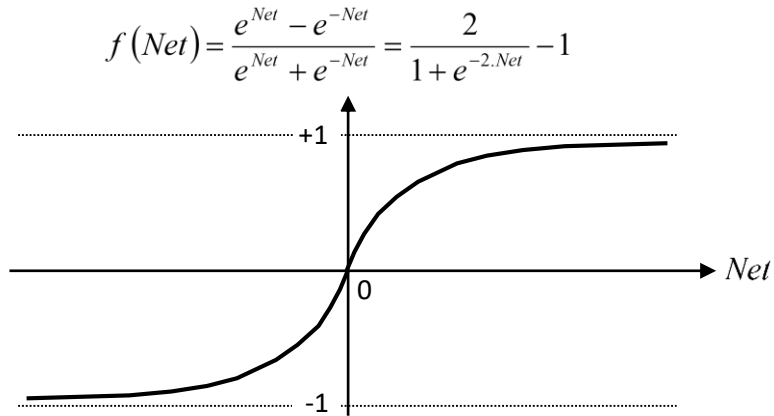
kullanılmaktadır. Çok katmanlı algılayıcılar gibi YSA modellerinde bu fonksiyonun türevi alınabilir ve sürekli bir fonksiyon olması şarttır. En çok kullanılan aktivasyon fonksiyonları sırasıyla Şekil 2.19., Şekil 2.20. ve Şekil 2.21.'de gösterilen lineer, sigmoid ve tanjant-sigmoid fonksiyonlarıdır (Sağıroğlu ve ark., 2003).



Şekil 2.19. Lineer fonksiyon.



Şekil 2.20. Sigmoid fonksiyonu.



Şekil 2.21. Tanjant-sigmoid fonksiyonu.

Günümüzde çok katmanlı algılayıcı modellerde en çok tercih edilen aktivasyon fonksiyonu sigmoid fonksiyonudur (Öztemel, 2003).

2.4.1.4. Yapay sinir ağları modelleri

YSA ağın yapısına göre, ileri beslemeli (feed-forward) ve geri beslemeli (feed-back) ağlar olarak iki alt başlıkta değerlendirilir:

İleri beslemeli YSA’nda, işlem elemanlarının çıktıları giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla ilerler. İşlem elemanları daima kendinden bir sonraki katmanla bağlantı kurarlar, aynı katman içerisinde bağlantı kurmazlar ve çıktıların geri dönüşü de yoktur (Öztemel, 2003). Çok katmanlı algılayıcılar, probabilistik ağlar ve LVQ (Learning Vector Quantization – Öğrenmeli Vektör Kuantalama) ağları bu tip ağlara örnek olarak verilebilir.

Geri beslemeli YSA’nda, işlem elemanlarının çıktıları sadece ileri doğru değil, aynı zamanda geriye doğru da gönderilmektedir. Ağın çıkışı veya ara katmanlardaki çıkışların, ağın giriş birimine veya kendinden bir önceki katmana geri beslendiği ağ yapısıdır (www.akanesen.com, 2018). Geri dönüşümlerin olması özellikle zaman gecikmelerinin dikkate alınmasını sağlar (Öztemel, 2003). Hopfield ağı tek katmanlı ve geri dönüşümlü, Elman ağı kısmi geri dönüşümlü YSA’dır.

YSA öğrenme kuralına göre, Hebb, Hopfield, delta ve Kohonen olarak dört alt başlıkta değerlendirilebilir:

Hebb kuralı bilinen ilk öğrenme algoritmasıdır. “Bir hücre (nöron) diğer bir hücreden bilgi alırsa ve her iki hücre de aktif ise (matematiksel olarak aynı işarete sahipse), bu hücreler arasındaki ağırlıklar kuvvetlendirilir” prensibine göre çalışmaktadır. Diğer öğrenme kurallarının temelini oluşturmuştur (Öztemel, 2003).

Hopfield kuralında işlem elemanlarının ağırlıklarının ne kadar kuvvetlendirilmesi ya da zayıflatılması gerektiği bir öğrenme katsayısıyla gerçekleştirilmektedir. “Eğer

beklenen çıktının ve girdinin her ikisi de aktif ise öğrenme katsayısı kadar ağırlık değeri artırılır, aksi durumlarda ise azaltılır” ilkesini kullanmaktadır. Öğrenme katsayısı çoğu zaman 0 ile 1 arasında sabit bir değer olarak alınır (Öztemel, 2003).

Delta kuralı, “ağın üretmesi gereken (beklenen) çıktılar ile ürettiği çıktılar arasındaki fark (hata) ağın işlem elemanlarının ağırlıklarına geri yayılımla dağıtılarak sürekli değiştirilir” ilkesine dayanmaktadır. Bu algoritma, hatanın karelerinin ortalamasını düşürmeyi hedeflemektedir (Öztemel, 2003).

Kohonen kuralı, “ağın elemanları (nöronlar) birbirleri ile yarıştırılır, kazanan (en büyük çıktıyı üreten) elemanların ağırlıkları güncellenir, böylece bu hücreler etrafındaki hücrelere karşı daha kuvvetli hale gelir” prensibi ile çalışmaktadır (Öztemel, 2003). Genellikle sınıflandırma işleminde kullanılmaktadır, kazanan eleman girdinin sınıfını göstermektedir. Bu kural hedef çıktıya gereksinim duymadığından danışmansız bir öğrenme yöntemidir.

YSA öğrenme stratejisine göre, danışmanlı (supervised), destekleyici (reinforcement) ve danışmansız (unsupervised) olarak üç alt başlıkta değerlendirilir:

Danışmanlı öğrenme stratejisinde, ağa öğrenilmesi istenen girdi ve çıktı değerleri aynı anda verilir. Verilen girdilere göre istenen çıkış değerlerini oluşturabilmek için ağ kendi ağırlıklarını günceller. Önce ağın çıktıları ile beklenen çıktılar arasındaki fark bulunur, bu farka göre her bir nörona düşen hata payı hesaplanır ve sonrasında nöronların ağırlıkları bu hata paylarına göre güncellenir. Böylece olayın girdileri ile çıktıları arasındaki ilişkiler öğrenilmiş olur. Çok katmanlı algılayıcılar bu tür öğrenme stratejisini kullanan ağlara örnek olarak verilebilir (Şen, 2004).

Destekleyici öğrenme stratejisinde, öğrenen sisteme yine bir öğretici (danışman) yardımcı olur. Bazı durumlarda ağın üretmesi gerektiği çıktıyı anlık vermek mümkün olamamaktadır, fakat sonrasında ağın ürettiği çıktının doğru ya da yanlış olduğu belirtilebilmektedir. Ağ, öğreticiden gelen bu bilgiyi dikkate alarak öğrenme sürecini gerçekleştirir (Öztemel, 2003). Örneğin, satranç oyunu öğretilen bir YSA yaptığı

hamlenin iyi mi kötü mü olduğunu anlık olarak ayırt edemez, yine de hamlesini yapar. Eğer oyunun sonunda oyun kazanıldıysa yapılan hamlelerin iyi, kaybedildiyse kötü olduğunu varsayacak ve sonraki oyunlarda benzer hamleleri iyi/kötü olarak değerlendirecektir. LVQ destekleyici öğrenme stratejisini kullanan bir YSA modelidir.

Danışmansız öğrenme stratejisinde, ağa sadece girdi değerleri verilir, herhangi bir beklenen çıktı bilgisi verilmez. Genellikle sınıflandırma problemlerinde kullanılan bir öğrenme biçimidir. Verilen girdi değerlerine göre ağın ilişkileri kendi kendine öğrenmesi beklenir. Öğrenme bittikten sonra, kullanıcılar tarafından çıktıların ne anlama geldiğini gösteren bir etiketlendirme yapılabilir (Öztemel, 2003). ART ve SOM bu stratejiyi kullanan ağlardır (Şen, 2004).

YSA öğrenme zamanına göre, statik (off-line) ve dinamik (on-line) olarak iki alt başlıkta değerlendirilir:

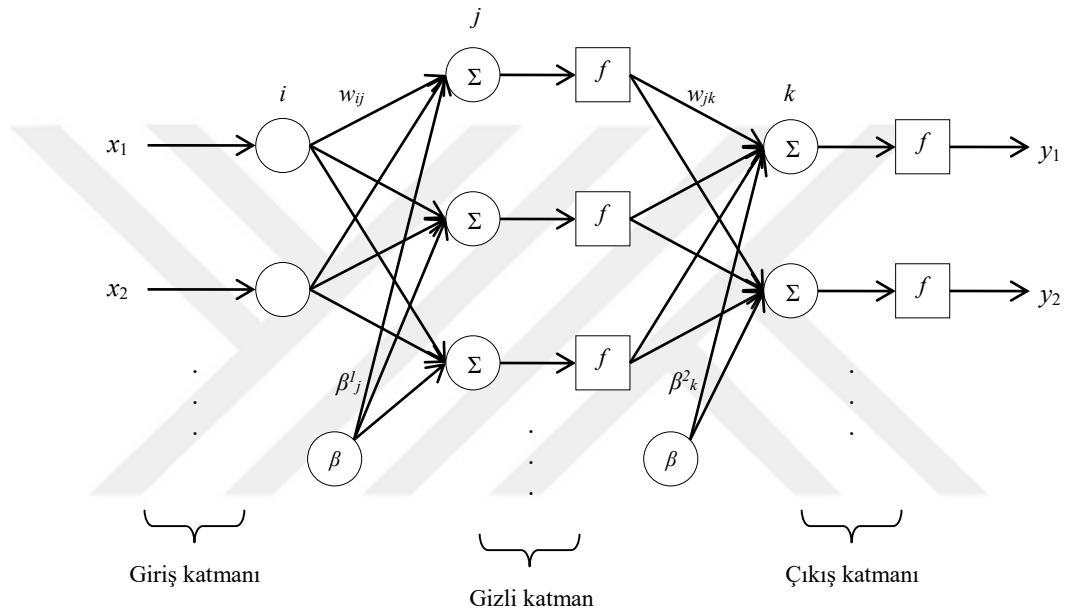
Statik öğrenmede, YSA kullanılmadan önce eğitilirler. Eğitim sonucunda elde edilen ağırlıklar ile ağ, daha sonra istenilen şekilde kullanılabilir. Bu bir internet uygulaması da olabilir, fakat kullanılma aşamasında hiçbir öğrenme işlemi yapılmaz, ağın ağırlıklarında değişiklik olmaz.

Dinamik öğrenmede ise, ilk eğitim aşaması bittikten sonra da, YSA'nın kullanıldığı süre zarfında bir taraftan öğrenme işlemine devam edilir. Sistemin zamanla çok farklı davranışlar sergileyebildiği durumlarda tercih edilmektedir. Çevrimiçi veya gerçek zamanlı öğrenme yapısını destekleyen donanım ve yazılımlar mevcuttur.

2.4.1.5. Çok katmanlı yapay sinir ağları

YSA'nın ilk modellerinden olan tek katmanlı algılayıcılar (perceptron) doğrusal problemleri çözebilme yeteneğine sahiptirler ancak doğrusal olmayan ilişkileri pek öğrenememektedirler. Bu sorun, çok katmanlı algılayıcıların geliştirilmesi ile çözüme kavuşmuştur. Çok katmanlı YSA, tek katmanlı YSA'nın art arda yer

almasıyla oluşur. Bu ağlar ileriye doğru bağlantılıdır ve yapısı giriş katmanı, ara (gizli) katman ya da katmanlar ve çıkış katmanından oluşmaktadır. Ara katman sayısının fazla olması ilişkilerin daha başarılı öğrenilmesini sağlayabilir. Ancak, hesaplama yükünü üstel bir biçimde artırır. Genellikle makul sayıda nöron içeren 1 ara katman problemin çözümü için yeterli gelmektedir. Dolayısıyla, uygulamalarda yaygın olarak 3 katmanlı YSA kullanılmaktadır, genel yapısı Şekil 2.22.'de gösterilmiştir.



Şekil 2.22. Üç katmanlı yapay sinir ağı yapısı.

Çok katmanlı YSA’nda, ilk önce giriş katmanı dışarıdan gelen girdileri ($x_1, x_2, \dots, x_n$) alır, işler ve gizli katmana gönderir. Giriş katmanındaki her bir işlem elemanı (nöron), bir sonraki ara katmandaki tüm işlem elemanlarıyla bağlantılıdır. Gizli katman, giriş katmanından gelen bilgileri işleyerek çıkış katmanına gönderir. Çok katmanlı YSA’nda birden fazla gizli katman ve her katmanda birden fazla işlem elemanı olabilir, sayısı genellikle deneme yanılma yöntemi ile bulunur. Yine, gizli katmandaki her bir işlem elemanı bir sonraki katmanda (gizli katmanda ya da çıkış katmanında) bulunan işlem elemanlarının hepsine bağlıdır. Çıkış katmanı, gizli katmandan gelen bilgileri işleyerek ağın cevabını ($y_1, y_2, \dots, y_m$) dışarıya sunar (Öztemel, 2003).

2.4.1.6. Çok katmanlı yapay sinir ağlarının eğitilmesi

Çok katmanlı YSA, danışmanlı öğrenme stratejisine göre çalışırlar. Yani, ağa girdi ve çıktı seti aynı anda verilerek öğretilir. Bu ağlar, öğrenme işlemini genelleştirilmiş delta kuralı ile gerçekleştirir. Bu kural, ağın çıktısının hesaplandığı ileriye doğru hesaplama ve ağırlıkların hata miktarına göre değiştirildiği geriye doğru hesaplama olmak üzere 2 safhadan oluşmaktadır.

İleri doğru hesaplama safhası, eğitim setindeki ilk örneğin giriş katmanından ağa gösterilmesi ile başlar. Önce, gelen girdiler hiçbir değişiklik olmadan ilk katmandaki çıktı olarak alınırlar, yani i . işlem elemanı ζ_i^1 'nin değeri x_i olarak belirlenir. Arkasından, sırasıyla ara katmanlardaki ve çıkış katmanındaki çıktı değerleri aşağıdaki formül kullanılarak hesaplanır:

$$\zeta_j^l = f \left(\sum_{i=1}^n w_{ij}^l \zeta_i^{l-1} + \beta_j^l \right), \quad (2.16)$$

burada l katman numarasını, j işlem elemanı numarasını, ζ_j^l l . katmandaki j . işlem elemanı çıktı değerini, f aktivasyon fonksiyonunu, n bir önceki katmandaki eleman sayısını, w_{ij} bir önceki katmanın i . elemanı ile mevcut katmanın j . elemanı arasındaki ağırlığı, β bias değerini temsil etmektedir. Uygulamada genellikle sigmoid aktivasyon fonksiyonu kullanılmaktadır. Ağın ağırlık ve bias değerleri başlangıçta rastgele değerler olarak alınırlar. İşleme, ağın çıkış katmanındaki değerler bulununcaya kadar devam edilir ve ileri hesaplama safhası tamamlanmış olur.

Geriye doğru hesaplama safhası, ağın ürettiği hatanın tespiti ve ağın ağırlıklarının değiştirilmesi işlemlerini içermektedir. Ağın verilen girdi değeri için ürettiği çıktı ζ , asıl bulması istenen eğitim setindeki çıktı değeri y ile ifade edilirse, çıkış katmanındaki k . işlem elemanı için oluşan hata $E_k = y_k - \zeta_k$ olacaktır. Ağın eğitilmesindeki amaç hata miktarını düşürmektir. Ancak, toplam hatayı nötralize edebilecek negatif hatayı önlemek hem de daha büyük hataya daha çok odaklanmak amacı ile hatanın karesi üzerinden işlem yapılır:

$$E_k = \frac{1}{2} \sum_{k=1}^m (y_k - \zeta_k)^2. \quad (2.17)$$

Hata, kendisine neden olan işlem elemanlarına dağıtılır. Önce çıkış katmanı ile ara katman arasındaki ağırlıklar güncellenir, sonra ara katmanlar arası veya ara katman ile giriş katmanı arasındaki ağırlıklar güncellenir.

l katmanındaki j . işlem elemanını k . işlem elemanına bağlayan bağlantının ağırlığındaki değişim miktarı Δw_{jk}^l olarak ifade edilirse, herhangi bir t . iterasyonda ağırlığın değişim miktarı aşağıdaki formül ile bulunur:

$$\begin{aligned} \Delta w_{jk}^l(t) &= -\eta \frac{\partial E_k}{\partial w_{jk}^l} + \alpha \Delta w_{jk}^l(t-1) = -\eta \frac{\partial E_k}{\partial \zeta_j^l} \frac{\partial \zeta_j^l}{\partial w_{jk}^l} + \alpha \Delta w_{jk}^l(t-1) \\ &= \eta \delta_k^l \zeta_j^l + \alpha \Delta w_{jk}^l(t-1). \end{aligned} \quad (2.18)$$

Benzer biçimde, bias ağırlıklarındaki değişim miktarının bulunması da gerekmektedir. l katmanındaki k . işlem elemanına ait bias değişim miktarı $\Delta \beta_k^l$, bu ünitenin çıktısının sabit ve 1 olması nedeniyle,

$$\begin{aligned} \Delta \beta_k^l(t) &= -\eta \frac{\partial E_k}{\partial \beta_k^l} + \alpha \Delta \beta_k^l(t-1) = -\eta \frac{\partial E_k}{\partial \zeta_j^l} \frac{\partial \zeta_j^l}{\partial \beta_k^l} + \alpha \Delta \beta_k^l(t-1) \\ &= \eta \delta_k^l + \alpha \Delta \beta_k^l(t-1) \end{aligned} \quad (2.19)$$

olacaktır. Yukarıdaki formüllerde kullanılan η öğrenme katsayısı, α ise momentum katsayısıdır. Öğrenme katsayısı, ağırlıkların değişim miktarını belirler. Büyük değerlerde ağırlık değişim miktarı artar, küçük değerlerde ise öğrenme zamanını arttırır. Değeri genellikle 0,2 ile 0,4 arasında tercih edilmektedir. Momentum katsayısı ise ağırlık değişim değerlerine önceki değişimlerin belli bir oranda eklenmesini sağlayarak, öğrenme esnasında momentum etkisi yaratır ve öğrenmenin yerel bir optimum noktada yavaşlayıp durma ihtimalini azaltır. Değeri genellikle 0,6 ile 0,8

arasında tercih edilmektedir (Öztemel, 2003). Eşitliklerdeki δ_k^l ise l katmanındaki k ünitesinin hatasını göstermektedir, çıkış katmanı için değeri şöyle hesaplanır:

$$\delta_k = f'(NET)E_k, \quad (2.20)$$

burada $f'(NET)$ aktivasyon fonksiyonunun türevidir. Sigmoid fonksiyonunun türevi:

$$\begin{aligned} \frac{ds(x)}{dx} &= \left(\frac{1}{1+e^{-x}} \right)^2 \frac{d}{dx} (1+e^{-x}) = \left(\frac{1}{1+e^{-x}} \right)^2 e^{-x} (-1) = \left(\frac{1}{1+e^{-x}} \right) \left(\frac{1}{1+e^{-x}} \right) (-e^{-x}) \\ &= \left(\frac{1}{1+e^{-x}} \right) \left(\frac{-e^{-x}}{1+e^{-x}} \right) = s(x)(1-s(x)), \end{aligned} \quad (2.21)$$

olarak hesaplanabilir. Dolayısıyla, sigmoid aktivasyon fonksiyonunun kullanılması durumunda hata değeri ara katmanlar için

$$\delta_k^l = \zeta_k^l (1 - \zeta_k^l) \sum_{j=1}^m (\delta_j^l \zeta_{jk}^l) \quad (2.22)$$

olacaktır. Değişim miktarı hesaplandıktan sonra, ağırlıkların t . iterasyondaki yeni değerleri şu şekilde bulunur:

$$w_{jk}^l(t) = w_{jk}^l(t-1) + \Delta w_{jk}^l(t). \quad (2.23)$$

Bias değerinin t . iterasyondaki ağırlıklarının yeni değerleri ise

$$\beta_k^l(t) = \beta_k^l(t-1) + \Delta \beta_k^l(t) \quad (2.24)$$

biçiminde hesaplanır. Böylece, ağırlık her bir ağırlığı verilen 1 girdi-çıkış değeri için hem ileri hem de geriye hesaplamaları yapılarak güncellenmiş olur. Ardından, bir sonraki örnek ağı gösterilir. Aynı işlemler, tüm örnekler için kabul edilebilir oranda doğru cevap üretinceye kadar yinelenir (Öztemel, 2003).

İleri beslemeli çok katmanlı YSA’nda en çok kullanılan öğrenme algoritması, yukarıda matematiksel formülleri ile verilen “geriye yayılım” algoritmasıdır. Bu algoritma, ağırlıkların toplam hata miktarını (performans kriterini) en aza indirecek şekilde ağırlıkları düzenleyebilmek için, performans fonksiyonunun gradyenini kullanır. Yani, hata değerleri bulunduğundan sonra, çıkış hatalarının türevi çıkış katmanından geriye doğru gizli katmanlara ve sonrasında giriş katmanına geçirilir. Ağırlıklar, performans fonksiyonunun azalması yönünde düzenlenir. Geriye yayılım öğrenme kuralı, en genel anlamıyla her katmandaki toplam karesel hatanın gradyeninin tahminine dayalı, gradyen azalması (steepest descent) algoritmasıdır. Fakat bu öğrenme yöntemi, çok sayıda veri içeren problemlerde çok yavaş kalmaktadır. Gradyen azalması algoritmasından daha hızlı, daha yüksek performanslı algoritmalar günümüzde geliştirilmiştir.

Hızlı öğrenme algoritmaları başlıca 2 kategori halinde incelenebilir. Birinci kategorideki algoritmalar geriye yayılım işleminde deneme yanılma yöntemini de kullanırlar ve standart gradyen azalması algoritmasından daha iyi sonuçlar elde ederler. Bu algoritmalar; momentum terimli geriye yayılım, öğrenme hızı değişen geriye yayılım ve esnek geriye yayılımdır. İkinci kategorideki algoritmalar ise standart sayısal optimizasyon yöntemini kullanırlar. Bu öğrenme algoritmaları; eşlenik gradyen, Newton ve Levenberg–Marquardt algoritmalarıdır.

Hızlı öğrenme algoritmaları, daha önce geliştirilmiş algoritmaların iyi yönlerini alıp, kötü yönlerinin iyileştirilmesi veya azaltılması ile ortaya çıkmaktadırlar. Örneğin, Levenberg–Marquardt algoritması, gradyen azalması ve Newton öğrenme algoritmalarının en iyi özelliklerini içinde barındırır, bazı kısıtlamaları ise ortadan kaldırır ve yavaş yakınsama problemlerinden etkilenmez.

Newton öğrenme yöntemleri, eşlenik gradyen öğrenme algoritmasının geliştirilmiş bir halidir. Gradyen birinci mertebe türevden oluşur. Newton öğrenmesi, Hessian matrisine dayanır. Bu matris, performans fonksiyonunun ağırlıklara göre ikinci dereceden türevlerini içerir. Hessian matrisi, ağırlık uzayının farklı doğrultularındaki gradyen değişimini ifade eder. Aşağıdaki formül ile değeri hesaplanır:

$$H(k) = \frac{\partial^2 E(k)}{\partial w^2(k-1)}, \quad (2.25)$$

burada H Hessian matrisini, E performans fonksiyonunu, w ise ağırlığın ağırlığını göstermektedir. Bu algorithma, performans fonksiyonu genellikle ortalama karesel hata olarak kullanılmaktadır:

$$E_{ort} = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{2} \sum_{k=1}^m E_k^2 \right), \quad (2.26)$$

burada N eğitim setinde bulunan toplam örnek sayısını, m ağırlık çıkış katmanı işlem elemanı sayısını, E_k ise k . çıktı değeri için hata miktarını temsil etmektedir. Daha önce de belirtildiği üzere, hata miktarı istenen değer y_k ile ağırlık çıkışı C_k arasındaki farktır:

$$E_k = y_k - C_k. \quad (2.27)$$

Ağırlıklar, Hessian matrisi hesaplandıktan sonra, tersi bulunarak yenilenebilir. Ancak bu adım, ileri beslemeli çok katmanlı YSA için hesaplanması zor ve karmaşık matris işlemleridir. Bunun yerine, Quasi-Newton yöntemi (Davidon, 1959) olarak adlandırılan daha basit bir model ile öğrenme gerçekleştirilir. Bu yöntemde, ikinci dereceden türev hesabı gerektirmeden algoritmanın her bir iterasyonunda Hessian matrisinin yaklaşık değeri kullanılır.

Levenberg-Marquardt algoritması (Levenberg, 1944; Marquardt, 1963) da Hessian matrisinin yaklaşık değerini kullanan bir algoritmadır, yaklaşık değer aşağıdaki formüldeki gibi hesaplanır:

$$H(k) = J^T(k)J(k) + \mu I, \quad (2.28)$$

burada J Jacobian matrisi, μ Marquardt parametresi, I ise birim matristir. Jacobian matrisi ağırlık hatalarının ağırlıklara göre birinci dereceden türevinden oluşmaktadır:

$$J(k) = \frac{\partial E(k)}{\partial w(k-1)}. \quad (2.29)$$

Jacobian matrisi, Hessian matrisinden daha kolay hesaplandığı için tercih edilmektedir. Ağıın gradyeni ise

$$g(k) = J^T(k)E(k) \quad (2.30)$$

olur ve ağırlıklar şu şekilde değıştirilir:

$$w_k(t) = w_k(t-1) - [H(k)]^{-1} g(k). \quad (2.31)$$

Marquardt parametresi μ ise küçük sabit bir sayıdır. Levenberg–Marquardt algoritması, Hessian matrisine eklenmiş küçük bir μ sabiti içerir. Yani $\mu = 0$ olarak alınır, yaklaşık Hessian matrisini kullanan Quasi–Newton algoritması ile aynı olurlar. Eğer μ 'nın değeri büyük bir sayı olarak alınır, yöntem küçük adımlı gradyen azalması haline dönüşür. Levenberg–Marquardt algoritmasının her iterasyonunda performans fonksiyonunun değeri sürekli azalır ve daima daha iyi öğrenme sağlanmış olur (Ngia, 2000; Oğuz, 2001; Bolat ve Kalenderli, 2003).

Çok katmanlı YSA'nın eğitilmesi ayrıca şu adımları içermektedir (Öztemel, 2003):

Örneklerin toplanması: Ağıın hem eğitilmesi hem de test edilmesi için daha önce gerçekleşmiş örneklerin toplanması gereklidir. Genellikle örneklerin %60 ile %80 arasında bir miktarı ağa eğitim seti olarak gösterilir ve ağ eğitilir. Daha sonra, kalan miktar (test seti) ile de ağıın performansı ölçülür. Ağıın asıl başarısını, hiç görmediğı örnekler karşısındaki davranışı belirler.

Verilerin ölçeklendirilmesi: YSA çıkışları, aktivasyon fonksiyonu sigmoid ise $[0 \ 1]$, tanjant-sigmoid ise $[-1 \ 1]$ arasında olmaktadır. Bu nedenle, giriş çıkış verilerinin ağıın eğitiminden önce böyle bir aralıkta kalacak biçimde ölçeklendirilmesi gerekir.

Giriş ve çıkış örnekleri teker teker, o işlem elemanının en büyük değerine bölünebilir ve böylece bu işlemin sonunda 0 ile 1 ya da -1 ile 1 arasında değerler alırlar.

Topolojik yapının belirlenmesi: Ağın girdi elemanı sayısı, gizli katman sayısı, her bir gizli katmandaki işlem elemanı sayısı ve çıktı elemanı sayısı bu aşamada belirlenir.

Parametrelerin belirlenmesi: Ağın öğrenme ve momentum katsayısı ile aktivasyon fonksiyonları bu aşamada belirlenir.

Ağırlıkların başlangıç değerlerinin atanması: Ağırlıklar ve bias değerleri başlangıçta genellikle rastgele değerlerle atanırlar. Başlangıç değerleri de ağın performansını etkileyebilmektedir. Bazen atanan o rastgele değerler öğrenmeyi güçleştirmekte hatta başarısızlığa sebep olabilmektedir. Her olumsuz eğitim de başlangıç değerleri ile ilgili değildir.

Örneklerin ağa sunulması ve ağırlıkların güncellenmesi: Eğitim setindeki örneklerin ağa gösterilme şekli de öğrenme performansını etkileyebilir. Ağa örnekler sıralı ya da rastgele düzeneğe göre verilebilir. Sunulan girdi için ileri hesaplama yapılarak ağın çıktı değeri hesaplanır. Gerçekleşen çıktı ile beklenen çıktı karşılaştırılır ve ağın ürettiği hata miktarı bulunur. Geri hesaplama yapılarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır. Ağın öğrenmesi tamamlanıncaya kadar örnekler ağa tekrar tekrar sunulur. Ağın gereğinden fazla eğitilmemesi de önemlidir. Ağın öğrenmesi için bir durdurma kriterinin olması gerekmektedir. Bu ise belli bir iterasyon sayısı veya test seti için üretilen hatanın kabul edilebilir bir düzeye inmesi olarak alınmaktadır.

Çok katmanlı YSA'nda, problemin çözümü için gereken ara katman sayısını ve ara katmanlardaki işlem elemanı sayısını belirleyen net bir yöntem bulunmamaktadır. En iyi topoloji deneme yanılma yöntemi ile bulunur. Bazen eksik sayıda bazen de fazla sayıda katman veya işlem elemanı kullanılır. Ağın büyütülmesi ya da budanması ile optimum çözüme ulaşılır (Öztemel, 2003).

2.4.2. Bulanık mantık

Bulanık mantık, ilk olarak 1965 yılında Lotfi A. Zadeh'in yayınladığı “bulanık kümeler” isimli makalesi ile ortaya çıktı. İnsanın ifade ve düşünme biçimini modellemeye çalışan bir yapay zekâ metodolojisidir. Konuya özellikle matematik ile ilgilenen bazı bilim adamları başlangıçta olumlu yaklaşmadılar. Zadeh çalışmalarına devam etti ve 1970’li yılların başlarında bulanık mantığın ve bulanık kontrolün temelini oluşturan yeni makaleler yayınladı. Mamdani ve Assilian 1975 yılında buhar kazanının kontrolünü, Holmblad ve Ostergaard 1978 yılında çimento üretiminde kullanılan değirmenin kontrolünü bulanık denetleyici ile gerçekleştirdi. Böylece, matematiksel modeli bilinmeyen sistemlerde bulanık mantık yönteminden dilsel değişkenlerin yardımıyla faydalanılabileceği gösterilmiş oldu. 1979’da Tsukamoto ve 1985’te Sugeno bulanık çıkarım yöntemleri geliştirildi. Çalışmalar kontrol üzerine yoğunlaştı. Yasunobu ve Miyomoto 1987 yılında Sendai metrosunda otomatik tren denetimini gerçekleştirdiler ve %10 enerji tasarrufu sağladılar. Yamachi Securities firması borsada bulanık mantık denetimi kullanarak 1988 yılındaki “kara pazar” olarak anılan krizi 18 gün öncesinde fark etti. 1989 yılında Japonya’da “Laboratory for International Fuzzy Engineering” laboratuvarı kuruldu. 1992 yılında bulanık mantık konulu ilk uluslararası IEEE konferansı San Diego’da düzenlendi ve 1993’te “IEEE Transactions on Fuzzy Systems” adlı dergi yayınlanmaya başlandı. 1980’li ve 1990’lı yıllarda bulanık kontrol trafik kavşağı, çamaşır makinası, robot kolu, klima, araba hareketi ve hızı, enerji gibi birçok sistemde başarıyla kullanıldı. Günümüzde, bulanık mantık her alanda uygulanabilen bir yapay zekâ tekniği haline gelmiştir.

Bulanıklık, kesinlik içermeyen bilgiyi yani belirsizliği ifade etmektedir. Bulanık mantık ise, dilsel niteleyiciler kullanarak kesinlikler yerine belirsizlikler ile çalışan bir yapay zekâ tekniğidir. Dilsel bir niteleyicinin değeri örneğin “soğuk”, “ılık” veya “sıcak” gibi sözcükler ile ifade edilir. Bu sayede karmaşık bir sistem, sözel değişkenler ile tanımlanabilir ve yönetilebilir. Bulanık mantığın klasik matematiksel yöntemlere göre avantajı, sistemin matematiksel modeline ihtiyaç duymaması, belirsizlikleri matematiksel olarak ifade edilebilmesi, kesinliklerle çalışmaması, çözülmesi güç problemleri genel bir yapıya kavuşturarak çözebilmesidir (Koç, 2002).

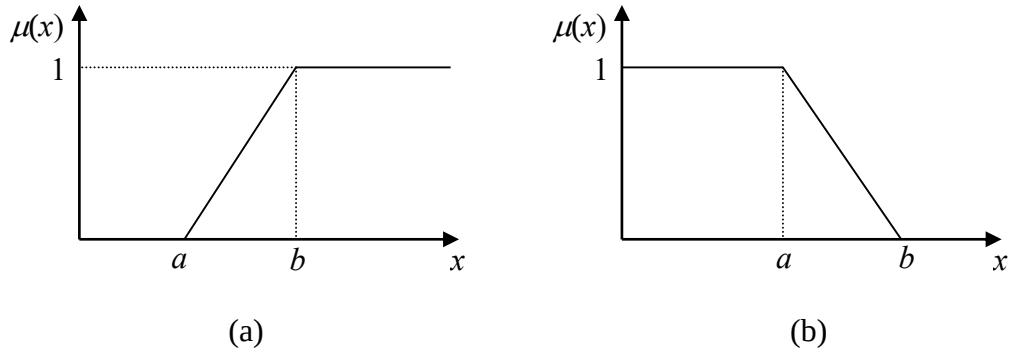
Bulanık mantığın temeli bulanık küme teorisine dayanmaktadır. Klasik kümelerde, bir varlık bir kümenin ya elemanıdır ya da değildir. Varlık, üyelik ilişkisi bakımından kümenin elemanı olduğu zaman “1”, olmadığında ise “0” değerine sahip olur. Bulanık küme teorisi, klasik küme teorisinin daha genel bir matematiksel yaklaşımıdır. Bulanık varlık kümesinde her bir eleman $[0, 1]$ aralığında bir üyelik değerine sahip olur. Yani, bir kümeye tam üye olma ile o kümeye üye olmama durumları arasında dereceli geçişe imkân sağlar. Diğer bir bakış açısıyla, kısmi üyeliğe olanak tanır. Bu nedenle, bulanık küme teorisi klasik küme teorisinin daha genişletilmiş bir hali olarak düşünülebilir (Temurtaş, 2000).

2.4.2.1. Üyelik fonksiyonu ve türleri

Bulanık mantık yönteminde, bir elemanın bir kümeye olan üyelik derecesi, özel olarak tanımlı üyelik fonksiyonu ile belirlenir. Klasik kümelerde elemanların üyelik dereceleri sadece 0 ya da 1 iken, bulanık kümelerde ise 0 ile 1 arasında sonsuz sayıda değişebilir. Klasik kümede; 0 üye olmamayı, 1 üye olmayı temsil eder. Bulanık kümede ise; 0 tam üye olmamayı, 0-1 üye olma derecesini, 0’a yakın değerler zayıf üye olmayı, 1 tam üye olmayı gösterir. Keskin kümelerdeki soğuk-sıcak, yakın-uzak, yavaş-hızlı, küçük-büyük, kısa-uzun, hafif-ağır, genç-yaşlı, ucuz-pahalı gibi ikili (var-yok) değişkenler, bulanık mantıkta “çok soğuk”, “soğuk”, “biraz soğuk”, “biraz sıcak”, “sıcak”, “çok sıcak” gibi dilsel niteleyicilerle daha gerçekçi biçimde esnetilebilir ve geçiş yumuşatılabilir (https://tr.wikipedia.org/wiki/Bulanık_mantık, 2015).

Üyelik fonksiyonları (membership functions) şekil ve matematiksel olarak değişik biçimlerde alınabilir. Sık kullanılanları: çizgi, üçgen, yamuk ve çan biçimli üyelik fonksiyonlarıdır.

Çizgisel olarak artan ve azalan üyelik fonksiyonları Şekil 2.23.’te gösterilmiştir. Bu fonksiyonların matematiksel ifadeleri sırasıyla Denklemler 2.32 ve 2.33’teki gibi tanımlanır.

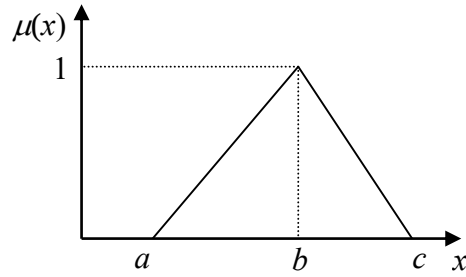


Şekil 2.23. Çizgi üyelik fonksiyonları (a) çizgisel artan fonksiyon, (b) çizgisel azalan fonksiyon.

$$f(x, a, b) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x. \end{cases} \quad (2.32)$$

$$f(x, a, b) = \begin{cases} 1, & x \leq a, \\ \frac{b-x}{b-a}, & a \leq x \leq b, \\ 0, & b \leq x. \end{cases} \quad (2.33)$$

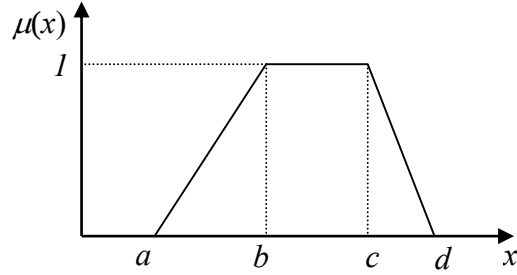
Şekil 2.24.'te üçgen üyelik fonksiyonu gösterilmiştir. Bu fonksiyonun matematiksel ifadesi ise Denklem 2.34'te verilmiştir.



Şekil 2.24. Üçgen üyelik fonksiyonu.

$$f(x, a, b, c) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ \frac{c-x}{c-b}, & b \leq x \leq c, \\ 1, & c \leq x. \end{cases} \quad (2.34)$$

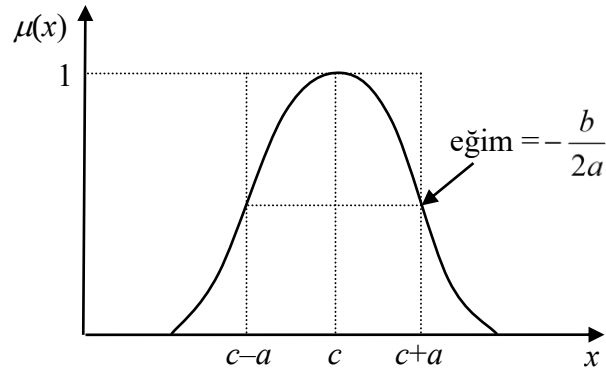
Yamuk üyelik fonksiyonu Şekil 2.25.'te, matematiksel ifadesi ise Denklem 2.35'te verilmiştir.



Şekil 2.25. Yamuk üyelik fonksiyonu.

$$f(x, a, b, c, d) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ \frac{d-x}{d-c}, & c \leq x \leq d, \\ 0, & d \leq x. \end{cases} \quad (2.35)$$

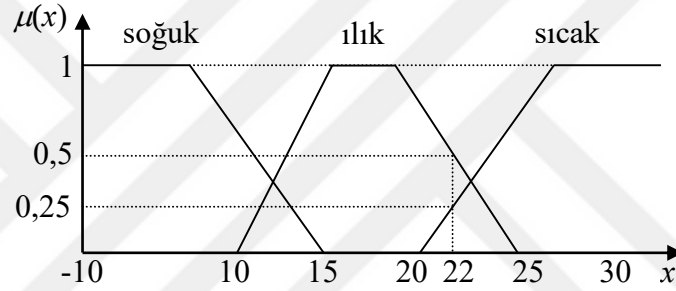
Çan biçimli üyelik fonksiyonu Şekil 2.26.'da gösterilmiştir. Bu fonksiyon Denklem 2.36'da verilen matematiksel ifade ile tanımlanmaktadır.



Şekil 2.26. Çan biçimli üyelik fonksiyonu.

$$f(x, a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}. \quad (2.36)$$

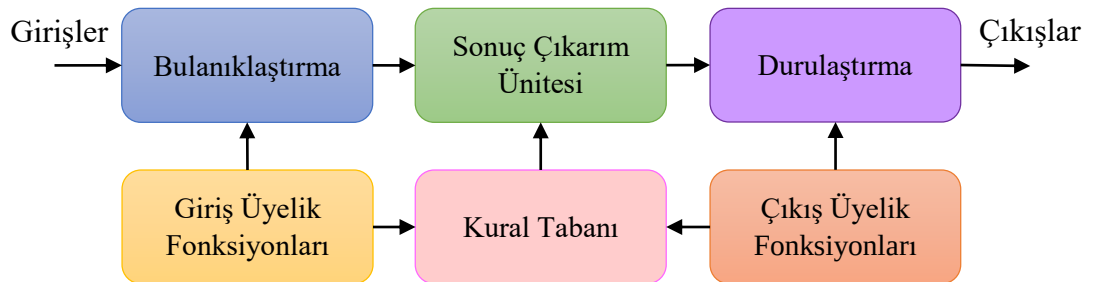
Bir x noktasının A bulanık kümesindeki üyelik derecesi $\mu_A(x)$ ile gösterilirse, $\mu_A(x) = 1$ konumu x 'in A bulanık kümesine ait tam bir elemanı olduğunu, $\mu_A(x) = 0$ konumu x 'in A bulanık kümesinin dışında olduğunu, $0 < \mu_A(x) < 1$ arasındaki konumu ise x 'in A bulanık kümesinin belirsiz bir ara değerinde olduğunu ifade eder. Örneğin bir ortamın sıcaklığı “soğuk”, “ılık”, “sıcak” isimli çizgi ve yamuk üyelik fonksiyonları ile Şekil 2.27.’deki gibi tanımlanabilir. Eğer ortamın sıcaklığı -5°C ise sıcaklık sadece “soğuk” bulanık kümesine girecek ve 1 oranında üye olacak, “ılık” ve “sıcak” bulanık kümelerine girmeyecek ve 0 oranında üye olacaktır. Ortamın sıcaklığı 22°C ise sıcaklık, “ılık” bulanık kümesine 0,5 oranında, aynı zamanda “sıcak” bulanık kümesine 0,25 oranında üye olacaktır.



Şekil 2.27. Ortamın sıcaklığını gösteren üyelik fonksiyonları.

2.4.2.2. Bulanık mantık tabanlı sistemlerin temel yapısı

Bulanık mantığa dayalı bir sistem Şekil 2.28.’de gösterildiği gibi genellikle bulanıklaştırma, kural tabanı, sonuç çıkarım ünitesi ve durulaştırma birimlerinden oluşur. Giriş ve çıkış üyelik fonksiyonları bunlara eşlik eder.



Şekil 2.28. Bulanık mantık tabanlı bir sistemin genel yapısı.

Bulanıklaştırma, bilginin bulanık sisteme ilk girişinin yapıldığı birimdir. Burada, gerçek sayısal değerler ölçek değişikliği ile dilsel değerlere dönüştürülür. Bu amaç doğrultusunda, bulanık kümeler ve giriş üyelik fonksiyonları kullanılır. Sonrasında sonuç çıkarım birimine gelen bilgiler, bulanık kurallardan oluşan kural tabanı ile işlenir. Kural tabanı, bulanık mantık tabanlı sistemlerin davranışını belirleyen bulanık “If – Then (Eğer – Öyleyse)” kurallarını içerir. Genellikle uzman kişilerin problemin yapısına ve sisteme ait bilgi ve tecrübesi ile oluşturulur. Son adımda ise elde edilen sonuçlar durulaştırma birimine gönderilir. Durulaştırıcı, bulanık çıkarım ünitesi ile elde edilen bulanık sonucu sayısal değere dönüştürür. Çıkış üyelik fonksiyonları kullanılarak yine ölçek değişikliği ile sistemin çıkış değerleri hesaplanır.

Bulanıklaştırma (fuzzification): Bulanık sistemlerde, sözel giriş-çıkış değişkenleri yeterli temsili sağlayacak kadar seviyelendirilerek bulanık alt kümelere ayrılır. Her bir alt küme bir üyelik fonksiyonu ile temsil edilir. Sayısal giriş değerlerinin üyelik fonksiyonları aracılığıyla bulanık değerlere dönüştürülmesi bulanıklaştırma işlemi ile yapılır.

Örneğin, elemanları x olan evrensel bir X kümesinin, A bulanık alt kümesine aitliği üyelik fonksiyonları ile belirlenir ve $\mu_A(x)$ üyelik fonksiyonu için $\mu_A(x): X \rightarrow [0, 1]$ aralığındaki kısmi üyelik derecesi hesaplanır. Üyelik fonksiyonunun değeri, x için A bulanık alt kümesine aitlik derecesini verir (Gülbağ, 2006).

Kural tabanı (rule base): Bulanık sistemler, çoğunlukla çok sayıda kuralı içinde barındıran bir bulanık kural tabanına sahiptir. Kurallar uzmanlık bilgilerini sebep-sonuç ilişkisi biçiminde temsil ederler. A ve B , sırasıyla E ve F evrensel kümelerinde bulanık kümeler ve sözel değişkenler $x \in E$ ve $y \in F$ olmak üzere, bir bulanık kural genellikle şu şekilde tanımlanır:

Eğer x A ise, o halde y B' dir. (2.37)

Bulanık bir kuralın eğer kısmında yer alan sözel değişken sayısı birden fazla ise, değişkenlere ait ifadeleri VE ya da VEYA operatörleri ile birbirine bağlamak mümkündür.

Çıkarım ünitesi (inference engine): Bu birimde, girişlerin aitlik derecelerine göre kural tabanındaki uygun kurallar belirlenir ve işlenir. Parça parça olan ilişkiler bir araya getirilerek sistemin girdiler altında nasıl bir çıktı vereceğini bulunur. Giriş üyelik değeri birden fazla kural kümesine karşılık gelirse, bu durumda kesişim, birleşim, tümleme gibi bulanık küme operatörleri ile işlemler yapılır.

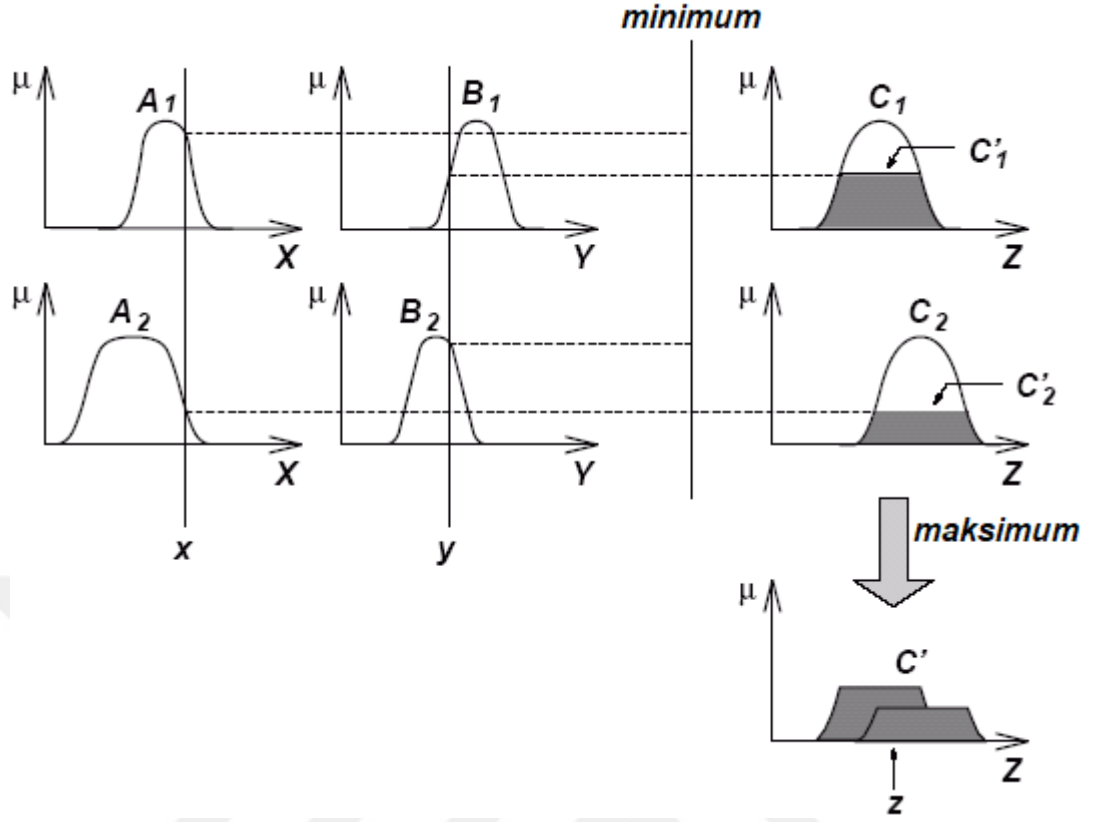
A ve B iki bulanık küme, μ_A ve μ_B sırasıyla A ve B 'nin üyelik fonksiyonları olmak üzere bulanık kümeler için birleşim işlemi $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$, kesişim işlemi $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$ ve tümleme işlemi $\mu_{A^c}(x) = 1 - \mu_A(x)$ biçiminde tanımlanır.

Mamdani ve Sugeno literatürde en çok tercih edilen bulanık çıkarım yöntemleridir.

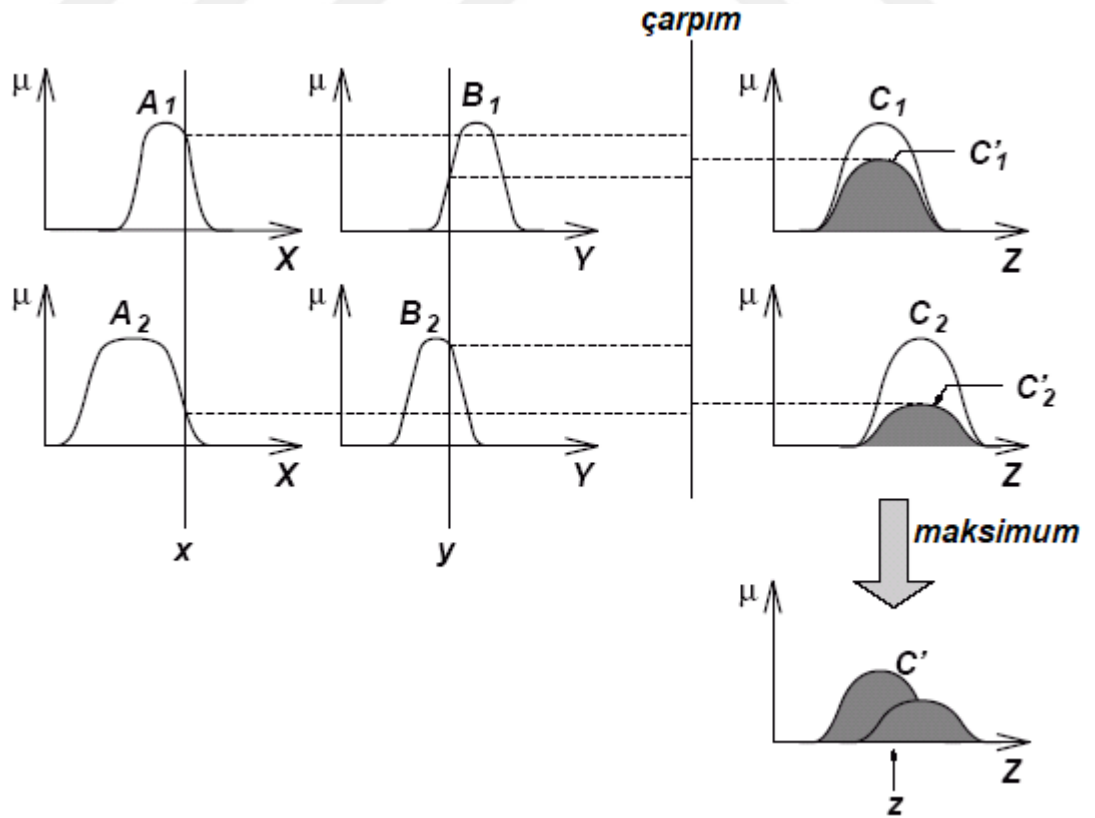
Mamdani bulanık çıkarım yönteminde; A_i , B_j ve C_k ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$; $k = 1, 2, \dots, q$), sırasıyla E , F ve G evrensel kümelerinde bulanık kümeler ve sözel değişkenler $x \in E$, $y \in F$ ve $z \in G$ olmak üzere 2 adet bulanık kural aşağıdaki gibi ifade edilebilir:

Eğer $x A_1$ ve $y B_1$ ise $z C_1$,
Eğer $x A_2$ ve $y B_2$ ise $z C_2$ 'dir. (2.38)

Sonrasında, x , y ve z sözel değişkenleri arasındaki ilişki, maksimizasyon, minimizasyon ve çarpım işlemleri ile bulunur. Sırasıyla Şekil 2.29.'da ve Şekil 2.30'da gösterildiği gibi kuralların koşul kısmında minimum operatörü ya da çarpım operatörü kullanılabilir. Kuralları birleştirmek için ise maksimum operatörü kullanılır.



Şekil 2.29. Mamdani maksimum-minimum bulanık çıkarım yöntemi.



Şekil 2.30. Maksimum-çarpım bulanık çıkarım yöntemi.

Temel bulanık çıkarım sisteminde, girişler bulanık veya kesin değerler olarak alınabilir fakat çıkışlar daima bulanık kümelerdir. Bazen, özellikle de bulanık çıkarım sisteminden denetleyici olarak faydalandığı durumlarda, çıkışların net sayısal değerlerine gereksinim duyulur. Mamdani bulanık çıkarım yönteminde kesin bir değer elde etmek için bir durulaştırma yöntemine ihtiyaç vardır.

Sugeno bulanık çıkarım yöntemi, Mamdani modelinin bir uyarlamasıdır. Takagi, Sugeno ve Kang tarafından önerilmiştir (Takagi ve Sugeno, 1985; Sugeno ve Kang, 1986). Girdi değerlerinin bulanıklaştırılması ve bulanık mantık işlemleri Mamdani bulanık çıkarım yöntemi ile birebir aynıdır. Aralarındaki fark, çıkış için bulanık üyelik fonksiyonları yerine kesin değerli polinomların kullanılmasıdır. Sugeno bulanık çıkarım yönteminde kullanılan bulanık kural yapısı şu şekilde ifade edilir:

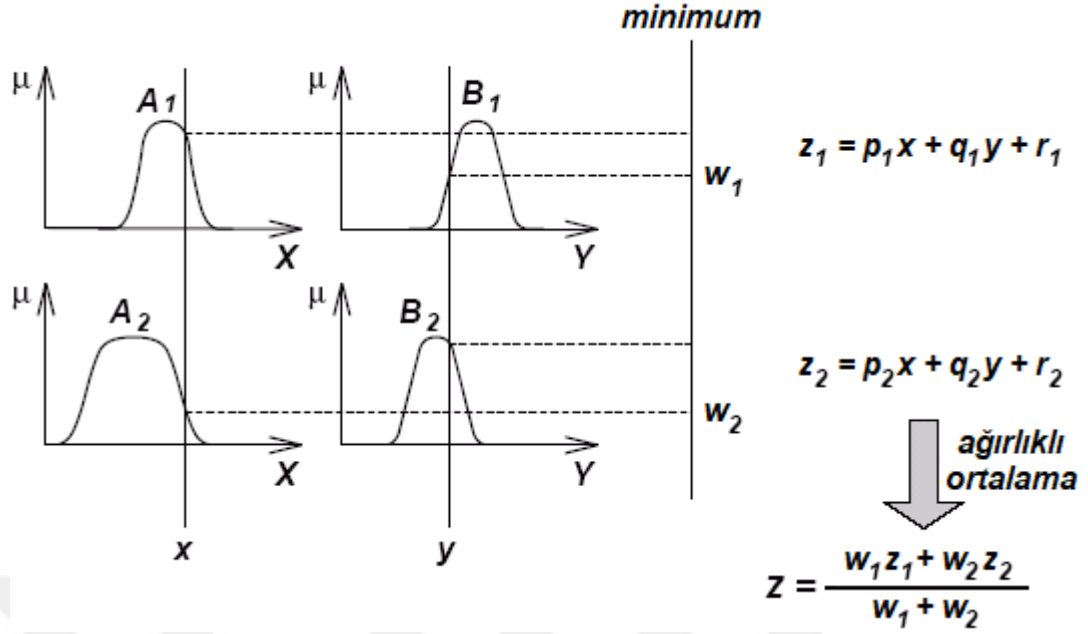
$$\text{Eğer } x \text{ } A \text{ ve } y \text{ } B \text{ ise } z = f(x, y), \quad (2.39)$$

burada koşul kısmındaki x ve y bulanık giriş değişkenlerinin değerlerini, A ve B giriş bulanık kümelerini, sonuç kısmındaki $z = f(x, y)$ kesin çıkış değerini temsil etmektedir. $f(x, y)$ ise giriş değerleri x ve y olan bir polinomdur. $f(x, y) = c$ gibi sabit bir değer olarak alınırsa “sıfırıncı dereceden Sugeno bulanık modeli”, $f(x, y) = ax + by + c$ gibi x ve y ’nin doğrusal bir fonksiyonu olarak alınırsa “birinci dereceden Sugeno bulanık modeli” biçiminde adlandırılır.

Birinci dereceden Sugeno bulanık modeli için 2 adet bulanık “Eğer–İse” kuralı bulunan tipik kural kümesi aşağıdaki biçimde ifade edilebilir:

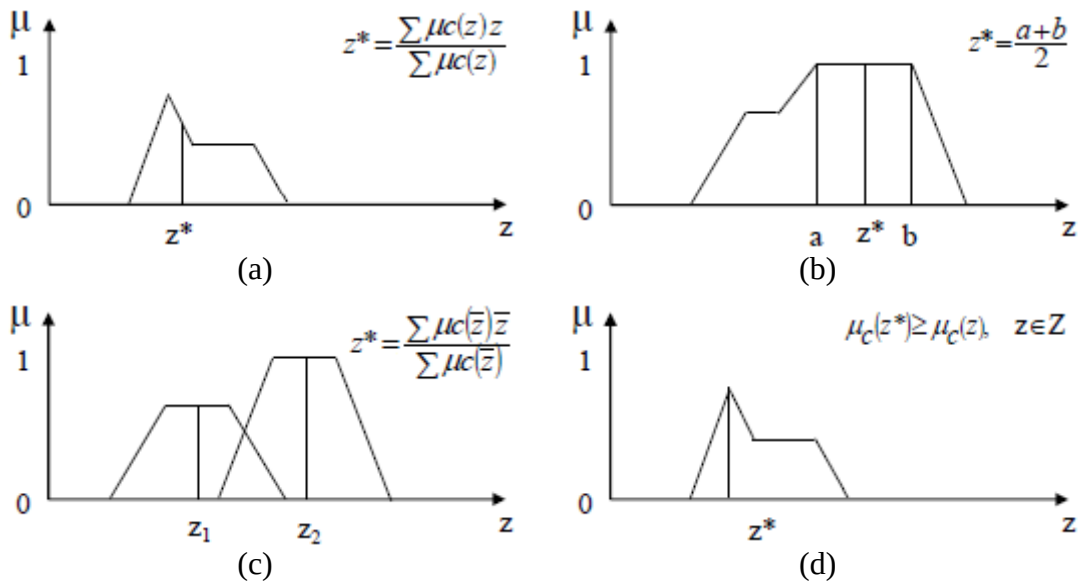
$$\begin{aligned} \text{Eğer } x \text{ } A_1 \text{ ve } y \text{ } B_1 \text{ ise } z_1 &= p_1x + q_1y + r_1, \\ \text{Eğer } x \text{ } A_2 \text{ ve } y \text{ } B_2 \text{ ise } z_2 &= p_2x + q_2y + r_2. \end{aligned} \quad (2.40)$$

Şekil 2.31.’de birinci dereceden Sugeno bulanık modeli için çıkarım işlem basamakları 2 kurala dayalı olarak gösterilmiştir. Çıkış değeri, ağırlıklı ortalama işlemi ile hesaplanır. Bu yöntemde sonuç, net sayısal değer olduğu için durulaştırma işlemine gereksinim duyulmaz.



Şekil 2.31. Sugeno bulanık çıkarım yöntemi.

Durulaştırma (defuzzification): Kullanılan bulanık çıkarım yöntemine göre işlem sonucunda istenilen bilginin bulanık hali ortaya çıkabilir. Pratik uygulamalar, sonuçların kesin değerlerde olmasını ister. Bulanık mantıkta sayısal değerlere dönüştürülme süreci durulaştırma ya da diğer adıyla netleştirme aşamasında olur. Bu aşamada, çıkarım sonucu elde edilen bulanık değerler tekrardan kesin değerlere dönüştürülür (<http://akabamuhendislik.com>, 2017).



Şekil 2.32. Durulaştırma yöntemleri (a) ağırlık merkezi, (b) en büyüklerin ortası, (c) ağırlıklı ortalama, (d) maksimum üyelik.

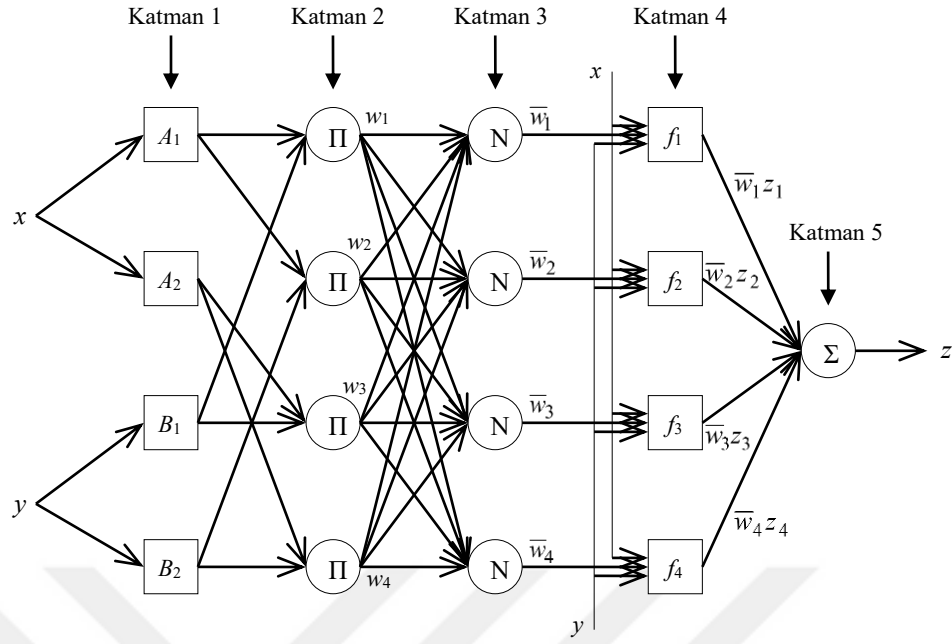
Literatürde birçok durulaştırma yöntemi kullanılmaktadır. Bunlardan en çok tercih edilenleri Şekil 2.32.'de gösterilen ağırlık merkezi yöntemi, en büyüklerin ortası yöntemi, ağırlıklı ortalama yöntemi ve maksimum üyelik yöntemidir. Ancak, en iyi durulaştırma yönteminin seçilmesi için sistematik bir yol yoktur, uygulamanın özellikleri dikkate alınarak yöntem belirlenir (Baykal ve Beyan, 2004).

2.4.3. Sinirsel-bulanık ağlar (ANFIS)

ANFIS, “Adaptive Neuro-Fuzzy Inference System” isminin baş harflerinden oluşan bir kısaltmadır. Literatürde uyarlamalı sinirsel-bulanık çıkarım sistemi, adaptif sinirsel-bulanık çıkarım sistemi veya kısaca sinirsel-bulanık ağlar gibi isimlerle kullanılmaktadır. Jang tarafından 1993 yılında önerilmiştir. ANFIS, Sugeno bulanık modelinin YSA öğrenme ilkeleriyle birlikte kullanılan uyarlamalı bir ağ yapısıdır. Hem YSA’ndan hem de bulanık mantıktan faydalanan melez öğrenme sistemidir. Önerildiği 1990’lı yıllarda, doğrusal olmayan sistemlerin modellenmesinde ve kaotik zaman serilerinin tahmininde başarıyla kullanılmıştır (https://tr.wikipedia.org/wiki/Uyarlamalı_ağ_tabanlı_bulanık_çıkartım_sistemi, 2015).

Bulanık sistemlerin veri seti aracılığıyla öğrenme kabiliyeti yoktur. Sinirsel-bulanık ağlarda ise, ağa ait parametreler güncellenir; bulanık üyelik fonksiyonu değerlerinin ve kural tabanının optimize edilmesi sağlanır. Optimizasyon işlemi YSA’nın geri yayılım algoritması ya da en küçük kareler yöntemi ile geri yayılım algoritmasının kombinasyonu ile yapılır. Böylece, Sugeno bulanık mantık modeline giriş ve çıkış değer örnekleri gösterilerek öğrenme yeteneği kazandırılmış olunur. YSA’nın içerisinde “Eğer-İse” yapısı yer almaz. Sinirsel-bulanık ağlar, bu dezavantajı içermeyen bir makine öğrenmesi sistemidir.

Sinirsel-bulanık ağlar, 5 katman halinde yerleştirilmiş düğümlerin birleşiminden oluşur. Her bir katman belli bir fonksiyonu gerçekleştirir (Tsoukalas ve Uhrig, 1996). Şekil 2.33.’te 2 girdiye ve 1 çıktıya karşılık gelen ANFIS yapısı verilmiştir.



Şekil 2.33. ANFIS mimarisi.

Katman 1: Bu katmanın girdileri sistemin girdileridir. Bu katmandaki her bir i düğümü, girdiler için tanımlanan üyelik fonksiyonlarıdır. Bu katmanın çıktıları ise üyelik dereceleridir. Üyelik fonksiyonları aşağıdaki gibi tanımlanabilir:

$$\mu_{A_i}(x), \quad i = 1, 2 \quad (2.41)$$

ya da

$$\mu_{B_{i-2}}(y), \quad i = 3, 4 \quad (2.42)$$

ki burada x ve y giriş değişkenlerini, A_i ve B_{i-2} ise değişkenlere ait bulanık kümeleri temsil etmektedir.

Katman 2: Bu katmanda yer alan düğümler Π ile etiketlenmiştir. Katmandaki i . düğüm, kendisine gelen değerlerin çarpımını çıkış olarak verir:

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y). \quad (2.43)$$

Katman 3: Bu katmanda yer alan düğümler N ile etiketlenmiştir. Bu katmanda her bir düğüme ait normalize edilmiş gerçekleştirme dereceleri bulunur. Katmanın i . düğümdeki gerçekleştirme derecesi, o kuralın gerçekleştirme seviyesinin tüm kuralların gerçekleştirme seviyeleri toplamına bölümü ile hesaplanır:

$$\bar{w}_i = \frac{w_i}{\sum_i w_i}. \quad (2.44)$$

Katman 4: Bu katmandaki her bir i düğümü, bir önceki katmandan gelen $\bar{w}_i$ değeri ile $z_i = p_i x + q_i y + r_i$ polinomunun çarpımını çıkış olarak verir:

$$f_i = \bar{w}_i (p_i x + q_i y + r_i). \quad (2.45)$$

Sıfırıncı dereceden Sugeno bulanık modelindeki gibi polinom sabit bir sayı olarak alınırsa,

$$f_i = \bar{w}_i r_i \quad (2.46)$$

olur, burada p_i , q_i ve r_i sonuç ya da çıkış parametreleri olarak ifade edilen parametre kümesidir.

Katman 5: Bu katmanda sinirsel-bulanık ağın çıkış değeri olan z , kendisine gelen tüm değerlerin toplamı ile hesaplanır:

$$z = \sum_i f_i. \quad (2.47)$$

Sinirsel-bulanık ağlarda sonuç, Sugeno çıkarım yönteminde olduğu gibi, net sayısal değer olduğu için ayrıca durulaştırma işlemi yapılmaz.

Böylelikle, tam olarak Sugeno bulanık modelinin işleyişine sahip, uyarlamalı bir ağ mimarisi yapılandırılmıştır (Özçalık ve Uygur, 2003).

Sinirsel-bulanık ağılara ait parametrelerin güncellenmesinde genellikle en küçük kareler yöntemi ile geri yayılım algoritmasının birleşimi olan hibrit öğrenme algoritması kullanılır.

2.5. Kaotik Sistemlerin Yapay Zekâ ile Kontrolü

Bir kontrol sisteminde, çıkışı istenilen değerde tutmak için sisteme uygulanacak girişi belirlemek gerekir. Hata, istenilen çıkış (referans) ile sistem çıkışı arasındaki farktır. Sistemin çıkışı ile referans (sistemin denge noktası) arasındaki hatayı minimize edebilecek giriş sinyali üretmek için, hata miktarı geri-beslenerek makine öğrenmesi içeren YSA'ndan faydalanılabilir. YSA'nın doğrusal olmasa bile sistem davranışının tümünü öğrenebilmesi ve sahip olduğu gürültü toleransı kaotik kontrol uygulamaları için bir avantajdır. Bir diğer bakış açısıyla, kontrolörlerin çok iyi performans göstermesi için optimal noktalarda çalıştırılmaları istenir. Bu amaç doğrultusunda kontrolörlerin parametreleri YSA, genetik algoritmalar gibi zeki optimizasyon algoritmaları ile hesaplanabilmektedir.

YSA ile kaotik sistemlerin kontrolü konusunda birçok çalışma yapılmıştır: Alsing ve arkadaşları (1994) Henon haritaları kaotik sisteminin kontrolü için OGY formülünü YSA'na öğretmişlerdir. Otowara ve Fan (1995) kaosun kontrolünü YSA ile gerçekleştirmiş, kaotik lojistik harita üzerine uygulamıştır. Suykens ve Vandewalle (1996) Chua kaotik devresinin kontrolü için geri dönüşümlü YSA kullanmışlardır. Weeks ve Burgess (1997) Henon haritaları kaotik sisteminin kontrolünü YSA ile gerçekleştirmişlerdir. Poznyak ve arkadaşları (1999) Lorenz, Duffing ve Chua kaotik sistemlerinin tanımlanması ve kontrolü için dinamik YSA'nı önermişlerdir. Kiss ve Gaspar (2000) iki farklı YSA modeli ile kaosun kontrolünü önermiş, kaotik otokatalitik tepkimeli bir sistem üzerinde kontrolü gerçekleştirmişlerdir. Hernandez ve arkadaşları (2001) Lorenz kaotik sisteminin kontrolü için YSA kullanımını bildirmişlerdir. Ramesh ve Narayanan (2001) Bonhoeffer–van der Pol kaotik sistemi için OGY tabanlı YSA kontrolörü geliştirmişler. Ge ve Wang (2002) belirsiz kaotik sistemlerin kontrolü için adaptif YSA üzerine çalışmış, Brusselator kaotik sistemi üzerinde uygulamış, sonuçları bazı diğer yöntemlerle karşılaştırmışlardır. Sanchez ve

Ricalde (2003) Rössler ve Chua kaotik sistemlerin sabitleştirilmesi, senkronizasyonu, kontrolü için geri dönüşümlü YSA'nı kullanmışlardır. Boukabou ve Mansuri (2005) bilinmeyen kaotik sistemlerin tahmine dayalı kontrolünü YSA ile uygulamışlardır. Shen ve Wang (2007) Henon ve Lorenz kaotik sistemlerin kontrolü için tek katmanlı YSA'nın yeterli olduğunu göstermişlerdir. Kuntanapreeda (2008) Lorenz kaotik sistemi için Thau gözlemci tabanlı YSA kontrolörü ile gerçekleştirmiştir. Yang ve arkadaşları (2009) Rössler kaotik sistemini kontrol etmek için YSA kullanımını incelemişlerdir. Sadeghpour ve arkadaşları (2012) doğrusal geri-besleme tabanlı bir YSA kaos kontrolü için geliştirmişler, Bonhoeffer-van der Pol kaotik sistemi üzerinde uygulamışlardır. Tang, Gao ve Liu (2014) ayrık zamanlı kaotik sistemler için YSA tabanlı nerdeyse optimal kontrol yöntemi kullanmışlardır. Aguilar-Mejia, Tapia-Olvera, Valderrabano-Gonzalez ve Cambero (2016) kaotik daimi mıknatıslı senkron motorun kontrolünü uyarlamalı YSA ile sağlamışlardır. Gökçe ve Uyaroglu (2017) Ikeda haritası ayrık kaotik sisteminin kontrolünü uyarlamalı YSA ile gerçekleştirmişlerdir.

Kim ve arkadaşları (2000) kaotik sistemlerin kontrolünde radyal tabanlı YSA önermişler, Duffing ve Lorenz kaotik sistemleri üzerinde gerçekleştirmişlerdir. Guo ve arkadaşları (2006) Lorenz kaotik sistemi için radyal tabanlı YSA kayma kipli kontrolör kullanmışlardır. Ishikawa ve arkadaşları (2007) Duffing ve Lorenz kaotik sistemlerinin kontrolünü radyal tabanlı YSA ile uygulamışlardır. Yan ve Liu (2010) kesirli Lü kaotik sisteminin kontrolünü radyal tabanlı YSA'nı kayma kipli kontrolör ile uygulamışlardır. Gao, Dong ve Ning (2015), Duffing ve Genesio kaotik sistemlerinin kontrolünü radyal tabanlı YSA ile gerçekleştirmiştir. Li (2016), bilinmeyen kısım için radyal tabanlı YSA, bilinen kısım için uyarlamalı kontrol kullanarak kaotik sistemlerin kontrolünü önermiştir.

Funke, Herrmann ve Der (1997) kaotik sistemlerin kontrolünü Kohonen ağırları ile gerçekleştirmişlerdir. Li ve Wang (2003) formülü bilinmeyen kaotik sistemlerin kontrolünde kestirim tabanlı YSA'nı kullanmışlardır. Song ve arkadaşları (2007) ayrık kaotik sistemlerin kontrolünü, Shen ve arkadaşları (2008) Lorenz kaotik sisteminin kontrolünü kestirim tabanlı YSA ile gerçekleştirmişlerdir.

Wang ve arkadaşları (2010) bilgi işleme için zaman gecikmeli kaotik YSA'ndaki kaosu kontrolünü incelemişlerdir.

Bulanık mantık ile kontrol, ilk olarak 1970'lerde önerilmiş ve yapısal olarak modellenmesi zor sistemler için kontrolör tasarlamak amacıyla kullanılmıştır (Zadeh, 1973). Sonrasında, bulanık kontrol önemli bir araştırma ve çalışma alanı haline gelmiştir. Endüstriyel alanda pratik ve verimli çok sayıda uygulama, geleneksel kontrol teorisi kadar iyi sonuçlar ürettiğini göstermiştir (Lee, 1990; Cox, 1992). Bununla birlikte şu anda bulanık mantık ile kontrol için sistematik bir tasarım prosedürü bulunmamaktadır. Bulanık mantığın kullanılması kapalı çevrim sistemin kararlılığını garanti etmez. Bu nedenle, klasik kontrol teorilerinin bulanıklaştırılması ile uygulanmaktadır. Bulanık mantık, kontrol parametrelerini daha kolay değiştirme imkânı sağlar, kontrolörleri daha anlaşılır ve avantajlı kılar. Birçok çalışmada bulanık kontrolörlerin tasarlanması, başka bir kontrol yöntemi tasarımı yapıldıktan sonra gerçekleştirilmektedir (Glawer ve Munighan, 1997).

Bulanık mantık ile kaotik sistemlerin kontrolü konusunda bazı çalışmalar yapılmıştır. Chang ve arkadaşları (2008b) SMIB (Single-Machine Infinite-Bus – Tek-Makine Sonsuz-Bara) güç sisteminin kontrolünü bulanık mantık tabanlı kontrolör tasarlayarak uygulamışlardır. Tam ve Li (2015) Sprott kaotik sisteminin kontrolünde bulanık mantık kullanmışlardır.

Alasty ve Salarieh (2005) ise Bonhoeffer–van der Pol osilatörünün OGY ve Pyragas yöntemi ile kontrolünü bulanık mantıkla gerçekleştirmişlerdir. Wu ve Wang (2007) Lorenz kaotik sisteminin bulanık mantık tabanlı adaptif kayma kipli kontrolünü uygulamışlardır. Mehran (2011) ise yayınladığı bildiride kaotik Chua devresinin kontrolünde bulanık mantık tabanlı OGY kaos kontrolünü araştırmıştır. Li ve arkadaşları (2012) bulanık geri-beslemeli kontrol ve bulanık uyarlamalı kontrol ile Lorenz–Stenflo kaotik sisteminin kontrolünü gerçekleştirmişlerdir.

Zhao ve arkadaşları (2011) kaotik güç sisteminin kontrolünü, Sarailoo ve arkadaşları (2012) hiperkaotik Chen sisteminin kontrolünü bulanık mantık tabanlı kayma kipli

kontrolörler ile uygulamışlardır. Son yıllarda, kaotik sistemlerin kontrolünde kayma kipli kontrol ile bulanık mantığın beraber kullanımı çalışmaları artmıştır (Song ve Song, 2016a; Abadi ve Balochian, 2017; Zirkohi, 2017).

Li ve arkadaşları (2002) Lorenz ve Henon kaotik sistemlerinin geri-beslemeli kontrolünde Takagi–Sugeno bulanık mantık kullanmışlardır. Umoh (2013) Rabinovic kaotik sisteminin, Precup ve arkadaşları (2014) Rössler kaotik sisteminin, Song ve Song (2016b) Liu kaotik sisteminin kontrolünü Takagi–Sugeno bulanık mantık yöntemi ile gerçekleştirmişlerdir.

Sinirsel-bulanık ağlar (ANFIS) kaosun kontrolünde son yıllarda kullanılmaya başlanmıştır. Verilerden makine öğrenmesine dayalı bir yöntem olduğundan, kurgusu YSA ile kontrole benzemektedir. Bagheri ve Moghaddam (2009) Lorenz kaotik probleminin çözümü için ANFIS kayma kipli kontrolör kullanmışlardır. Chen ve arkadaşları Lorenz ve Genesio kaotik sistemlerinin senkronizasyonunu uyarlamalı sinirsel-bulanık çıkarım ile gerçekleştirmişlerdir (Chen, 2009; Chen ve Chen, 2009). Ginarsa ve arkadaşları (2009) güç sistemlerindeki kaosun kontrolünü ANFIS tabanlı tümleşik kontrolör ile uygulamışlardır. Yeh ve arkadaşları (2011) zaman gecikmeli damper kaotik sisteminin kontrolünde sinirsel-bulanık ağları kullanmışlardır. Tooranjipour ve Vatankhah (2018) kaotik sistemin kontrolünde uyarlamalı sinirsel-bulanık ağları önermişlerdir.

Diğer yapay zekâ teknikleri ile kaotik sistemlerin kontrolüne literatürde daha az rastlanılmaktadır. Liu ve Liu (2006) Lorenz kaotik sisteminin kontrolünü uyarlamalı ters kontrol tabanlı destek vektör makineleri ile gerçekleştirmişlerdir. Premalatha ve VanajaRanjan (2008) doğrusal olmayan elektrik devrelerindeki kaosun kontrolü için genetik algoritma kullanmışlardır.

BÖLÜM 3. KAOTİK SİSTEMLERİN KONTROLÜ

3.1. Chua Kaotik Sisteminin Kontrolü

Chua kaotik sisteminin kontrolü sırasıyla klasik kontrol yöntemlerinden kayma kipli kontrol ile, yapay zekâ tekniklerinden yapay sinir ağları ile ve her ikisinin birleşimi ile uygulanmış, bilgisayar simülasyonlarından elde edilen sonuçlar karşılaştırmalı grafiklerle sunulmuştur.

3.1.1. Kayma kipli kontrol

Bu bölümde kaotik Chua devresinin kayma kipli kontrolü gerçekleştirilmiştir. Denklem 2.2’de verilen Chua kaotik sistemine kayma kipli kontrolün uygulanması amacıyla x ve z durum değişkenlerine u_{KKK1} ve u_{KKK2} kontrol fonksiyonları eklenerek yeniden aşağıdaki biçimde yazılabilir:

$$\begin{cases} \dot{x} = \alpha(y - x - f(x)) + u_{KKK1}, \\ \dot{y} = x - y + z, \\ \dot{z} = -\beta y + u_{KKK2}. \end{cases} \quad (3.1)$$

Yönlendirme kontrolünün yapılacağı sabit denge noktası $(\bar{x}, \bar{y}, \bar{z})$ biçiminde gösterilirse, bu durumda yörüngelerdeki durum hataları $e_1 = x - \bar{x}$, $e_2 = y - \bar{y}$ ve $e_3 = z - \bar{z}$ olur. Bu sayede, durum değişkenleri $x = e_1 + \bar{x}$, $y = e_2 + \bar{y}$ ve $z = e_3 + \bar{z}$ olarak tanımlanabilir. Chua kaotik sistemi sadece $(0, 0, 0)$ denge noktasına sahip olduğundan $x = e_1$, $y = e_2$ ve $z = e_3$ olur. Denklem sistemi 3.1’in durum hata dinamik denklemleri aşağıdaki gibi elde edilir:

$$\begin{cases} \dot{e}_1 = \alpha(e_2 - e_1 - f(e_1)) + u_{KKK1}, \\ \dot{e}_2 = e_1 - e_2 + e_3, \\ \dot{e}_3 = -\beta e_2 + u_{KKK2}. \end{cases} \quad (3.2)$$

Denklem 3.2'deki e_1 ve e_3 hata dinamikleri sıfıra eşitlendiğinde, $\dot{e}_2 = -e_2$ olur. Bu, e_2 hata dinamiklerinin de sıfıra yakınsayacağı anlamına gelmektedir.

Sistem dinamiklerinin kararlı olması için bir kayma yüzeyi seçilmelidir. Sistemin kararlı olmasına imkân tanıyan bir kayma yüzeyini elde etmek için, sistem doğrusal forma dönüştürülür ve kayma yüzeyi katsayıları bu doğrusal form üzerinden hesaplanır.

Uygun kayma yüzeyleri aşağıdaki gibi alınabilir:

$$\begin{cases} s_1 = e_1 + k_1 e_2, \\ s_2 = e_3 + k_2 e_2, \end{cases} \quad (3.3)$$

burada k_1 ve k_2 pozitif sabit parametrelerdir.

Kayma kipli kontrolün başarıyla gerçekleşmesi için $s\dot{s} < 0$ şartını sağlamalıdır. Bu koşulu karşılamak için, kayma kipli kontrol fonksiyonları aşağıdaki gibi alınabilir:

$$\begin{cases} u_{KKK1} = -\alpha(e_2 - e_1 - f(e_1)) - k_1(e_1 - e_2 + e_3) - k_3 s_1 - k_4 \text{sign}(s_1), \\ u_{KKK2} = \beta e_2 - k_2(e_1 - e_2 + e_3) - k_3 s_2 - k_4 \text{sign}(s_2), \end{cases} \quad (3.4)$$

burada k_3 ve k_4 sabit parametrelerdir. k_3 değerinin büyük alınması, kayma yüzeyine erişim zamanını azaltır fakat çatırtıya yol açar, k_4 değerinin küçük alınması ise çatırtıyı azaltır ama kayma yüzeyine erişim süresi artar. Denklem 3.4'te geçen 'sign' ise işaret fonksiyonudur:

$$\text{sign}(s) = \begin{cases} 1, & s > 0, \\ 0, & s = 0, \\ -1, & s < 0. \end{cases} \quad (3.5)$$

Kayma kipli kontrol sinyallerinin ana amacı sistemin asimptotik kararlılığını sağlayabilmek için sistemi kayma yüzeyi üzerinde tutmaktır. Denklem 3.4'te tasarlanan kontrol sinyalleri, sistem 3.2'nin kayma yüzeyi $s = 0$ 'da kalmasını sağlar.

Kayma yüzeyinin türevi aşağıdaki denklemdeki gibi elde edilir:

$$\dot{s} = -k_3 s - k_4 \text{sign}(s). \quad (3.6)$$

Elde edilen kontrol sisteminin kararlılığının araştırılması amacıyla, Lyapunov fonksiyonu $V = 0,5s^2$ olarak seçilmiştir, Lyapunov fonksiyonunun birinci dereceden türevi aşağıda verilmiştir:

$$\begin{aligned} \dot{V} &= s\dot{s} \\ &= -k_3 s^2 - k_4(s)\text{sign}(s). \end{aligned} \quad (3.7)$$

$s^2 \geq 0$ ve $s.\text{sign}(s) = |s| \geq 0$ olduğundan dolayı, $k_3 > 0$ ve $k_4 > 0$ iken $\dot{V} \leq 0$ olur. Bu şartlar, tasarlanan kayma yüzeyi s 'in asimptotik olarak sıfır hata noktasına stabilize olacağını garanti eder.

3.1.2. Yapay sinir ağları ile kontrol

Kontrol sistemleri teorisi, dinamik sistemlerin bir ya da birkaç değişkeninin arzu edilen sınırlar arasında tutulmasını sağlayacak biçimde sistemin analiz ve tasarımını kapsar. Kontrol edilecek sistemin doğru bir matematiksel modeli biliniyorsa kontrol problemi, istenilen denetim çıkışını üretecek matematiksel bir kontrol fonksiyonu tasarlamaktır. Doğrusal sistemler için zaman ve frekans domeninde çeşitli kontrol yöntemleri geliştirilmiştir. Bazı çalışmalarda ise parametreleri bilinmeyen doğrusal sistemlerin kontrolü için çeşitli uyarlamalı kontrol yapıları incelenmiştir. Sonuçta

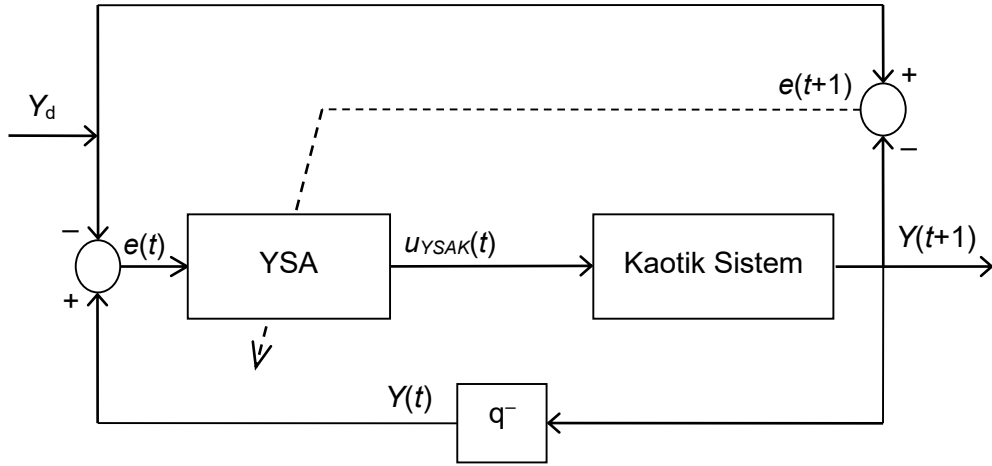
sistemin modeli ile bazı ön bilgilerin olması durumunda kararlı uyarlama kontrol kurallarının belirlenebileceği gösterilmiştir. Çalışmalar gürültü adı verilen birtakım bozucu şartlar altında da kontrol sisteminin öz yeteneğini korumasını sağlayacak kuralların oluşturulmasına doğru yönelmiştir.

Doğrusal olmayan sistemlerin uyarlamalı kontrolünde ise YSA, yapısı ve özellikleri nedeniyle tercih edilmektedir. YSA ile sistem kontrolünde, doğrusal sistemlerin denetiminde kullanılan uyarlamalı kontrol yapıları esas alınmış ve doğrusal olmayan sistemler için YSA'nın kontrol yetenekleri bazı çalışmalarda incelenmiştir.

Uyarlamalı kontrolde, doğrudan ve dolaylı uyarlamalı kontrol olmak üzere başlıca iki çeşit kontrol yapısı karşımıza çıkmaktadır. Doğrudan uyarlamalı kontrolde, sistemin düz modeli kullanılmaksızın sistemin arzu edilen şekilde davranmasını sağlayacak kontrolör parametreleri ayarlanır. Dolaylı kontrolde ise sistemin düz modelinin parametreleri kullanılarak kontrol isteklerini sağlayacak şekilde kontrolör parametreleri ayarlanır.

Ayrıca, sistem için arzu edilen çıkışın, sistemin dinamiklerine uygun bir referans modelden sağlandığı kontrol sistemleri, “model referans kontrol” olarak adlandırılır ve bu kontrol sistemleri de doğrudan ve dolaylı uyarlamalı kontrol yapıları ile incelenir.

Doğrudan kontrolde temel amaç, sistemin doğru bir ters modelinin belirlenmesidir. Ters model doğrusal olmayan sistemlerde net bilinmemektedir. Bu nedenle, sistemin ters YSA modeli kontrolör olarak kullanılır. Kontrolör YSA, referans girişten yararlanarak sistem girişini belirlemelidir. YSA model, sistemin tersini tam olarak temsil edebilirse sistemin çıkışı, referans girişe eşit olacaktır (Gökbulut, 1998). Şekil 3.1.'de temel bir kaotik sistemin YSA ile doğrudan kontrol blok diyagramı gösterilmektedir.



Şekil 3.1. Yapay sinir ağı ile kaotik sistemlerin kontrol modeli.

Şekil 3.1.'de $Y(t)$ o andaki kaotik sistemin aktif değeri, $Y(t+1)$ sistemin çıktısı ya da bir sonraki değeri ve Y_d sistemin denge noktasıdır. $e(t)$ mevcut hatayı temsil etmektedir. Kontroldeki $e(t+1)$ çıktı hatası, $Y(t+1) - Y_d$ formülü ile hesaplanır. Modelde amaç, $\lim_{t \rightarrow \infty} e(t+1) \cong 0$ olacak biçimde $u_{YSAK}(t)$ kontrol girdilerini belirlemektir. Bunun için, sistemin mevcut değeri ile denge noktası arasındaki fark yani hata geri-beslenerek YSA tabanlı kontrolöre öğretilir. Bu eğitim yolu Şekil 3.1.'de kesikli çizgi ile gösterilmiştir. Eğitim başarılı bir şekilde tamamlandıktan sonra, aynı model eğitim işlemi yapılmadan sistemin istenilen denge noktasına kontrolü için kullanılır.

Kaotik Chua devresinin kontrolünü sağlamak için sistemin x ve z durum değişkenlerine YSA tabanlı kontrolörler eklenmiştir. Bu gerçeklemede, kontrolör katsayısı YSA'ndaki ağırlıklara karşılık gelmektedir. Chua kaotik sistemi üzerine uygulanan kontrol modelinin formülü Denklem 3.8'de verilmiştir:

$$\begin{cases} \dot{x} = \alpha(y - x - f(x)) + u_{YSAK1}, \\ \dot{y} = x - y + z, \\ \dot{z} = -\beta y + u_{YSAK2}. \end{cases} \quad (3.8)$$

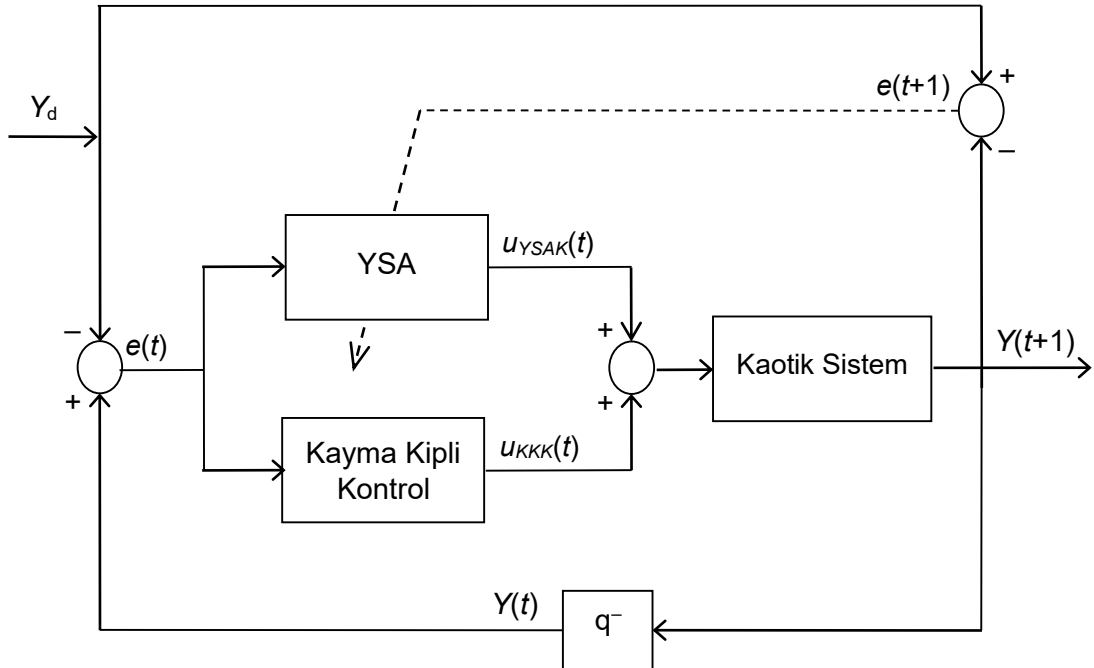
Şekil 3.1.'deki modeldeki gibi, sistemin x , y ve z durum değişkenlerindeki hata YSA'na giriş olarak uygulanmış ve YSA çıkışı x ve z durum değişkenine eklenecek

kontrol fonksiyonu değeri olarak alınmıştır. x ve z durum değişkenlerine eklenen kontrol değeri ile modelin ürettiği x ve z durum değişkenleri arasındaki hata Levenberg–Marquardt öğrenme algoritması ile minimize edilir. Böylece, Chua kaotik sisteminin x ve z durum değişkenleri ile denge noktası arasındaki fark sıfıra yaklaştırılırken sistemin diğer durum değişkeni olan y de sıfıra yakınsamaktadır.

Kontrolün başarıyla sağlanması için, eğitilmiş YSA kontrolör bir döngü ile tekrardan YSA kontrolörlü sistem çıkışı ile denge noktası arasındaki hatanın geri beslenmesi ile eğitilmiştir. Döngü 10 iterasyon çalıştırıldığında kontrol için yeterli durum sağlanmış ve YSA kontrolörler test setine göre sırasıyla 0,000000039879 ve 0,000000064047 ortalama karesel hatalara sahip olmuştur.

3.1.3. Kayma kipli yapay sinir ağları ile kontrol

Kaotik Chua devresinin kontrolünü sağlamak için sistemin x ve z durum değişkenlerine hem kayma kipli kontrolörler hem de YSA tabanlı kontrolörler eklenmiştir. Tasarlanan kontrol modelinin diyagramı Şekil 3.2.'de gösterilmiştir.



Şekil 3.2. Kayma kipli kontrol ve YSA ile tasarlanan kaotik sistemlerin kontrol modeli.

Tasarlanan hem kayma kipli kontrol hem de YSA tabanlı kontrol modelinin Chua kaotik sistemi için formülü Denklem 3.9’da verilmiştir:

$$\begin{cases} \dot{x} = \alpha(y - x - f(x)) + u_{KKK1} + u_{YSAK1}, \\ \dot{y} = x - y + z, \\ \dot{z} = -\beta y + u_{KKK2} + u_{YSAK2}. \end{cases} \quad (3.9)$$

Şekil 3.1.’de gösterilen YSA ile kontrol modeline benzer biçimde, bu sefer içinde kayma kipli kontrol fonksiyonu da yer alan bir model, kaotik sistemlerin kontrolü için tasarlanmıştır. Şekil 3.2.’deki modelde, yine sistemin x , y ve z durum değişkenlerindeki hata YSA’na giriş olarak uygulanmıştır. Aynı biçimde, YSA çıkışı x ve z durum değişkenine eklenecek kontrol fonksiyonları değerleridir. Yine Levenberg–Marquardt öğrenme algoritması kullanılmıştır. Kontrolün başarıyla sağlanması için, gene eğitilmiş YSA kontrolörler bir döngü ile tekrardan kayma kipli ve YSA kontrolörlü sistem çıkışı ile denge noktası arasındaki hatanın geri beslenmesi ile eğitilmiştir. Döngü 2 iterasyon çalıştırıldığında kontrol için yeterli durum sağlanmış, YSA kontrolörler test setine göre sırasıyla 0,000000001054 ve 0,0000000006648 ortalama karesel hataya sahip olmuştur.

3.1.4. Simülasyon sonuçları

Bilgisayar benzetimleri ile $x_0 = 0$, $y_0 = 0$, $z_0 = 0,6$ başlangıç şartlarına sahip Chua kaotik sistemi parametre değerleri $\alpha = 15,6$, $\beta = 25,58$, $a = -8/7$ ve $b = -5/7$ alınarak kayma kipli kontrol, YSA ve bu iki yöntemin birleşimi ile kontrolü denetlenmiştir. Kayma kipli kontrol parametre değerleri $k_1 = 1$, $k_2 = 1$, $k_3 = 1$ ve $k_4 = 0,1$ olarak alınmıştır.

Chua kaotik sisteminin belirtilen t_k zamanında denge noktasına kontrolünün sağlanması, kayma kipli kontrol, YSA ile kontrol ve kayma kipli YSA ile kontrol sonuçlarının karşılaştırılması amacıyla ‘chua_k.m’ dosyası oluşturulmuştur. Bu kod kayma kipli kontrolü içeren ‘chuaf_kkk’, YSA ile kontrolü içeren ‘chuaf_ysak’ ve

kayma kipli YSA ile kontrolü içeren ‘chuaf_kkk_ysak’ fonksiyonlarını kullanmaktadır. Matlab dilinde yazılan Program Kodları 3.1–3.4 aşağıda verilmiştir.

Program Kodu 3.1. chua_k.m dosyası

```
clear;

global alfa beta a b;
global tk xd yd zd k1 k2 k3 k4;
global YSAK_f1 YSAK_f2 maks_t1 maks_t2;

t0 = 0;
tf = 50;

tk = 30;

alfa = 15.6;
beta = 25.58;
a = -8/7;
b = -5/7;

xd = 0;
yd = 0;
zd = 0;

k1 = 1;
k2 = 1;
k3 = 1;
k4 = 0.1;

x0 = [0 0 0.6]';

tic
[tout, xout] = ode45('chuaf_kkk', [t0, tf], x0);
toc

load 'chua_kntrl_ysak_10.mat';
tic
[tout2, xout2] = ode45('chuaf_ysak', [t0, tf], x0);
toc

load 'chua_kntrl_kkk_ysak_2.mat';
tic
[tout3, xout3] = ode45('chuaf_kkk_ysak', [t0, tf], x0);
toc

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,1), 'b-', tout2, xout2(:,1), 'g--', tout3, xout3(:,1),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
```

```

ylabel('x', 'FontSize', 12, 'FontWeight', 'bold');
legend('Kayma Kipli Kontrol', 'YSA ile Kontrol', 'Kayma Kipli YSA ile
Kontrol');

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,2), 'b-', tout2, xout2(:,2), 'g--', tout3, xout3(:,2),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('y', 'FontSize', 12, 'FontWeight', 'bold');
legend('Kayma Kipli Kontrol', 'YSA ile Kontrol', 'Kayma Kipli YSA ile
Kontrol');

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,3), 'b-', tout2, xout2(:,3), 'g--', tout3, xout3(:,3),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('z', 'FontSize', 12, 'FontWeight', 'bold');
legend('Kayma Kipli Kontrol', 'YSA ile Kontrol', 'Kayma Kipli YSA ile
Kontrol');

```

Program Kodu 3.2. chuaaf_kkk.m dosyası

```

function xdot = chuaaf_kkk(t, x)

global alfa beta a b;
global tk xd yd zd k1 k2 k3 k4;

if t < tk
    uKKK1 = 0;
    uKKK2 = 0;
else
    e = [x(1) - xd; x(2) - yd; x(3) - zd];

    s1 = e(1) + k1*e(2);
    uKKK1 = -alfa*(e(2) - e(1) - (b*e(1) + 0.5*(a - b)*(abs(e(1) + 1) -
abs(e(1) - 1)))) - k1*(e(1) - e(2) + e(3)) - k3*s1 - k4*sign(s1);

    s2 = e(3) + k2*e(2);
    uKKK2 = beta*e(2) - k2*(e(1) - e(2) + e(3)) - k3*s2 - k4*sign(s2);
end;

xdot1 = alfa*(x(2) - x(1) - (b*x(1) + 0.5*(a - b)*(abs(x(1) + 1) - abs(x(1)
- 1)))) + uKKK1;
xdot2 = x(1) - x(2) + x(3);
xdot3 = -beta*x(2) + uKKK2;

xdot = [xdot1; xdot2; xdot3];

```

Program Kodu 3.3. chuaif_ysak.m dosyası

```
function xdot = chuaif_ysak(t, x)

global alfa beta a b;
global tk xd yd zd;
global YSAK_f1 YSAK_f2 maks_t1 maks_t2;

if t < tk
    uYSAK1 = 0;
    uYSAK2 = 0;
else
    uYSAK1 = maks_t1 * sim(YSAK_f1, [(x(1)-xd) (x(2)-yd) (x(3)-zd)]');
    uYSAK2 = maks_t2 * sim(YSAK_f2, [(x(1)-xd) (x(2)-yd) (x(3)-zd)]');
end;

xdot1 = alfa * (x(2) - x(1) - (b * x(1) + 0.5 * (a - b) * (abs(x(1) + 1) -
abs(x(1) - 1)))) + uYSAK1;
xdot2 = x(1) - x(2) + x(3);
xdot3 = -beta * x(2) + uYSAK2;

xdot = [xdot1; xdot2; xdot3];
```

Program Kodu 3.4. chuaif_kkk_ysak.m dosyası

```
function xdot = chuaif_kkk_ysak(t, x)

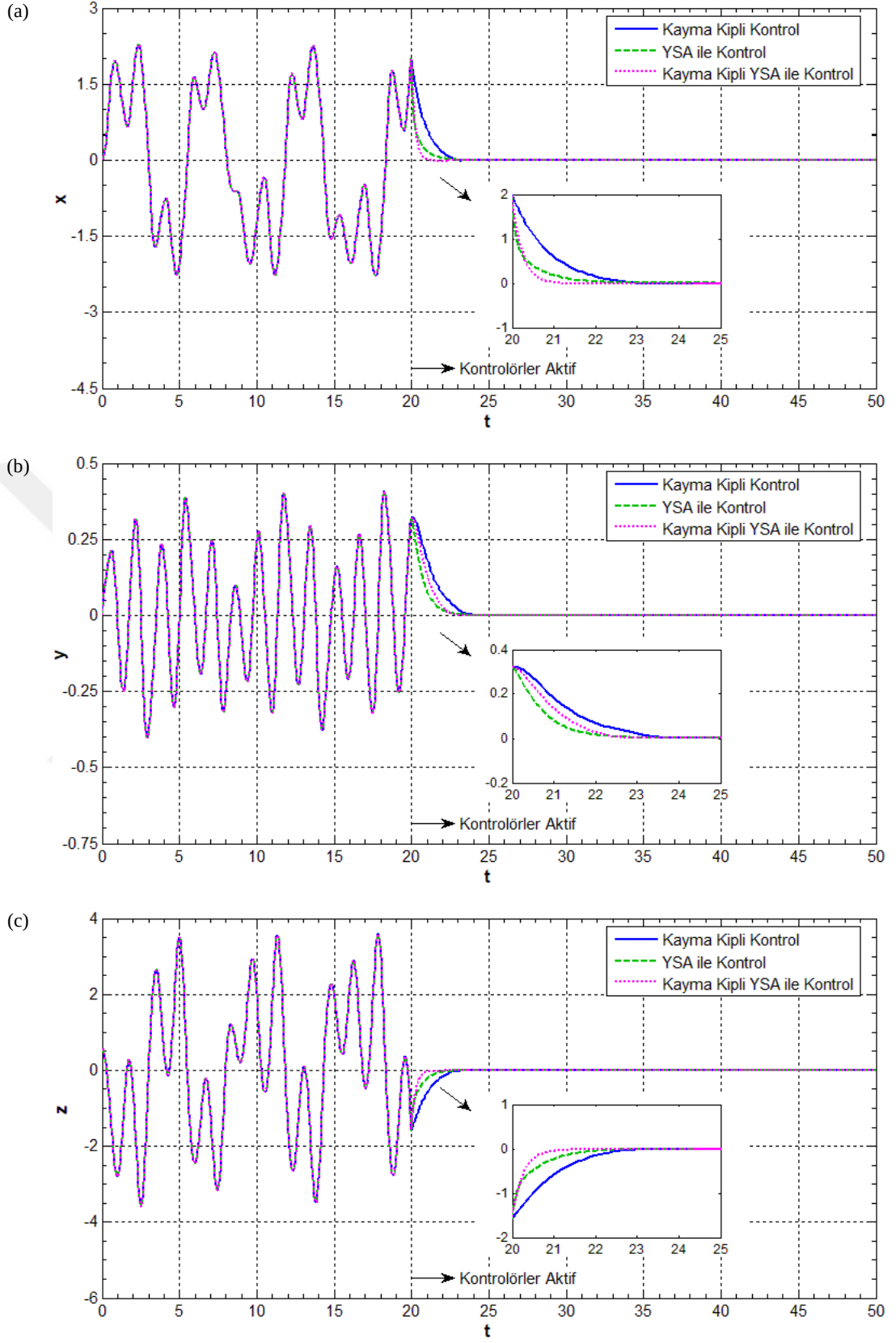
global alfa beta a b;
global tk xd yd zd k1 k2 k3 k4;
global YSAK_f1 YSAK_f2 maks_t1 maks_t2;

if t < tk
    uKKK1 = 0;
    uKKK2 = 0;
    uYSAK1 = 0;
    uYSAK2 = 0;
else
    e = [x(1) - xd; x(2) - yd; x(3) - zd];
    s1 = e(1) + k1*e(2);
    uKKK1 = -alfa*(e(2) - e(1) - (b*e(1) + 0.5*(a - b)*(abs(e(1) + 1) -
abs(e(1) - 1)))) - k1*(e(1) - e(2) + e(3)) - k3*s1 - k4*sign(s1);
    uYSAK1 = maks_t1 * sim(YSAK_f1, [(x(1)-xd) (x(2)-yd) (x(3)-zd)]');
    s2 = e(3) + k2*e(2);
    uKKK2 = beta*e(2) - k2*(e(1) - e(2) + e(3)) - k3*s2 - k4*sign(s2);
    uYSAK2 = maks_t2 * sim(YSAK_f2, [(x(1)-xd) (x(2)-yd) (x(3)-zd)]');
end;

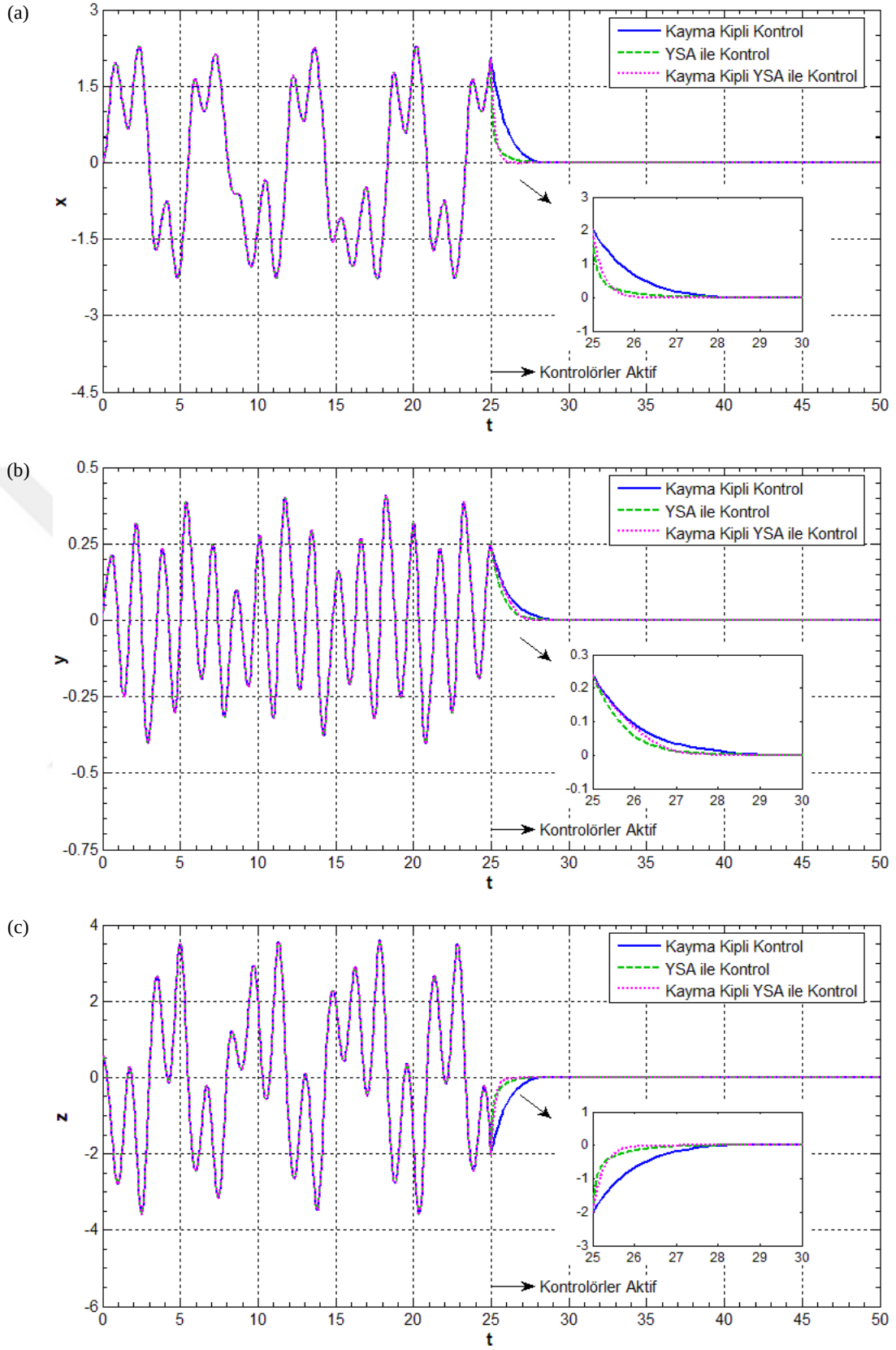
xdot1 = alfa*(x(2) - x(1) - (b*x(1) + 0.5*(a - b)*(abs(x(1) + 1) - abs(x(1)
- 1)))) + uKKK1 + uYSAK1;
xdot2 = x(1) - x(2) + x(3);
xdot3 = -beta*x(2) + uKKK2 + uYSAK2;

xdot = [xdot1; xdot2; xdot3];
```

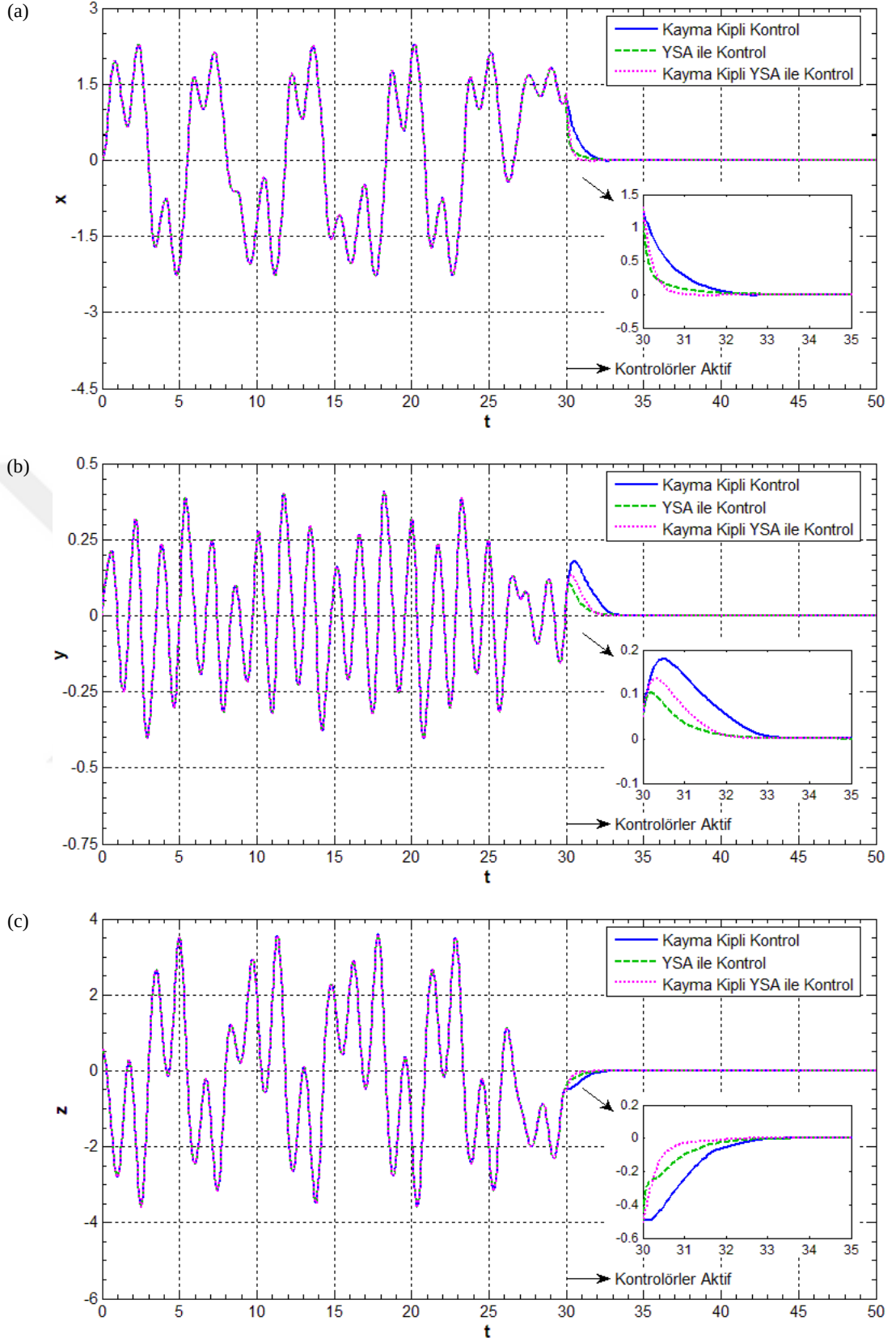
Kontrolörler $t = 20$, $t = 25$ ve $t = 30$ anlarında aktive edilmiştir ve sonuçlar sırasıyla Şekil 3.3., Şekil 3.4. ve Şekil 3.5.'te karşılaştırmalı olarak gösterilmiştir.



Şekil 3.3. Kontrolörler $t = 20$ anında aktive edildiğinde Chua kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 3.4. Kontrolörler $t = 25$ anında aktive edildiğinde Chua kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekiller 3.3.–3.5.’teki benzetim sonuçları hem kayma kipli kontrol, hem klasik YSA, hem de kayma kipli YSA yöntemleri ile kaotik Chua sisteminin $E(0, 0, 0)$ denge noktasına kontrolünün başarılı bir biçimde gerçekleştiğini göstermektedir. Şekiller karşılaştırmalı olarak kontrol sonuçlarını göstermektedir. Kontrolörler $t = 20$ anında aktive edildiğinde, kontrol ilk olarak $t \geq 22$ ’de kayma kipli YSA ile sağlanmıştır. Kontrolörler $t = 25$ anında aktive edildiğinde, kontrol ilk olarak $t \geq 27$ ’de yine kayma kipli YSA ile sağlanmıştır. Kontrolörler $t = 30$ anında aktive edildiğinde, kontrol ilk olarak $t \geq 32$ ’de yine kayma kipli YSA ile sağlanmıştır. Dolayısıyla, simülasyon sonuçları göstermiştir ki kayma kipli YSA kontrol, kayma kipli kontrol ve klasik YSA ile kontrol yöntemlerinden daha efektif kaos kontrol sonuçları üretmektedir.

3.2. Bonhoeffer–Van Der Pol Kaotik Sisteminin Kontrolü

Bonhoeffer–van der Pol kaotik sisteminin kontrolü sırasıyla klasik kontrol yöntemlerinden pasif kontrol ile, yapay zekâ tekniklerinden bulanık mantık ile ve her ikisinin birleşimi ile uygulanmış, bilgisayar simülasyonlarından elde edilen sonuçlar karşılaştırmalı grafiklerle sunulmuştur.

3.2.1. Pasif kontrol

Bu bölümde kaotik Bonhoeffer–van der Pol osilatörünün kontrolü pasif kontrol yöntemi ile gerçekleştirilmiştir. Denklem 2.9’da verilen kaotik Bonhoeffer–van der Pol sisteme pasif kontrolün uygulanması amacıyla x durum değişkenine u_{PK} kontrol fonksiyonu eklenerek yeniden aşağıdaki biçimde yazılabilir:

$$\begin{cases} \dot{x} = -z + s_1 x + \tanh(s_2 x) + u_{PK}, \\ \dot{y} = z - s_3 y, \\ \dot{z} = x - y. \end{cases} \quad (3.10)$$

Kontrol fonksiyonu u eklenen denklem $Y = x$, ve diğer denklemler $Z_1 = y$, $Z_2 = z$, $Z = [Z_1 \ Z_2]$, dönüşümleri uygulanarak aşağıdaki gibi tekrardan yazılır:

$$\begin{cases} \dot{Z}_1 = Z_2 - s_3 Z_1, \\ \dot{Z}_2 = Y - Z_1, \\ \dot{Y} = -Z_2 + s_1 Y + \tanh(s_2 Y) + u_{PK}. \end{cases} \quad (3.11)$$

Böylece Denklem 3.11, pasif kontrol teoreminin

$$\begin{aligned} \dot{Z} &= f_0(Z) + p(Z, Y)Y, \\ \dot{Y} &= b(Z, Y) + a(Z, Y)u_{PK} \end{aligned} \quad (3.12)$$

fonksiyonlarına dönüştürülebilir.

$f_0(Z)$ fonksiyonu Z denklemindeki Y dışındaki değerleri içerir:

$$f_0(Z) = \begin{bmatrix} Z_2 - s_3 Z_1 \\ -Z_1 \end{bmatrix}. \quad (3.13)$$

$p(Z, Y)$ fonksiyonu Z denkleminin Y 'ye göre türevi alınarak elde edilir:

$$p(Z, Y) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (3.14)$$

$b(Z, Y)$ fonksiyonu Y denklemlerinin u dışındaki değerlerini içerir:

$$b(Z, Y) = -Z_2 + s_1 Y + \tanh(s_2 Y). \quad (3.15)$$

Bu durumda

$$a(Z, Y) = 1 \quad (3.16)$$

olur.

Depo fonksiyonu aşağıdaki gibi seçilirse:

$$V(Z, Y) = W(Z) + \frac{1}{2} Y^2, \quad (3.17)$$

ki burada

$$W(Z) = \frac{1}{2} (Z_1^2 + Z_2^2), \quad (3.18)$$

$f_0(Z)$ fonksiyonunun pozitif tanımlı Lyapunov fonksiyonudur ve $W(0) = 0$ 'dır. $V(Z, Y)$ 'nin türevi alınırsa

$$\begin{aligned} \frac{d}{dt} V(Z, Y) &= \frac{\partial W(Z)}{\partial Z} \dot{Z} + Y\dot{Y} \\ &= \frac{\partial W(Z)}{\partial Z} f_0(Z) + \frac{\partial W(Z)}{\partial Z} p(Z, Y)Y + b(Z, Y)Y + a(Z, Y)Y u_{PK} \end{aligned} \quad (3.19)$$

elde edilir. Hata sistemi, minimum aşamalı olarak aşağıdaki denklemdeki gibi alınabilir:

$$\frac{d}{dt} W(Z) f_0(Z) \leq 0. \quad (3.20)$$

Bu durumda Denklem 3.20'deki eşitsizlik aşağıdaki hale dönüşür:

$$\frac{d}{dt} V(Z, Y) \leq \frac{\partial W(Z)}{\partial Z} p(Z, Y)Y + b(Z, Y)Y + a(Z, Y)Y u_{PK}. \quad (3.21)$$

Sistem 3.12'nin sıfır hatadaki dinamiği sistemin iç dinamiklerini temsil eder ki bu da sistemin dış dinamiğinin sabit $Y = 0$ yani $\dot{Z} = f_0(Z)$ olması ile ilişkilidir. Denklem 3.13'e göre $W(Z)$ 'nin türevi alınırsa,

$$\dot{W}(Z) = \frac{d}{dt}W(Z) = \frac{\partial W(Z)}{\partial Z} f_0(Z) = \begin{bmatrix} Z_1 & Z_2 \end{bmatrix} \begin{bmatrix} Z_2 - s_3 Z_1 \\ -Z_1 \end{bmatrix} = -s_3 Z_1^2 \quad (3.22)$$

olur. Bonhoeffer–van der Pol kaotik sisteminin s_3 parametresi pozitif olduğundan Denklem 3.22 kesin negatif tanımlıdır. $f_0(Z)$ fonksiyonunun Lyapunov fonksiyonu olan $W(Z)$ asimptotik kararlıdır. Dolayısıyla, denklem sistemi 3.11’in sıfıra yakınsayacağı kanıtlanmıştır.

Pasif kontrol yönteminde u_{PK} kontrol fonksiyonu aşağıdaki formüle göre bulunur (Mahmoud ve Abualnaja, 2013):

$$u_{PK} = a(Z, Y)^{-1} \left[-b(Z, Y) - \left(\frac{\partial W(Z)}{\partial Z} p(Z, Y) \right)^T - \alpha Y + v \right]. \quad (3.23)$$

Daha önceki Denklemler 3.14–3.16’daki fonksiyonlar Denklem 3.23’te yerine konursa:

$$u_{PK} = Z_2 - s_1 Y - \tanh(s_2 Y) - Z_2 - \alpha Y + v \quad (3.24)$$

olarak bulunur. Burada α ve v pasif kontrolün sistem parametreleridir. $Y = x$, $Z_1 = y$ ve $Z_2 = z$ dönüşümleri geri alınır ve eşitlik sadeleştirilirse

$$u_{PK} = -s_1 x - \tanh(s_2 x) - \alpha x + v \quad (3.25)$$

elde edilir. Pasif kontroldeki u_{PK} kontrol fonksiyonu kaotik sistemleri $(0, 0, 0)$ orijin noktası dışında α ve v parametrelerine bağlı olarak farklı noktalara da yönlendirebilmektedir. Böyle bir durumda v fonksiyonu devreye girmektedir. Bonhoeffer–van der Pol kaotik sisteminin sıfır noktası dışında başka bir denge noktası olmadığı için $v = 0$ olarak alınmalıdır, ayrıca değerinin hesaplamasına gerek yoktur. Sonuç olarak, pasif kontrolörlü Bonhoeffer–van der Pol kaotik sistemi 3.10 Denklem 3.25 ile tamamlanmıştır.

3.2.2. Bulanık mantık ile kontrol

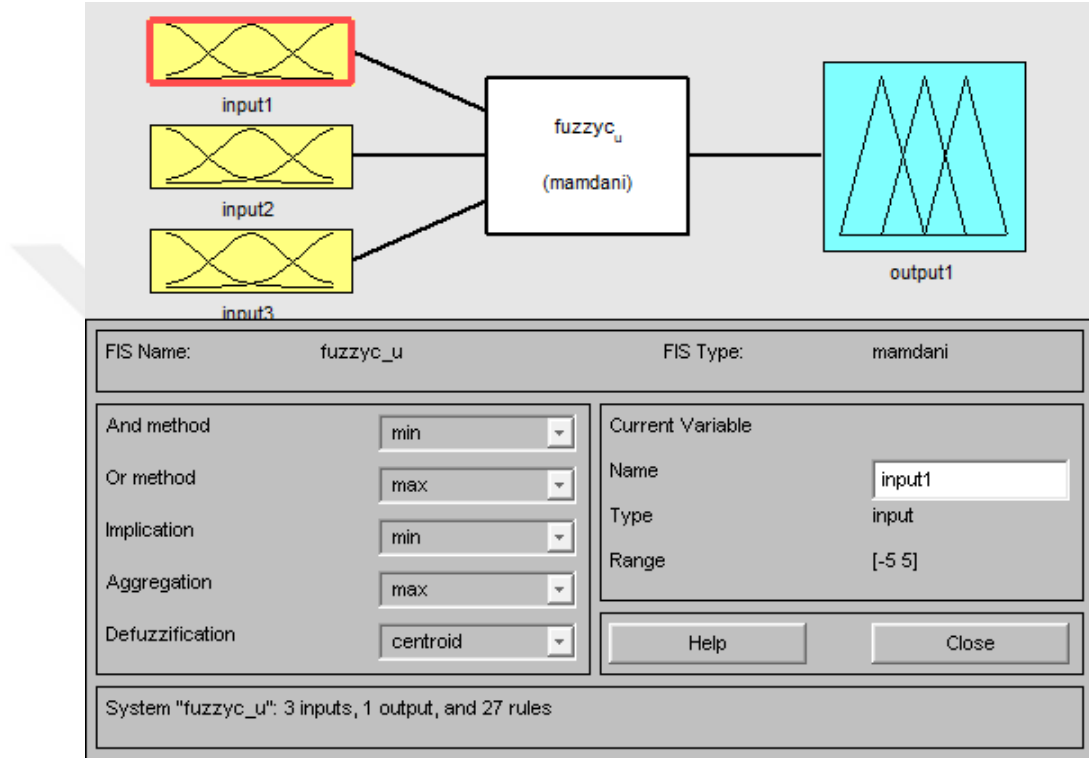
Klasik ve modern kontrol teorisinde kontrolü yapılacak sistemin kesin ve tam matematiksel modeline gerek duyulur. Bulanık mantık kontrolcöleri ise, kesin ve tam matematiksel formöllere ihtiyaç duymazlar. Bu yüzden, modellenmesi ve denetlenmesi zor olan karmaşık sistemlerde bulanık mantıktan faydalanmak sıkça tercih edilir hale gelmiştir. Yapısının uyarlanabilir özelliğe olması, değişken parametrelere ve belirsizliklere sahip sistemlerin kontrolünde bulanık mantığı avantajlı kılmakta, kontrol zamanını kısaltan etkin cevaplar üretmesini sağlamaktadır.

Bulanık mantık ile denetimi avantajlı kılan, uzman sistem operatörlerinin bilgi, sezgi ve deneyimlerinden faydalanılmasına imkân tanınmasıdır. Uzmanlar, kontrol davranışını “küçük”, “hızlı”, “çok” gibi sözel terimler ile ve “Eğer–İse” (If–Then) komutlarından oluşan kurallar ile belirlerler. Kontrol işlemi de oluşturulan bu bilgi tabanı sayesinde gerçekleşir.

Bulanık kontrol sistemleri bulanıklaştırma, kural tabanı, sonuç çıkarım ünitesi ve durulaştırma olmak üzere 4 temel birimden oluşmaktadır.

Bulanıklaştırma birimi, sayısal girdi değerlerini sözel ifadelerden oluşan bulanık üyelik fonksiyonları değerlerine dönüştürür. Kural tabanı, “Eğer–İse” biçiminde ifade edilen uzman kişilerin sistemi kontrol etme yöntemini belirledikleri bulanık kurallardan oluşur. Sonuç çıkarım ünitesi, önce bulanıklaştırma biriminden gelen değerleri kural tabanındaki bulanık kurallar ile eşleştirir, sonra birden fazla kurala denk gelen durumlar için uygun yorumlamayı yapar ve bulanık ya da net bir kontrol sonucu üretir. Durulaştırıcı, çıkarım ünitesinden gelen bulanık sonucu, denetimi gerçekleştirilen sisteme uygulanacak sayısal kontrol değerine çevirir. Bulanık mantık kontrolcölerinin tasarımındaki en büyük zorluk, uygun üyelik fonksiyonlarının ve doğru bulanık mantık kurallarının tanımlanmasıdır (Lin ve Lee, 1994).

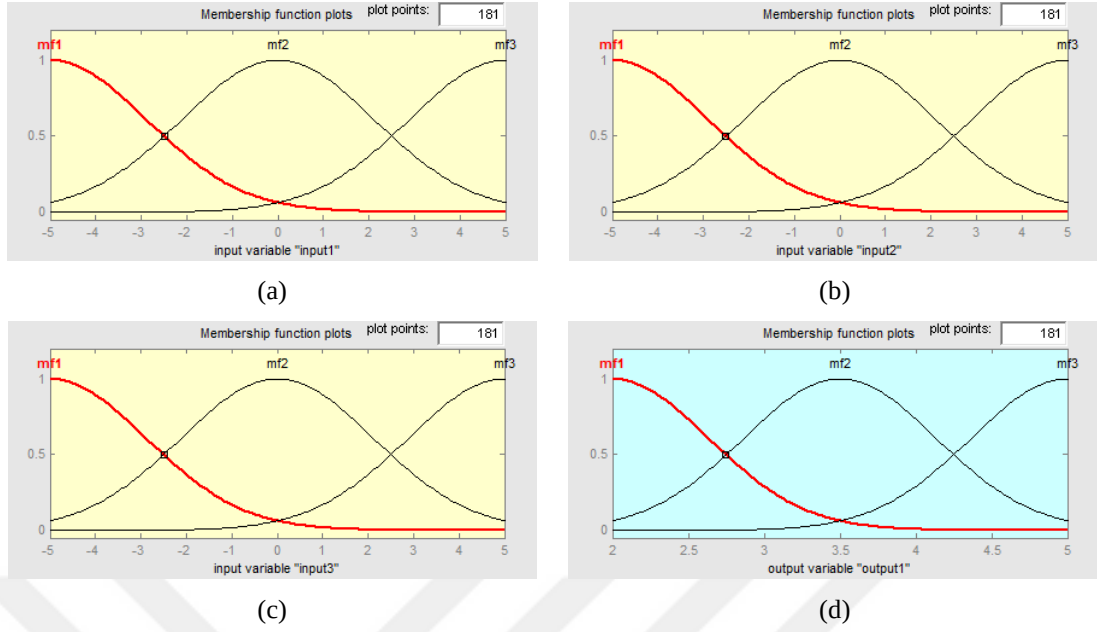
Bonhoeffer–van der Pol kaotik sisteminin kontrolü için, pasif kontrol yöntemine de eşdeğer olması amacıyla Denklem 3.26’da gösterildiği gibi sistemin sadece x durum değişkenine bir bulanık kontrolör eklenmiştir. Bulanık kontrolörün giriş değişkenleri sistem hataları e_1 , e_2 ve e_3 , çıkış değişkeni ise kontrol sinyali değeridir. Bulanık mantık denetleyicisinin giriş ve çıkışını gösteren blok yapısı Şekil 3.6.’da verilmiştir.



Şekil 3.6. Bulanık mantık denetleyici blok yapısı.

$$\begin{cases} \dot{x} = -z + s_1 x + \tanh(s_2 x) + u_{BK}, \\ \dot{y} = z - s_3 y, \\ \dot{z} = x - y. \end{cases} \quad (3.26)$$

Giriş-çıkış değişkenleri üyelik fonksiyonları Şekil 3.7.’de gösterildiği gibi çan biçimli olarak tercih edilmiştir. Bunlarda girişlerde mf1: Negatif (N), mf2: Orta (O), mf3: Pozitif (P); çıkışta mf1: Az (A), mf2: Orta (O), mf3: Çok (Ç) anlamlarına gelen üçer adet fonksiyon kullanılmıştır.

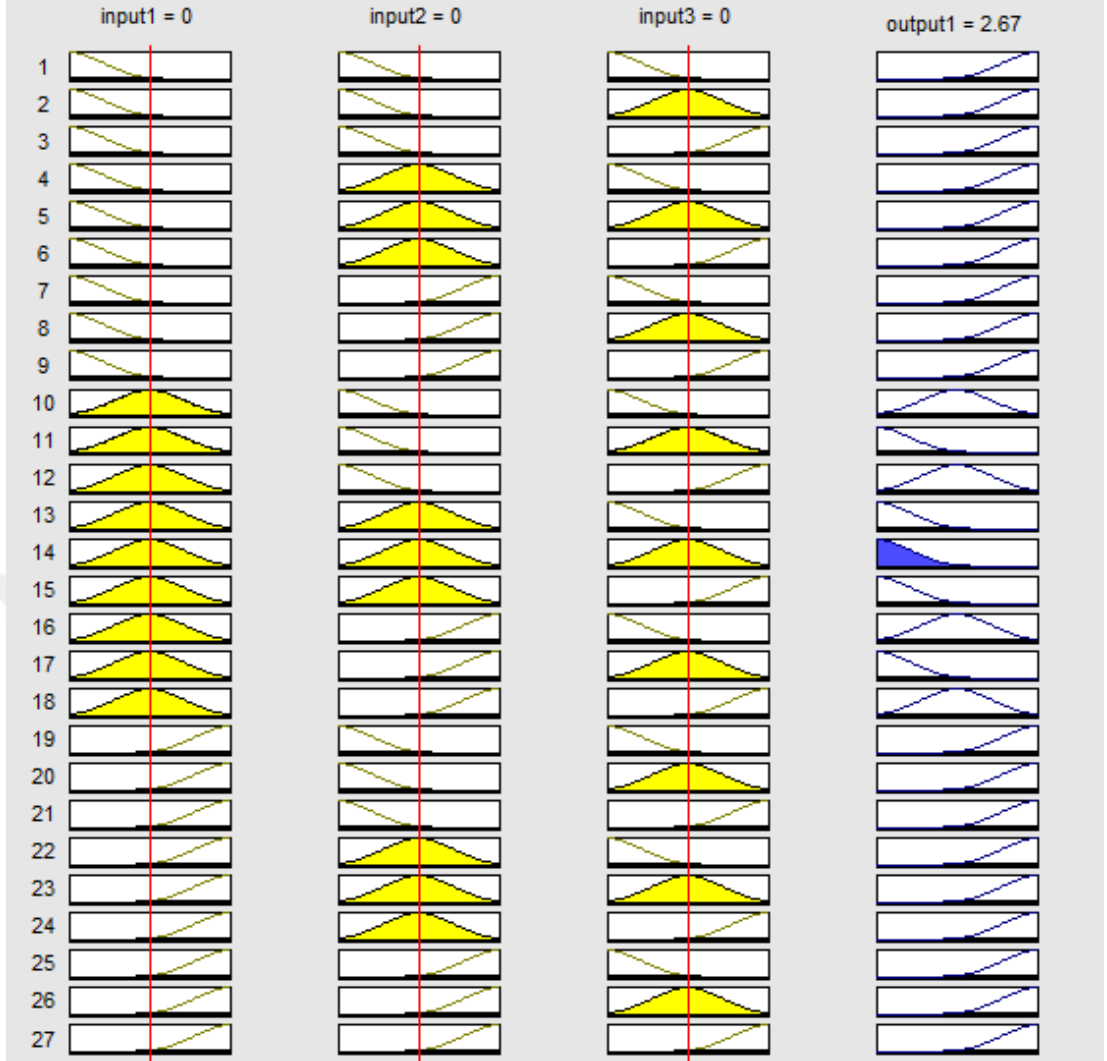


Şekil 3.7. Bulanık kontrolör üyelik fonksiyonları (a) e_1 giriş değişkeni, (b) e_2 giriş değişkeni, (c) e_3 giriş değişkeni, (d) çıkış değişkeni.

Bulanık kontrolör kural tabanı, sözel hüküm cümlelerinden oluşmuş bulanık kurallar kümesini içerir. Bulanık kurallar genellikle çizelge yardımı ile gösterilirler. Çizelgeyi oluştururken amaç, zamanla hatanın 0 olmasını sağlayacak bulanık kuralları belirlemektir. Bundan dolayı, e_1 , e_2 ve e_3 hataları 0'a doğru yaklaştıkça kontrol sinyali değerinin de 0'a doğru azalması gerekir. Tablo 3.1. ve Şekil 3.8.'de görüldüğü gibi çıkış sinyali merkezdeki 0 hata noktasından köşelere göre simetriktir.

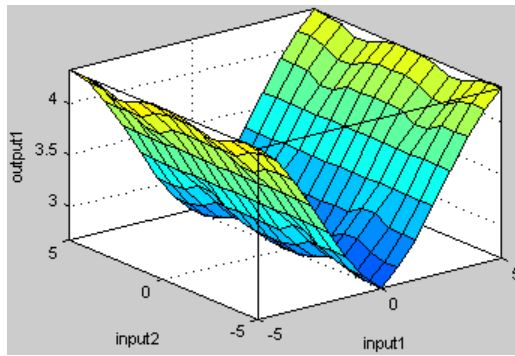
Tablo 3.1. Bulanık kontrolör kuralları

$e_1 - e_2 / e_3$	N	O	P
N-N	Ç	Ç	Ç
N-O	Ç	Ç	Ç
N-P	Ç	Ç	Ç
O-N	O	A	O
O-O	A	A	A
O-P	O	A	O
P-N	Ç	Ç	Ç
P-O	Ç	Ç	Ç
P-P	Ç	Ç	Ç

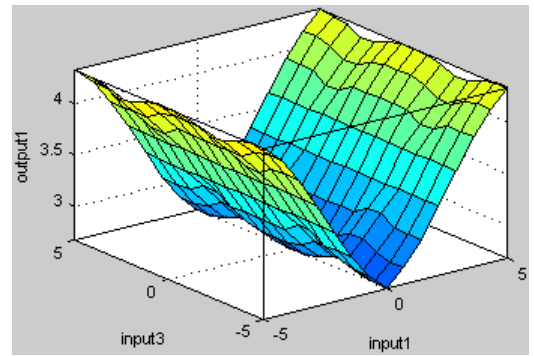


Şekil 3.8. Bulanık kontrolör kuralları.

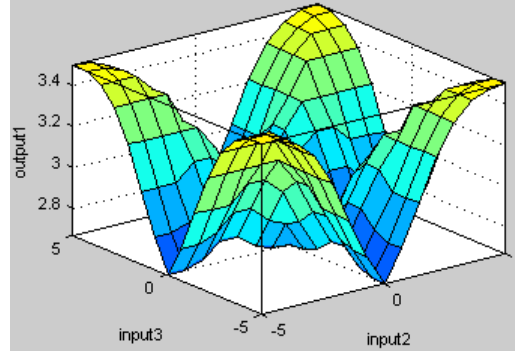
Bulanık kontrolörün girişlerin çıkış değişkeni üzerinde etkisini gösteren 3-boyutlu yüzey diyagramları Şekil 3.9.'daki gibidir:



(a)



(b)



(c)

Şekil 3.9. Bulanık kontrolör yüzey diyagramları (a) e_1-e_2-u , (b) e_1-e_3-u , (c) e_2-e_3-u .

Bulanık kontrolörün durulaştırılması için ağırlık merkezi yöntemi tercih edilmiştir.

3.2.3. Pasif bulanık mantık ile kontrol

Bu gerçeekte, pasif kontrol yönteminde bulunan kontrolör katsayısı bulanıklaştırılmıştır. Denklem 3.27’de gösterildiği gibi, pasif kontrol sinyali kullanılmış ve bulanıklaştırılmıştır. Yani, $u_{PK} = -s_1x - \tanh(s_2x) - \alpha x + v$ pasif kontrol sinyali bulanık mantık ile işlenmiştir. Bulanık kontrolörün giriş değişkenleri sistem hataları e_1 , e_2 ve e_3 , çıkış değişkeni ise kontrol sinyali katsayısı α (*alfa*)’dır.

$$\begin{cases} \dot{x} = -z + s_1x + \tanh(s_2x) + u_{PBK}, \\ \dot{y} = z - s_3y, \\ \dot{z} = x - y. \end{cases} \quad (3.27)$$

Giriş-çıkış değişkenleri üyelik fonksiyonları, bulanık kontrol yönteminde Şekil 3.7.’deki gibi çan biçimli üçer fonksiyon olarak alınmıştır. Kural tabanı da Tablo 3.1.’de verildiği gibi aynen kullanılmıştır. Dolayısıyla, Şekil 3.8.’deki bulanık kurallar ve Şekil 3.9.’daki yüzeyler, pasif bulanık kontrolörde aynen geçerlidir. Yine Mamdani bulanık çıkarım sistemi kullanılmıştır. Durulaştırıcı da gene önceki bulanık kontroldeki gibi ağırlık merkezi yöntemi olarak tercih edilmiştir. Aynı üyelik fonksiyonlarının, kural tabanının, sonuç çıkarım mekanizmasının ve durulaştırma yönteminin seçilme nedeni adil bir karşılaştırmaya imkân tanımaktır.

3.2.4. Simülasyon sonuçları

Bilgisayar benzetimleri ile $x(0) = 0,5$, $y(0) = -1$, $z(0) = 0,5$ başlangıç şartlarına sahip Bonhoeffer–van der Pol kaotik sistemi parametre değerleri $s_1 = 1$, $s_2 = 1$ ve $s_3 = 1,2$ alınarak pasif kontrolü, bulanık mantık ile kontrolü ve pasif bulanık mantık ile kontrolü denetlenmiştir. Pasif kontrol parametre değerleri $\alpha = 1$ ve $v = 0$ olarak alınmıştır.

Bonhoeffer–van der Pol kaotik sisteminin belirtilen t_k zamanında denge noktasına kontrolünün sağlanması, pasif kontrol, bulanık mantık ile kontrol ve pasif bulanık mantık ile kontrol sonuçlarının karşılaştırılması amacıyla ‘bonhoeffer_k.m’ dosyası oluşturulmuştur. Bu kod pasif kontrolü içeren ‘bonhoefferf_pk’, bulanık mantık ile kontrolü içeren ‘bonhoefferf_bk’ ve pasif bulanık mantık ile kontrolü içeren ‘bonhoefferf_pbk’ fonksiyonlarını kullanmaktadır. Ayrıca, Matlab’ın bulanık mantık arayüzü kullanılarak ‘fuzzyc_u’ adında yukarıda belirtilen özelliklere sahip bir bulanık denetleyici oluşturulmuştur. Matlab dilinde yazılan Program Kodları 3.5–3.8 aşağıda verilmiştir.

Program Kodu 3.5. bonhoeffer_k.m dosyası

```
clear;

global s1 s2 s3;
global tk alfa v blnk_k;

t0 = 0;
tf = 100;

tk = 60;

s1 = 1;
s2 = 1;
s3 = 1.2;

alfa = 1;
v = 0;

blnk_k = readfis('fuzzyc_u');

x0 = [0.5 -1 0.5]';

tic
```

```

[tout, xout] = ode45('bonhoefferf_pk', [t0, tf], x0);
toc

tic
[tout2, xout2] = ode45('bonhoefferf_bk', [t0, tf], x0);
toc

tic
[tout3, xout3] = ode45('bonhoefferf_pbk', [t0, tf], x0);
toc

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,1), 'b-', tout2, xout2(:,1), 'g--', tout3, xout3(:,1),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('x', 'FontSize', 12, 'FontWeight', 'bold');
legend('Pasif Kontrol', 'Bulanık Kontrol', 'Pasif Bulanık Kontrol');

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,2), 'b-', tout2, xout2(:,2), 'g--', tout3, xout3(:,2),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('y', 'FontSize', 12, 'FontWeight', 'bold');
legend('Pasif Kontrol', 'Bulanık Kontrol', 'Pasif Bulanık Kontrol');

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,3), 'b-', tout2, xout2(:,3), 'g--', tout3, xout3(:,3),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('z', 'FontSize', 12, 'FontWeight', 'bold');
legend('Pasif Kontrol', 'Bulanık Kontrol', 'Pasif Bulanık Kontrol');

```

Program Kodu 3.6. bonhoefferf_pk.m dosyası

```

function xdot = bonhoefferf_pk(t, x)

global s1 s2 s3;
global tk alfa v;

if t < tk
    uPK = 0;
else

```

```

    uPK = -s1*x(1) - tanh(s2*x(1)) - alfa*x(1) + v;
end;

xdot1 = -x(3) + s1*x(1) + tanh(s2*x(1)) + uPK;
xdot2 = x(3) - s3*x(2);
xdot3 = x(1) - x(2);

xdot = [xdot1; xdot2; xdot3];

```

Program Kodu 3.7. bonhoefferf_bk.m dosyası

```

function xdot = bonhoefferf_bk(t, x)

global s1 s2 s3;
global tk blnk_k;

if t < tk
    uBK = 0;
else
    uBK = -evalfis([x(1) x(2) x(3)], blnk_k)*x(1);
end;

xdot1 = -x(3) + s1*x(1) + tanh(s2*x(1)) + uBK;
xdot2 = x(3) - s3*x(2);
xdot3 = x(1) - x(2);

xdot = [xdot1; xdot2; xdot3];

```

Program Kodu 3.8. bonhoefferf_pbk.m dosyası

```

function xdot = bonhoefferf_pbk(t, x)

global s1 s2 s3;
global tk alfa v blnk_k;

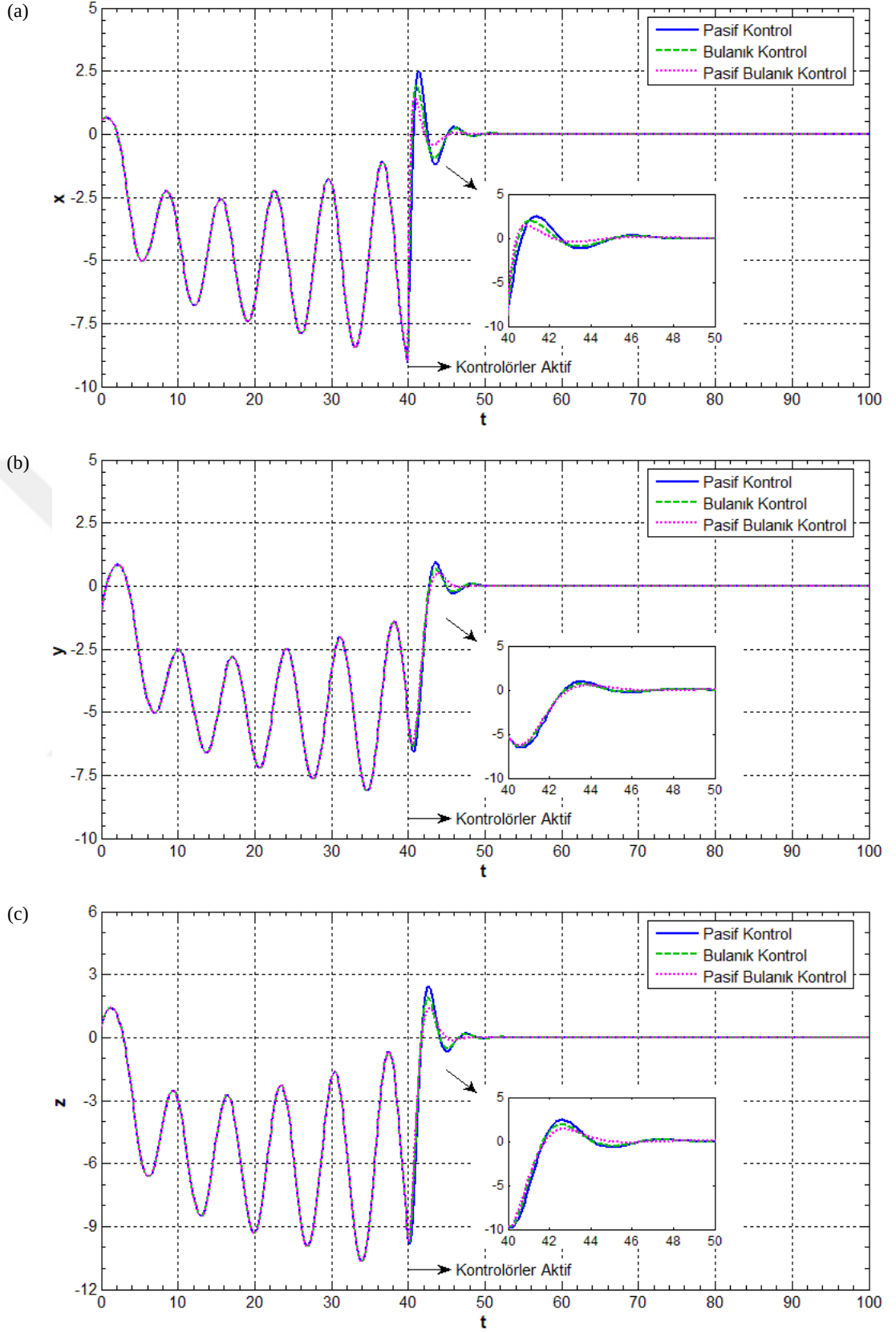
if t < tk
    uPBK = 0;
else
    uPBK = -s1*x(1) - tanh(s2*x(1)) - evalfis([x(1) x(2) x(3)], blnk_k)*x(1) + v;
end;

xdot1 = -x(3) + s1*x(1) + tanh(s2*x(1)) + uPBK;
xdot2 = x(3) - s3*x(2);
xdot3 = x(1) - x(2);

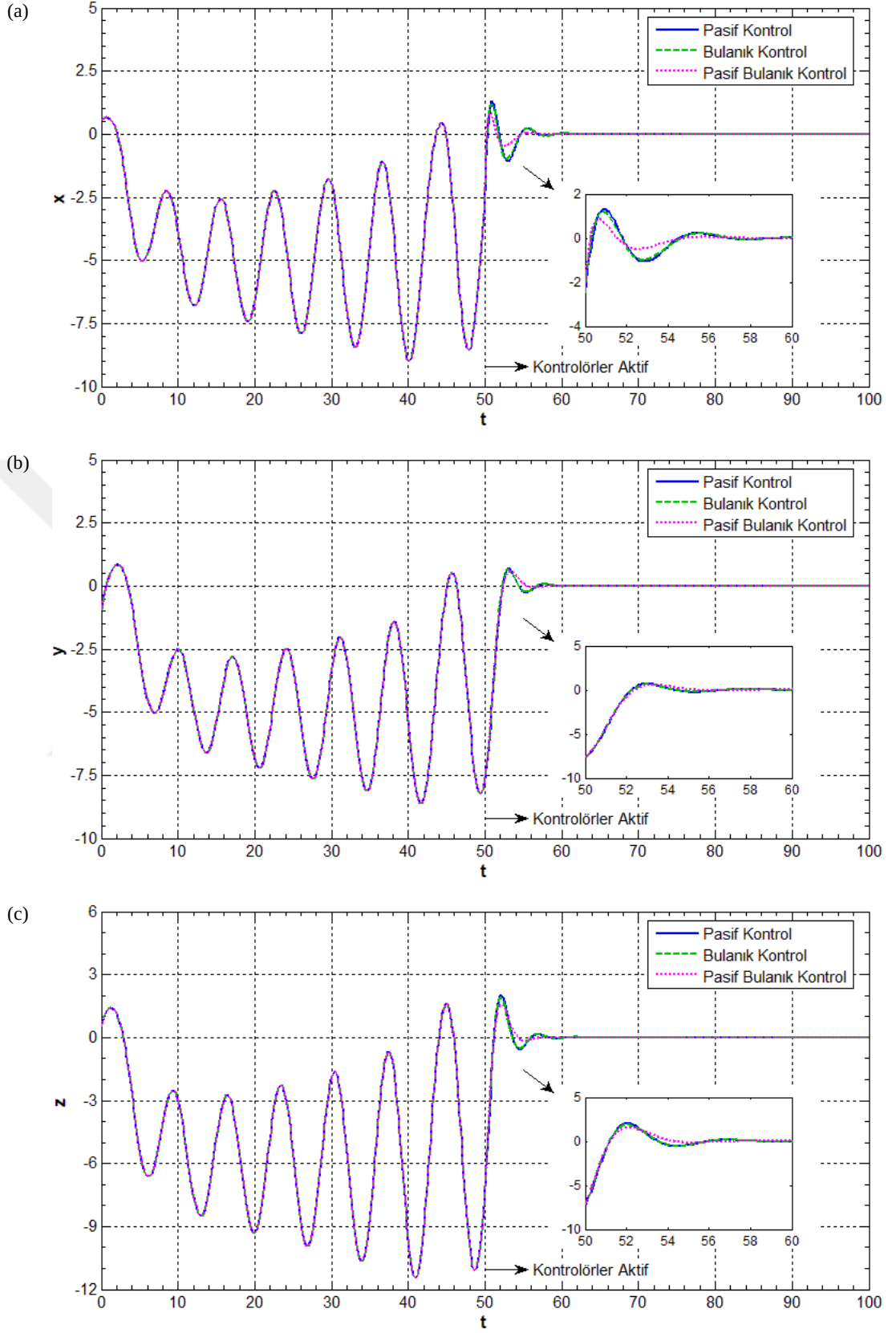
xdot = [xdot1; xdot2; xdot3];

```

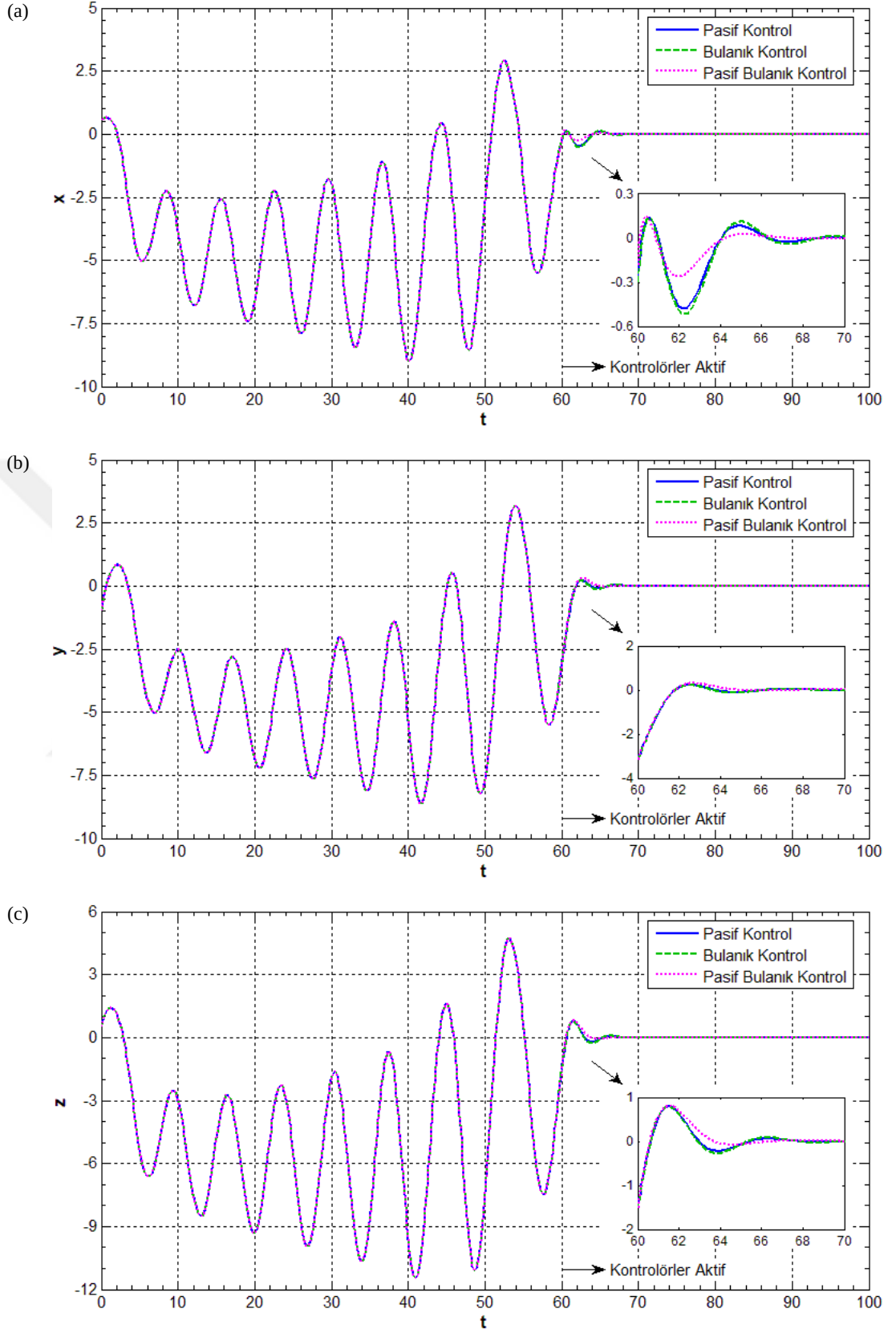
Kontrolörler $t = 40$, $t = 50$ ve $t = 60$ anlarında aktive edilmiştir ve sonuçlar sırasıyla Şekil 3.10., Şekil 3.11. ve Şekil 3.12.’de karşılaştırmalı olarak gösterilmiştir.



Şekil 3.10. Kontrolörler $t = 40$ anında aktive edildiğinde Bonhoeffer–van der Pol kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 3.11. Kontrolörler $t = 50$ anında aktive edildiğinde Bonhoeffer–van der Pol kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 3.12. Kontrolörler $t = 60$ anında aktive edildiğinde Bonhoeffer–van der Pol kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.

Şekiller 3.10.–3.12.’deki benzetim sonuçları hem pasif kontrol, hem bulanık mantık hem de pasif bulanık kontrol yöntemleri ile kaotik Bonhoeffer–van der Pol sisteminin $E(0, 0, 0)$ denge noktasına kontrolünün gerçekleştiğini göstermektedir. Bu şekillerden, pasif bulanık mantık tabanlı kontrol yönteminin pasif kontrole ve bulanık kontrole göre daha hızlı kontrolü gerçekleştirdiği görülmektedir. Kontrolörler $t = 40$ anında aktive edildiğinde, pasif bulanık kontrolör ile $t \geq 46$ civarlarında, pasif kontrolör ve bulanık kontrolör ile $t \geq 48$ civarlarında; kontrolörler $t = 50$ anında aktive edildiğinde, pasif bulanık kontrolör ile $t \geq 56$ civarlarında, pasif kontrolör ve bulanık kontrolör ile $t \geq 58$ civarlarında; kontrolörler $t = 60$ anında aktive edildiğinde ise, pasif bulanık kontrolör ile $t \geq 67$ civarlarında, pasif kontrolör ve bulanık kontrolör ile $t \geq 69$ civarlarında Bonhoeffer–van der Pol kaotik sisteminin kontrolü sağlanmıştır. Pasif kontrol yönteminin bulanıklaştırılması, kontrol zamanını hızlandırmıştır.

3.3. Colpitts Kaotik Sisteminin Kontrolü

Colpitts kaotik sisteminin kontrolü sırasıyla klasik kontrol yöntemlerinden doğrusal geri-beslemeli kontrol ile, yapay zekâ tekniklerinden sinirsel-bulanık ağlar ile ve her ikisinin birleşimi ile uygulanmış, bilgisayar simülasyonlarından elde edilen sonuçlar karşılaştırmalı grafiklerle sunulmuştur.

3.3.1. Doğrusal geri-beslemeli kontrol

Colpitts kaotik sisteminin denge noktalarına kontrolü için, denklem sistemi 2.11’e doğrusal geri-beslemeli kontrol uygulanabilir. Kontrol edilecek Colpitts sisteminin x , y ve z durum değişkenlerine sırasıyla u_{DGBK1} , u_{DGBK2} ve u_{DGBK3} kontrolörleri eklenerek yeniden aşağıdaki biçimde yazılabilir:

$$\begin{cases} \dot{x} = y - aF(z) + u_{DGBK1}, \\ \dot{y} = c - x - by - z + u_{DGBK2}, \\ \dot{z} = (y - dz)/\varepsilon + u_{DGBK3}. \end{cases} \quad (3.28)$$

u_{DGBK1} , u_{DGBK2} ve u_{DGBK3} doğrusal geri-beslemeli kontrol sinyalleri aşağıdaki gibi tanımlanır:

$$\begin{aligned} u_{DGBK1} &= -k_1(x - \bar{x}), \\ u_{DGBK2} &= -k_2(y - \bar{y}), \\ u_{DGBK3} &= -k_3(z - \bar{z}), \end{aligned} \quad (3.29)$$

burada $(\bar{x}, \bar{y}, \bar{z})$ kontrolü sağlanmış Colpitts sistemi 3.28'in bir denge noktası, k_1 , k_2 , k_3 pozitif geri-besleme kazançlarıdır. Denklem sistemi 3.29, sistem 3.28'e dâhil edildiğinde, sistem aşağıdaki hale dönüşür:

$$\begin{cases} \dot{x} = y - aF(z) - k_1(x - \bar{x}), \\ \dot{y} = c - x - by - z - k_2(y - \bar{y}), \\ \dot{z} = (y - dz) / \varepsilon - k_3(z - \bar{z}). \end{cases} \quad (3.30)$$

$F(z)$ denklemini

$$F(z) = \begin{cases} e - 1 - z, & z < e - 1, \\ 0, & z \geq e - 1, \end{cases} \quad (3.31)$$

gözönünde bulundurulursa, Colpitts kaotik sistemi 3.30'un Jacobi matrisi $z < e - 1$ için aşağıdaki gibi olur:

$$J = \begin{pmatrix} -k_1 & 1 & a \\ -1 & -(b + k_2) & -1 \\ 0 & 1/\varepsilon & -(d/\varepsilon + k_3) \end{pmatrix}. \quad (3.32)$$

Jacobi matrisi $z \geq e - 1$ için aşağıda verilmiştir:

$$J = \begin{pmatrix} -k_1 & 1 & 0 \\ -1 & -(b + k_2) & -1 \\ 0 & 1/\varepsilon & -(d/\varepsilon + k_3) \end{pmatrix}. \quad (3.33)$$

Kontrol parametresi $k_1 = k_2 = k_3 = k$ olarak kabul edildiğinde ve Colpitts kaotik sisteminin parametre değerleri $a = 30$, $b = 4/5$, $\varepsilon = 1$, $d = 2/25$ alındığında, Matris 3.32'nin karakteristik denklemi:

$$\lambda^3 + (3k + 22/25)\lambda^2 + (3k^2 + (44/25)k + 258/125)\lambda + k^3 + (22/25)k^2 + (258/125)k + 752/25 = 0 \quad (3.34)$$

olur. Matris 3.33'ün karakteristik denklemi ise:

$$\lambda^3 + (3k + 22/25)\lambda^2 + (3k^2 + (44/25)k + 258/125)\lambda + k^3 + (22/25)k^2 + (258/125)k + 2/25 = 0 \quad (3.35)$$

olur. Routh–Hurwitz kriterine göre, kontrolü sağlanacak sistemin kararlılığı, özdeğerlerle ilgili aşağıdaki şartların sağlanması ile gerçekleşir:

$$\begin{aligned} (i) & c_1 > 0, c_2 > 0, c_3 > 0, \\ (ii) & c_1 c_2 > c_3, \end{aligned} \quad (3.36)$$

burada c_1 , c_2 ve c_3 aşağıdaki eşitlikte verilen Jacobi matrisi karakteristik denklemi katsayılarıdır:

$$\lambda^3 + c_1 \lambda^2 + c_2 \lambda + c_3 = 0. \quad (3.37)$$

Kontrol parametreleri $k_1 = k_2 = k_3 = k = 1$ olarak alınır, $c_1 = 97/25$, $c_2 = 853/125$, c_3 ise Denklem 3.34'e göre $4253/125$, Denklem 3.35'e göre $503/125$ olarak bulunur. Bu değerlere göre $c_1 \times c_2 = 26,4771$ olarak hesaplanabilir ve Denklem 3.34'deki $c_3 = 34,024$ değerinden daha küçük olduğu, dolayısıyla Routh–Hurwitz kriteri (ii)'yi sağlamadığı görülmektedir.

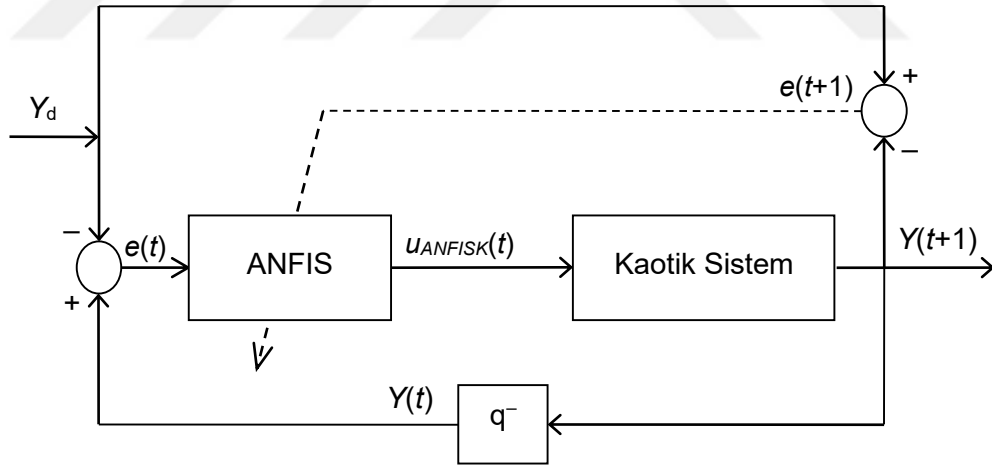
Kontrol parametreleri $k_1 = k_2 = k_3 = k = 2$ olarak alınır, $c_1 = 172/25$, $c_2 = 2198/125$, c_3 ise Denklem 3.34'e göre $5716/125$, Denklem 3.35'e göre $1966/125$ olarak bulunur. Bu değerlere göre $c_1 \times c_2 = 120,9779$ olarak hesaplanabilir ve hem Denklem

3.34'deki $c_3 = 45,728$ hem de Denklem 3.35'deki $c_3 = 15,728$ değerinden büyüktür. Buradan her iki karakteristik denklemin birden Routh–Hurwitz kriterlerini sağladığı görülmektedir. Sonuç olarak, kontrol parametreleri $k_1 = 2$, $k_2 = 2$ ve $k_3 = 2$ olarak alındığında, Colpitts sistemi 3.30 asimptotik kararlı olarak denge noktasına yakınsar. Böylece Colpitts kaotik sisteminin kontrolünün sağlanacağı kanıtlanmıştır.

3.3.2. Sinirsel-bulanık ağlar ile kontrol

Kaotik Colpitts osilatörünün kontrolünü sağlamak için sistemin x , y ve z durum değişkenlerine ANFIS tabanlı kontrolörler eklenmiştir. Uygulanan kontrol modelinin formülü Denklem 3.38'de, diyagramı ise Şekil 3.13.'te verilmiştir:

$$\begin{cases} \dot{x} = y - aF(z) + u_{ANFISK1}, \\ \dot{y} = c - x - by - z + u_{ANFISK2}, \\ \dot{z} = (y - dz) / \varepsilon + u_{ANFISK3}. \end{cases} \quad (3.38)$$



Şekil 3.13. ANFIS ile kaotik sistemlerin kontrol modeli.

Şekil 3.13.'te $Y(t)$ o andaki Colpitts kaotik sisteminin aktif değeri, $Y(t+1)$ sistemin çıktısı ya da bir sonraki değeri ve Y_d sistemin denge noktasıdır. $e(t)$ mevcut hatayı temsil etmektedir. Kontroldeki $e(t+1)$ çıktı hatası, $Y(t+1) - Y_d$ formülü ile hesaplanır. Modelde amaç, $\lim_{t \rightarrow \infty} e(t+1) \cong 0$ olacak biçimde $u_{ANFISK}(t)$ kontrol girdilerini belirlemektir. Bunun için, Colpitts sisteminin mevcut değeri ile denge noktası arasındaki fark yani hata geri-beslenerek ANFIS tabanlı kontrolöre öğretilir. Bu

eğitim yolu Şekil 3.13.'te kesikli çizgi ile gösterilmiştir. Eğitim başarılı bir şekilde tamamlandıktan sonra, aynı model eğitim işlemi yapılmadan sistemin istenilen denge noktasına kontrolü için kullanılır.

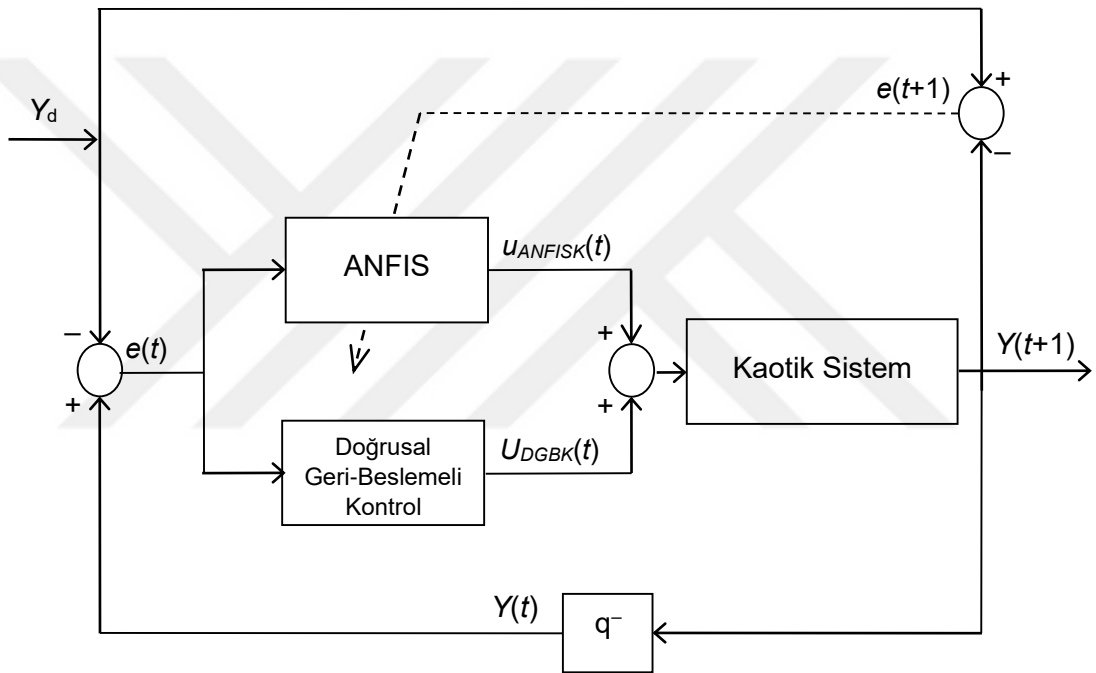
Kontrolör ANFIS'in giriş değişkenleri, $e(t)$ ile temsil edilen sistemin durum değişkenleri ile istenen nokta arasındaki hata miktarlarıdır. Çıkış değişkeni ise $u_{ANFISK}(t)$ kontrol sinyali. ANFIS kontrolörün eğitimi için Matlab programı kullanılmıştır. Bu işlem önce 'genfis1' komutu ile sağlanmaktadır. Tüm giriş değişkenleri için 5 adet üçgen ve yamuk üyelik fonksiyonuna göre daha yumuşak geçiş hatlarına sahip Gaussian (gaussmf) üyelik fonksiyonu tercih edilmiştir. Böylece, ANFIS tabanlı kontrolör 125 kural içermektedir. Çıkış değişkeni için doğrusal (linear) üyelik fonksiyonu tercih edilmiştir, diğer alternatifi ise sabit (constant) alınmasıdır. Eğitim işlemine 'anfis' komutu ile devam edilmiştir. Bu komut Sugeno sonuç çıkarım sistemine sahip ANFIS bulanık kural kümesi parametrelerinin değerlerini verilere göre ayarlamaktadır. ANFIS ağına ait parametrelerin güncellenmesinde hibrit öğrenme ya da geri yayılım algoritması kullanılır. Hibrit öğrenme algoritması en küçük kareler yöntemi ile geri yayılım yönteminin birleşimidir. Eğitim aşamasında, 'anfis' komutu hibrit öğrenme algoritması 10 iterasyon sıfır hata toleransı ile tercih edilmiştir. Bu yüzden eğitme işlemi maksimum iterasyon sayısına erişildiğinde tamamlanmıştır. Diğer parametreler için varsayılan değerler kullanılmıştır. Kontrolün başarıyla sağlanması için, eğitilmiş ANFIS kontrolörler bir döngü ile tekrardan ANFIS kontrolörlü sistem çıkışı ile denge noktası arasındaki hatanın geri beslenmesi ile eğitilmiştir. Döngü 3 iterasyon çalıştırıldığında kontrol için yeterli durum sağlanmış ve ANFIS kontrolörler sırasıyla 0,0000554039, 0,0000499057, 0,0000460294 ortalama karesel hatalara sahip olmuştur.

3.3.3. Doğrusal geri-beslemeli sinirsel-bulanık ağlar ile kontrol

Kaotik Colpitts devresinin kontrolünü sağlamak için sistemin durum değişkenlerine hem doğrusal geri-beslemeli kontrolörler hem de ANFIS tabanlı kontrolörler eklenmiştir. Uygulanan kontrol modelinin formülü Denklem 3.39'da verilmiştir:

$$\begin{cases} \dot{x} = y - aF(z) + u_{DGBK1} + u_{ANFISK1}, \\ \dot{y} = c - x - by - z + u_{DGBK2} + u_{ANFISK2}, \\ \dot{z} = (y - dz)/\varepsilon + u_{DGBK3} + u_{ANFISK3}. \end{cases} \quad (3.39)$$

Şekil 3.13.'te gösterilen ANFIS ile kontrol modeline benzer biçimde, bu sefer içinde doğrusal geri-beslemeli kontrol fonksiyonu da yer alan bir model, kaotik sistemlerin kontrolü için tasarlanmıştır. Colpitts kaotik sistemi üzerinde uygulanmıştır. Uygulanan kontrol modelinin diyagramı Şekil 3.14.'te gösterilmiştir.



Şekil 3.14. Doğrusal geri-beslemeli kontrol ve ANFIS ile tasarlanan kaotik sistemlerin kontrol modeli.

Kullanılan ANFIS'te yine tüm giriş değişkenleri için 5 adet Gaussian üyelik fonksiyonu, çıkış değişkeni için doğrusal üyelik fonksiyonu tercih edilmiştir. Böylece, ANFIS tabanlı kontrolör yine 125 kural içermektedir. Eğitim aşamasında, 'anfis' komutu hibrit öğrenme algoritması 10 iterasyon sıfır hata toleransı ile tercih edilmiştir. Diğer parametreler için ANFIS'in Matlab'daki varsayılan değerleri kullanılmıştır. Kontrolün başarıyla sağlanması için, eğitilmiş ANFIS kontrolörler bir döngü ile tekrardan doğrusal geri-beslemeli ve ANFIS kontrolörlü sistem çıkışı ile denge noktası arasındaki hatanın geri beslenmesi ile eğitilmiştir. Döngü 3 iterasyon

çalıştırıldığında kontrol için yeterli durum sağlanmış, ANFIS kontrolörler sırasıyla 0,0000501451, 0,0000511105 ve 0,0000541016 ortalama karesel hatalara sahip olmuştur.

3.3.4. Simülasyon sonuçları

Bilgisayar benzetimleri ile $x(0) = 7,02$, $y(0) = 2,06$, $z(0) = 9,09$ başlangıç şartlarına sahip Colpitts kaotik sistemi parametre değerleri $\varepsilon = 1$, $a = 30$, $b = 0,8$, $c = 20$, $d = 0,08$ ve $e = 10$ alınarak doğrusal geri-beslemeli kontrolü, ANFIS ile kontrolü ve doğrusal geri-beslemeli ANFIS ile kontrolü denetlenmiştir. Doğrusal geri-beslemeli kontrol parametre değerleri $k_1 = k_2 = k_3 = 2$ olarak alınmıştır.

Colpitts sisteminin belirtilen t_k zamanında denge noktasına kontrolünün sağlanması, doğrusal geri-beslemeli kontrol, ANFIS ile kontrol ve doğrusal geri-beslemeli ANFIS ile kontrol yöntemlerinin karşılaştırılması amacıyla Program Kodu 3.9'daki 'colpitts_k.m' dosyası oluşturulmuştur. Bu kod doğrusal geri-beslemeli kontrolü içeren 'colpittsf_dgbk', ANFIS ile kontrolü içeren 'colpittsf_anfisk' ve doğrusal geri-beslemeli ANFIS ile kontrolü içeren 'colpittsf_dgbk_anfisk' fonksiyonlarını kullanmaktadır. Program Kodları 3.9–3.12 aşağıda verilmiştir.

Program Kodu 3.9. colpitts_k.m dosyası

```
clear;

global a b c d e efs;
global tk xd yd zd k1 k2 k3;
global Anfis_f1 Anfis_f2 Anfis_f3;

t0 = 0;
tf = 50;

tk = 30;

a = 30;
b = 0.8;
c = 20;
d = 0.08;
e = 10;
efs = 1;

kk = a*(e - 1) / (a + d);
xd = c - (b*d + 1)*kk;
```



```

yd = d*kk;
zd = kk;

k1 = 2;
k2 = 2;
k3 = 2;

x0 = [7.02 2.06 9.09]';

tic
[tout, xout] = ode45('colpittsf_dgbk', [t0, tf], x0);
toc

load 'colpitts_kntrl_anfis_3.mat';

tic
[tout2, xout2] = ode45('colpittsf_anfisk', [t0, tf], x0);
toc

load 'colpitts_kntrl_dgbk_anfis_3.mat';

tic
[tout3, xout3] = ode45('colpittsf_dgbk_anfisk', [t0, tf], x0);
toc

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,1), 'b-', tout2, xout2(:,1), 'g--', tout3, xout3(:,1),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('x', 'FontSize', 12, 'FontWeight', 'bold');
legend('Geri-Beslemeli Kontrol', 'ANFIS ile Kontrol', 'Geri-Beslemeli ANFIS
ile Kontrol');

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,2), 'b-', tout2, xout2(:,2), 'g--', tout3, xout3(:,2),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);
box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('y', 'FontSize', 12, 'FontWeight', 'bold');
legend('Geri-Beslemeli Kontrol', 'ANFIS ile Kontrol', 'Geri-Beslemeli ANFIS
ile Kontrol');

figure('Color', [1 1 1], 'Position', [5.0 164.0 925.0 430.0]);
set(axes, 'FontSize', 11);
hp=plot(tout, xout(:,3), 'b-', tout2, xout2(:,3), 'g--', tout3, xout3(:,3),
'm:');
set(hp(2), 'Color', [0 0.8 0]);
set(gca, 'XMinorTick', 'on', 'YMinorTick', 'on');
set(hp, 'LineWidth', 2);

```

```

box on;
grid on;
xlabel('t', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('z', 'FontSize', 12, 'FontWeight', 'bold');
legend('Geri-Beslemeli Kontrol', 'ANFIS ile Kontrol', 'Geri-Beslemeli ANFIS
ile Kontrol');

```

Program Kodu 3.10. colpittsf_dgbk.m dosyası

```

function xdot = colpittsf_dgbk(t, x)

global a b c d e efs;
global tk xd yd zd k1 k2 k3;

if t < tk
    uDGBK1 = 0;
    uDGBK2 = 0;
    uDGBK3 = 0;
else
    uDGBK1 = -k1*(x(1) - xd);
    uDGBK2 = -k2*(x(2) - yd);
    uDGBK3 = -k3*(x(3) - zd);
end;

if x(3) < e-1
    fz = e - 1 - x(3);
else
    fz = 0;
end;

xdot1 = x(2) - a*fz + uDGBK1;
xdot2 = c - x(1) - b*x(2) - x(3) + uDGBK2;
xdot3 = (x(2) - d*x(3)) / efs + uDGBK3;

xdot = [xdot1; xdot2; xdot3];

```

Program Kodu 3.11. colpittsf_anfisk.m dosyası

```

function xdot = colpittsf_anfisk(t, x)

global a b c d e efs;
global tk xd yd zd;
global Anfis_f1 Anfis_f2 Anfis_f3;

if (t < tk)
    uANFISK1 = 0;
    uANFISK2 = 0;
    uANFISK3 = 0;
else
    uANFISK1 = evalfis([x(1)-xd x(2)-yd x(3)-zd], Anfis_f1);
    uANFISK2 = evalfis([x(1)-xd x(2)-yd x(3)-zd], Anfis_f2);
    uANFISK3 = evalfis([x(1)-xd x(2)-yd x(3)-zd], Anfis_f3);
end;

if (x(3) < e-1)
    fz = e - 1 - x(3);
else
    fz = 0;
end;

```

```

end;

xdot1 = x(2) - a*fz + uANFIS1;
xdot2 = c - x(1) - b*x(2) - x(3) + uANFIS2;
xdot3 = (x(2) - d*x(3)) / efs + uANFIS3;

xdot = [xdot1; xdot2; xdot3];

```

Program Kodu 3.12. colpittsf_dgbk_anfisk.m dosyası

```

function xdot = colpittsf_dgbk_anfisk(t, x)

global a b c d e efs;
global tk xd yd zd k1 k2 k3;
global Anfis_f1 Anfis_f2 Anfis_f3;

if t < tk
    uDGBK1 = 0;
    uDGBK2 = 0;
    uDGBK3 = 0;
    uANFISK1 = 0;
    uANFISK2 = 0;
    uANFISK3 = 0;
else
    uDGBK1 = -k1*(x(1) - xd);
    uDGBK2 = -k2*(x(2) - yd);
    uDGBK3 = -k3*(x(3) - zd);
    uANFISK1 = evalfis([x(1)-xd x(2)-yd x(3)-zd], Anfis_f1);
    uANFISK2 = evalfis([x(1)-xd x(2)-yd x(3)-zd], Anfis_f2);
    uANFISK3 = evalfis([x(1)-xd x(2)-yd x(3)-zd], Anfis_f3);
end;

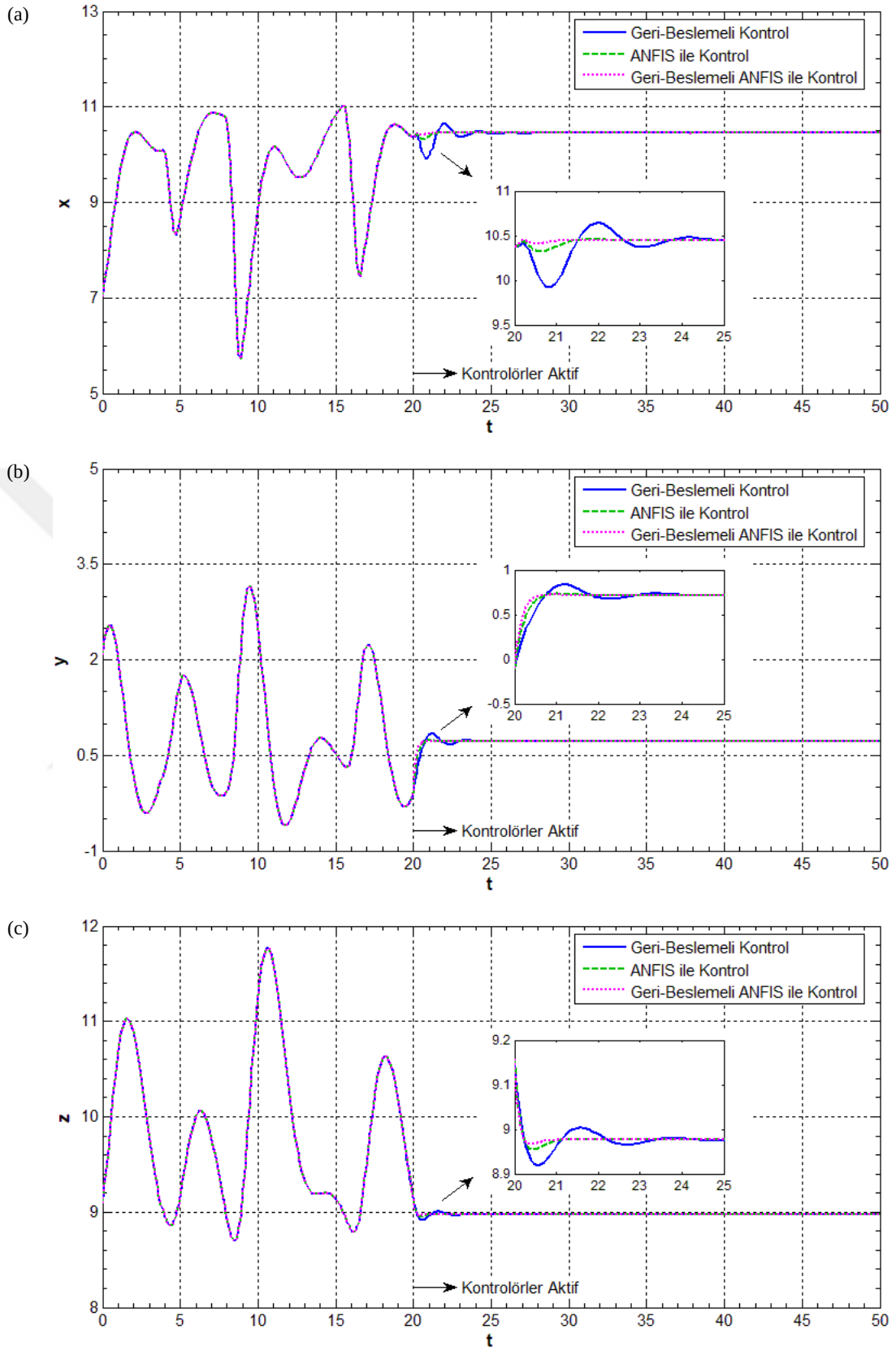
if x(3) < e-1
    fz = e - 1 - x(3);
else
    fz = 0;
end;

xdot1 = x(2) - a*fz + uDGBK1 + uANFIS1;
xdot2 = c - x(1) - b*x(2) - x(3) + uDGBK2 + uANFIS2;
xdot3 = (x(2) - d*x(3)) / efs + uDGBK3 + uANFIS3;

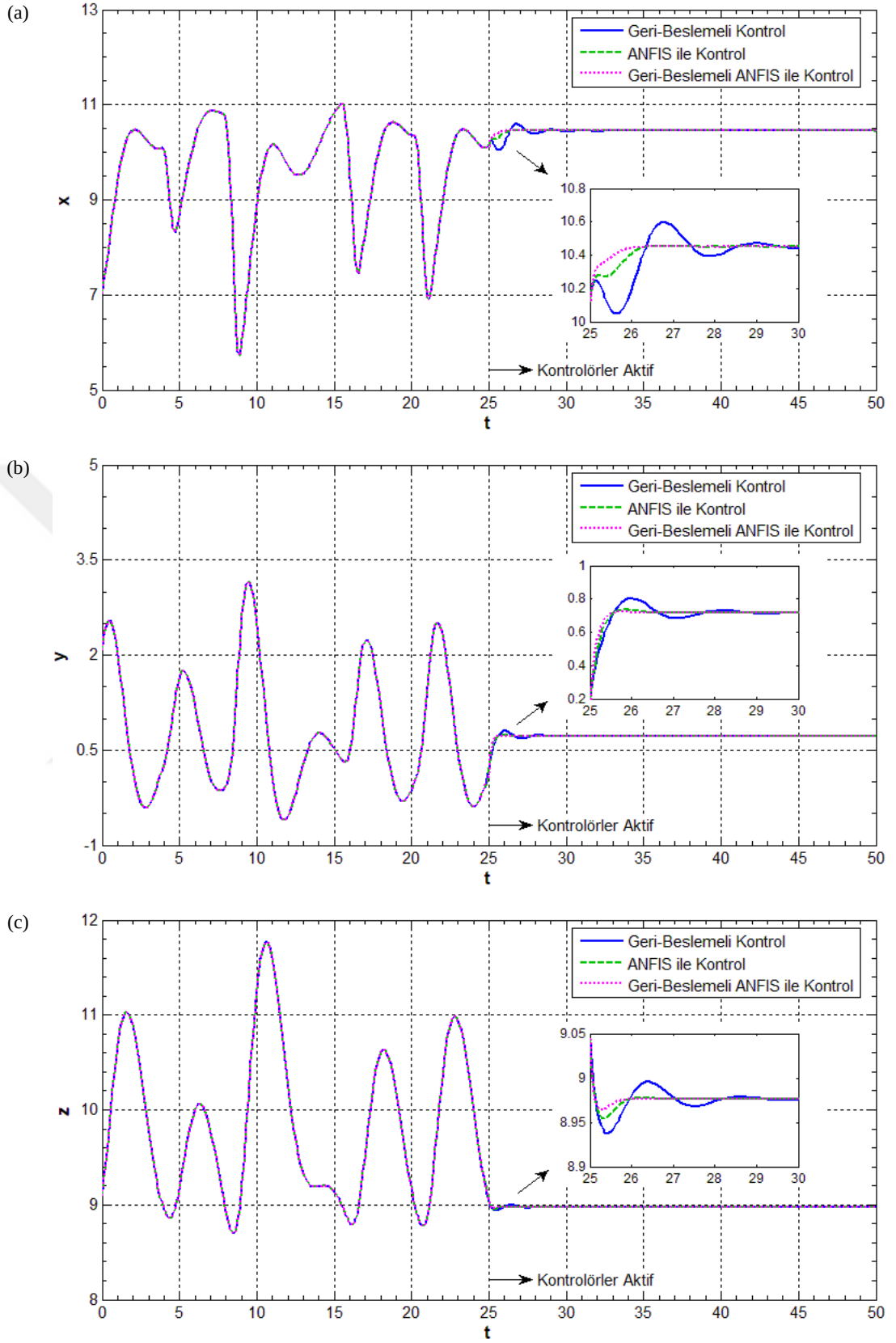
xdot = [xdot1; xdot2; xdot3];

```

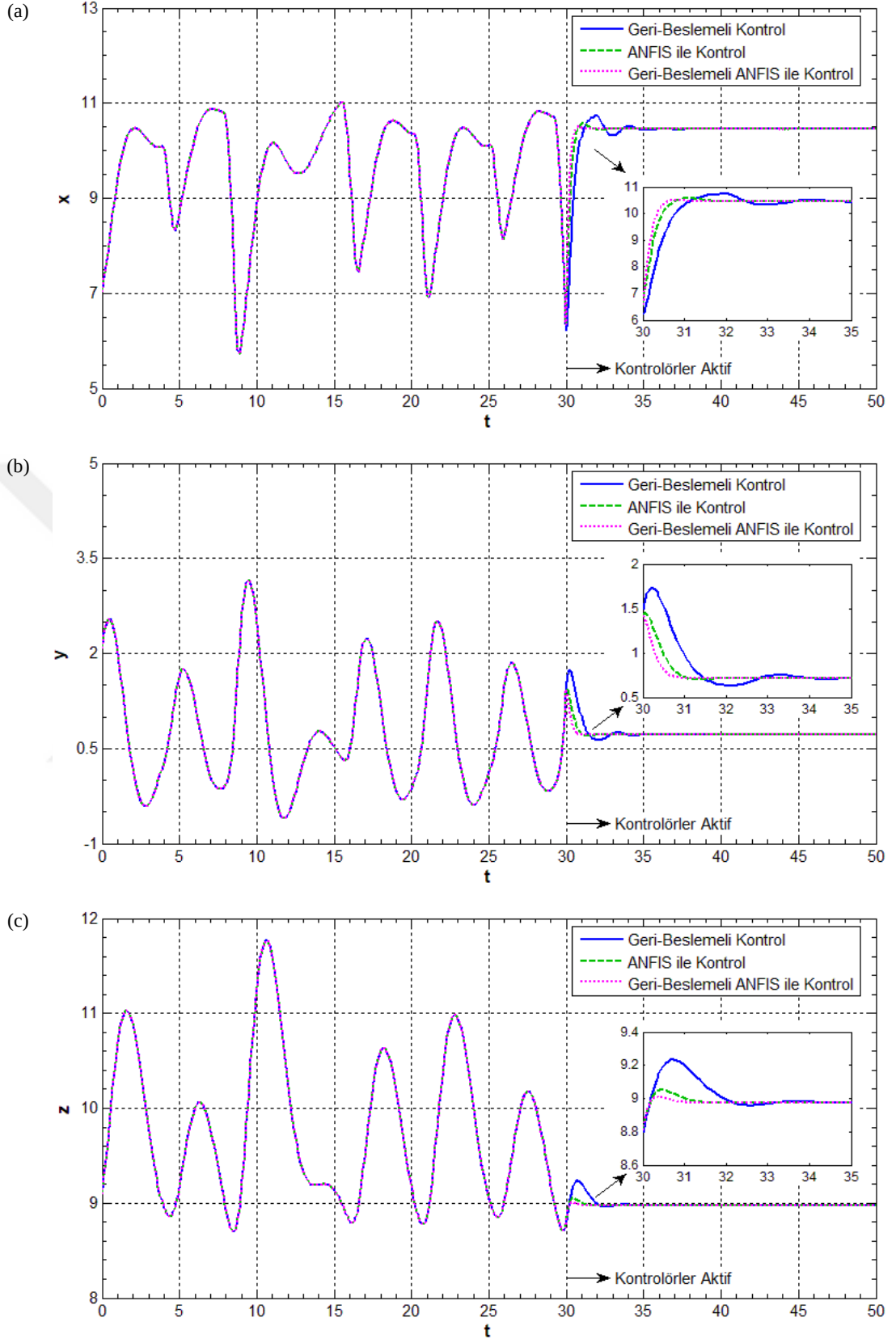
Kontrolörler $t = 20$, $t = 25$ ve $t = 30$ anlarında aktive edilmiştir ve sonuçlar sırasıyla Şekil 3.15., Şekil 3.16. ve Şekil 3.17.’de karşılaştırmalı olarak gösterilmiştir.



Şekil 3.15. Kontrolörler $t = 20$ anında aktive edildiğinde Colpitts kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 3.16. Kontrolörler $t = 25$ anında aktive edildiğinde Colpitts kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.



Şekil 3.17. Kontrolörler $t = 30$ anında aktive edildiğinde Colpitts kaotik sistemi kontrol sonuçları (a) x sinyali, (b) y sinyali, (c) z sinyali.

Şekil 3.15.–3.17.’deki benzetim sonuçları hem doğrusal geri-beslemeli kontrol hem klasik ANFIS hem de doğrusal geri-beslemeli ANFIS ile kontrol yöntemleri ile kaotik Colpitts sisteminin $E(10,449, 0,718, 8,976)$ denge noktasına kontrolünün gerçekleştiğini göstermektedir. Bu grafiklere göre, kontrolörler $t = 20$ anında aktive edildiğinde, kontrol ilk olarak $t \geq 21$ ’de, kontrolörler $t = 25$ anında aktive edildiğinde, kontrol ilk olarak $t \geq 26$ ’da, kontrolörler $t = 30$ anında aktive edildiğinde ise, kontrol ilk olarak $t \geq 31$ ’de hep doğrusal geri-beslemeli ANFIS ile kontrol yöntemiyle sağlanmıştır. ANFIS ile kontrol de benzer sonuçlar üretmesine rağmen sıfır hatayı biraz daha geç sağlamaktadır. Doğrusal geri-beslemeli kontrol ile denge noktasına ulaşmak yaklaşık 10 birim zamanı bulmaktadır. Dolayısıyla, simülasyon sonuçları göstermiştir ki doğrusal geri-besleme tabanlı ANFIS kontrol, doğrusal geri-beslemeli kontrol ve klasik ANFIS ile kontrol yöntemlerinden daha efektif kaos kontrol sonuçları üretmektedir.

BÖLÜM 4. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, elektronik tabanlı kaotik sistemlerin daha etkin kontrolü için klasik kontrol yöntemlerinin yanısıra yapay zekâ metodolojilerinden faydalanılması da amaçlanmıştır. Yaygın kullanılan Chua, Bonhoeffer–van der Pol ve Colpitts kaotik devreleri kontrol yöntemlerinin uygulanması için tercih edilmiştir. Klasik kaos kontrol yöntemlerinden kayma kipli kontrol, pasif kontrol ve geri-beslemeli kontrol yöntemleri ile kaotik sistemlerin denge noktasında doğrusallaşması sağlanmıştır. Yapay zekâ metodolojilerinden ise kaos kontrolünde yapısı uygun olduğu için daha çok tercih edilen ve daha iyi sonuçlar üreten YSA, bulanık mantık ve sinirsel-bulanık ağlar kullanılmıştır.

Bu tezde, ilk önce Chua kaotik sistemin kontrolü, kayma kipli kontrol ve YSA tabanlı kontrol yöntemleri ile uygulanmıştır. Ayrıca, yeni bir yaklaşım olarak bu iki yöntemin bir arada kullanımı ile kaosu kontrolü önerilmiş ve uygulanmıştır.

Bonhoeffer–van der Pol kaotik sistemin kontrolü ise, pasif kontrol ve bulanık mantık tabanlı kontrol yöntemleri ile uygulanmıştır. Ayrıca, yeni bir yaklaşım olarak pasif bulanık mantık kullanımı ile kaosu kontrolü önerilmiş ve uygulanmıştır.

Son olarak Colpitts kaotik sistemin kontrolü, doğrusal geri-beslemeli kontrol ve ANFIS tabanlı kontrol yöntemleri ile uygulanmıştır. Ayrıca, yeni bir yaklaşım olarak bu iki yöntemin bir arada kullanımı ile kaosu kontrolü önerilmiş ve uygulanmıştır.

Kayma kipli kontrol, pasif kontrol ve doğrusal geri-beslemeli kontrol kaotik sistemlerin kontrolünde başvuru ve sistemin denge noktasına yakınsamasını etkin hızlarda gerçekleştiren klasik kontrol yöntemleridir.

Chua kaotik sisteminin kontrolü için, Levenberg–Marquardt öğrenme algoritmasını kullanan, çok katmanlı, ileri beslemeli YSA tasarlanmıştır. YSA kullanılmasının diğer yapay zekâ yöntemlerine göre üstünlüğü, daha az sayıda ve işlemci tarafından daha hızlı gerçekleştirilen toplama, çarpma gibi işlemlerle hesaplama yaparak, kaosun kontrolünü sağlayabilmesidir. YSA'nın eğitimde, minimum hataya daha az iterasyonla ulaşıldığı dolayısıyla daha iyi sonuçlar ürettiği için Levenberg–Marquardt algoritması tercih edilmiştir. YSA kullanımı, öğrenme başarısından dolayı daha kapsamlı ve etkin kaos kontrol çözümlerinin elde edilmesini de sağlamaktadır.

Netice olarak, Chua kaotik sisteminin kontrolü kayma kipli kontrol, YSA ile kontrol ve her iki yöntemin bir arada kullanımı ile uygulanmıştır, Matlab ortamında sonuçlar birbiri ile karşılaştırılmıştır. Yapılan incelemelerde önerilen kayma kipli YSA ile kontrol yönteminin sistemi referans noktasına daha hızlı ulaştırdığı görülmüştür.

Sonraki çalışmalarda, önerilen bu ikili zeki model, klasik kontrol algoritması değiştirilerek uygulanabilir. Ayrıca, yapılan literatür araştırmasında, kaotik sistemlerin PNN, GRNN, SOM gibi bazı YSA modelleri ile gerçekleştirilmesinin incelenmediği fark edilmiştir. Bu konularda da araştırmalar yapılabilir.

Bonhoeffer–van der Pol kaotik sisteminin kontrolü için, bulanık mantık tabanlı kontrolörler tasarlanmıştır. Sistemin referans noktasına uzaklıkları bulanık kontrole giriş olarak verilmiş, girişler üçer adet çan biçimli üyelik fonksiyonları ile bulanıklaştırılmış, gerekli olan kontrol davranışları mantık çerçevesinde “eğer-ise” kuralları ile oluşturulmuş, Mamdani sonuç çıkarım mekanizması kullanılmış ve durulaştırıcı olarak ağırlık merkezi yöntemi tercih edilmiştir. Bulanık mantık esnek modellemeye olanak tanımaktadır. Bilgisayarın da yardımıyla, insan düşünce sistemine yakın bir anlayış ile kaosun kontrolü sağlanmıştır. Direkt olarak bulanık kontrolör ile sistemin kontrolünün yanısıra, klasik kontrol yöntemlerinden biri olan pasif kontrolörün bulanıklaştırılması ile de uygulanmıştır. Literatürde bulanık kayma kipli kontrolörlere rastlanırken, pasif kontrol yönteminin bulanıklaştırılması ilk defa gerçekleştirilmiştir.

Netice olarak, Bonhoeffer–van der Pol kaotik sisteminin kontrolü pasif kontrol, bulanık mantık ile kontrol ve pasif kontrolörün bulanıklaştırılması ile uygulanmıştır, Matlab ortamında sonuçlar birbiri ile karşılaştırılmıştır. Yapılan incelemelerde pasif kontrolün ve bulanık mantık ile kontrolün referans noktaya ulaştırma süresinin pasif bulanık mantık ile kontrolden daha uzun olduğu gözlemlenmiştir.

Sonraki çalışmalarda, önerilen pasif bulanık kontrol yöntemi ile kaotik sistemlerin senkronizasyonu ve anti-senkronizasyonu gerçekleştirilebilir. Senkronizasyon ve anti-senkronizasyon sonuçlarının daha hızlı başarılması güvenli haberleşme uygulamalarında avantaj sağlayacaktır.

Colpitts kaotik sisteminin kontrolü için, bir diğer makine öğrenme algoritması olan ANFIS tabanlı kontrolörler tasarlanmıştır. ANFIS, iç mimari olarak bulanık mantık kullanan, parametre değerlerinin YSA ile ayarlandığı, sinirsel-bulanık ağlardır. YSA’ndakine benzer modeller ile kaosun kontrolü uygulanmıştır.

Elektronik devre gibi sistemlerde, gürültü adı verilen bazı belirsiz küçük değerler de dışarıdan sisteme dâhil olurlar. Sistem kaotikse etkisi büyük olur ve klasik kontrol yöntemlerinde problemlere yol açabilir. Makine öğrenmesi içeren YSA, ANFIS gibi yapay zekâ teknikleri ile kontrol, formüllerle değil, verilerle çalıştığından gürültü toleransına sahip olurlar ve bu da ayrı bir avantajdır.

Netice olarak, Colpitts kaotik sisteminin kontrolü doğrusal geri-beslemeli kontrol, ANFIS ile kontrol ve her iki yöntemin bir arada kullanımı ile uygulanmıştır, Matlab ortamında sonuçlar birbiri ile karşılaştırılmıştır. Önerilen doğrusal geri-beslemeli ANFIS ile kontrolün diğer iki yöntemden daha iyi sonuçlar verdiği görülmüştür.

ANFIS ile kontrol üzerine literatürde az sayıda çalışma karşımıza çıkmaktadır. Kaotik sistemlerin kontrolünde iyi sonuçlar ürettiği tespit edilmiştir. Daha yaygın kullanılması araştırmacılara önerilmektedir.

KAYNAKLAR

- Abadi, A. S. A., Balochian, S. 2017. Chaos control of the power system via sliding mode based on fuzzy supervisor. *International Journal of Intelligent Computing and Cybernetics*, 10(1): 68–79.
- Aguilar-Mejia, O., Tapia-Olvera, R., Valderrabano-Gonzalez, A., Cambero, I. R. 2016. Adaptive neural network control of chaos in permanent magnet synchronous motor. *Intelligent Automation and Soft Computing*, 22(3): 499–507.
- Alasty, A., Salarieh, H. 2005. Controlling the chaos using fuzzy estimation of OGY and Pyragas controllers. *Chaos, Solitons & Fractals*, 26(2): 379–392.
- Al-Khawaja, S. 2011. Synchronization of chaos in two mesoscopic shunted resistive–capacitive–inductive Josephson junctions by means of a nonlinear van der Pol oscillator. *Physica Scripta*, 83(2): 025006.
- Alrifai, M. T., Zribi, M. 2018. Sliding mode control of chaos in a single machine connected to an infinite bus power system. *Mathematical Problems in Engineering*, 2018: 2703684.
- Alsing, P. M., Gavrielies A., Kovanis, V. 1994. Using neural networks for controlling chaos. *Physical Review E*, 49(2): 1225–1232.
- Alvarez–Ramirez, J. 1994. Nonlinear feedback for controlling the Lorenz equation. *Physical Review E*, 50(3): 2339–2342.
- Bagheri, A., Moghaddam, J. J. 2009. Decoupled adaptive neuro-fuzzy (DANF) sliding mode control system for a Lorenz chaotic problem. *Expert Systems with Applications*, 36: 6062–6068.
- Baker, G. L., Gollub, J. P. 1990. *Chaotic Dynamics: An Introduction*. Cambridge University Press, Cambridge, İngiltere.
- Baykal, N., Beyan, T. 2004. *Bulanık Mantık İlke ve Temelleri*. Bıçaklar Kitabevi, Ankara, Türkiye.
- Bolat, S., Kalenderli, Ö. 2003. Levenberg-Marquardt algoritması kullanılan yapay sinir ağı ile elektrot biçim optimizasyonu. *International XII. Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN 2003*, Çanakkale, Türkiye.
- Boukabou, A., Mansouri, N. 2005. Neural predictive control of unknown chaotic systems. *Nonlinear Analysis: Modelling and Control*, 10(2): 95–106.

- Cenys, A., Tamasevicius, A., Baziliauskas, A., Krivickas, R., Lindberg, E. 2003. Hyperchaos in coupled Colpitts oscillators. *Chaos, Solitons & Fractals*, 17(2–3): 349–353.
- Chang, J. F., Hung, M. L., Yang, Y. S., Liao, T. L., Yan, J. J. 2008a. Controlling chaos of the family of Rössler systems using sliding mode control. *Chaos, Solitons & Fractals*, 37(2): 609–622.
- Chang, W. J., Ku, C. C., Huang, P. H. 2008b. Fuzzy controller design with passivity performance for single machine infinite bus power systems. *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, Nanjing, Çin Halk Cumhuriyeti, 2217–2222.
- Chen, C. S. 2009. Quadratic optimal neural fuzzy control for synchronization of uncertain chaotic systems. *Expert Systems with Applications*, 36(9): 11827–11835.
- Chen, C. S., Chen, H. H. 2009. Robust adaptive neural-fuzzy-network control for the synchronization of uncertain chaotic systems. *Nonlinear Analysis: Real World Applications*, 10: 1466–1479.
- Chen, M. Y., Han, Z. Z. 2003. Controlling and synchronizing chaotic Genesio system via nonlinear feedback control. *Chaos, Solitons & Fractals*, 17(4): 709–716.
- Chua, L. O., Komuro, M., Matsumoto, T. 1986. The double scroll family. *IEEE Transactions on Circuits and Systems*, 33(11): 1072–1118.
- Colpitts, E. H. 1918. Oscillation generator. U.S. Patent, No: 1624537.
- Cox, E. 1992. Fuzzy fundamentals. *IEEE Spectrum*, 58–61.
- Davidon, W. C. 1959. Variable metric method for minimization. Technical Report, Argonne National Laboratories, Ill., ANL-5990.
- Demirkol, A. Ş. 2007. Kaotik osilatör girişli ADC tabanlı rastgele sayı üretici. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Bölümü, Yüksek Lisans Tezi.
- Efe, M. Ö., Kaynak, O., Yu, X., Wilamowski, B. M. 2000. Gauss merkezci taban fonksiyonlu sinir ağları ile doğrusal olmayan sistemlerin kayma kipli denetimi. Otomatik Kontrol Ulusal Toplantısı, TOK'2000, Hacettepe Üniversitesi, Ankara, Türkiye, 29–36.
- Emelyanov, S. V. 1967. Variable Structure Control Systems. Nauka, Moskova, Rusya.
- Emiroğlu, S., Uyaroğlu, Y. 2010. Control of Rabinovich chaotic system based on passive control. *Scientific Research and Essays*, 5(21): 3298–3305.
- Funke, M., Herrmann, M., Der, R. 1997. Controlling low-dimensional chaos: Determination and stabilization of unstable periodic orbits by Kohonen neural nets. *International Journal of Adaptive Control and Signal Processing*, 11(6): 489–499.

- Gambino, G., Lombardo, M. C., Sammartino, M. 2006. Global linear feedback control for the generalized Lorenz system. *Chaos, Solitons & Fractals*, 29(4): 829–837.
- Gao, S. G., Dong, H. R., Ning, B. 2015. Neural adaptive control of uncertain chaotic systems with input and output saturation. *Nonlinear Dynamics*, 80(1–2): 375–385.
- Ge, S. S., Wang, C. 2002. Uncertain chaotic system control via adaptive neural design. *International Journal of Bifurcation and Chaos*, 12(5): 1097–1109.
- Ginarsa, I. M., Soeprijanto, A., Purnomo, M. H. 2009. Controlling chaos using ANFIS-based Composite Controller (ANFIS-CC) in power systems. *International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering, ICICI-BME, Bandung, Endonezya*, 1–5.
- Glawer, J. S., Munighan, J., 1997. Designing fuzzy controllers from a variable structures standpoint. *IEEE Transactions on Fuzzy Systems*, 5(1): 138–144.
- Gleick, J. 1995. *Kaos (Yeni Bir Bilim Teorisi)*. Çeviren: F. Üçcan, Tübitak Yayınevi, Ankara, Türkiye.
- Gökbulut, M. 1998. Fırçasız doğru akım motorlarının yapay sinir ağları ile uyarlamalı denetimi. *Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi*.
- Gökçe, K., Uyaroğlu, Y. 2017. An adaptive neural network control scheme for stabilizing chaos to the stable fixed point. *Information Technology and Control*, 46(2): 219–227.
- Guo, H., Lin, S., Liu, J. 2006. A radial basis function sliding mode controller for chaotic Lorenz system. *Physics Letters A*, 351: 257–261.
- Gülbağ, A. 2006. Yapay sinir ağı ve bulanık mantık tabanlı algoritmalar ile uçucu organik bileşiklerin miktarsal tayini. *Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Bölümü, Doktora Tezi*.
- Hanbay, D., Türkoğlu, İ., Demir, Y. 2007. Chua devresinin yapay sinir ağı ile modellenmesi. *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 19(1): 67–72.
- Harb, A. M., Smadi, I. A. 2004. On fuzzy control of chaotic systems. *Journal of Vibration and Control*, 10: 979–993.
- Haykin, S. 2005. *Neural Networks, A Comprehensive Foundation*. Second Edition. Upper Saddle River, New Jersey, Prentice Hall, Amerika Birleşik Devletleri.
- Hernandez, C., Martinez, A., Mingo, L. F., Castellanos, J. 2001. Controlling Lorenz chaos with neural networks. *Advances in Scientific Computing, Computational Intelligence and Applications*, 302–309.
- Hu, J., Qiu, Y., Lu, H. 2016. Adaptive robust nonlinear feedback control of chaos in PMSM system with modeling uncertainty. *Applied Mathematical Modelling*, 40(19–20): 8265–8275.

- Ishikawa, Y., Masukake, Y., Ishida, Y. 2007. Control of chaotic dynamical systems using RBF networks. *World Academy of Science, Engineering and Technology*, 25: 175–178.
- Jang, J. S. R., 1993. ANFIS: Adaptive network based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3): 665–685.
- Jiang, Z. P., 2002. Advanced feedback control of the chaotic Duffing equation. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, 49(2): 244–249.
- Kemih, K., Filali, S., Benslama, M., Kimouche, M. 2006. Passivity-based control of chaotic Lu system. *International Journal of Innovative Computing, Information and Control*, 2(2): 331–337.
- Kennedy, M. P. 1994. Chaos in the Colpitts oscillator. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 41(11): 771–774.
- Kılıç, R. 1996. Elektronik devrelerdeki kaos olayının bilgisayar simülasyonları ile incelenmesi. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Anabilim Dalı, Yüksek Lisans Tezi.
- Kim, K. B., Park, J. B., Choi, Y. H., Chen G. 2000. Control of chaotic dynamical systems using radial basis function network approximators. *Information Sciences*, 130: 165–183.
- Kiss, I. Z., Gaspar, V. 2000. Controlling chaos with artificial neural network: Numerical studies and experiments. *The Journal of Physical Chemistry A*, 104: 8033–8037.
- Koç, M. L. 2002. Taş dolgu dalgakıranların yapay sinir ağları, bulanık mantık sistemleri ve genetik algoritma ile ön tasarımı ve güvenilirlik analizi. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi.
- Kuntanapreeda, S. 2008. An observer-based neural network controller for chaotic Lorenz system. *Advances in Computation and Intelligence, Lecture Notes in Computer Science*, 5370: 608–617.
- Küçükkefe, Y., Kaypmaz, A. 2010. Bir güç sisteminde Hopf çatallanmaları ve salınımların sönümlendirilmesi. *İTÜ Dergisi/D Mühendislik*, 9(1): 15–24.
- Kyprianidis, I. M., Papachristou, V., Stouboulos, I. N., Volos, C. K. 2012. Dynamics of coupled chaotic Bonhoeffer – van der Pol oscillators. *WSEAS Transactions on Systems*, 9(11): 516–526.
- Lai, B. C., He, J. J., 2018. Dynamic analysis, circuit implementation and passive control of a novel four-dimensional chaotic system with multiscroll attractor and multiple coexisting attractors. *Pramana – Journal of Physics*, 90(3): 33.
- Lee, C. C. 1990. Fuzzy logic in control systems and fuzzy logic controller. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2): 404–434.
- Levenberg, K. 1944. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2: 164–168.
- Li, D. J. 2016. Adaptive neural stabilization control for unified chaotic systems with full state constraints. *Journal of Vibration and Control*, 22(1): 121–128.

- Li, H. Q., Liao, X. F., Lei, X. Y. 2012. Two fuzzy control schemes for Lorenz-Stenflo chaotic system. *Journal of Vibration and Control*, 18(11): 1675–1682.
- Li, M., Liu, C. X. 2010. Sliding mode control of a new chaotic system. *Chinese Physics B*, 19(10): 100504.
- Li, Z., Park, J. B., Chen, G., Joo, Y. H., Choi, Y. H. 2002. Generating chaos via feedback control from a stable TS fuzzy system through a sinusoidal nonlinearity. *International Journal of Bifurcation and Chaos*, 12(10): 2283–2291.
- Li, C. L., Tong, Y. N. 2013. Adaptive control and synchronization of a fractional-order chaotic system. *Pramana – Journal of Physics*, 80(4): 583–592.
- Li, D., Wang, Z. 2003. Control of unknown chaotic systems based on neural predictive control. *Communications in Theoretical Physics*, 40(4): 439–442.
- Li, G. H., Zhou, S. P., Yang, K. 2007. Controlling chaos in Colpitts oscillator. *Chaos, Solitons & Fractals*, 33(2): 582–587.
- Liao, X., Yu, P. 2006. Chaos control for the family of Rössler systems using feedback controllers. *Chaos, Solitons & Fractals*, 29(1): 91–107.
- Lin, C. T., Lee, C. S. G. 1994. Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems. *IEEE Transactions on Fuzzy System*, 2(1): 46–63.
- Liu, D., Liu, H. 2006. Chaos control in Lorenz systems based on adaptive inverse control of support vector machines. *International Symposium on Intelligent Control*, IEEE, Munich, Almany, 2024–2029.
- Lorenz, E. N. 1963. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2): 130–141.
- Luo, S. H., Li, S. B., Tajaddodianfar, F. 2018. Adaptive chaos control of the fractional-order arch MEMS resonator. *Nonlinear Dynamics*, 91(1): 539–547.
- Mahmoud, E. E., Abualnaja, K. M. 2013. Comments on "Passive and impulsive synchronization of a new four-dimensional chaotic system". *International Journal of Scientific & Engineering Research*, 4(11): 1618–1620.
- Mahmoud, G. M., Arafa, A. A., Abed-Elhameed, T. M., Mahmoud, E. E. 2017. Chaos control of integer and fractional orders of chaotic Burke-Shaw system using time delayed feedback control. *Chaos, Solitons & Fractals*, 104: 680–692.
- Mahmoud, G. M., Mahmoud, E. E., Arafa, A. A. 2013. Passive control of n-dimensional chaotic complex nonlinear systems. *Journal of Vibration and Control*, 19(7): 1061–1071.
- Marquardt, D. 1963. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2): 431–441.
- Mehran, Y. Z. 2011. Fuzzy OGY chaos control approach and its application in Chua circuit. *5th International Conference on Bioinformatics and Biomedical Engineering*, Wuhan, Çin Halk Cumhuriyeti.

- Mei, S., Lixin, T., Qiahg, J. 2007. Controlling chaos in the energy resource system by time-delayed feedback. 6th International Conference on Management, Wuhan, Çin Halk Cumhuriyeti, 1714–1719.
- Namajhas, A., Tamaievitius, A. 1995. Modified Wien-bridge oscillator for chaos. *Electronic Letters*, 31(5): 335–336.
- Ngia, S. H. 2000. Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm. *IEEE Transactions on Signal Processing*, 48: 1915–1927.
- Nishiuchi, Y., Ueta, T., Kawakami, H. 2006. Stable torus and its bifurcation phenomena in a simple three-dimensional autonomous circuit. *Chaos, Solitons & Fractals*, 27(4): 941–951.
- Njah, A. N., Vincent, U. E. 2009. Synchronization and anti-synchronization of chaos in an extended Bonhöffer-van der Pol oscillator using active control. *Journal of Sound and Vibration*, 319(1–2): 41–49.
- Oğuz, M. 2001. Yalıtkan maddelerde elektriksel delinme dayanımının yapay sinir ağları ile belirlenmesi. İstanbul Teknik Üniversitesi, İTÜ, Yüksek Lisans Tezi.
- Ojo, K. S., Njah, A. N., Ogunjo, S. T. 2013. Comparison of backstepping and modified active control in projective synchronization of chaos in an extended Bonhöffer-van der Pol oscillator. *Pramana – Journal of Physics*, 80(5): 825–835.
- Otawara, K., Fan, L. T. 1995. Controlling chaos with an artificial neural network. *International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, Yokohama, Japonya, Proceedings of 1995 IEEE International Conference on Fuzzy Systems, 4: 1943–1948.
- Ott, E., Grebogi, C., Yorke, J. A. 1990. Controlling chaos. *Physical Review Letters*, 64(11): 1196–1199.
- Özçalık, H. R., Uygur, A. F. 2003. Dinamik sistemlerin uyumlu sinirsel-bulanık ağ yapısına dayalı etkin modellenmesi. *KSÜ Fen ve Mühendislik Dergisi*, 6(1): 36–46.
- Öztemel, E. 2003. Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul, Türkiye.
- Pan, L., Zhou, W. N., Fang, J. A., Li, D. Q. 2010. Analysis of linear and adaptive feedback synchronization in a new unified chaotic system. *International Journal of Adaptive Control and Signal Processing*, 24(8): 708–716.
- Pehlivan, İ., Uyaroğlu, Y., Yalçın, M. A., Ferikoğlu, A. 2007. Sprott_94_A kaotik sisteminin senkronizasyonu ve bilgi gizlemede kullanılması. II. Information Security and Cryptology Conference with International Participation, ISCTURKEY, Ankara, Türkiye, 230–237.
- Poznyak, A. S., Yu, W., Sanchez, E. N. 1999. Identification and control of unknown chaotic systems via dynamic neural networks. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 46(12): 1491–1495.

- Precup, R. E., Tomescu, M. L., Dragos, C. A. 2014. Stabilization of Rossler chaotic dynamical system using fuzzy logic control algorithm. *International Journal of General Systems*, 43(5): 413–433.
- Premalatha, L., Vanajaranjan, P. 2008. Control of chaos in nonlinear switching circuits by selection of optimal system parameters using genetic algorithm. *IEEE International Conference on Industrial Technology, ICIT, Chengdu, Çin Halk Cumhuriyeti*, 1461–1466.
- Pyragas, K. 1992. Continuous control of chaos by self-controlling feedback. *Physics Letters A*, 170(6): 421–428.
- Qi, D., Zhao, G., Song, Y. 2004. Passive control of Chen chaotic system. *5th World Congress on Intelligent Control and Automation, WCICA, Hangzhou, Çin Halk Cumhuriyeti*, 1284–1286.
- Ramesh, M., Narayanan, S. 2001. Chaos control of Bonhoeffer-van der Pol oscillator using neural networks. *Chaos, Solitons and Fractals*, 12: 2395–2405.
- Sadeghpour, M., Khodabakhsh, M., Salarieh, H. 2012. Intelligent control of chaos using linear feedback controller and neural network identifier. *Communication Nonlinear Science and Numerical Simulation*, 17: 4731–4739.
- Sağıroğlu, Ş., Beşdok, E., Erler, M. 2003. *Mühendislikte Yapay Zeka Uygulamaları – I. Ufuk Kitap Kirtasiye-Yayıncılık, Kayseri, Türkiye.*
- Saini, S., Saini, J. S. 2014. GA optimized time delayed feedback control of chaos in a memristor based chaotic circuit. *IEEE Symposium Series on Computational Intelligence for Engineering Solutions, Orlando, Florida, Amerika Birleşik Devletleri*, 74–80.
- Sanchez, E. N., Ricalde, L. J. 2003. Chaos control and synchronization, with input saturation, via recurrent neural networks. *Neural Networks*, 16(5–6): 711–717.
- Sarailoo, M., Rahmani, Z., Rezaie, B. 2012. Fuzzy sliding mode control for hyper chaotic Chen system. *Advances in Electrical and Computer Engineering*, 12(1): 85–90.
- Setoudeh, F., Dousti, M. 2014. Chaos behavior of voltage control oscillator based on Colpitts oscillator. *Control Engineering and Applied Informatics*, 16(4): 91–98.
- Shen L., Wang, M. 2007. Adaptive control of chaotic systems based on a single layer neural network. *Physics Letters A*, 368(5): 379–382.
- Shen, L., Wang, M., Liu, W., Sun, G. 2008. Prediction based chaos control via a new neural network, *Physics Letters A*, 372: 6916–6921.
- Song, Y., Chen, Z., Yuan, Z. 2007. Neural network nonlinear predictive control based on tent-map chaos optimization. *Chinese Journal of Chemical Engineering*, 15(4): 539–544.
- Song, X. N., Song, S. 2016a. Fuzzy sliding mode control for fractional-order unified chaotic system. *IEEE International Conference on Information and Automation, Ningbo, Çin Halk Cumhuriyeti*, 1090–1095.

- Song, S., Song, X. N. 2016b. T-S fuzzy control for fractional order Liu chaotic system with uncertain parameters. IEEE International Conference on Information and Automation, Ningbo, Çin Halk Cumhuriyeti, 1090–1095.
- Strogatz, S. 2001. Nonlinear Dynamics and Chaos. Westview Press, Cambridge, İngiltere.
- Sugeno, M., Kang, G. 1986. Structure identification of fuzzy model. Fuzzy Sets and Systems, 28: 329–346.
- Suykens, J. A. K., Vandewalle, J. 1996. Control of a recurrent neural network emulator for the double scroll. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 43(6): 511–514.
- Şen, Z. 2004. Yapay Sinir Ağları İlkeleri. Su Vakfı Yayınları, İstanbul, Türkiye.
- Takagi, T., Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics, 15: 116–132.
- Tam, L. M., Li, S. Y. 2015. Novel-fuzzy-model based modeling and control of nonlinear chaotic systems with uncertainty. International Conference on Advanced Robotics and Intelligent Systems, Taipei, Tayvan.
- Tang, L., Gao, Y., Liu, Y. J. 2014. Adaptive near optimal neural control for a class of discrete-time chaotic system. Neural Computing & Applications, 25(5): 1111–1117.
- Temurtaş, F. 2000. Kimyasal sensör dizilerinde yapay sinir ağları ve bulanık mantık uygulamaları: Gazların sınıflandırılması ve gaz konsantrasyonlarının belirlenmesi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Bölümü, Doktora tezi.
- Tooranjipour, P., Vatankhah, R. 2018. Adaptive critic-based quaternion neuro-fuzzy controller design with application to chaos control. Applied Soft Computing, 70: 622–632.
- Tsoukalas, L. H., Uhrig, R. E. 1996. Fuzzy and Neural Approaches in Engineering. Jhon Wiley & Sons Inc., New York, Amerika Birleşik Devletleri.
- Umoh, E. A. 2013. Chaos control of the complex Rabinovich system via Takagi-Sugeno fuzzy controller. IEEE International Conference on Emerging and Sustainable Technologies for Power and ICT in a Developing Society, Owerri, Nijerya, 217–222.
- Vaidyanathan, S., Rhif, A. 2017. A novel four-leaf chaotic system, its control and synchronisation via integral sliding mode control. International Journal of Modelling Identification and Control, 28(1): 28-39.
- Wang, H., Han, Z. Z., Xie, Q. Y., Zhang, W. 2009. Sliding mode control for chaotic systems based on LMI. Communications in Nonlinear Science and Numerical Simulation, 14(4): 1410–1417.
- Wang, X., Li, X. 2010. Feedback control of the Liu chaotic dynamical system. International Journal of Modern Physics B, 24(3): 397–404.

- Wang, Q., Meng, L. J., Shao, K. Y., Han, X., Huang, H., Qian, K. 2017. A new fractional order chaotic system with 4-scroll chaotic attractors and its adaptive control. 36th Chinese Control Conference, Dalian, Çin Halk Cumhuriyeti, 462–468.
- Wang, T., Wang, K., Jia, N. 2010. Chaos control of a time-delay chaotic neural network for information processing. IEEE International Conference on Information and Automation, ICIA, Harbin, Çin Halk Cumhuriyeti, 305–308.
- Weeks, E. R., Burgess, J. M. 1997. Evolving artificial neural networks to control chaotic systems. *Physical Review E*, 56(2): 1531–1540.
- Wu, L., Wang, C. 2007. Adaptive fuzzy sliding mode control of Lorenz chaotic system. *Frontiers of Electrical and Electronic Engineering in China*, 2(3): 277–281.
- Yan, X., Liu, D. 2010. Chaos control of fractional order Lü system via RBF neural network sliding mode controller. 8th World Congress on Intelligent Control and Automation, WCICA, Jinan, Çin Halk Cumhuriyeti, 5099–5104.
- Yang, L. X., Zhang, Z. R., Zhang, J. G. 2009. Using a neural network controller to control chaos in the Rossler system. *International Conference on Artificial Intelligence and Computational Intelligence*, AICI '09, Shanghai, Çin Halk Cumhuriyeti, 50–53.
- Yau, H. T., Yan, J. J. 2004. Design of sliding mode controller for Lorenz chaotic system with nonlinear input. *Chaos, Solitons & Fractals*, 19(4): 891–898.
- Yeh, K., Chen, C. W., Lo, D. C., Liu, K. F. R. 2011. Neural-network fuzzy control for chaotic tuned mass damper systems with time delays. *Journal of Vibration and Control*, 18(6): 785–795.
- Yu, W. 1999. Passive equivalence of chaos in Lorenz system. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, 46(7): 876–878.
- Zadeh, L. A. 1965. Fuzzy sets. *Information and Control*, 8: 338–353.
- Zadeh, L. A. 1973. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1): 28–44.
- Zadeh, L. A., 1975. The concept of a linguistic variable and its application to approximate reasoning, Parts 1, 2, and 3. *Information Sciences*, 8: 199–249, 8: 301–357, 9: 43–80.
- Zeng, Y. M., Singh, S. N. 1997. Adaptive control of chaos in Lorenz system. *Dynamics and Control*, 7(2): 143–154.
- Zhao, H., Ma, Y., Liu, S., Yue, Y. 2011. Fuzzy sliding mode variable structure control of chaotic power system with uncertainty. *Journal of Computational Information Systems*, 7(6): 1959–1966.
- Zhou, W. N., Pan, L., Li, Z., Halang, W. A. 2009. Non-linear feedback control of a novel chaotic system. *International Journal of Control, Automation and Systems*, 7(6): 939–944.

- Zirkohi, M. M. 2017. Model reference type-2 fuzzy sliding mode control for a novel uncertain hyperchaotic system. *Journal of Intelligent & Fuzzy Systems*, 32(1): 389–400.
- Zribi, M., Alshamali, S. 2012. Synchronization of the extended Bonhoffer-Van der Pol oscillators. *Mathematical Problems in Engineering*, 2012: 962680.
- <http://akabamuhendislik.com/bulanik-mantiga-giris-2/>. Bulanık mantığa giriş 2. Erişim Tarihi: Nisan 2017.
- <http://www.akanesen.com/2017/09/ileri-ve-geri-beslemeli-aglar.html>. İleri ve geri beslemeli ağlar. Erişim Tarihi: Temmuz 2018.
- <http://www.buzlu.org/kaos-ve-kaos-teorisi/>. Kaos ve kaos teorisi. Erişim Tarihi: Mart 2016.
- <http://www.evrenindili.com/component/content/article/101-dostlarimiz/263-kaos-kuram-ve-kaotik-sistemler?directory=194>. Kaos kuramı ve kaotik sistemler. Erişim Tarihi: Mart 2016.
- <https://fikirjeneratoru.com/cesitli-yapay-zeka-tekniklerinin-karsilastirilmesi/>. Çeşitli yapay zeka tekniklerinin karşılaştırılması. Erişim Tarihi: Mayıs 2017.
- <http://ozguraktekin.blogspot.com/2015/10/kaos-kontrolu.html>. Kaos kontrolü. Erişim Tarihi: Mart 2016.
- https://en.wikipedia.org/wiki/control_of_chaos. Control of chaos. Erişim Tarihi: Mart 2016.
- https://tr.wikipedia.org/wiki/Bulanık_mantık. Bulanık mantık. Erişim Tarihi: Kasım 2015.
- https://tr.wikipedia.org/wiki/Uyarlamalı_ağ_tabanlı_bulanık_çıkartım_sistemi. Uyarlamalı ağ tabanlı bulanık çıkarım sistemi. Erişim Tarihi: Kasım 2015.

ÖZGEÇMİŞ

Uğur Erkin KOCAMAZ, 02.01.1979 tarihinde Malatya'nın Yeşilyurt ilçesinde doğdu. İlk ve orta eğitimini sırasıyla Sakarya Büyükgazi İlkokulu ve Sakarya Anadolu Lisesi'nde gördü. Kocaeli Körfez Fen Lisesi'nde başladığı lise eğitimini Sakarya Atatürk Lisesi'nde tamamladı. 1996 yılında başladığı, 1 yıl İngilizce hazırlık eğitimi de gördüğü Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği bölümünden 2001 yılında mezun oldu. 2004'de Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği bilim dalında yüksek lisans öğrenimi bitirdi ve Sakarya Üniversitesi Elektrik-Elektronik Mühendisliği anabilim dalında doktora öğrenimine başladı. 2003 yılı Aralık ayında Sakarya Üniversitesi'nde fen bilimleri enstitüsü kadrosu ile araştırma görevlisi olarak mesleki kariyerine başladı. 2012 yılı Aralık ayında öğretim görevlisi olarak Uludağ Üniversitesi Karacabey Meslek Yüksekokulu'na geçti. Halen bu görevini sürdürmektedir.