T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

LYAPUNOV ÜSTELLERİNİN YAPAY SİNİR AĞLARI KULLANILARAK TAHMİNİ

Hazırlayan
Koray Fethi PARLAK

Danışman
Dr. Öğr. Üyesi İlker DALKIRAN

Yüksek Lisans Tezi

Haziran 2019
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

LYAPUNOV ÜSTELLERİNİN YAPAY SİNİR AĞLARI KULLANILARAK TAHMİNİ

(Yüksek Lisans Tezi)

Hazırlayan
Koray Fethi PARLAK

Danışman
Dr. Öğr. Üyesi İlker DALKIRAN

Haziran 2019
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

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

Koray Fethi PARLAK

"Lyapunov Üstellerinin Yapay Sinir Ağları Kullanılarak Tahmini" adlı Yüksek Lisans / ~~Doktora~~ tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan
Koray Fethi PARLAK

Danışman
Dr. Öğr. Üyesi İlker DALKIRAN

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

Dr. Öğr. Üyesi İlker DALKIRAN danışmanlığında **Koray Fethi PARLAK** tarafından hazırlanan "**Lyapunov Üstellerinin Yapay Sinir Ağları Kullanılarak Tahmini** " adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

.

25/06/2019

JÜRİ:

Danışman	: Dr. Öğr. Üyesi İlker DALKIRAN	.....
Üye	: Prof. Dr. Kenan DANIŞMAN	.....
Üye	: Dr. Öğr. Üyesi Selim SOYLU	.....

ONAY:
Bu tezin kabulü Enstitü Yönetim Kurulunun 09/07/2019 tarih ve 2019/40-19 sayılı kararı ile onaylanmıştır.

09/07/2019

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

ÖNSÖZ / TEŞEKKÜR

Yazmış olduğum bu Yüksek Lisans Tezinin; yürütülmesinde ve sonuçlandırılmasında yardım ve katkılarını esirgemeyen değerli hocam Sayın Dr. Öğr. Üyesi İlker DALKIRAN'a teşekkürü bir borç bilirim.

Ayrıca Yüksek Lisans sürecinde sabırla yanımda olan ve beni her konuda destekleyen aileme teşekkür ederim.

Koray Fethi PARLAK
Kayseri, Haziran 2019

LYAPUNOV ÜSTELLERİNİN YAPAY SİNİR AĞLARI KULLANILARAK TAHMİNİ

Koray Fethi PARLAK

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Haziran 2019

Danışman: Dr. Öğr. Üyesi İlker DALKIRAN

ÖZET

Doğadaki bir çok sistemin fiziksel davranışını önceden tahmin edebilmemize rağmen, hava olayları veya borsa gibi tahmin edilemeyen sistemler de vardır. Doğrusal olmayan, deterministik, belirli matematiksel kriterleri karşılayan ve düzensiz görünümlü olan bu sistemler kaotik olarak adlandırılırlar. Kaosun temel özelliklerinden birisi başlangıç koşullarına hassas bağımlılıktır ve yakın başlangıç şartlarına sahip iki yörüngenin zamanla farklılaşmasını ölçen en büyük Lyapunov Üsteli, kaosun kullanışlı bir göstergesidir. Uygun parametrelere sahip kaotik sistemler, başlangıç koşullarındaki küçük bir değişimden dolayı çok farklı yönlere evrilebilirler.

Bu çalışmada Lorenz, Chua ve Rössler sistemlerinin en büyük Lyapunov üstelleri, farklı parametre (Lorenz sistemi için; Sigma, Beta, Rho, Chua sistemi için; Alfa, Beta, a, b, ve Rössler sistemi için ise a, b ve c) değerleri için hesaplanmış ve elde edilen sonuçlar, farklı Yapay Sinir Ağları (YSA) modelleri için eğitim ve test verisi olarak kullanılmıştır. Hesaplaması uzun süre alan en büyük Lyapunov üstelleri YSA'lar ile daha kısa bir sürede, küçük bir hatayla tahmin edilmiştir. Ayrıca YSA algoritmaları, katmanlardaki nöron sayıları ve aktivasyon fonksiyonlarının sonucu nasıl etkilediği incelenmiştir. Böylelikle bir sistemin çeşitli parametre değerlerinde kaotik olup olmadığı YSA kullanılarak tahmin edilmiştir.

Anahtar Kelimeler: Kaos, En Büyük Lyapunov Üsteli, Yapay Sinir Ağı, Lorenz, Chua, Rössler.

ESTIMATION OF LYAPUNOV EXPONENTS USING ARTIFICIAL NEURAL NETWORKS

Koray Fethi PARLAK

Erciyes University, Institute of Natural and Applied Sciences

M.Sc. Thesis, June 2019

Supervisor: Asst. Prof. İlker DALKIRAN

ABSTRACT

Although we can predict in advance physical behaviours of many systems in nature, there are also many systems like weather or stock exchanges that are not predictable. These nonlinear, deterministic, disorderly looking systems that satisfy certain mathematical criteria are called chaotic. One of the main features of chaos is sensitive dependence to initial conditions and the largest Lyapunov exponent is a useful indicator of chaos due to the fact that it measures how two orbits that have close initial conditions differ in time. Chaotic systems with appropriate parameters evole in many different directions by small changes in initial conditions.

In this study, the largest Lyapunov exponents of Lorenz, Chua and Rössler systems for different parameter values (Sigma, Beta, Rho for Lorenz system; Alfa, Beta, a, b for Chua system and a, b, c for Rössler system) are calculated and then obtained results are used as training and test data for a combination of different artificial neural network (ANN) models. The largest Lyapunov exponent, which its calculation may take so long time, is predicted by ANNs with a small error in a shorter time. Also it is examined that how ANN algorithms, number of neurons in layers and activation functions effect the result. Thus, its is predicted by using ANNs whether a system in various parameter values is chaotic or not.

Keywords: Chaos, Largest Lyapunov Exponent, Artificial Neural Network, Lorenz, Chua, Rössler.

İÇİNDEKİLER

BİLİMSEL ETİĞE UYGUNLUK SAYFASI ii
KABUL VE ONAY SAYFASI iii
ÖNSÖZ iv
ÖZET v
ABSTRACT vi
İÇİNDEKİLER vii
KISALTMALAR VE SİMGELER x
TABLOLAR LİSTESİ xii
ŞEKİLLER LİSTESİ xiii

1. BÖLÜM
GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

Genel Bilgiler ve Literatür Çalışması 1

2. BÖLÜM
KAOS TEORİSİ

2.1. Giriş 4
2.2. Kaos Teorisi ile İlgili Kavramlar 6
2.3. Kaotik Zaman Serisi Analizi 11
2.3.1. Faz Uzayının Yeniden İnşası 12
2.3.2. Zaman Gecikmesinin Seçilmesi 13
2.3.3. Gömme Boyutunun Seçilmesi 14
2.3.4. Durağanlık Analizi 16
2.3.5. Vekil Veri Metodu (Surrogate Data Method) 16
2.4. Boyut Kavramı 16

3. BÖLÜM
LYAPUNOV ÜSTELLERİ

3.1. Giriş 21
3.2. Tanım 21
3.3. Diferansiyel Denklemlerin Çözümü 22
3.4. Runge Kutta Metotları 23
3.5. Lorenz Sistemi 24
3.6. Rössler Sistemi 29

3.7. Chua Devresi 31
3.8. Yörünge Ayırma (Orbit Separation) Metodu 35

4. BÖLÜM
YAPAY SİNİR AĞLARI

4.1. Giriş 39
4.2. Tarihçe 39
4.3. Yapay Sinir Ağlarının Avantaj ve Dezavantajları 40
4.4. Yapay Sinir Ağlarının Kullanım Alanları 41
4.5. Biyolojik Nöron 42
4.6. Yapay Nöron 42
4.7. Aktivasyon Fonksiyonları 44
4.8. Yapay Sinir Ağlarının Sınıflandırılması 44
4.8.1. Öğrenme Algoritmasına Göre Yapay Sinir Ağları 44
4.9. Yapay Sinir Ağı Modelleri 45
4.10. Geri Yayılım Algoritması 45
4.11. Ortalama Karesel Hata 48
4.12. En Küçük Ortalama Kareler (Least Mean Squares) Algoritması 51
4.13. Geri Yayılım Algoritmasının Dezavantajları ve Sezgisel Yaklaşımlar 52
4.13.1. Momentum 53
4.13.2. Değişken Öğrenme Oranı 53

5. BÖLÜM
BULGULAR

Bulgular..... 56

6. BÖLÜM
TARTIŞMA-SONUÇ ve ÖNERİLER

Tartışma-Sonuç ve Öneriler..... 70

KAYNAKLAR……...73

ÖZGEÇMİŞ…….……...…...82

KISALTMALAR VE SİMGELER

Sembol	Anlamı	Birimi
T	Zaman gecikmesi	--
m	Gömme boyutu	--
P	Ölçülme olasılığı	--
I	Ortalama karşılıklı bilgi	--
R	Öklid farkı	--
ε	Kutu kenar uzunluğu	--
N	Kartezyen uzayı boyutu	--
D_0	Kutu sayım boyutu	--
r	Doğruluk oranı	--
S	Bilgi boyutu gerekli bilgi biti	--
D_I	Bilgi boyutu	--
D_C	Korelasyon boyutu	--
λ	Lyapunov üsteli	--
δ	Ayrılma vektörü	--
h	Zaman adımı	--
n	İterasyon sayısı	--
G	Nonlineer eleman karakteristik eğimi	--
B	Karakteristik eğrisi kırılma noktası	--
YSA	Yapay Sinir Ağı	--

w	Nöron ağırlığı	--
b	Nöron bias değeri	--
η	Öğrenme oranı	--
δ	Yerel gradyent	--
R	Giriş korelasyon matrisi	--
h	Çapraz korelasyon	--
x^*	Minimum hata kareler ortalaması	--
α	Öğrenme oranı	--
g	Gradyent	--
EKOK	En Küçük Ortalama Kare algoritması	--
J	Jakobyen	--
λ	Marquardt parametresi	--
I	Tanımlama matrisi	--
LM	Levenberg-Marquardt	--
EBLÜ	En Büyük Lyapunov Üsteli	--
LLE	En Büyük Lyapunov Üsteli	--
σ	Prandtl sayısı	--
tansig	Hiperbolik Tanjant Sigmoid	--
logsig	Logaritmik Sigmoid	--

TABLOLAR LİSTESİ

Tablo 3.1.	Butcher Tablosu genel formu [69].	23
Tablo 3.2.	4. Dereceden Runge Kutta Metodu için Butcher Tablosu [69].	24
Tablo 3.3.	8. Dereceden Runge Kutta yöntemi için Butcher Tablosu [68].	25
Tablo 3.4.	Tablo 3.3' de kullanılan w değerleri [68].	25
Tablo 5.1.	Lorenz sistemi parametre setlerinin oluşturulması için kullanılan parametre aralıkları.	57
Tablo 5.2.	Chua sistemi parametre setlerinin oluşturulması için kullanılan parametre aralıkları.	57
Tablo 5.3.	Rössler sistemi parametre setlerinin oluşturulması için kullanılan parametre aralıkları.	57
Tablo 5.4.	Lorenz, Chua ve Rössler sistemleri için standart parametreler [70, 75, 81].	57
Tablo 5.5.	Kullanılan Matlab eğitim algoritmaları.	59
Tablo 5.6.	Chua sistemi için elde edilen en iyi sonuçlar (MSE).	62
Tablo 5.7.	Lorenz sistemi için elde edilen en iyi sonuçlar (MSE).	62
Tablo 5.8.	Rössler sistemi için elde edilen en iyi sonuçlar (MSE).	63
Tablo 5.9.	Chua sistemi için elde edilen en iyi 5 algoritmada en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranları.	65
Tablo 5.10.	Lorenz sistemi için elde edilen en iyi 5 algoritmada en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranları.	65
Tablo 5.11.	Rössler sistemi için elde edilen en iyi 5 algoritmada en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranları.	65
Tablo 5.12.	1 adet en büyük Lyapunov üstelinin hesaplanması için gereken süre (saniye).	67
Tablo 5.13.	Eğitim algoritmalarına göre 1 adet LLE tahmini yapılabilmesi için gereken saniye cinsinden eğitim ve test süreleri (Epoch:500, İlk katman nöron sayısı: 5, İkinci katman nöron sayısı:5, İlk katman aktivasyon fonksiyonu: logsig, İkinci katman aktivasyon fonksiyonu: logsig).	68

ŞEKİLLER LİSTESİ

Şekil 2.1. Doğadan kaotik örnekler: (a) Satürn'ün kaotik yörüngeli halkaları (b) Fraktal benzeri yapılar oluşturan fitoplanktonların bulunduğu bir denizin uydu görüntüsü (c) Kutuplarını kaotik bir şekilde tersine çeviren Dünya'nın geomanyetik alanı [6]. 5

Şekil 2.2. Damlayan musluk deneyi [44]. 6

Şekil 2.3. Musluk ağzındaki su damlasının titreşimi [45]. 6

Şekil 2.4. Duffing osilatörünün zaman serisi ve faz uzayı [48]. 8

Şekil 2.5. Lojistik sistem çatallaşma diyagramı [49]. 9

Şekil 2.6. (a) Yörüngelerin tek bir noktaya doğru çekildiği sabit noktalı kaotik olmayan çeker (b) Yörüngelerin iki farklı noktaya doğru çekildiği iki sabit noktalı kaotik olmayan çeker (c) Yörüngelerin halka benzeri bir çeker setine doğru çekildiği kaotik olmayan limit çevrim çekeri (d) Lorenz sistemine ait kaotik Lorenz çekeri [53]. 10

Şekil 2.7. Mandelbrot kümesi [52]. 10

Şekil 2.8. Faz uzayında bir limit döngü [3]. 13

Şekil 2.9. Lorenz sistemi için zaman gecikmesinin fonksiyonu olarak ortalama karşılıklı bilgi [50]. 14

Şekil 2.10. $\tilde{N}(\varepsilon)$ kutuları a) İki nokta için b) Bir eğri parçası için c) Kapalı bir eğrinin içinde kalan alan için [63]. 18

Şekil 2.11. 48 Olası durumlu bir faz uzayı ve eşit ayrık zamanlarla örneklenmiş bir yörünge [2]. 19

Şekil 3.1. [0; 2; 0] başlangıç değerleri ve $\sigma = 10$, $r = 28$ ve $b = 8/3$ parametre değerleri için; (a) X-Y-Z düzlemindeki Lorenz çekeri (b) X-Y düzlemi Lorenz çekeri (c) Y-Z düzlemi Lorenz çekeri (d) X-Z düzlemi Lorenz çekeri [70]. 27

Şekil 3.2. X-Y-Z düzlemindeki Lorenz çekeri (Parametreler: $\sigma = 10$, $r = 28$, $b = 8/3$) (a) Başlangıç değerleri: $x = 0$, $y = 1$, $z = 0$ (b) Başlangıç değerleri: $x = 0$, $y = 1{,}01$, $z = 0$ [73]. 27

Şekil 3.3. Lorenz sisteminin $x_1(t)$ ve $x_2(t)$ grafiği. 28

Şekil 3.4. Lorenz sisteminin elektronik devre olarak gerçekleştirilmesi [74]. 28

Şekil 3.5. [0,1; -0,1; 0,1] başlangıç değerleri ve a = 0,2, b = 0,2, c = 5,7 parametre değerleri için; (a) X-Y-Z düzlemindeki Rössler çekeri (b) X-Z düzlemi Rössler çekeri (c) X-Y düzlemi Rössler çekeri (d) Y-Z düzlemi Rössler çekeri [76]. 30

Şekil 3.6. Rössler devresi [77]. 31

Şekil 3.7. Chua devresi. [79]. 31

Şekil 3.8. R_0 direnci eklenmiş Chua devresi [80]. 32

Şekil 3.9. Parçalı-lineer direncin v-i karakteristiği [80]. 33

Şekil 3.10. [0,1; 0,1; 0,1] başlangıç değerleri ve $\alpha = 9$, $\beta = 100/7$, $a = -8/7$ ve $b = -5/7$ parametre değerleri için; (a) X-Y-Z düzlemi Chua çekeri (b) Y-Z düzlemi Chua çekeri (c) X-Y düzlemi Chua çekeri (d) X-Y düzlemi Chua çekeri [82]. 34

Şekil 3.11. Başlangıç şartları birbirine yakın iki yörüngenin ayrılması [68]. 36

Şekil 3.12. Yörüngelerin referans mesafe (ε) ile gösterilmesi [68]. 36

Şekil 4.1. Biyolojik nöronun yapısı [99]. 42

Şekil 4.2. Yapay nöron modeli [100]. 43

Şekil 4.3. Yapay sinir ağı modeli [109]. 49

Şekil 4.4. (a) Öğrenme oranı olması gerekenden büyük seçildiği için hedefine ulaşamayan bir yörünge ve karesel hatası (b) Aynı başlangıç şartlarına ve öğrenme oranına sahip fakat momentum katsayısı eklenmiş yörünge ve karesel hatası [109]. 54

Şekil 5.1. Lorenz, Chua ve Rössler sistemlerinin her biri için ayrı ayrı uygulanan YSA kombinasyonları. 58

Şekil 5.2. Farklı Sigma değerleri için Lorenz sisteminin sahip olduğu LLE değerleri. 60

Şekil 5.3. $x_0 = 0$, $y_0 = 0$ ve $z_0 = 0$ başlangıç değerleri ve Beta = 8/3, Rho = 28 parametre değerlerine karşılık Sigma = 10 (Mavi) ve Sigma = 10,1 (Kırmızı) değerleri için Lorenz sisteminin ürettiği x(t) grafiği. 60

Şekil 5.4. Lorenz sisteminin parametre değişikliği halinde birbirlerine göre grafiği (Başlangıç değerleri: 0 , 1,01 , 0 Diferansiyel denklem çözüm hassasiyeti: 0,000001 Zaman adımı = 15) X ekseni : Sigma = 10 Y ekseni: Sigma = 10,1 61

Şekil 5.5. Chua sistemi için en iyi sonucun gerçek LLE/ağ çıkışı karşılaştırma grafiği. 63
Şekil 5.6. Lorenz sistemi için en iyi sonucun gerçek LLE/ağ çıkışı karşılaştırma grafiği. 64
Şekil 5.7. Rössler sistemi için en iyi sonucun gerçek LLE/ağ çıkışı karşılaştırma grafiği. 64
Şekil 5.8. Chua sistemi için elde edilen en iyi sonucun veri başına karesel hata - MSE karşılaştırması. 66
Şekil 5.9. Lorenz sistemi için elde edilen en iyi sonucun veri başına karesel hata - MSE karşılaştırması. 66
Şekil 5.10. Rössler sistemi için elde edilen en iyi sonucun veri başına karesel hata - MSE karşılaştırması. 67
Şekil 5.11. Chua sistemi için nöron sayısı artırımının MSE cinsinden hataya etkisi. 69
Şekil 5.12. Lorenz sistemi için nöron sayısı artırımının MSE cinsinden hataya etkisi. 69
Şekil 5.13. Rössler sistemi için nöron sayısı artırımının MSE cinsinden hataya etkisi. 69

1. BÖLÜM
GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

Deterministik bir sistemdeki stokastik davranış olarak tanımlanabilen kaos, son zamanların en çok ilgi çeken ve hızlı bir şekilde genişleyen bilimsel araştırma alanlarından bir tanesidir. Kaotik sistemlerin ilgi çekmesinin nedenleri; düzensiz görünen zaman serilerinin deterministik olup olmadığını anlama isteği, modellemeyi kolaylaştırma, kısa vadeli tahminlerin daha doğru yapılabilmesi ve uzun vadeli sistemlerde doğru tahmin yapmanın mümkün olup olmadığının araştırılmasıdır [1, 2]. Kaosun en önemli özelliklerinden biri, başlangıç koşullarına hassas bağımlılık olduğu için başlangıç şartlarına duyarlılığın bir ölçüsü olan en büyük Lyapunov üsteli kaotikliğin en önemli göstergelerindendir [3].

Modern bilim dünyasında "kaos" kelimesini ilk kez Tien-Yien Li ve James Yorke 1975'de yazdıkları "Period Three Implies Chaos" isimli çalışmalarında kullanmalarına rağmen [4] kaos teorisinin bilim dünyasındaki kökleri çok daha öncelere dayanmaktadır. 17. yüzyılın ortalarında diferansiyel denklemleri keşfeden Newton, iki cisim problemi olarak da bilinen Dünya ve Güneş'in yörüngesel hareketlerini hesaplamıştır. Newton'dan sonra gelen bilim adamları, üç cisim problemini yani Ay, Dünya ve Güneş'in yörüngesel hareketlerinin nasıl olacağını çözmeye çalışmışlar fakat bu çalışmanın neticesinde çözümün neredeyse imkansız olduğu sonucuna ulaşmışlardır. Bilim dünyasının ihtiyaç duyduğu bakış açısı, 19. yüzyılın sonlarında Poincare ile gelmiştir. Üç cisim problemini çözmeye çalışan bilim adamı, ne sabit bir noktaya yakınsayan ne de sonsuza kadar artan, periyodik olmayan yörüngelerin var olabileceğini keşfederek kaos teorisine göz kırpmıştır. 1898'de Jacques Hadamard, kaotik olduğu kanıtlanan ilk dinamik sistem olan Hadamard dinamik sistemini yayınlamıştır [5]. Daha sonraki yılarda Birkhoff, Kolmogorov ve Smale gibi isimler, doğrusal olmayan diferansiyel denklemler konusunda çeşitli çalışmalar yapmışlardır. 20. yüzyılın ikinci

yarısında bilgisayar teknolojisinde gelişmeler yaşanması kaos teorisinin yeni bir evreye girmesini sağlamıştır. 1963'de Lorenz, Royal McBee LGP-30 bilgisayarıyla başlangıç şartlarına hassas bağımlılık ilkesini ve Lorenz dinamik sistemini keşfetmiştir [6]. 1960 ve 1970'lerde Mandelbrot, fraktalları sistematikleştirip popüler hale getirirken, Feigenbaum, May, Libchaber ve diğer birçok bilim adamı, kaos teorisini farklı uygulama alanlarında geliştirmişlerdir [7, 8]. 1980'lerde Kaliforniya Üniversitesinden Leon Chua, yalnızca bir adet doğrusal olmayan eleman içermesine rağmen kaotik çeker oluşturabilen basit bir elektronik devre geliştirmiştir [9]. Bilgisayarların daha da hızlanmasıyla günümüze kadar sürekli gelişen bir bilim dalı olan kaosun uygulama alanları, mühendislikten psikolojiye, sağlık bilimlerinden finansa kadar yayılmıştır [6].
Lyapunov üstelleri konusunda literatürde yer alan çalışmalara bakıldığında ise kaotiklik göstergesi sayıların varlığının Oseledec [10] teoremiyle ispatlandığı fakat Lyapunov üstellerini kullanan ilk çalışmanın Henon ve Helies [11] tarafından sunulduğu görülür [12]. Sürekli sistemlerde Lyapunov üstellerinin hesaplanması için temel sayılabilecek algoritmalar Benettin [13] ile Shimada ve Nagashima [14] tarafından bulunmuş daha sonra ise bu çalışmalar Benettin [15] ve Wolf [16] tarafından geliştirilmiştir. Süreksizlikler ve zaman gecikmeleri olan sistemlerde Lyapunov üstellerinin hesaplanması (tahmini) için ise Takens prosedürü temel oluşturur [17]. Wolf ve arkadaşları [18] ve Sano ve Sawada [19] bu yaklaşımı temel alarak nümerik algoritmalar geliştirmiş daha sonra Eckmann ve arkadaşları [20], Rosenstein ve arkadaşları [21] ve Parlitz [22] algoritmaları daha da iyileştirmişlerdir. Kaotik bir sistemin tüm Lyapunov üstellerini hesaplamak zaman alabileceği için sadece en önemlisi olan en büyük Lyapunov üstelini hesaplamak üzere Rosenstein ve arkadaşları [21] ve Kantz [23] tarafından algoritmalar geliştirilmiştir. Literatürde hesaplama tekniklerini geliştirmeyi amaçlayan daha yakın zamanlı çeşitli çalışmalarda mevcuttur [12, 24-29].

Kaos teorisi ve Yapay Sinir Ağları (YSA'ları) literatürde oldukça geniş yer bulmuş iki konudur. Kaos ve YSA'ların ikisine de değinen çalışmalar araştırıldığında, kaotik sistemlerin YSA'lar ile modellendiği görülmüştür. Örneğin; Chua devresi [30] veya MOS transistörlü, kuartz osilatörlü elektronik devreler [31] YSA ile modellenmiş, kriptoloji için YSA temelli kaotik jeneratör tasarlanmış [32], Chua devresinin dinamikleri YSA ile tahmin edilmiştir [33]. YSA'lar ve Lyapunov üstelleri tarandığında

ise; YSA'lar ile Lyapunov üstellerinin modellendiği [34], Lyapunov spektrumunun YSA ile tahmin edildiği [34-37], YSA'ların başarılı eğitimi için geliştirilen bir tekniğin Lyapunov üstellerinin tahmininde kullanıldığı [38], gürültülü bir sistemde Lyapunov üstellerinin parametrik olmayan tahmininin YSA ile yapıldığı [39] görülür. Yapay sinir ağlarında değerlendirme kriteri olarak Lyapunov üstellerinin kullanıldığı örneklere bakıldığında ise; YSA ve Lyapunov üstelleri kullanılarak kısmi epilepsi hastalarındaki elektrokardiyografik değişikliklerin saptandığı [40], elektro-ansefalogramlardan elde edilen veriler ile YSA ve Lyapunov üstellerinin yardımı ile epileptik nöbetlerin otomatik saptandığı [41] ve aynı şeklide epileptik nöbetlerin Lyapunov üstelleri ve YSA kullanılarak tahmin edildiği [42] görülür.

Bu tez çalışmasında üç farklı kaotik osilatör modeline ait farklı parametre ve başlangıç şartları için en büyük Lyapunov üstelleri nümerik olarak hesaplanmış ve bu değerlerin birleşimi YSA modellerinin eğitimi aşamasında kullanılmıştır. Eğitim aşaması tamamlandıktan sonra ise en büyük Lyapunov üstellerinin YSA modelleri ile kabul edilebilir hatalarla tahmini yapılmıştır. Bu sayede YSA modellerinin genelleme yaparak hesaplanmamış en büyük Lyapunov üstellerini daha kısa sürede tahmin etmesi planlanmıştır.

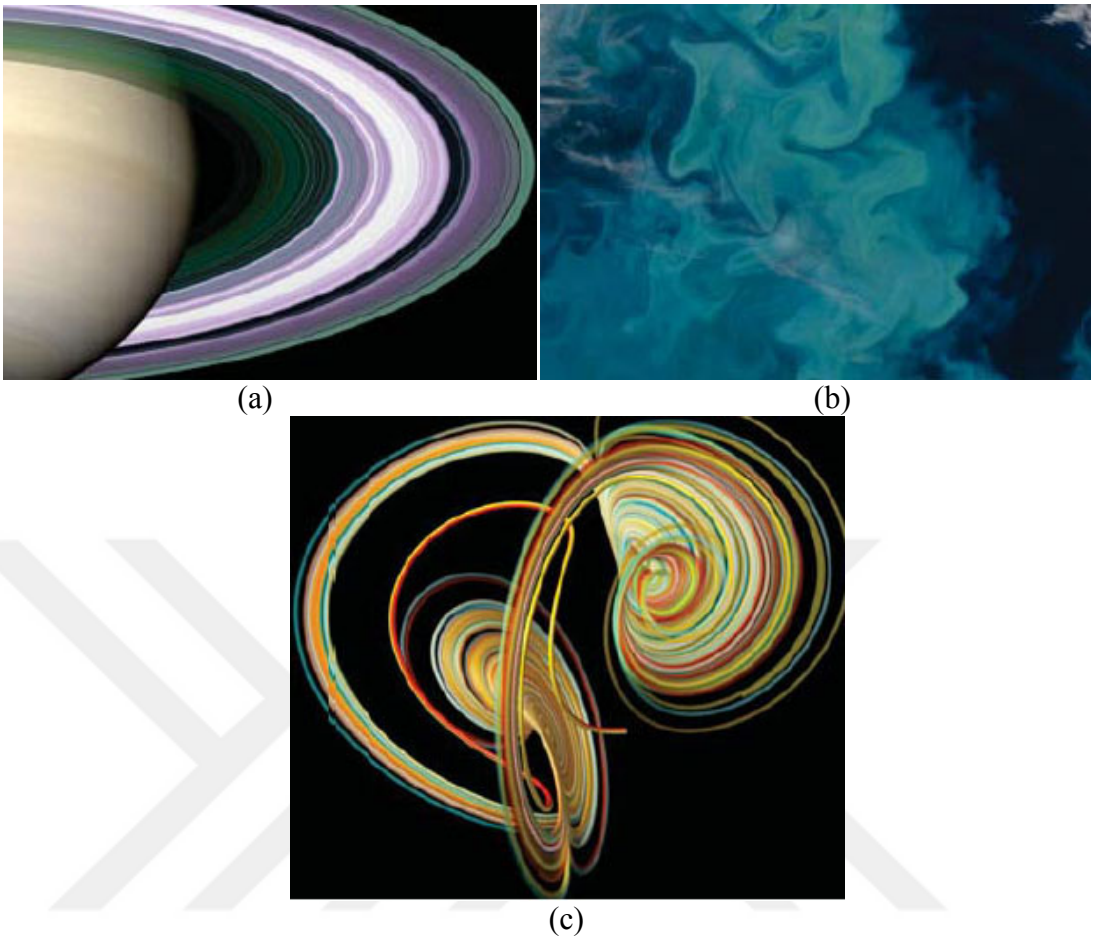
Bu tez çalışmasının ikinci bölümünde kaos teorisi hakkında genel bilgiler verilmiş, kaotik zaman serilerinin analizi ve sistemlerdeki boyut kavramı konularında bilgiler aktarılmıştır. Üçüncü bölümde, en büyük Lyapunov üstelini hesaplama yöntemi anlatılmış daha sonra da en büyük Lyapunov üsteli hesaplanacak sistemler hakkında bilgiler verilmiştir. Bunun yanı sıra hesaplama esnasında kullanılan diferansiyel denklemlerin çözümünden bahsedilmiştir. Dördüncü bölümde, Yapay sinir ağları (YSA) çeşitleri, geri yayılım algoritması ve bu tezde Matlab programı ile kullanılan eğitim algoritmalarından bahsedilmiştir. Beşinci bölümde elde edilen hesaplama sonuçları aktarılmıştır. Altıncı bölümde ise bu sonuçlar tartışılmıştır.

2. BÖLÜM
KAOS TEORİSİ

2.1. Giriş

Son 30 yılda oldukça ilgi çekici hale gelen kaos kavramı, mühendislikten ekonomiye varıncaya kadar bir çok bilim dalında kendine yer bulmuştur. Temel anlamı düzensizlik, karmaşa olan kaos kelimesi bilimsel literatürde bu temel anlamından çok daha fazlasını ifade eder. Kaos kelimesinin evrensel olarak kabul görmüş bir tanımı olmasa da; kaos, " *Deterministik bir sistemde başlangıç koşullarına hassas bağımlılık gösteren uzun süreli aperiyodik davranış*" [7] veya " *Deterministik doğrusal olmayan bir sistemde var olan ve belirli özel matematiksel kriterleri sağlayan, düzensiz görünen, uzun süreli, destekli evrim*" [2] olarak tanımlanabilir.

Yaşanan bazı fiziksel olaylar, formüller yardımı ile önceden bilinebilirken bazıları ise başlangıç şartlarına hassas bağımlı oldukları için kaotik davranış sergilerler ve önceden bilinemezler. Başlangıç koşullarına hassas bağımlılık, kaosun en önemli özelliklerinden biridir. Bu özelliğin anlamı kaotik bir sistemin başlangıç şartlarındaki ufak bir sapmanın süreç içerisinde büyük değişimlere yol açmasıdır. Bu davranış, Edward Lorenz tarafından tanımlanan "*Kelebek etkisi*" tabiri ile de ifade edilebilir. Kaos teorisinin önde gelenlerinden olan Lorenz, bir meteoroloji uzmanıydı [43]. Hava olaylarının simülasyonu sırasında verilerini bilgisayarına girerken ondalık hanesi daha uzun olan sayıları 3 haneye yuvarlayan bilim adamı, beklenmedik çok ciddi sonuçlarla karşılaştı [6]. Yani küsüratlardaki yuvarlamalar bile sonuca etki etmişti. Bu keşif daha sonra "*Amazonlarda kanat çırpan bir kelebeğin Amerika Birleşik Devletlerinde kasırgaya yol açabileceği*" şeklinde bir metafora dönüştürüldü ve günümüzde oldukça popüler olan "*Kelebek etkisi*" ifadesi literatüre katılmış oldu [6]. Şekil 2.1'de doğada bulunan bazı kaotik yapılara örnekler verilmiştir.

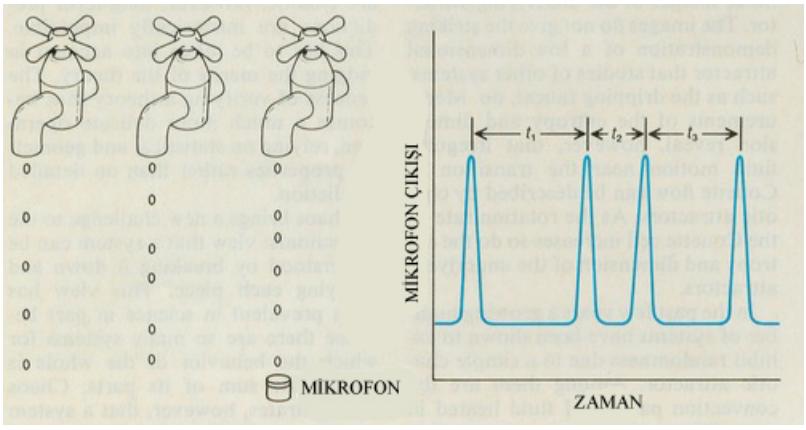


(a) (b)

(c)

Şekil 2.1. Doğada var olan kaotik örnekler: (a) Satürn'ün kaotik yörüngeli halkaları (b) Fraktal benzeri yapılar oluşturan fitoplanktonların bulunduğu bir denizin uydu görüntüsü (c) Kutuplarını kaotik bir şekilde tersine çeviren Dünya'nın geomanyetik alanı [6].

Günlük hayattaki kaotik davranışlara, bir musluğun damlaması örnek olarak gösterilebilir. Shaw tarafından 1984'de yapılan çalışmada zaman aralıkları incelenen su damlalarının kaotik davranış sergilediği ispat edilmiştir. Şekil 2.2'de verilen deneyde, değişik akış hızlarında musluğun damlaması sağlanmış ve su damlaları arasındaki süreler kaydedilmiştir [44].



Şekil 2.2. Damlayan musluk deneyi [44].

Akış hızı belli bir seviyeyi aşana kadar aralıkların periyodik olduğu ama musluk yeteri kadar hızlı damladığında periyodikliğin bozulup kaotik bir davranışın belirdiği gözlemlenmiştir. Kaotikliğin nedeni musluk ağzındaki su damlalarının titreşimi ile açıklanabilir. Şekil 2.3'de gösterilen titreşim, sonraki su damlasının başlangıç koşullarını etkilemekte, akış hızının artmasıyla da kaotiklik meydana gelmektedir [44].



Şekil 2.3. Musluk ağzındaki su damlasının titreşimi [45].

2.2. Kaos Teorisi İle İlgili Kavramlar

Dinamik sözcüğü canlı, hareketli, değişken gibi anlamlara gelirken; şimdiki durumunu geçmişteki durumlarına göre belirleyen, zaman içerisinde değişen, hareket eden sistemlere, dinamik sistemler denir. Örneğin her saat mevcut sayısını ikiye katlayan bir

bakteri kültürü dinamik bir sistemdir. Belirli bir zamandaki popülasyonu, geçen zamana ve başlangıçtaki bakteri sayısına bağlı olan dinamik sistem, determine (gerekirci) özellik taşır, yani sistemin başlangıç şartları ve matematiksel eşitliği (veya eşitlikleri) bilinirse, herhangi bir zaman noktasındaki değeri hesaplanabilir. Deterministik sistemlerde rastgelelik söz konusu değildir [2, 46].

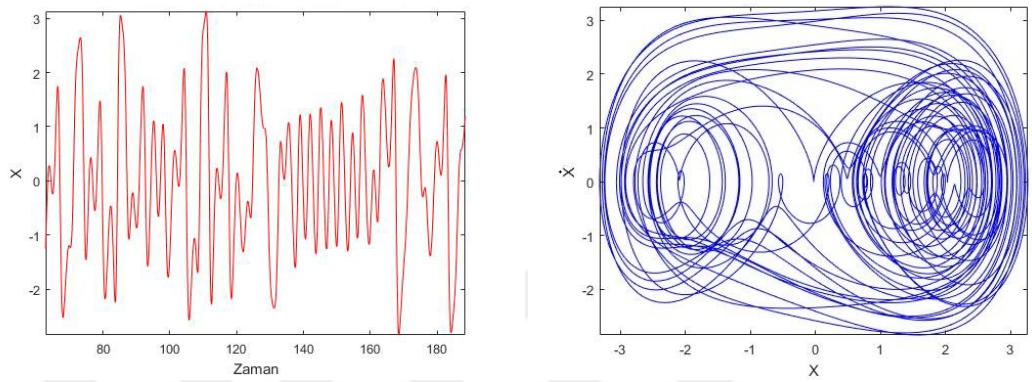
Dinamik sistemler, ölçme işleminin nasıl yapıldığına göre, ayrık zamanlı veya sürekli zamanlı olarak ikiye ayrılırlar. Sistem, belirli zaman aralıklarıyla ölçülürse, fark denklemleri ile tanımlanabilir ve ayrık zamanlı dinamik sistem olarak adlandırılır. Ölçüm işleminin aralıksız olarak yapıldığı sürekli zamanlı dinamik sistemler ise diferansiyel denklemler kullanılarak ifade edilir. Ayrık zamanlı sistemlerde adımlar, yani iterasyonlar bir haritada (map) ilerlerler. Sürekli zamanlı dinamik sistemlerde ise zamanın ve başlangıç değerinin bir fonksiyonu olan ilerlemeye, diferansiyel denklemin akışı (flow) denir [2, 46, 47].

Zamana göre değişen büyüklüklerin sürekli veya belirli aralıklarla ölçülmesiyle zaman serileri elde edilir. Zaman serilerinde değişkenlerin zamana göre ele alınmasından farklı olarak, sistemdeki değişkenlerin birbirlerine göre durumunu gösteren bir noktaya ise o sistemin fazı denir. Muhtemel tüm fazların bulunduğu grafiğe de faz uzayı denir. Sistemlere zaman serilerinden farklı bir gözle bakmamızı sağlayan faz uzayları, kaos teorisinde oldukça sık kullanılır. Bunun temel nedeni ise zaman serisinde görülemeyen şekillerin faz uzayında görülebilmesidir. Ayrıca zaman serilerinin oldukça uzun olabilmesine karşın faz uzayları, tüm sistemi tek grafikte yansıtabilir [2, 3]. Kaotik davranış sergileyen bir osilatör olan ve δ, α, β, ω katsayılar, x' ise x'in birinci dereceden türevi olmak üzere Denklem 2.1'de verilen Duffing osilatörünün zaman serisine ve faz uzayına ait Matlab programında hazırlanmış görselleri Şekil 2.4'de verilmiştir [48].

$$x''+\delta x'+\alpha x+\beta x^3 = \gamma \cos(\omega t) \tag{2.1}$$

Sadece miktar ifade edebilen uzunluk, sıcaklık, zaman ve hız gibi fiziksel bazı büyüklükler, skaler değerler olarak adlandırılırken, miktarın yanında yön bilgisi de içeren büyüklüklere ise vektörel değerler denir. Kaos teorisinde faz uzayındaki noktalar birer vektördür. Bu çalışmada ilerdeki sayfalarda bahsedilecek olan ve kaotik davranışın

en önemli göstergelerinden biri olan Lyapunov üstellerinin hesaplanmasında iki vektör arasındaki açıya ihtiyaç duyulur. Yine bu amaç için kullanılan birbirlerine dik (ortogonal) vektörleri belirleme işlemine ise ortogonalizasyon denir [2].



(a) (b)

Şekil 2.4. Duffing osilatörünün (a) zaman serisi ve (b) faz uzayı [48].

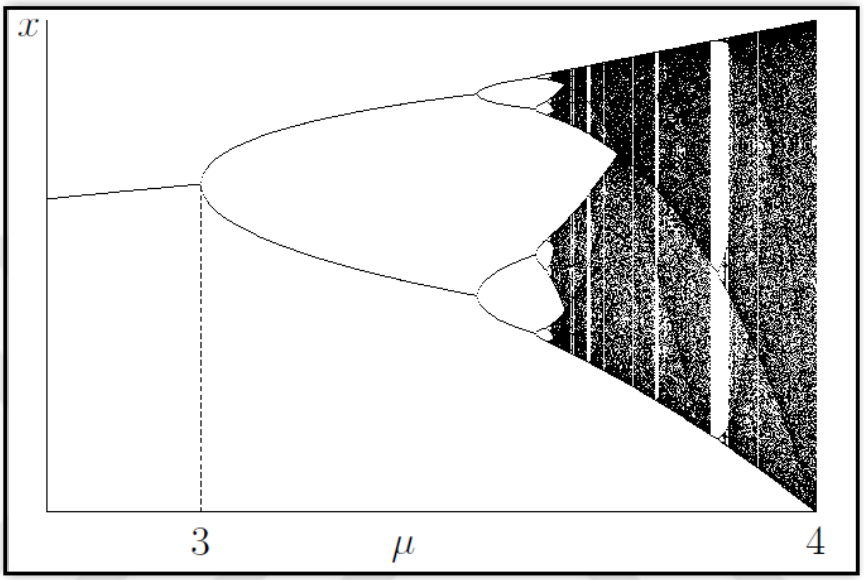
Sistemlerin kendi davranışlarını niteliksel olarak değiştirdiği parametre değerlerinde çatallaşmalar (bifurcation) meydana gelir. Nicelikten farklı olarak niteliksel değişim sistem kararlılığının değişimini ifade etmektedir. Yani çatallaşmalar sistemin kararlılığını yitirmesiyle ortaya çıkar. Çatallaşmaların meydana geldiği değerlere ise çatallaşma noktaları denir. Çatallaşma diyagramları, sistemlerin bir parametreye nasıl bağlı olduğunu grafiksel olarak gösteren diyagramlardır. Örneğin, Denklem 2.2'de verilen Lojistik sistemin $x \in [0, 1]$ değişkeni , $\mu \in [2.4, 4]$ parametresi ve $n = 0, 1, 2,\ldots$ için çatallaşma diyagramı Şekil 2.5'de verilmiştir [49].

$$x_{n+1} = \mu x_n(1-x_n) \tag{2.2}$$

Şekil 2.5'de görüldüğü üzere çıkış; $3 \leq \mu \leq 4$ şartını sağlayan μ değerine çok hassas bir biçimde bağımlıdır. Eğer çatallaşma diyagramı kullanılmazsa $3 \leq \mu \leq 4$ şartını sağlayan her bir μ değeri için ayrı ayrı grafik çizilmesi gerekir. Bu yaklaşım ise tatmin edici bir analiz biçimi değildir [49].

Faz uzayında, yörüngeleri kendine doğru çeken nokta veya noktalara çeker denir. Kaotik olmayan sistemlerde noktalar, çemberler veya düz yüzeyler gibi çekerler bulunurken, kaotik sistemlerde kararsız davranış gösteren yörüngeler, periyodik olarak

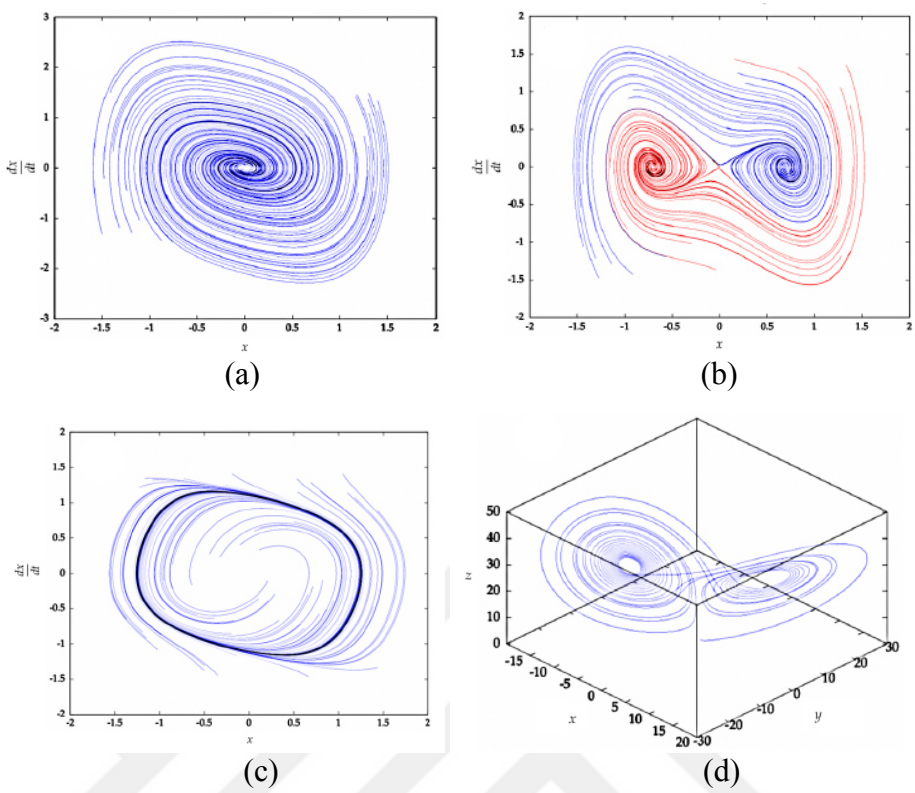
hareket etmemesine rağmen belirli bir şekil boyunca gezinirler. Yani sistem kaotik olmasına rağmen faz uzayındaki çekerinin ana hatları bellidir. Böyle garip bir davranış sergiledikleri için kaotik çekerlere "*garip çekerler*" de denir [2, 50]. Şekil 2.6'da çeşitli sistemlere ait çekerler verilmiştir.



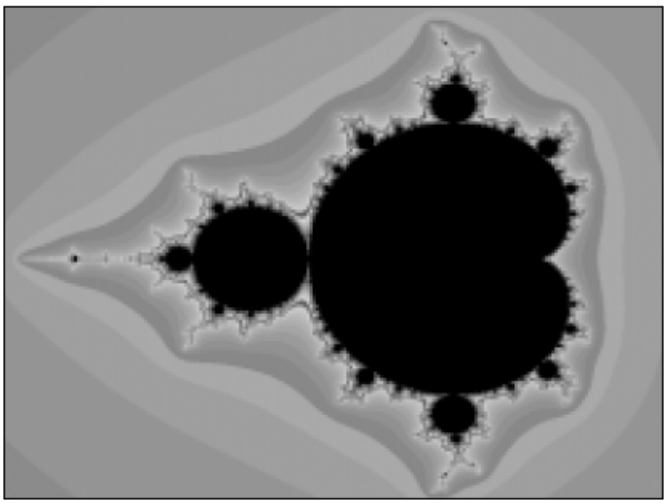
Şekil 2.5. Lojistik sistem çatallaşma diyagramı [49].

Her ne kadar genel geçer bir tanımı olmasa da, yakınlaştıkça basitleşmeyen geometrik şekillere fraktal denir. 1960'larda IBM'de matematikçi olarak çalışan Mandelbrot tarafından sunulan fraktalların; tam sayı değil kesirli boyuta sahip olmaları, çevre uzunluklarının ölçülememesi, sürekli olmalarına rağmen türevlerinin alınamaması ve ölçek değişimine rağmen yapılarının aynı kalması gibi çeşitli özellikleri vardır. Öklid geometrisinin klasik mekaniği açıklamak için üstlendiği rolü, kaos teorisi için fraktallar üstlenmiştir. Öyle ki fraktallara benzer bir biçimde kaotik çekerler de, çeşitli zaman ve uzunluk ölçekleri için genellikle tekrar eden bir yapı sergilerler [46, 51].

En çok bilinen fraktal yapılarından birisi Mandelbrot kümesidir. Karmaşık düzlemdeki $z(n+1) = z(n)z(n)+c$ dönüşümünde sabit bir c sayısı için $\{z(n) : n \in N\}$ yörüngesini sonlu yapan $z(0)$ elemanlarının oluşturduğu kümeye Julian dolgusu denir. Merkezden kopmuş parçaların atıldığı tek parça halindeki her Julian dolgusu ise bir Mandelbrot kümesidir. Kenar uzunluğu sonsuz olan Mandelbrot kümesinin alanı, tam olarak bilinmemektedir. Şekil 2.7'de bir Mandelbrot kümesi görülmektedir [52].



Şekil 2.6. (a) Yörüngelerin tek bir noktaya doğru çekildiği sabit noktalı kaotik olmayan çeker (b) Yörüngelerin iki farklı noktaya doğru çekildiği iki sabit noktalı kaotik olmayan çeker (c) Yörüngelerin halka benzeri bir çeker setine doğru çekildiği kaotik olmayan limit çevrim çekeri (d) Lorenz sistemine ait kaotik çeker [53].



Şekil 2.7. Mandelbrot kümesi [52].

2.3. Kaotik Zaman Serisi Analizi

Kaotik zaman serisi analizinde kullanılacak olan sinyal, sabit uzaysal (spatial) noktada ölçülen tekil bir skaler veridir. Başlangıç zamanı t_0 , örnekleme zamanı ise τ_s alındığında ölçülen bu değere $s(t_0+n\tau_s) = s(n)$ denebilir. Sinyalimiz doğrusal olmayan bir devredeki gerilim veya akım, akışkan dinamiğindeki yoğunluk, hız veya atmosferik bir alandaki sıcaklık olabilir [50].

Analizi yapılacak olan sinyalde gürültünün olup olmadığı tespit edilmeli, gürültü varsa azaltılmalıdır. Doğrusal sistemlerin analizinde, bu işlem, Fourier uzayında (domain) geniş bantlı gürültünün içinden keskin, dar bantlı sinyalleri ayırarak yapılabilir ama kaotik sistemlerin analizinde sinyal ve gürültü birbirinden zaman domeninde ayrılmalıdır. Bu ayrımın yapılabilmesi için üst üste binmiş sinyallerin, birbirlerinden farklarının görülebilmesini sağlayacak şekilde karakterize edilmesi gerekmektedir [50]. Doğrusal olmayan sinyali, gürültüden ayırma işlemleri, aşağıda belirtilen üç ayrı durum için farklılık gösterir [3, 50];

- Sinyalin alındığı sistemin denklemleri biliniyorsa,
- Daha önce ölçülmüş gürültüsüz sinyal mevcutsa,
- Ne sistem denklemleri ne de gürültüsüz sinyal mevcutsa,

Gözlemlenen gürültülü sinyalin $s(n) = s_1(n) + s_2(n)$ şeklinde olduğunu varsayalım. Burada $s_1(n)$ gürültüsüz sinyal, $s_2(n)$ ise ayırmak istenilen gürültü olsun. Gürültüsüz sinyalin alındığı kaynak sistemin denklemleri biliniyorsa, denklemler herhangi bir başlangıç şartıyla başlatılıp koşturularak sistemin özellikleri ile ilgili bilgiler elde edilebilir fakat kaotik sistemler, başlangıç şartlarına hassas bağlı olduğu için gürültüsüz sinyal dizisi, gürültülü sinyalin içerisinden tam olarak tespit edilemeyebilir. Çünkü denklemler için belirlenen başlangıç koşulları ile gözlemlenen sinyalin başlangıç koşulları farklı olabilir. Bundan dolayı denklemleri bilinen sistemin yörüngesine en yakın yörüngeyi, gürültülü sinyal içerisinden seçme amacı taşıyan bir kalite veya doğruluk (accuracy) fonksiyonu kullanılır. Daha önceden ölçülmüş bir sinyalin varlığı durumda ise temiz sinyal, referans olarak kullanılır. Son durumda, yani eğer sistem

denklemleri ve gürültüsüz sinyal mevcut değil ise yerel polinom haritaları yapılır veya doğrusal filtreler, yerel ya da global şekilde kullanılır [50].

2.3.1. Faz uzayının yeniden inşası

Belousoz-Zhabotinsky kimyasal reaksiyonları ile ilgili bir araştırmada 20' den fazla diferansiyel denklemden üretilen zaman serileri incelenmiş ve içlerinden birinin sistemin kaotik yapısını tamamıyla açıklayabildiği gösterilmiştir. Faz uzayını yeniden inşa etme düşüncesi, bu örnekte de görüldüğü gibi, incelenecek tek bir zaman serisinin sistemdeki diğer değişkenlerden izler taşıyabileceği olgusuna dayanmaktadır [54].
Faz uzayının yeniden inşası için Packard ve arkadaşları tarafından kullanılan türev koordinatları metodu veya Takens tarafından geliştirilen zaman gecikmeli koordinatlar metodu tercih edilmiştir [3]. Türevler gürültüye duyarlı olduğu için türev koordinatları metodu deneysel veriler için çok kullanışlı değildir [55]. Bundan dolayı bu çalışmada sadece tek bir değişkeni izleyip, zaman gecikmeli kopyalar kullanılarak, sistem dinamiklerini hesaplamayı mümkün kılan zaman gecikmeli koordinatlar metodu tercih edilecektir. Gözlemlenen bir dinamik sistemin faz uzayının yeniden inşası için zaman gecikmeli koordinatların kullanılması fikri, eş zamanlı olarak Kaliforniya Üniversitesinden bir grup (Packard ve arkadaşları) ve David Ruelle tarafından ortaya atılmıştır. Metodla ilgili önce Takens daha sonra da Sauer ve arkadaşları tarafından iyileştirmeler önerilmiştir [50].

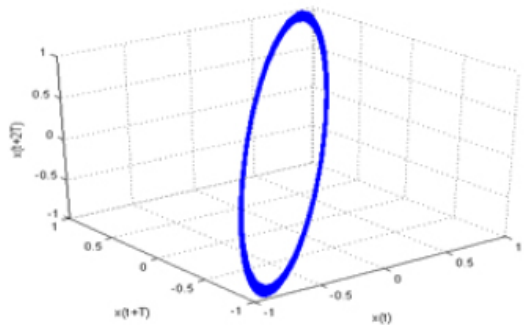
Takens teoremine göre eğer bir zaman serisi, d bir tamsayı olmak üzere d boyutlu bir çekerin elemanı ise, bu çekerin topolojik özellikleri, $m \geq 2d+1$ olmak üzere m boyutlu bir faz uzayında sinyalin gecikmeli kopyalarından oluşan çekerin topolojik özelliklerine eşittir [3].

$$\vec{y}_i = (x(i\Delta t), x(i\Delta t + T), x(i\Delta t + 2T),...,x(i\Delta t + (m-1)T)) \quad (2.3)$$

Denklem 2.3'deki T'ye zaman gecikmesi, m'ye ise o sistemin gömme boyutu denir. Farklı T ve m değerleri için farklı çekerler oluşur [3].

Bir periyodik sistemin kararlı olması durumunda faz uzayındaki eğrileri, "*limit döngü*" de denen kapalı bir göz şeklindedir. Çekeri, limit döngü şeklinde olan sistemlerin

eğrilerinin yakınsama ve ıraksama oranını gösteren Lyapunov üsteli sıfırdır. Lyapunov üsteli, sıfırdan büyük olan kaotik sistemlerin çekerlerinde ise yörüngeler aynı nokta üzerinden birden fazla kez geçmezler ve tüm faz uzayını doldurmazlar. Şekil 2.8'de faz uzayında bir limit döngü görülmektedir [3].



Şekil 2.8. Faz uzayında bir limit döngü [3].

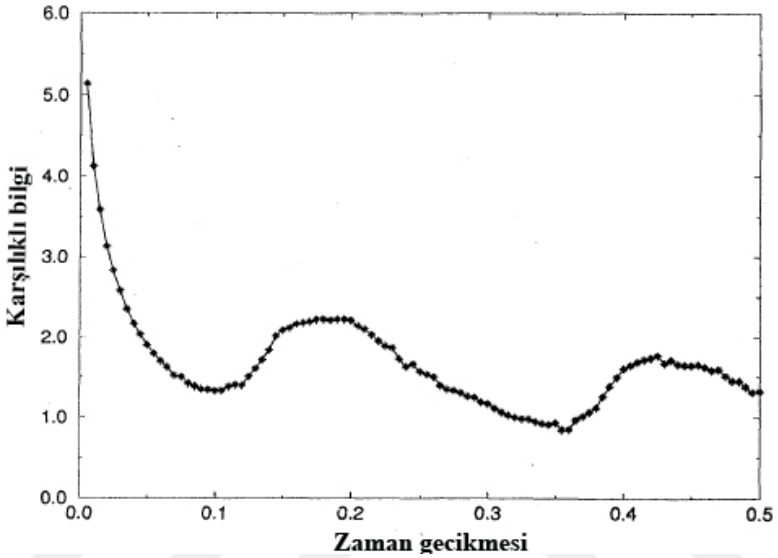
2.3.2. Zaman Gecikmesinin Seçilmesi

Faz uzayının yeniden inşası esnasında zaman gecikmesinin doğru olarak seçilmesi önem arz etmektedir. Şayet zaman gecikmesi T, olması gerekenden küçük seçilirse gömülen zaman serisinin komşu iki koordinatı birbirine çok yakın olacağından birbirlerinden ayırt edilemezler, buna karşın zaman gecikmesinin çok büyük seçilmesi durumunda ise istatistiksel olarak birbirinden bağımsızlaşan koordinatlar üzerinde oluşan çeker yörüngesi, birbiriyle ilişkisi olmayan iki yön üzerinde çizdirilmiş olur [3, 50].

Doğrusal olmayan sistemlerde gecikme değerinin uygun şekilde seçilmesi için ortalama karşılıklı bilgi (average mutual information) fonksiyonu kullanılır. A ve B iki sistem, a_i ve b_k bu sistemlerde ölçülmesi muhtemel değerler olmak üzere, tüm A setinde a'nın ölçülme olasılığı $P_A(a)$, tüm B setinde b'nin ölçülme olasılığı $P_B(b)$ ve benzer şekilde a ve b'nin ölçülme olasılığı $P_{AB}(a,b)$ ise ortalama karşılıklı bilgi, aşağıdaki şeklide tanımlanır [50];

$$I_{AB}(T)=\sum_{a_i,b_k} P_{AB}(a_i,b_k)\log_2\left[\frac{P_{AB}(a_i,b_k)}{P_A(a_i),P_B(b_k)}\right] \tag{2.4}$$

Fraser ve Swinney yaptıkları çalışmada [56] karşılıklı bilginin T zaman gecikmesine göre çizdirildiği grafikteki ilk minimum değerin, seçilmesi gereken uygun zaman gecikmesi olduğunu göstermişlerdir. Şekil 2.9'da Lorenz sistemi için zaman gecikmesinin fonksiyonu olarak ortalama karşılıklı bilgi grafiği gösterilmiştir [50].



Şekil 2.9. Lorenz sistemi için zaman gecikmesinin fonksiyonu olarak ortalama karşılıklı bilgi [50].

2.3.3. Gömme Boyutunun Seçilmesi

Faz uzayının yeniden inşasındaki amaç, incelenen sistemin görünümündeki tüm belirsizlikleri gidermeye yetecek kadar büyük bir Öklid uzayı oluşturmaktır. Böylelikle izlenilen boyutta, birbirinin üzerine binmiş veya çok yakın olan alakasız iki nokta birbirinden ayrılmış olur. Örneğin 1 boyutlu çizgi halindeki bir çekerin 2 boyutlu bir uzayda izlendiğini düşünelim. Bu durumda çizginin kendisini kesip kesmediği anlaşılamaz. Bu durum anlaşılamadığı için de birbirine yakın iki noktanın gerçekten komşu olup olmadığı bilinemez. Bu belirsizlikler, çekerin 3 boyutta çizdirilmesiyle son bulur. Çeker daha yüksek bir boyutta da görülebilir; fakat boyut arttıkça hesaplama maliyeti üstel olarak artmaktadır ve bunun yanı sıra incelenecek dinamiklerin bulunmadığı gereksiz yere artırılmış boyutta gürültü bulunabilir. Bu nedenlerle tüm belirsizlikleri giderecek en uygun boyutun seçilmesi önemlidir ve seçilecek bu boyutun tam sayı olması zorunlu değildir [50].

Gömme boyutunun seçilmesi için, tekil değer analizi, sistem değişmezlerinin doyumu, doğru vektör alanları ve yanlış en yakın komşular metodları kullanılabilir [50]. Sık kullanılan yöntemlerden biri olan yanlış en yakın komşular (false nearest neighbors) metodu incelendiğinde;

$$y(k) = [x(k), x(k + T),..., x(k + (m -1)T)] \tag{2.5}$$

Eşitliğinde y(k), m boyutundaki bir vektör ve y^{NN} (k) bu vektörün en yakın komşusu olmak üzere, bu iki nokta arasında ki Öklid farkı $R_m(k)$ şöyle ifade edilir;

$$R_m(k)^2 = \left[x(k)-x^{NN}(k)\right]^2+\left[x(k+T)-x^{NN}(k+T)\right]^2+\ldots+\left[x\left(k+T(m-1)\right)-x^{NN}(k+T(m-1))\right]^2 \tag{2.6}$$

$R_m(k)$'nın m+1'inci boyuttaki Öklid farkı ile şöyle bir ilişkisi vardır;

$$R^2_{m+1}(k) = R^2_m(k)+\left[x(k+mT)-x^{NN}(k+mT)\right]^2 \tag{2.7}$$

Buradan hareketle R_T önceden belirlenen eşik değeri olmak üzere;

$$\frac{|x(k+mT)-x^{NN}(k+mT)|}{R_m(k)} > R_T \tag{2.8}$$

Eşitsizliği var ise, yani belirlenen eşik değer aşılmışsa, k noktasındaki komşuların yanlış olduğu değerlendirilir. Eşik değerin aşılmadığı durumda ise, komşuların doğru olduğu ve bu boyutun gömme boyutu olduğu anlaşılır. Eşik değerinin seçimine bağlı olarak farklı gömme boyutlarının elde edilmesi bir dezavantaj olduğundan Cao'nun bir çalışmasında uygun eşik belirlemek için bir yöntem sunulmuştur [3].

Kısaca özetlenecek olursa; yanlış en yakın komşular tekniğinde belirli iki nokta arasındaki fark, gözlem altında tutularak boyut artırılır. Boyutun artması ile farkta artıyorsa henüz istenilen gömme boyutuna ulaşılmadığı anlaşılır. Fakat boyut artmasına

rağmen noktalar arasındaki fark artmıyorsa artık boyutu artırmanın bir anlamı olmadığı ve ulaşılan boyutun gömme boyutu olduğu sonucuna varılır.

2.3.4. Durağanlık Analizi

İster doğrusal olsun, ister olmasın zaman serisi analizlerinde ölçülecek sistem için bir çeşit durağanlık varsayımında bulunmak zorunludur. Bundan dolayı sistemin durağan olmaması, yani ölçüm süreci esnasında dinamiklerin değişmesi istenmeyen sonuçlara yol açar. Durağanlığın tespit edilebilmesi için çeşitli teknikler geliştirilmiştir. Schreiber tarafından sunulan çapraz tahmin tekniğinde, zaman serisinin parçaları birbirleriyle karşılaştırılarak, bir parçadan diğer parçanın tahmin edilebilirliğinin göstergesi olan çapraz tahmin hatası (cross-prediction error) veri tabanında kayda alınıp, analiz edilir [57].

2.3.5. Vekil Veri Metodu (Surrogate Data Method)

Bazı durumlarda kaotik olmayan zaman serileride sanki kaotikmiş gibi görülebilir. Böyle durumlarda, ölçülmüş değerlerden oluşan zaman serilerinin doğrusal olup olmadığını tespit edebilmek için çeşitli yöntemler geliştirilmiştir. Bu yöntemlerin en çok kullanılanlarından biride Theiler ve arkadaşları tarafından geliştirilen Vekil Veri metodudur [59]. Bu teknikte öncelikle mevcut verilerden gözlemlenebilir ve doğrusal olmayan bir λ değeri hesaplanır. Bu λ değeri korelasyon boyutu, Lyapunov üsteli, tahmin hatası vb. olabilir. Daha sonra λ baz alınarak bir hükümsüz hipotez (null hypothesis) belirlenir. Doğrusal bir yapıya sahip olan bu hipotez, orjinal verileri test etmek için kullanılacak vekil verilerle bağıntılıdır. Vekil veriler, doğrusallığı test edilecek zaman serileri ile istatistiksel olarak aynı özellikte ama determinizmi ayıklanmış verilerdir. Bu verilerin oluşturulabilmesi için eldeki seri kullanılarak üretilen çeşitli seriler rastgele olarak karıştırılır. Oluşturulan vekil veriler önceden belirlenen bir oranda (örneğin %95) test edilen verilerle uyuşmuyorsa, sistemin doğrusal olmadığı anlaşılır [3, 54, 58, 59].

2.4. Boyut Kavramı

Bir çeker Lyapunov üsteli ve boyut gibi bazı kavramlar kullanılarak nitelendirebilir. Bu kavramlar, çekerin içinde bulunduğu koordinat sisteminden ve yörüngenin başlangıç

şartlarındaki değişiminden bağımsız olduğu için, deneysel veriyi değerlendirmede kullanılabilir. Ayrıca somut birer sayı olduklarından dolayı zaman serilerinin birbirleriyle karşılaştırılması için elverişlidirler [60]. Lyapunov üsteli, bu çalışmanın ileri safhalarında ayrıntılı bir şekilde anlatılacak olup bu bölümde boyut kavramına değinilecektir.

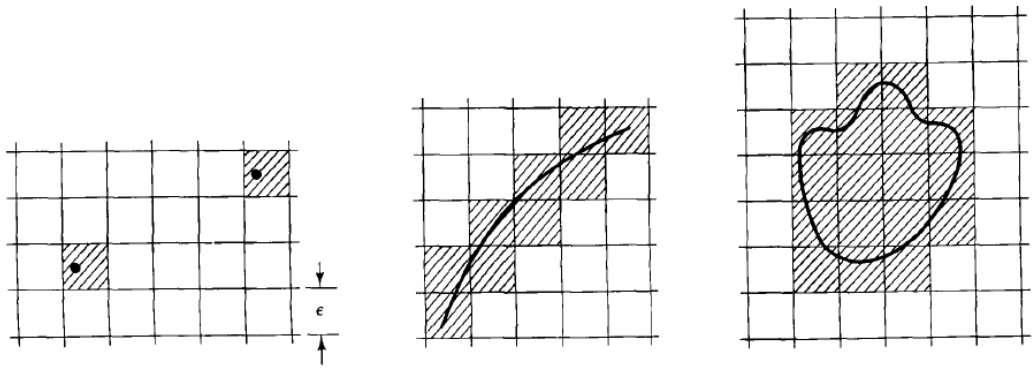
Boyut, geometrik bir cismin öz benzerliğinin ölçüsüdür. Noktalar sıfır, çizgiler bir, yüzeyler iki boyutlu iken kaotik çekerler genelde kesirli boyuta sahiptirler. Garip çekerlerin boyutunu belirlemek için çeşitli yollar var olup, kutu sayım boyutu (box-counting dimension), bilgi boyutu (information dimension) ve korelasyon boyutu (correlation dimension) sayısal hesaplama için uygun boyut türleridir [60].

Kapasite boyutu olarak da bilinen kutu sayım boyutunda, çeker yörüngesini kaplamak için ne kadar kutu gerektiği sayılır. Kutuların sayılması ile çekerin geometrik özellikleri hakkında bilgi edinilirken, yörüngenin bir kutuya ne kadar sıklıkla uğradığı bilinmez. Yani bazı kutular daha yoğun yörünge barındırırken bazılarından yörünge sadece bir defa geçiyor olabilir. Yörünge sıklığı bilgisinden yoksun olunması kutu sayım boyutunun dezavantajlarından biridir. Ayrıca kutu sayma sürecinin fazla işlem gücü gerektirmesi, yöntemin diğer bir dezavantajıdır [61, 62].

N boyutlu bir kartezyen uzayında, her bir kenarı ε uzunluğunda N boyutlu küpler (eğer N=2 ise kareler) bulunduğu düşünülürse, git gide küçülen ε değerine sahip $\tilde{N}(\varepsilon)$ küplerinin sayılması gerekir. Bu durumda kutu sayım boyutu [63];

$$D_0 = \lim_{\varepsilon \to 0} \frac{\ln \tilde{N}(\varepsilon)}{\ln (1/\varepsilon)} \tag{2.9}$$

Olarak verilir. Şekil 2.10'da kutu sayımı ile ilgili örnek bir çizim bulunmaktadır.



(a) (b) (c)

Şekil 2.10. a) İki nokta için b) Bir eğri parçası için c) Kapalı bir eğrinin içinde kalan alan için $\tilde{N}(\varepsilon)$ kutuları [63].

Bilgi boyutu, kutu sayım boyutunun yörüngelerin kutuları ziyaret sıklığını da belirten, genelleştirilmiş bir halidir. Bu boyut türünde bir kutunun konumunu belirli hassasiyette tanımlamak için kaç bit bilgi gerektiği hesaplanır. Bir çizgi üzerindeki bir noktayı $r = 2^{-k}$ doğruluğu (accuracy) ile belirlemek için k bit bilgi gerekir. Kare üzerindeki noktayı aynı doğruluk ile belirlemek için 2k (her eksen için bir k), aynı şekilde küp için 3k bit bilgi gerekir. Genellenecek olursa d boyutlu bir kutunun r doğruluğunda konumunu belirlemek için $S(r) = -d \log_2 r$ bit bilgi gerekir [61, 62].

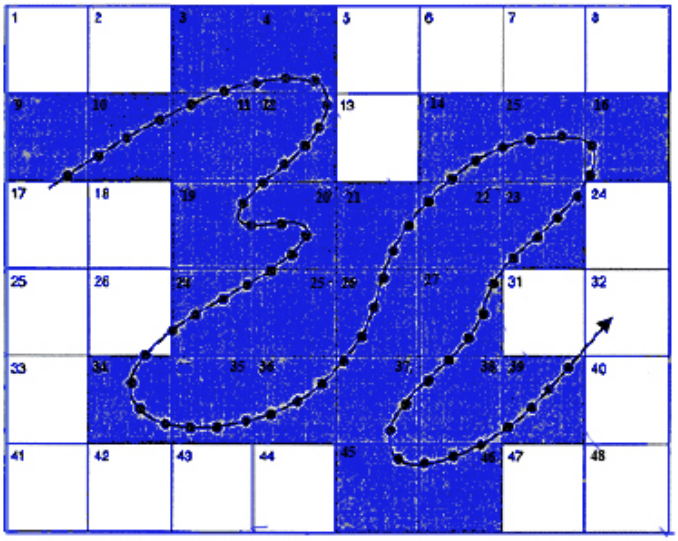
Gerekli bilgi biti değeri S(r) aynı zamanda yörüngenin kutulara denk gelme olasılıkları ile de ilişkilidir. Bir kutuyu belirlemek için gerekli olan ortalama bilgi miktarı Denklem 2.10'daki gibidir.

$$S(r) = -\sum_i P_i \log_2 P_i \tag{2.10}$$

Burada P_i i. kutunun olasılık ölçeğidir. Böylece bilgi boyutu aşağıdaki gibi tanımlanabilir [62];

$$D_I = \lim_{r \to 0} \frac{-S(r)}{\log_2 r} = \lim_{r \to 0} \frac{\sum_i P_i \log_2 P_i}{\log_2 r} \tag{2.11}$$

Bilgi boyutu hesaplanırken her kutu eşit olasılığa sahip olursa, bilgi boyutu kutu sayım boyutuna eşit olur, ama genellikle durum böyle değildir ve bir çekerin kutu sayım boyutu, bilgi boyutundan büyüktür [64]. Bir yörüngenin 48 olası durumlu bir faz uzayındaki kutuları ziyaret sıklığının değişken olabildiği örnek bir çizimle Şekil 2.11'de gösterilmiştir [2].



Şekil 2.11. 48 Olası durumlu bir faz uzayı ve eşit ayrık zamanlarla örneklenmiş bir yörünge [2].

Grassberger ve Procaccia tarafından geliştirilen ve bilgi boyutuna göre hesaplaması daha kolay olan korelasyon boyutu ise bir diğer popüler boyut türüdür. Rastgele seçilen iki noktanın birbirlerine belirli bir mesafe uzaklıkta olma olasılığını ölçen korelasyon boyutu, çekerin faz uzayını nasıl doldurduğunun değerlendirebilmesini sağlayacak olan noktaların dağılma şeklini açıklayabilir [60, 65]. Çeker, her biri ε yarıçapında hücrelerle kaplandığında $P_i(\varepsilon)$, i. hücrede bir nokta bulunma olasılığını gösteriyorsa, korelasyon boyutu;

$$D_C = \lim_{\varepsilon \to 0} \sum_i \frac{\log\left(\sum_i P_i(\varepsilon)^2\right)}{\log(\varepsilon)} \qquad (2.12)$$

olarak tanımlanır. Buradaki $\sum_i P_i(\varepsilon)^2$ değeri, ε yarıçapındaki bir hücrede bir nokta çiftinin bulunma olasılığıdır ve bu değeri hesaplamak için işlemsel açıdan verimli bir algoritma olan Grassberger-Procaccia algoritması kullanılabilir [65].

3. BÖLÜM
LYAPUNOV ÜSTELLERİ

3.1. Giriş

Dinamik bir sistemin, Lyapunov üstelleri ve özellikle de en büyük Lyapunov üsteli, o sistemin kaotikliğini tanımlamak için en çok kullanılan yollardan biridir. Rus matematikçi Aleksandr Mikhailovich Lyapunov tarafından litaretürde sunulan Lyapunov üstelleri, birbirine yakın iki komşu yörüngenin zamanla nasıl farklılaştığını gösterir [2]. Pozitif bir Lyapunov üsteline sahip olan iki yakın yörünge, hızlı bir şekilde birbirinden ayrılır ve farklı davranışlar sergilerler. Yani pozitif Lyapunov üsteline sahip bir sistem, başlangıç şartlarına hassas bağımlılık gösteren bir yapıdadır ve kaotiktir. Negatif Lyapunov üsteline sahip bir sistemde ise farklı başlangıç şartlarına sahip yörüngeler sabit bir noktaya veya yörüngeye doğru çekilecek ve kaotik davranış sergilemeyeceklerdir [2].

3.2. Tanım

Lyapunov üstelleri, birbirine yakın iki komşu yörüngenin zaman içindeki yakınsama veya ıraksama oranıdır. Faz uzayındaki ayrılma, vektör olarak ifade edilip, $\delta X(0)$ olarak gösterilirse t kadar zaman sonra Denklem 3.1'de verilen ifade sağlanmış olur.

$$\|\delta X(t)\| = e^{\lambda t}\|\delta X(0)\| \tag{3.1}$$

Burada $\delta X(t)$ ayrılma vektörü ve λ ise Lyapunov üstelidir [66].

3.3. Diferansiyel Denklemlerin Çözümü

Mühendislikteki birçok olgu, türevlendiği takdirde daha kolay ifade edilebilmektedir. Örneğin Denklem 3.2'de tarif edilen Newton'un ikinci kanununa göre m kütleli bir cisim a ivmesine sahipse üzerine F kuvveti etki eder.

$$F = ma \tag{3.2}$$

İvmenin hızın zamanla değişimi olduğunu ve hızında birim zamandaki yer değiştirmeye eşit olduğunu bildiğimizden;

$$a = \frac{dv}{dt} = \frac{d^2s}{dt^2} \tag{3.3}$$

şeklinde yazılabilir. Denklem 3.3 yeniden yazılırsa;

$$\frac{f}{m} = a = \frac{d^2s}{dt^2} \tag{3.4}$$

şeklinde adi bir diferansiyel denklem haline getirilebilir [67].

Bu tez çalışmasında incelenen Lorenz ve Rössler sistemleri ve Chua devresinin matematiksel modelleri, diferansiyel denklemlerden oluşmaktadır. Sistemlerin en büyük Lyapunov üstellerini hesaplayabilmek için öncelikle bu sistemleri oluşturan diferansiyel denklemlerin analitik veya nümerik yöntemlerle çözülmesi gerekmektedir [68].

Diferansiyel denklemleri nümerik olarak çözmek için çeşitli metotlar bulunmaktadır. Bu yöntemlerden bir tanesi, denklemleri Taylor Serisine açmaktır. Bu arada Taylor serisi yöntemi, Euler ve Runge Kutta metotlarının temelini teşkil eder. Diferansiyel denklemi Taylor Serisinin birinci dereceden terimine kadar açmaya Euler metodu, 4. dereceden terimine kadar açmaya 4. dereceden Runge Kutta metodu, 8. dereceden terimine kadar açmaya ise 8. dereceden Runge Kutta metodu denir. Diferansiyel denklem sistemlerini Taylor Serisine açarken terim sayısı arttıkça ideal sonuca daha da yaklaşılmakta fakat işlem yükü ve karmaşıklığı artmaktadır [68].

Bu çalışmada Lorenz ve Rössler sistemleri için 4. Dereceden Runge Kutta metodu, Chua devresi için ise 8. Dereceden Runge Kutta metodu kullanılmıştır.

3.4. Runge Kutta Metotları

Runge Kutta yöntemiyle diferansiyel denklemlerin çözümü bu tez çalışmasının ana omurgasını teşkil etmediğinden, detaylandırılmadan sadece metotların tanımı yapılacaktır. Taylor serisi yöntemi, Euler ve Runge Kutta metodları hakkında ayrıntılı bilgi literatürde mevcuttur [67-69].

Runge Kutta metodlarının s tamsayı $a_{21}, a_{31}, a_{32} \ldots a_{s1}, a_{s2} \ldots a_{s,\,s-1}$, $b_1 \ldots b_s$, $c_2 \ldots c_s$ reel katsayılar olmak üzere genel formu aşağıdaki gibidir;

$$k_1 = f(x_n, y_n)$$
$$k_2 = f(x_n + c_2 h,\ y_n + h a_{12} k_1)$$
$$k_3 = f(x_n + c_3 h,\ y_n + h(a_{31} k_1 + a_{32} k_2))$$
$$\vdots$$
$$k_n = f(x_n + c_s h,\ y_n + h(a_{s1} k_1 + \ldots + a_{s,s-1} k_{s-1})) \qquad (3.5)$$

Burada

$$y_{n+1} = y_n + h(b_1 k_1 + \ldots + b_s k_s) \qquad (3.6)$$

şeklinde tanımlanmıştır. h adım aralığı olup, a, b ve c katsayıları Butcher Tablosuna göre belirlenir. Butcher Tablosunun genel formu Tablo 3.1' deki, 4. Dereceden Runge Kutta metodu için Butcher Tablosu da Tablo 3.2' deki gibidir.

Tablo 3.1. Butcher Tablosu genel formu [69].

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
⋮	⋮	⋮	⋮		
c_s	a_{s1}	a_{s2}	…	$a_{s,\,s-1}$	
	b_1	b_2	…	b_{s-1}	b_s

Tablo 3.2. 4. Dereceden Runge Kutta Metodu için Butcher Tablosu[69].

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

Böylelikle 4. Dereceden Runge Kutta formülasyonu aşağıdaki gibi elde edilir;

$$k_1 = f(x_n, y_n)$$
$$k_2 = f(x_n+1/2\ h, y_n+1/2\ k_1)$$
$$k_3 = f(x_n+1/2\ h, y_n+1/2\ k_2)$$
$$k_4 = f(x_n+h, y_n+k_3) \qquad (3.7)$$

Ayrıca

$$y_{n+1} = y_n+h\left(\frac{1}{6}k_1+\frac{2}{6}k_2+\frac{2}{6}k_3+\frac{1}{6}k_4\right) \qquad (3.8)$$

şeklinde yazılabilir. Bu tez çalışmasında, Chua devresine ait diferansiyel denklemlerin nümerik çözümünde kullanılacak olan 8. dereceden Runge Kutta metodu için Butcher Tablosu Tablo 3.3‘ deki gibidir. Burada kullanılan w katsayıları Tablo 3.4’deki gibidir.

Aynı şekilde Denklem 3.5’deki Runge Kutta metodu genel formu k = 8 değeri için açılıp, Tablo 3.3’deki değerler a, b ve c şeklinde belirlenirse, 8. Dereceden Runge Kutta formülasyonu elde edilir. Bu tez çalışmasının ana konusu olmadığı için 8. Dereceden Runge Kutta formülasyonunun açık hali verilmemiştir.

3.5. Lorenz Sistemi

Bir meteoroloji uzmanı olan Edward Lorenz kaos teorisinin matematiksel altyapısına önemli bir katkı sunmuştur [43]. Ünlü “*Kelebek etkisi*” terimini ilk kez kullanan Lorenz [6], hava simülasyonu verilerini bilgisayarına girerken işlem rahatlığı olsun diye

nümerik değerlerin küsüratlarını virgülden sonra 3 haneye kadar yuvarlamış ve hiç beklemediği değişik sonuçlar elde edince başlangıçtaki küçük değişikliklerin sonuçta büyük farklara yol açabileceğini gözlemlemiştir [6]. Simülasyon modelini 3 adet adi diferansiyel denklem ile elde edilebilecek şekilde sadeleştiren Lorenz, bu çalışmasını "*Deterministic Nonperiodic Flow*" adıyla 1962'de literatüre sunmuştur [70]. Daha sonra Tucker, sistemin bir garip çekere sahip olduğunu ispatlamıştır [71].

Tablo 3.3. 8. Dereceden Runge Kutta yöntemi için Butcher Tablosu [68].

$½ - w_2$	w_1	$w_1' - w_3 + w_4'$	$w_1' - w_3 - w_4'$	$w_1 - w_5$
$½ - w_2'$	$w_1 - w_3' + w_4$	w_1'	$w_1' - w_5'$	$w_1 - w_3' - w_4$
$½ + w_2'$	$w_1 + w_3' + w_4$	$w_1' + w_5'$	w_1'	$w_1 + w_3' - w_4$
$½ - w_2$	$w_1 + w_5$	$w_1' + w_3 + w_4'$	$w_1' + w_3 - w_4'$	w_1
	$2w_1$	$2w_1'$	$2w_1'$	$2w_1$

Tablo 3.4. Tablo 3.3'de kullanılan w değerleri [68].

$w_1 = \frac{1}{8} - \frac{\sqrt{30}}{144}$	$w_1' = \frac{1}{8} + \frac{\sqrt{30}}{144}$
$w_2 = \frac{1}{2}\sqrt{\frac{15+2\sqrt{30}}{35}}$	$w_2' = \frac{1}{2}\sqrt{\frac{15-2\sqrt{30}}{35}}$
$w_3 = w_2(\frac{1}{6} + \frac{\sqrt{30}}{24})$	$w_3' = w_2(\frac{1}{6} - \frac{\sqrt{30}}{24})$
$w_4 = w_2(\frac{1}{21} + \frac{5\sqrt{30}}{168})$	$w_4' = w_2'(\frac{1}{21} - \frac{5\sqrt{30}}{168})$
$w_5 = w_2 - 2w_3$	$w_5' = w_2' + 2w_3'$

Lorenz sisteminde bulunan diferansiyel denklemlerin 3 değişkeni vardır. Bunlar Prandtl sayısı "σ", Rayleigh sayısı "r" ve sistemin fiziksel boyutuyla ilgili bir parametre olan "b" dir [70, 72]. Lorenz diferansiyel denklem sistemi;

$$\frac{dx}{dt} = x'$$
$$\frac{dy}{dt} = y' \qquad (3.9)$$
$$\frac{dz}{dt} = z'$$

olmak üzere

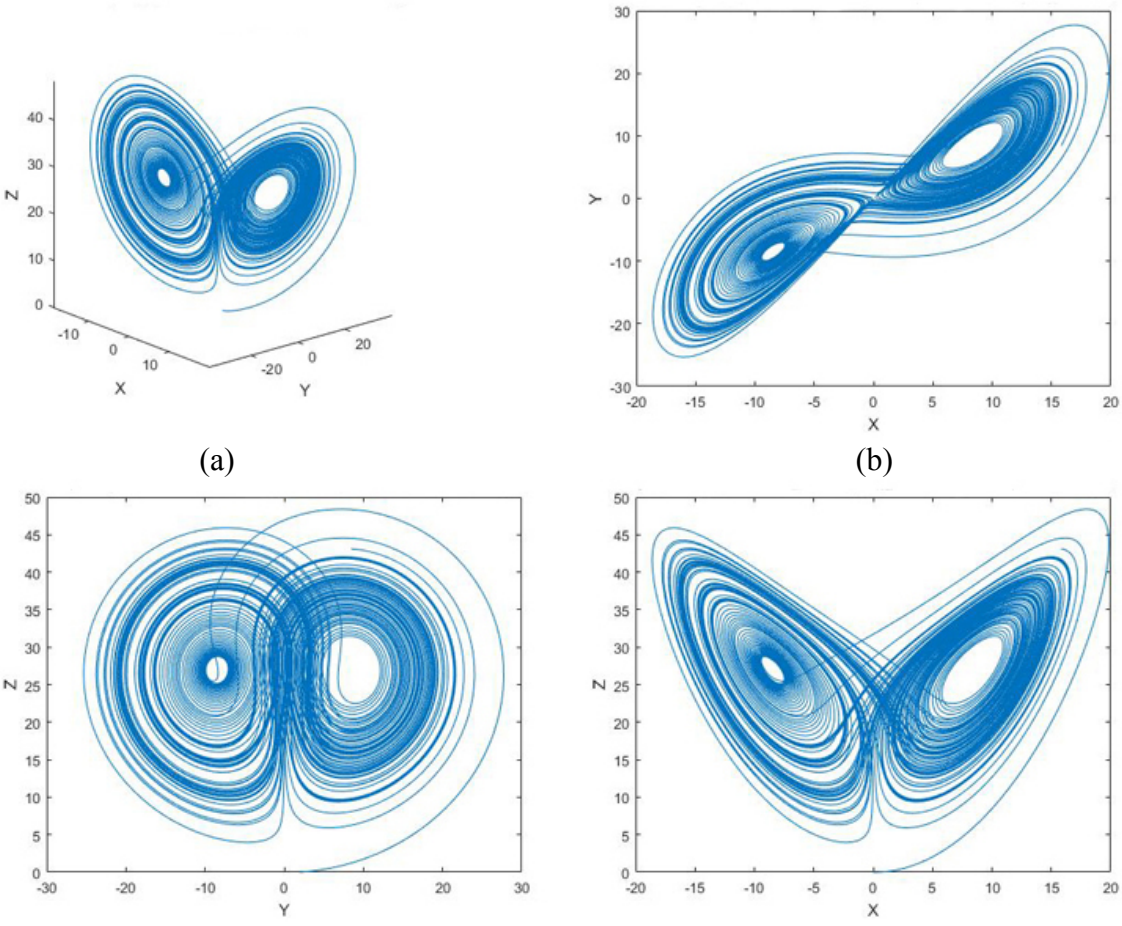
$$x' = \sigma (y - x)$$
$$y' = rx - y - xz \qquad (3.10)$$
$$z' = xy - bz$$

şeklindedir.

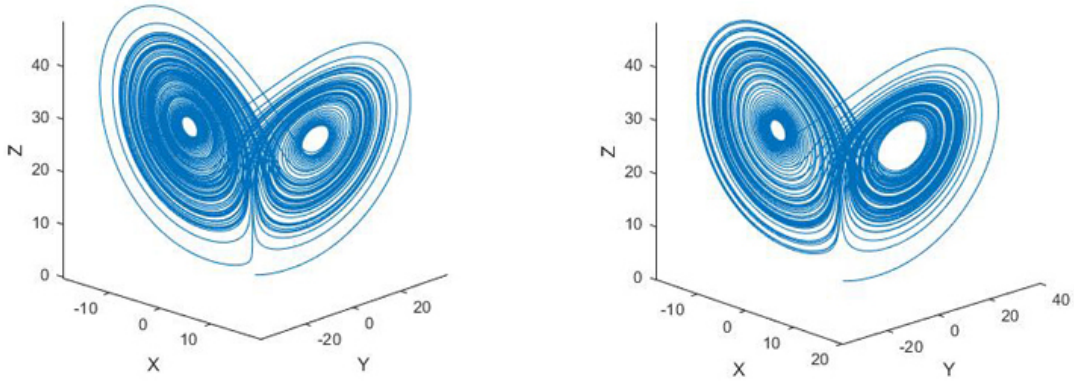
Denklem 3.10'da verilen diferansiyel denklem takımı, $\sigma = 10$, $r = 28$ ve $b = 8/3$ parametre değerleri ve $x_0 = 0$, $y_0 = 2$, $z_0 = 0$ başlangıç değerleri için çözdürülüp faz uzayına aktarılırsa Şekil 3.1'deki gibi bir çeker grafiği elde edilir.

Kaotik davranışın ana özelliklerinden birisinin başlangıç şartlarına hassas bağlılık olduğu daha önce belirtilmişti. Bu, Lorenz çekeri üzerinde de gözlemlenebilir. Eğer Lorenz sistemine ait diferansiyel denklem takımı, parametre değerleri değiştirilmeden ($\sigma = 10$, $r = 28$, $b = 8/3$) sadece başlangıç şartlarından birisi $x_0 = 0$, $y_0 = 1.01$, $z_0 = 0$ şeklinde %1' lik bir oranla değiştirilerek çözdürüldüğünde elde edilen yeni sonuçlar Şekil 3.2'de verilmiştir.

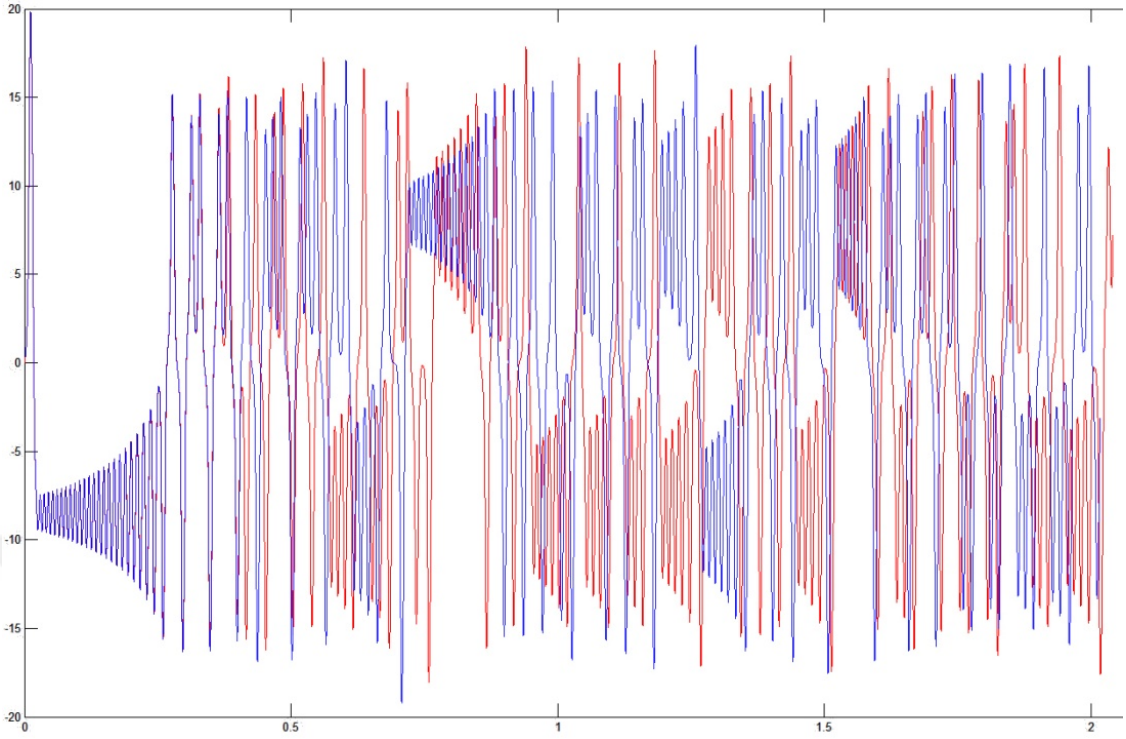
Şekil 3.1 ve Şekil 3.2 karşılaştırıldığında genel çeker formları benzerdir ama Şekil 3.3'de verilen zamana göre değişim grafikleri incelendiğinde başlangıç değerlerindeki ufak bir farktan dolayı sistemin zamana göre bambaşka bir yol izlediği açıktır. Burada $\sigma = 10$, $r = 28$, $b = 8/3$ alınmıştır. Ayrıca başlangıç şartları olarak $x_0 = 0$ ve $z_0 = 0$ alınmış fakat y_0 değeri ikinci durumda %1 farkla değiştirilmiştir. Buna göre $y_0 = 1$ değeri için mavi renkli görsel, $y_0 = 1{,}01$ için kırmızı renkli görsel elde edilmiştir.



Şekil 3.1. [0; 2; 0] başlangıç değerleri ve σ = 10, r = 28 ve b = 8/3 parametre değerleri için; (a) X-Y-Z düzlemindeki Lorenz çekeri (b) X-Y düzlemi Lorenz çekeri (c) Y-Z düzlemi Lorenz çekeri (d) X-Z düzlemi Lorenz çekeri [70].

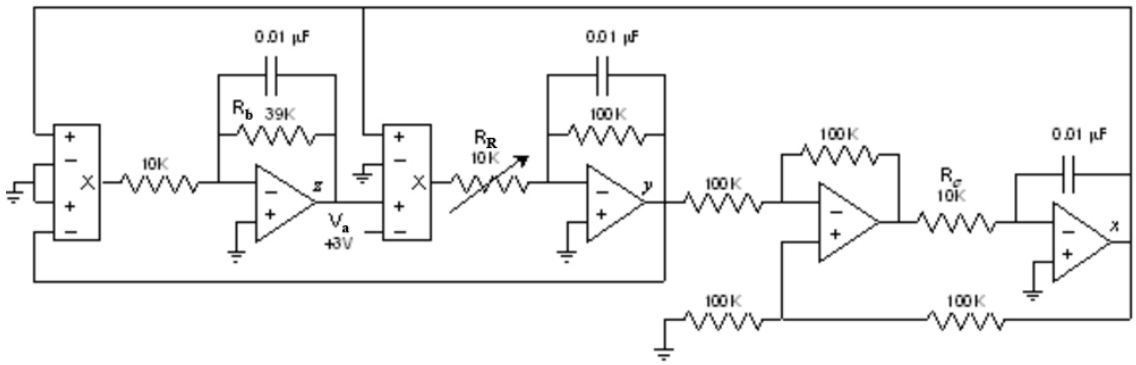


Şekil 3.2. X-Y-Z düzlemindeki Lorenz çekeri (Parametreler:σ = 10, r = 28, b = 8/3) (a) Başlangıç değerleri: x = 0, y = 1, z = 0 (b) Başlangıç değerleri: $x_0 = 0$, $y_0 = 1{,}01$, $z_0 = 0$ [73].



Şekil 3.3. Lorenz sisteminin $x_1(t)$ ve $x_2(t)$ grafiği

Lorenz sisteminin elektronik devre olarak gerçekleştirimi Şekil 3.4' de verilmiştir. Burada çarpıcılar AD633 entegre devresi, işlemsel yükselteçler ise TL082 entegre devresi veya muadili olup devre ±15V ile beslenmiştir [74].



Şekil 3.4. Lorenz sisteminin elektronik devre olarak gerçekleştirilmesi [74].

Lorenz sisteminin parametreleri olan r, σ ve b parametreleri, devredeki direnç değerleri ve V_a voltajı değiştirilerek aşağıdaki şekilde belirlenir;

$$r = \frac{10k\Omega}{aR_R}$$
$$\sigma = \frac{100k\Omega}{R_\sigma}$$
$$b = \frac{100k\Omega}{aR_b} \quad (3.11)$$

Burada a değeri;

$$a = \frac{V}{V_a} \quad (3.12)$$

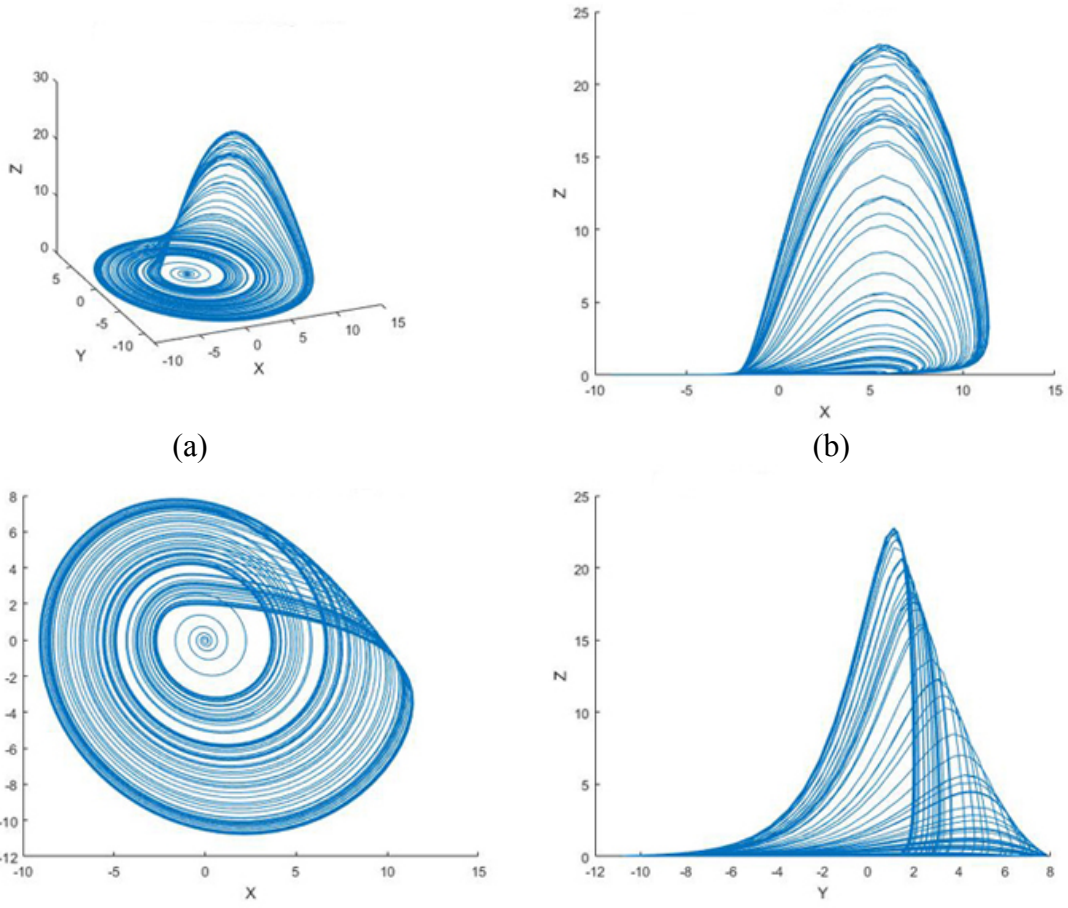
şeklinde ayarlanır [74].

3.6. Rössler Sistemi

Lorenz sistemine göre daha basit bir diferansiyel denklem takımı ile kaotik çeker üretmeyi başaran Alman bilim adamı Otto Rössler, bu çalışmasını 1976'da literatüre sunmuştur. Rössler sistemi şu şekilde tanımlanmıştır [75];

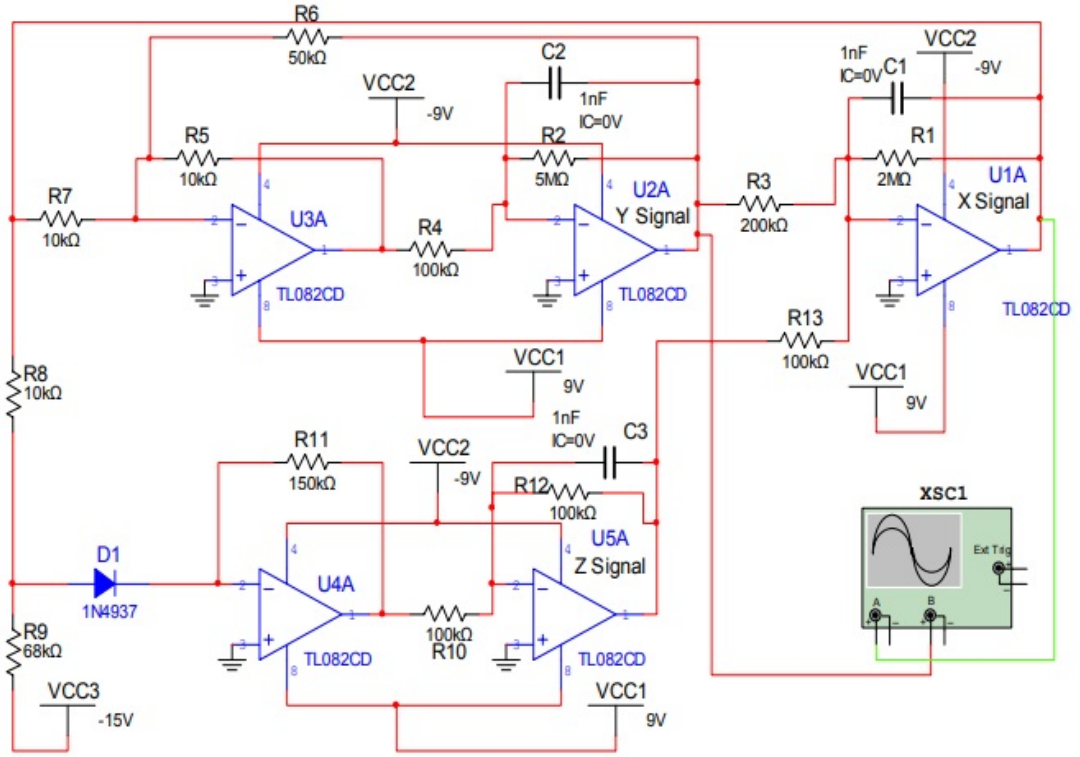
$$x' = -y - z$$
$$y' = x + ay \quad (3.13)$$
$$z' = b + z(x - c)$$

Üç boyutlu Lorenz ve Rössler sistemleri, eşit terim sayılarına sahip olmalarına karşın Lorenz sistemi 2 doğrusal olmayan terim barındırırken (xz ve xy), Rössler sistemi 1 doğrusal olmayan terim barındırmaktadır (zx). Rössler sisteminin a = 0,2, b = 0,2, c = 5,7 parametre değerleri ve [0.1; -0.1; 0.1] başlangıç değerleri için ürettiği çekerler Şekil 3.5'deki gibidir.



Şekil 3.5. [0,1; -0,1; 0,1] başlangıç değerleri ve a = 0.2, b = 0.2, c = 5,7 parametre değerleri için; (a) X-Y-Z düzlemindeki Rössler çekeri (b) X-Z düzlemi Rössler çekeri (c) X-Y düzlemi Rössler çekeri (d) Y-Z düzlemi Rössler çekeri [76].

Rössler sisteminin devre gerçekleştirimi Şekil 3.6'da verilmiştir [77].

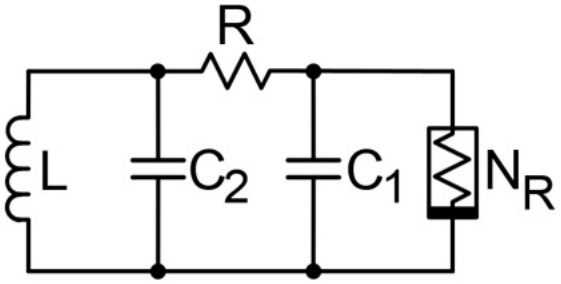


Şekil 3.6. Rössler devresi [77].

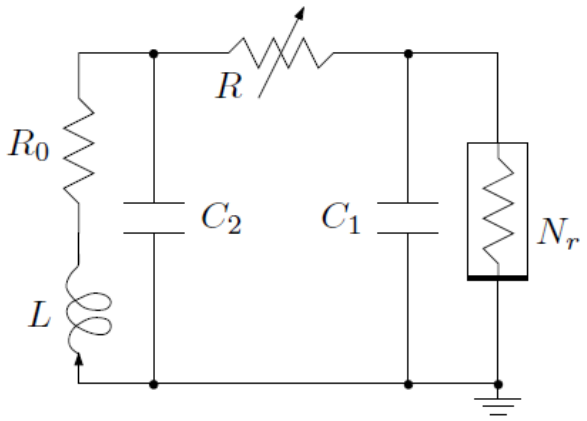
3.7. Chua Devresi

Kaotikliği öncelikle bilgisayar simülasyonlarında keşfedilen [9] daha sonra da Zhong ve Ayrom tarafından deneysel olarak kanıtlanan [78] Chua devresi, biri hariç tüm elemanları doğrusal olan bir devredir. Devrenin doğrusal olmayan elemanı, parçalı-doğrusal özellik gösteren özel bir yapıdaki dirençtir.

Şekil 3.7'de gösterilen Chua devresine R_0 direnci eklenirse, Şekil 3.8'de gösterilen ve literatürde Chua osilatörü olarak da geçen devre elde edilir.



Şekil 3.7. Chua devresi. [79].



Şekil 3.8. R_0 direnci eklenmiş Chua devresi [80].

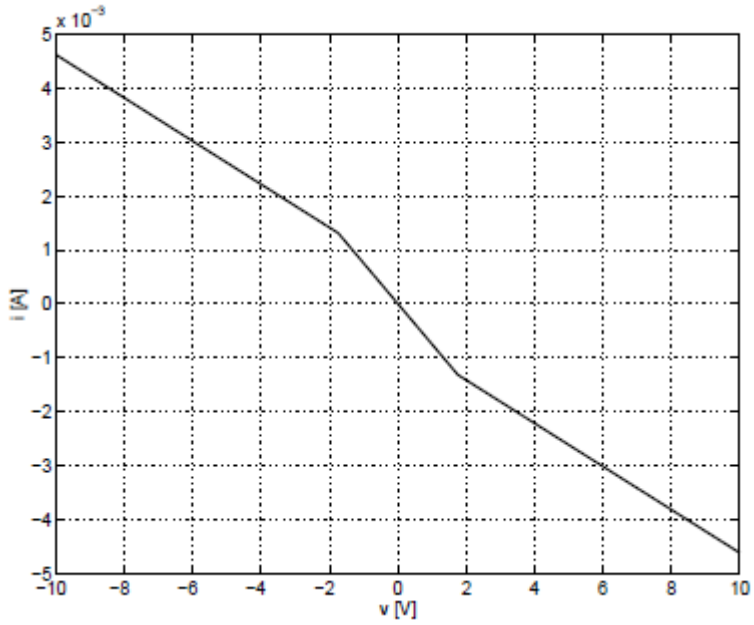
Şekil 3.8'de verilen Chua devresi üzerinde devre analiz yöntemleri uygulanarak aşağıda belirtilen diferansiyel denklemler elde edilir:

$$\begin{aligned} C_1 \frac{dVc_1}{dt} &= \frac{1}{R} V_{C2} (-V_{C1}) - f(V_{C1}) \\ C_2 \frac{dVc_2}{dt} &= I_L - \frac{1}{R}(V_{C2} - V_{C1}) \\ L \frac{dI_L}{dt} &= -V_{C2} - \dot{I}_L R_0 \end{aligned} \tag{3.14}$$

Burada f fonksiyonu aşağıdaki gibi tanımlanır;

$$f(V_{C1}) = G_b V_{C1} + \frac{1}{2}(G_a - G_b) * \left(|V_{C1} + B_p| - |V_{C1} - B_p| \right) \tag{3.15}$$

Burada belirtilen G_a ve G_b doğrusal olmayan elemanın karakteristik grafiğindeki eğimler olup $+B_p$ ve $-B_p$ karakteristik eğrinin kırılma noktalarıdır. Şekil 3.9'da parçalı-doğrusal direncin karakteristiği gösterilmiştir [80].



Şekil 3.9. Parçalı-lineer direncin v-i karakteristiği [80].

Devre üzerinde nümerik işlemler yapabilmek için Denklem 3.16'daki kabuller yapılır ve Denklem 3.17 ile Denklem 3.18 elde edilir.

$$x = \frac{V_{C1}}{B_p} \qquad y = \frac{V_{C2}}{B_p} \qquad z = \frac{R_{i_L}}{B_p}$$

$$\alpha = \frac{C_2}{C_1} \qquad \beta = \frac{C_2 R^2}{L} \qquad \tau = \frac{t}{RC_2} \tag{3.16}$$

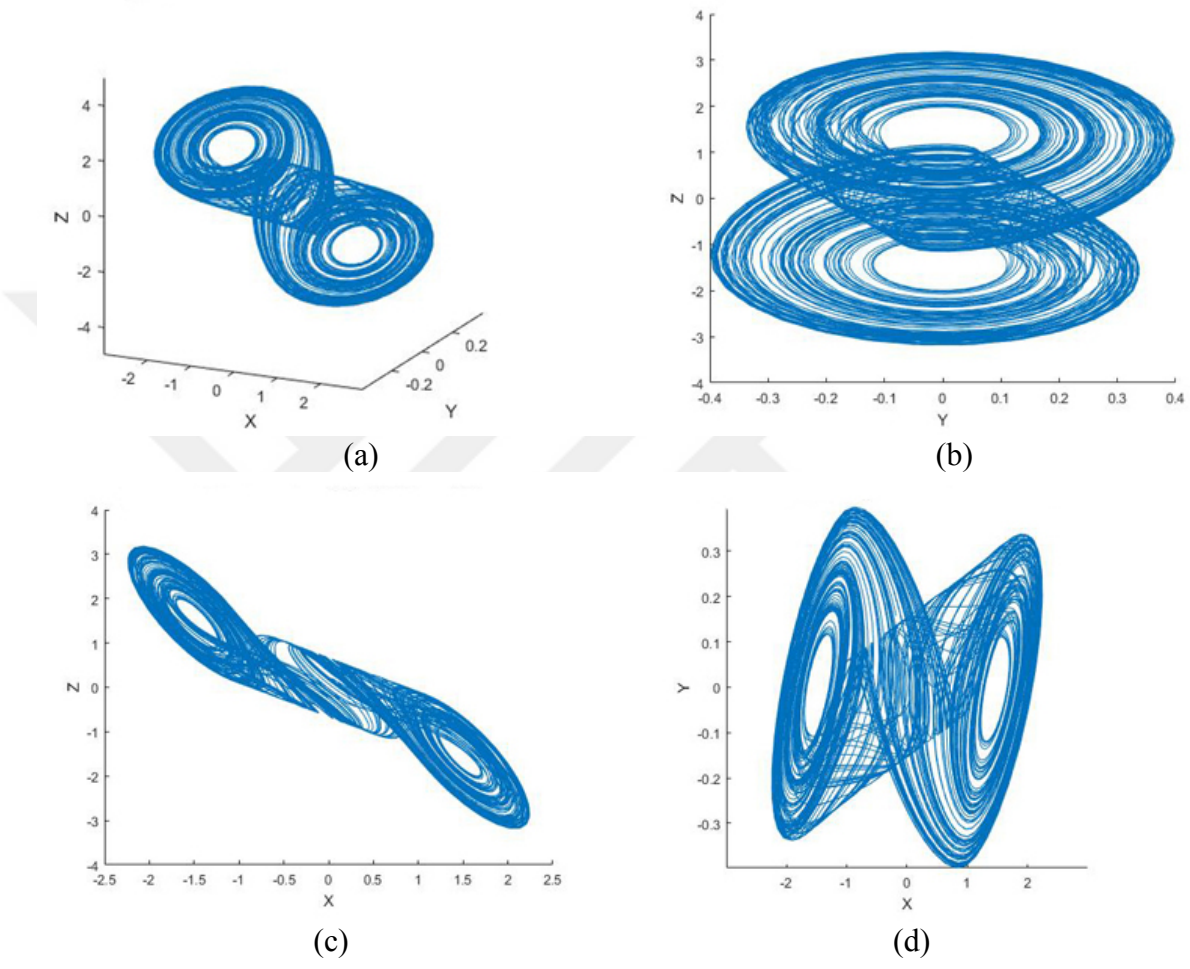
$$a = G_a R \qquad b = G_b R \qquad c = B_p$$

$$\begin{aligned} x' &= \alpha\,(y - x - f(x)) \\ y' &= x - y + z \\ z' &= -\beta y \end{aligned} \tag{3.17}$$

$$f(x) = bx + 0.5(a - b) * (\,|x+c| - |x-c|) \tag{3.18}$$

Parçalı direnç karakteristiğinin kırılma noktalarından elde edilen x = -1 ve x = 1 değerleri için Denklem 3.18'deki f(x) fonksiyonu aşağıdaki biçimde yazılabilir [80].

$$f(x)=\begin{cases} bx+a-b, & x \geq 1 \\ ax, & -1 \leq x \leq 1 \\ bx-a+b, & x \leq -1 \end{cases} \quad (3.19)$$



Şekil 3.10. [0,1; 0,1; 0,1] başlangıç değerleri ve $\alpha = 9$, $\beta = 100/7$, $a = -8/7$ ve $b = -5/7$ parametre değerleri için; (a) X-Y-Z düzlemi Chua çekeri (b) Y-Z düzlemi Chua çekeri (c) X-Y düzlemi Chua çekeri (d) X-Y düzlemi Chua çekeri [82].

Chua devresi $\alpha = 9$, $\beta = 100/7$, $a = -8/7$ ve $b = -5/7$ değerleri için Matsumoto'nun "*double scroll*" [81] olarak adlandırdığı bir çekere sahip olur. $x_0 = 0{,}1$, $y_0 = 0{,}1$ ve $z_0 = 0{,}1$ başlangıç değerleri için bu çeker Şekil 3.10'da verilmiştir.

3.8. Yörünge Ayırma (Orbit Separation) Metodu

Dinamik bir sistemin kaotikliğinin analizi için zaman serileri üzerinden hesaplanan parametreleri kullanan birçok yöntem vardır. En yaygın yöntemler; Poincare Haritalama, Güç Spektrumları, Lyapunov Üstelleri ve Fraktal Boyut Analizidir [83].

Yörünge Ayırma (Orbit Separation) metodu, birbirine yakın iki komşu yörüngenin zamanla nasıl farklılaştığını gösteren en büyük Lyapunov üstelini hesaplayabilmek için Sprott tarafından geliştirilmiş bir metottur. Bu tez çalışmasında kullanılan Lyapunov üstelleri, Yörünge Ayırma metodu kullanılarak hesaplanmıştır. Daha ayrıntılı bilgi için [84] numaralı kaynağa bakılabilir.

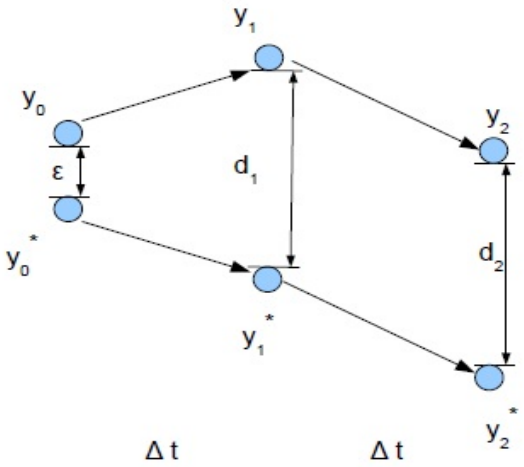
Yörünge ayırma metodunda herhangi dinamik bir sistem için uygun parametre ve başlangıç değerleri seçilir ve zamana (iterasyon) göre yörüngesi hesaplanır. Daha sonra aşağıdaki işlem adımları uygulanır;

- Aynı noktada bulunan iki yörünge takip edilir.
- İkinci yörünge ε kadar ötelenir.
- Normalizasyon işlemi uygulanır.
- Normalizasyon işlemi sonucunda bulunan değerler kullanılarak yörüngeler arasındaki ayrılmanın doğal logaritması alınır.
- Her iterasyon için hesaplanan logaritmalar üst üste toplanır.
- Elde edilen değer, iterasyon sayısı ve zaman adımı sayısına bölünür ve en büyük Lyapunov üsteli elde edilir.

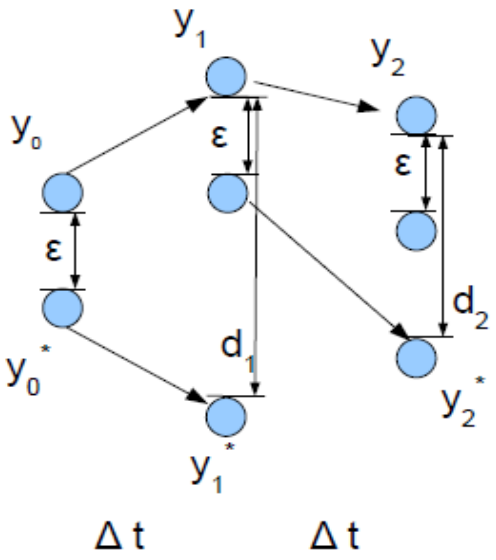
Gerçeğe yakın bir değer elde etme olasılığını artırmak için;

- İşlem çok sayıda rastgele başlangıç değeri için tekrarlanır.
- Zaman adımlarının baştan belli bir kısmı hesaba katılmaz.

Ayrık dinamik sistemlerdeki fark denklemleri, harita (map) olarak adlandırılır. Yörünge ise haritanın her iterasyonda hareket ettiği noktalar setine verilen addır [68]. Başlangıçta birbirine yakın iki yörüngenin Şekil 3.11'deki gibi birbirinden ayrıldığı düşünülürse;



Şekil 3.11. Başlangıç şartları birbirine yakın iki yörüngenin ayrılması [68].



Şekil 3.12. Yörüngelerin referans mesafe (ε) ile gösterilmesi [68].

İki yörünge arasındaki fark ε kadar iken bir zaman adımı sonra d_1 kadar, iki zaman adımı sonra ise d_2 kadar olmaktadır. Yörünge Ayırma metodunda, yörüngeler arasındaki mesafe, farklılaşmalarının ortalaması olarak hesaplanmaktadır. Bu hesaplamayı yapabilmek için öncelikle bir referans mesafe (ε) oranına ihtiyaç duyulur. Şekil 3.12'de yörüngeler referans mesafe ile birlikte gösterilmiştir.

Yörünge ayırma metodunda yörüngeler arasındaki ortalama ayrılma hesaplandığı için ikinci yörüngenin birinci yörüngeye göre konumu her adım için ayrı ayrı, Denklem 3.20'de gösterildiği gibi hesaplanır. Burada $\{ k \in Z : k \geq 1 \}$ olmak üzere $c_k = y_k - y_k^*$ ve $d_k = |c_k|$ dir [68].

$$y_k^* = y_k + \frac{\varepsilon}{d_k} c_k \tag{3.20}$$

Bu işlem, normalizasyon prosedürü olarak adlandırılır. Normalizasyon prosedüründeki referans mesafe ve sonrasında kullanılan algoritmadaki zaman adımı ne kadar küçük seçilirse ideal sonuca o kadar yaklaşılır fakat tahmin edileceği gibi bu durumda da hesaplamalar daha fazla zaman alır [68].

Daha sonra her bir zaman adımı için yörüngeler arasındaki ayrılmanın doğal logaritması (e tabanına göre) Denklem 3.21'de gösterildiği gibi bulunur.

$$L_f = \log\left(\left|\frac{d_1}{\varepsilon}\right|\right) \tag{3.21}$$

Bir sonraki adım ise hesaplanan logaritmanın daha önce hesaplanan logaritmaların üzerine eklenmesidir. Bunun için Denklem 3.22 her iterasyonda tekrarlanır.

$$L_s = L_s + L_f \tag{3.22}$$

Bu toplam alındıktan sonra en büyük Lyapunov üsteli, h zaman adımıyla n iterasyon için şu şekilde hesaplanır:

$$\lambda_1 = \frac{L_s/n}{h} \tag{3.23}$$

En büyük Lyapunov üstelinin yörünge ayırma metoduyla hesaplanması bu şekilde olmakla birlikte gerçek değere en yakın sonucun elde edilebilmesi için iki hususa dikkat edilmesi gerekmektedir. Bunlardan birincisi, bahsedilen hesaplama adımlarını çok sayıda başlangıç değeri için tekrarlamaktır. Yörünge ayırma metodunda başlangıç değeri rastgele seçildiği için işlem aşamaları ne kadar çok

tekrar edilirse gerçeğe en yakın sonuca ulaşma ihtimali o kadar artar. Önemli olan ikinci husus ise yörüngeler arasındaki ayrılmalar hesaplanırken başlangıçtan belli bir zamana kadar elde edilen değerlerin hesaba katılmamasıdır. Uzun süreli çalışma durumunda periyodik olmayan davranış sergileyen yörüngeler, başlangıçta bir süre periyodik davranış sergileyebildikleri için, ilk değerlerin hesaba katılması ortalamayı olumsuz etkileyebilecektir. Böyle bir durum için kaç iterasyondan sonra hesaplamanın başlatılacağı konusunda kesin bir kural yoktur. Farklı sistemler için deneme yanılma yoluyla çeşitli değerler bulunabilir [68].

4. BÖLÜM
YAPAY SİNİR AĞLARI

4.1. Giriş

Biyolojik bir yapı olarak insan vücudu, içinde bulunduğu çevre ile sürekli bir etkileşim halindedir. Dış dünyadan gelen verilere göre kararlar alınmakta, vücut kendini güncellemekte ve gerekli tepkiler üretilmektedir. İnsan vücudunda algılama, karar alma, icra etme görevlerini sinir sistemi gerçekleştirir. Bilim dünyası, bu işleri yapabilecek bir yapay zeka tasarlamak ve uygulamak için çalışmış ve çalışmalarına devam etmektedir [1].

İnsan beyni konvansiyonel bilgisayarlardan oldukça farklı bir biçimde hesaplamalar yapar. Beyne ulaşan verilerin işlenmesi bir hayli karmaşık, doğrusal olmayan ve paralel bir biçimde gerçekleştirilmektedir [85]. Bilgisayarlar, sayısal işlemlerde insan beynine göre oldukça hızlı olmalarına rağmen, görüntü algılama, planlama, öğrenme gibi işlemlerde başarısız olmaktadırlar. Bunun yanı sıra bilgisayarlarda tek bir komutun yanlış olması sonucu işlemler dururken, insan beyninde bir sinir hücresinin işlevini yitirmesi halinde dahi beyin çalışmaya devam eder. Yapay sinir ağları, beynin bu çalışma sistemine benzetilerek tasarlanan yapılardır [86].

Bir yapay sinir ağı (YSA), tek bir bütün içinde katmanlar halinde paralel olarak dağıtılmış basit işleme ünitelerinden oluşan bir işlemcidir. YSA'lar, elektronik komponentler kullanılarak veya bir yazılım vasıtasıyla gerçekleştirilebilirler [85].

4.2. Tarihçe

İlk yapay sinir hücresi, Warren McCulloch ve Walter Pitts tarafından 1943 yılında tanıtılmıştır [87]. 1949'da Donald Hebb, Hebb kuralını ortaya koymuştur [88]. Hebb

kuralı; YSA'lardaki bağlantı sayısının değiştirilmesi durumunda ağın öğrenebileceğini ileri sürer [86]. 1959 senesinde Bernard Widrow ve doktora öğrencisi Ted Hoff LMS (Least Mean Squares) algoritmasını geliştirmişlerdir ve bu algoritma daha sonra geliştirilecek olan ADALINE (ADaptive LInear NEuron) ve MADALINE (Multiple Adaline) modellerinin temelini oluşturmuştur [89]. 1969 yılında Marvin Minsky ve Seymour Papert, günümüz YSA modellerinde kullanılan hücre yapısının temellerini atmıştır [90].

1970' lerde tasarlanan algoritmaların mevcut problemlerin çözümünde yetersiz kalması ve teknolojinin kısıtlı olması sonucu YSA'lar hakkındaki çalışmalar yavaşlamıştır. 1980' li yılların ikinci yarısıyla beraber YSA'lar ile ilgili gelişmeler tekrar hız kazanmış ve günümüze kadar gelmiştir [1].

4.3. Yapay Sinir Ağlarının Avantaj ve Dezavantajları

YSA'ların çok çeşitli avantajları vardır. Bunlardan en önemlisi öğrenme yeteneğidir. Öğrenme, aslında ağ içerisindeki sinir hücrelerinin ağırlıklarının değiştirilmesidir. Bu ağırlık değişimi sonucunda ağın çıkış değerlerinin idealde olması gereken değerlere yakınsaması beklenir. Öğrenme kabiliyeti, ağa genelleme yapabilme imkanıda sağlar. Genelleme yapmak, daha önceden karşılaşılmamış giriş değerleri içinde çıkış üretebilmektir ki YSA'ların bir diğer önemli özelliği de budur. Genelleme yapabilme yeteneği, ağa giren verilerin eksik olması durumunda bile uygun çıkışlar üretmeyi sağlayabilir [1, 91].

Yapay sinir ağlarının bir diğer özelliği ise uyarlanabilirliktir. Değişken giriş değerleri karşısında eğitilmiş bir YSA, bu değişkenliğe kendini adapte edebilir. Yani, bir işlemi yapmak için eğitilmiş bir YSA modeli, işlemdeki değişikliklere göre tekrar eğitilebilmektedir. Bu adaptasyon için ayrıca dış bir müdahaleye (örneğin bir programcının kod yazmasına) gerek duyulmaz [86, 91].

Yapay sinir ağı hücresi, yapısı gereği doğrusal değildir. YSA'lar da, bu sinir hücrelerinden oluştuğu için doğrusal olmayan bir yapıdadırlar. YSA'ların bu özelliği, doğrusal olmayan verilerin (örneğin ekonomik veriler) analizinde bir avantaj sağlamaktadır [1, 86].

Verileri paralel olarak işleme, YSA'ların bir diğer özelliğidir. Seri veri işleme sürecinde herhangi bir birimin arızalanması tüm sistemi durdururken, paralel yapıda arızalanan birim haricindeki diğer birimler veri işlemeye devam eder. Ayrıca paralel işlemede, birden fazla kolda aynı anda işlem yapıldığından toplamdaki işlem hızı artmakta, problem daha kısa sürede çözülmektedir. Buna karşılık paralel yapının donanımsal olarak gerçekleştirimi, seri yapıya göre daha zor olabilmektedir [1, 86].

Geniş veri seti gereksinimi YSA'ların dezavantajlarından biridir. Ağ eğitimi ve eğitilen ağın test edilmesi için belirli genişlikte bir veri setine ihtiyaç duyulur. Bu genişlik uygulamaya göre değişmektedir. Bir diğer dezavantaj ise bazı uygulamalarda problemin basitliğine karşın YSA'nın oldukça karmaşık kalmasıdır. Ayrıca yine uygulamaya ve işlemci gücüne göre değişmekle birlikte bazı durumlarda ağın eğitilmesi ve test edilmesi uzun zaman alabilmektedir [86, 91].

4.4. Yapay Sinir Ağlarının Kullanım Alanları

Gerek geliştirilmesinde gerekse uygulamasında disiplinler arası bir konu olan yapay sinir ağları, döviz kuru tahmini [91], kalp rahatsızlıklarının sınıflandırılması [92], karakter tanıma [93], ses sinyali işleme [94], otomobil satış tahmini [95], enerji yük tahmini [96] ve karbon monoksit emisyonlarının modellenmesi [97] gibi birçok farklı alanda kullanılmaktadır.

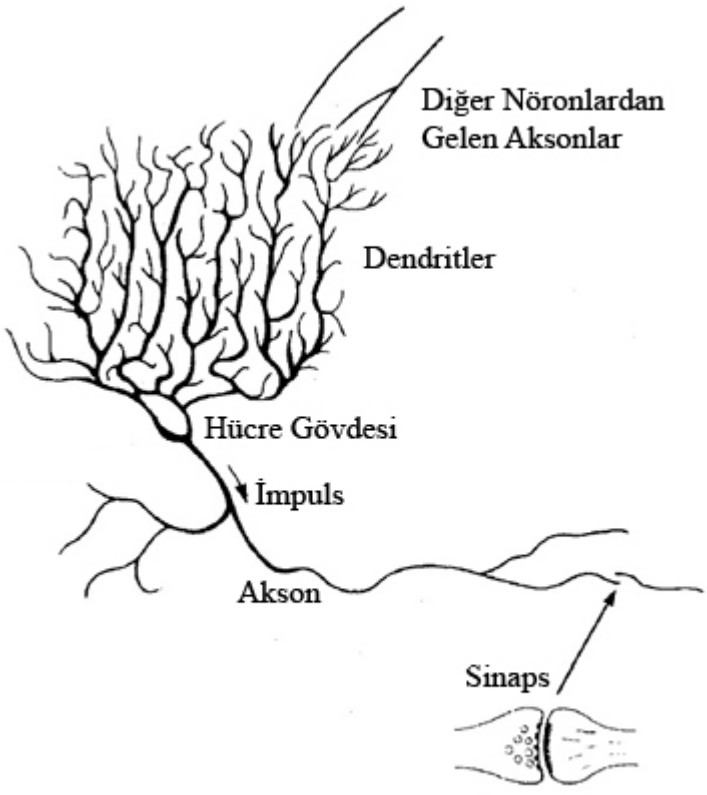
Öğrenebilme, doğrusal olmayan verileri işleyebilme, eksik bilgiyle çalışabilme, gürültüye karşı iyi bir performans gösterme, hafızası olma ve paralel işlem yapabilme gibi özellikleri bulunan YSA'ların bazı genel kullanım alanları şu şekildedir [98, 86];

- Sinyal işleme
- Örüntü tanıma / konuşma tanıma
- Zaman serilerinin analizi
- Sınıflandırma
- Tahmin
- Ekonomik verilerin analizi
- Tıbbi sinyallerin analizi

- Kontrol/denetim

4.5. Biyolojik Nöron

İnsan sinir sistemini oluşturan sinir hücrelerine nöron adı verilir. Bir nöron, 3 ana kısımdan oluşur. Bunlar akson, dentrit ve aynı zamanda soma olarak da adlandırılan hücre gövdesidir. Dendritler sayıca çok, dallanmış bir yapıya sahiplerken akson uzun, silindirik bir yapıdadır. Sinir hücrelerinden iletilen impulsların (elektriksel sinyaller) yönü dendritlerden hücre gövdesine, hücre gövdesinden ise aksona doğrudur. Akson ucu ile dendrit ucu arasındaki bağlantı bölümüne sinaps denir [99]. Şekil 4.1'de bir nöronun yapısı görülmektedir.

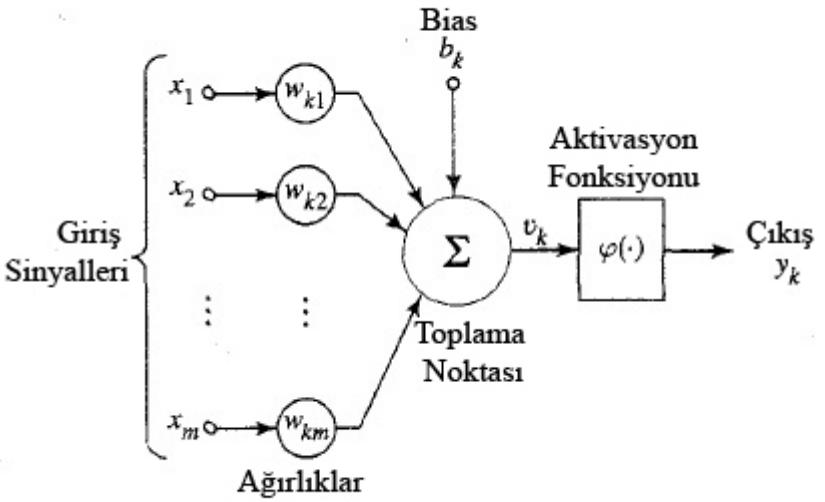


Şekil 4.1. Biyolojik nöronun yapısı [99].

4.6. Yapay Nöron

Sinir sisteminde olduğu gibi, yapay sinir ağlarında da temel işlem birimine nöron denilmektedir. Şekil 4.2' de yapay bir nöron modeli görülmektedir [100]. Bu model 3 temel kısımdan oluşmaktadır:

- Birinci kısım, sinir sistemindeki sinapslara karşılık gelen her biri ağırlıklandırılmış bağlantı noktalarıdır. Her bir x_i giriş sinyali, k nöronunda w_{ki} ağırlığı ile çarpılmaktadır.
- Bir toplayıcı, bu ağırlıklandırılmış giriş sinyallerini toplamaktadır.
- Toplayıcı çıkışı, bir aktivasyon fonksiyonuna tabi tutulmakta ve nöron çıkışını oluşturmaktadır.



Şekil 4.2. Yapay nöron modeli [100].

Şekil 4.2'de aynı zamanda ağırlıklandırılmış girişlere eklenen bias değeride gösterilmiştir. YSA'larda öğrenmeye yardımcı olması için kullanılan ve sabit bir değer olan bias değeri, giriş sinyalleri 0 (sıfır) olsa bile aktivasyon fonksiyonuna giren sinyalin sıfır olmasını önler [100].

Giriş sayısı *m* olan *k* nöronunun çıkışını matematiksel olarak aşağıdaki gibi ifade edilebilir [100].

$$v_k = \sum_{i=1}^{m} w_{ki} x_i$$

$$y_k = \varphi(v_k + b_k) \qquad (4.1)$$

4.7. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları veya diğer deyişle transfer fonksiyonları, her yapay nöronda bulunur ve nöronların, giriş değerlerine karşılık üretecekleri çıkış değerlerini belirlerler. Bunlardan bazıları Sert Sınırlayıcı, Logaritmik Sigmoid, Doğrusal ve Hiperbolik Tanjant Sigmoiddir [101].

4.8. Yapay Sinir Ağlarının Sınıflandırılması

Çok çeşitli şekillerde tasarlanabilecek olan YSA'lar yine çeşitli şekillerde sınıflandırılabilirler. Bu tez çalışmasında, ağ mimarilerine ve öğrenme algoritmalarına göre sınıflandırma konuları kısaca ele alınacaktır. Literatürde çeşitli ağ mimarilerinden bahsedilse de, burada ileri beslemeli ve geri beslemeli ağ mimarileri ele alınacaktır.

Geri beslemesi olmayan, bilgi akışı girişten çıkışa doğru olan ağlara ileri beslemeli ağlar denir. Bu tip ağlar, katmanlı bir yapıda düzenlenmişlerdir ve aynı katmandaki hücrelerin birbirleri ile bağlantıları yoktur. Bir katmandaki hücre sadece bir sonraki katmana veri gönderebilir. İleri beslemeli ağlar, öğrenme algoritması olarak genelde geri yayılım öğrenme algoritmalarını kullanmaktadırlar [86, 99].

İleri beslemeli ağların aksine dinamik yapıya sahip, çıkıştan veya ara katmanlardan girişe veya önceki ara katmanlara doğru bilgi akışı olan YSA'lar, geri beslemeli olarak tanımlanırlar. İleri beslemeli ağlardan farklı olarak sonuç, daha önceki sonuçlara bağlı olduğu için hafızaya sahip oldukları söylenebilir. Bu özelliklerinden dolayı önceden tahmin uygulamaları için uygundurlar. Hopfield, Elman ve Jordan ağları geri beslemeli ağlardan bazılarıdır [66, 86].

4.8.1. Öğrenme Algoritmasına Göre Yapay Sinir Ağları

Yapay sinir ağları, kendilerine gösterilen örnek değerlere bakarak öğrenme işlemini gerçekleştirirler. Esasında öğrenme, bağlantı noktalarındaki ağırlıkların değiştirilmesidir. Çeşitli öğrenme metodları bulunan YSA'larda temel olarak danışmanlı (supervised) ve danışmansız (unsupervised) öğrenme mevcuttur [86].

Danışmanlı öğrenme çeşidinde, ağa uygulanan girişlerle birlikte bu girişlere karşılık üretilen çıkışlar da örnek olarak gösterilir. Öğrenme algoritması, bu örneklere bakarak ağ ağırlıklarını değiştirir. Ağa örnek gösterilen çıkış değeri ile ağın ürettiği çıkış değeri arasındaki hata önceden belirlenen bir eşik değerinin altına ininceye kadar ağırlıklar değiştirilir. Eşik değer bir danışman tarafından tayin edilir.

Danışmanlı öğrenmenin aksine danışmansız öğrenmede YSA'ya sadece giriş değerleri örnek olarak gösterilir. Öğrenme algoritması, giriş verileri arasındaki benzerliklere bakarak ağ ağırlıklarını değiştirir. Bu tip öğrenmede algoritma, benzer dokuları ayırabilecek şekilde ağ ağırlıklandırması yapar [66].

Levenberg-Marquardt, Esnek Yayılım, Ölçeklendirilmiş Eşleştirmeli Eğim, Grup Öğrenme, Çevrim içi Öğrenme, BFGS (Broyden–Fletcher–Goldfarb–Shanno) ve Bayesyen Regülarizasyon algoritmaları yaygın kullanılan öğrenme algoritmalarına örnek olarak verilebilir [74, 79, 102-107].

4.9. Yapay Sinir Ağı Modelleri

Literatürde çok çeşitli yapay sinir ağları bulunmasına rağmen, Çok Katlı Perseptronlar (Multi Layer Perceptrons, MLP) başta olmak üzere, LVQ Ağları (Learning Vector Quantization), SOM Ağları (Self Organising Map), Hopfield Ağları, Elman Ağları ve Jordan Ağları en çok kullanılan ağlardandır [1].

4.10. Geri Yayılım Algoritması

Geri yayılım algoritması Least Mean Squared Algorithm yani en küçük ortalama kareler algoritmasının (LMS) genelleştirilmiş halidir. Danışmanlı öğrenmeyi kullanan bu algoritmada sinaptik ağırlıkları, istenilen çıkışla halihazırdaki hesaplanmış çıkış arasındaki ortalama karesel hatanın minimize edileceği şekilde değiştirilir. Örnekler, ağa gösterilip eğitim bittikten sonra ağırlıklar, yeni giriş değerlerine göre çıkışları hesaplamak için sabitlenir [108].

İstenilen değer d ve çıkış değeri y olmak üzere j'inci nöronun *n*'inci iterasyonda ki hatası Eşitlik 4.2'deki gibi hesaplanır [100];

$$e_j(n) = d_j(n) - y_j(n) \tag{4.2}$$

Nöron başına düşen ortalama karesel hata $\frac{1}{2}e_j^2(n)$ olarak tanımlanırsa ve "C" çıkış nöronlarını belirtirken toplam hata (ε);

$$\varepsilon(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{4.3}$$

ve toplam örnek sayısı N ile gösterilecek olursa normalize edilmiş ortalama karesel hata aşağıdaki gibi bulunur.

$$\varepsilon_{ort} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sum_{j\in C} e_j^2(n) \tag{4.4}$$

Geri yayılım algoritmasında öğrenme işleminin temel amacı, ortalama karesel hatayı (ε_{ort}) en aza indirmek için sinaptik ağırlıkları ve bias değerleri gibi çeşitli parametreleri ayarlamaktır. Bu ayarlamalar, eğitim esnasında ağa sunulan örnekler baz alınarak hesaplanan hata değerlerine göre yapılır [100].

Sinaptik ağırlığı *w*, aktivasyon fonksiyonu φ ve *j, i*'den sonra gelen nöron olmak üzere;

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n)y_i(n) \tag{4.5}$$

olur iken çıkış aşağıdaki gibi hesaplanır.

$$y_j(n) = \varphi_j\left(v_j(n)\right) \tag{4.6}$$

Geri yayılım algoritmasında, sinaptik ağırlıklarda yapılan düzeltme ile hatanın sinaptik ağırlıklarına göre kısmi türevi orantısaldır. Kısmi türev alınırken kullanılabilen zincir kuralı [100] uygulanır ise;

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{4.7}$$

elde edilir. Eşitlik 4.7'deki kısmi türevlerin karşılıkları yazılır ise;

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = e_j(n)\varphi_j'\left(v_j(n)\right) y_i(n) \tag{4.8}$$

elde edilir [100].

Geri yayılım algoritmasında öğrenme oranı parametresi η olmak üzere sinaptik ağırlığına uygulanacak düzeltme, $\Delta w_{ji}(n)$ delta kuralına göre aşağıdaki gibi yazılabilir [99];

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \tag{4.9}$$

Eşitlik 4.8 kullanılarak Eşitlik 4.9 yeniden yazılırsa;

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{4.10}$$

Elde edilir. Burada δ, yerel gradyent değeri olup nöronun çıkış katmanında veya gizli (ara) katmanda olmasına göre farklı şekilde hesaplanır. Çıkış katmanında bulunan bir nöron için yerel gradyent;

$$\delta_j(n) = e_j(n)\varphi_j'\left(v_j(n)\right) \tag{4.11}$$

olur iken, nöron ara katmanda ise yerel gradyent;

$$\delta_j(n) = \varphi_j'\left(v_j(n)\right) \sum_k \delta_k(n)\, w_{kj}(n) \tag{4.12}$$

şeklinde elde edilir. Karışıklık olmaması için gizli katman nöronu j, çıkış nöronu ise k notasyonu ile gösterilmiştir. Gizli katman nöronu için yerel gradyentin nasıl elde edildiği ile ilgili ayrıntılı bilgi [100] numaralı kaynaktan elde edilebilir.

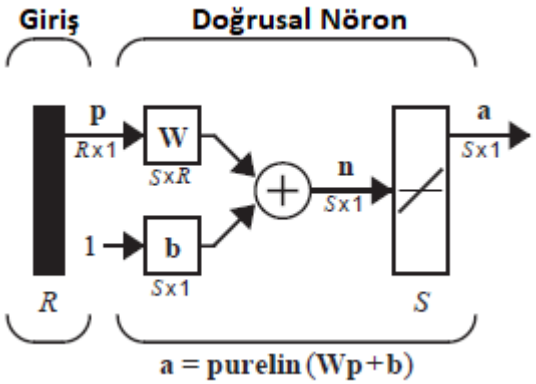
Geri yayılım algoritmasında çıkışa dair hesaplamalar iki aşamada yapılır. Bunlar, girişten çıkışa doğru giden ileri yönlü geçiş aşaması ve çıkıştan girişe doğru giden geri yönlü geçiş aşamasıdır. İleri yönlü geçişte ağın fonksiyon sinyalleri nöron nöron hesaplanırken, geri yönlü geçişte, her nöron için yerel gradyent hesaplanır. İleri yönlü geçişte, ağırlıklar değiştirilmezken, geri yönlü geçişte hata sinyali üretilip sinaptik ağırlıklar değiştirilir [100].

Geri yayılım algoritmasının tasarımı, çözülmek istenen problemin türüne göre ve daha iyi performans elde etmek için kullanıcı tarafından değiştirilebilir. Aşağıda algoritmanın başlıca tasarım etmenleri belirtilmiştir [98, 108].

- Ağırlıkların başlangıç değerleri
- Aktivasyon fonksiyonu
- Eğitim setindeki örnek sayısı
- Ağırlıkların her örnek için ayrı ayrı mı yoksa tüm örnekler gösterildikten sonra tek bir seferde mi değiştirileceği
- Öğrenme oranının değeri
- Eğitimin ne kadar sürede ve hangi kriter sağlanınca biteceği
- Ara katman sayısı ve her ara katmandaki düğüm sayısı

4.11. Ortalama Karesel Hata

Tek katmanlı ve tek nöronlu bir yapıya sahip ve doğrusal transfer fonksiyonu bulunan bir yapay sinir ağı modeli Şekil 4.3'de görülmektedir.



Şekil 4.3. Yapay sinir ağı modeli [109].

Burada "p" giriş değerleri, "b" bias değeri ve " $_1$w" ağırlık değerlerinin 1. sırası olmak üzere, ayarlama yapılacak değerler tek bir "x" vektörü şeklinde yazılmak istenirse;

$$x = \begin{bmatrix} {}_1w \\ b \end{bmatrix} \tag{4.13}$$

Benzer şekilde bias girişi 1 alınır ve bir "z" giriş vektörü oluşturulursa;

$$z = \begin{bmatrix} p \\ 1 \end{bmatrix} \tag{4.14}$$

Ortaya konan ağın çıkışı;

$$a = {}_1w^T p+b$$
$$a = x^T z \tag{4.15}$$

şeklinde yazılabilir. Böylelikle t eğitimde kullanılacak hedef değerler olmak üzere, ağ için ortalama karesel hata aşağıdaki şekilde yazılabilir [109];

$$F(x) = E[e^2] = E\left[(t-a)^2\right] = E\left[(t-x^T z)^2\right] \tag{4.16}$$

Burada E[], deterministik sinyaller için zaman ortalaması olan beklenen değerdir [109]. Denklem 4.16 yeniden düzenlenirse;

$$F(x) = E[t^2-2tx^Tz+x^Tzz^Tx] = E[t^2]-2x^TE[tz]+x^TE[zz^T]x \quad (4.17)$$

Burada;

$$c = E[t^2],\ h = E[tz] \text{ ve } R = E[zz^T] \quad (4.18)$$

olmak üzere Denklem 4.17 aşağıdaki formda ifade edilebilir;

$$F(x) = c-2x^Th+x^TRx \quad (4.19)$$

Burada R değeri, giriş korelasyon matrisi iken h değeri giriş vektörü ile onun hedefi arasındaki çapraz korelasyondur [109].

İkinci dereceden tek değişkenli (kuadratik) bir fonksiyonun durgunluk noktası, yani azalmayı veya artırmayı durdurduğu nokta, o fonksiyonun gradyenti alınıp sıfıra eşitlenerek bulunabilir.

Denklem 4.19'un gradyenti alınırsa;

$$\nabla F = \nabla(c - 2x^Th+x^TRx\ ) = -2h+2Rx \quad (4.20)$$

ve alınan gradyent sıfıra eşitlenirse;

$$-2h+2Rx = 0 \quad (4.21)$$

elde edilir. Eğer korelasyon matrisi pozitif tanımlı ise yalnızca 1 adet durgunluk noktası olacaktır;

$$x^* = R^{-1}h \quad (4.22)$$

Burada dikkat edilmesi gereken nokta, çözümün tek olmasının yalnızca korelasyon matrisine bağlı olmamasıdır. Giriş vektörlerinin karakteride bir veya daha fazla çözüm olup olmayacağını etkiler [109].

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

Yapay sinir ağlarının eğitilmesi, aslında bir optimizasyon problemidir. Bu optimizasyonda ortalama karesel hata kriter olarak alınırsa buradaki temel hedef minimum noktayı bulmak olur. İstenilen minimum nokta, Denklem 4.23 ile doğrudan bulunabilir veya R'nin tersinin hesaplanması istenmiyorsa, Denklem 4.21'deki gradyent ile birlikte En Dik İniş (Steepest Descent) algoritması kullanılabilir. Fakat genelde h ve R'nin hesaplanması elverişli olmaz. Bundan dolayı, içerisinde tahmini gradyent kullanılan bir yaklaşık en dik iniş algoritması tercih edilir. Bu tip gradyente stokastik gradyent de denir [109].

Ortalama karesel hatayı tahmin etmek için Denklem 4.16'daki hata kareleri beklentisi ile her k. iterasyondaki karesel hata yer değiştirilirse;

$$\hat{F}(x) = (t(k)-a(k))^2 = e^2(k) \tag{4.23}$$

ve gradyenti alınırsa;

$$\hat{\nabla}F(x) = \nabla e^2(k) = -2e(k)z(k) \tag{4.24}$$

elde edilir. Burada z giriş vektörüdür.

En dik iniş metodu, Denklem 4.25'deki gibi tanımlanabilir. Burada α_k öğrenme oranı, g_k ise gradyentdir.

$$x_{k+1} = x_k - a_k g_k \tag{4.25}$$

Denklem 4.25 ortalama karesel hataya uyarlanırsa;

$$x_{k+1} = x_k - a\nabla F(x)|_{x=x_k} \tag{4.26}$$

elde edilir.

Denklem 4.24 ile Denklem 4.26 birleştirilirse;

$$x_{k+1} = x_k + 2ae(k)z(k) \tag{4.27}$$

denklemi elde edilir. Bu denklemde x ve z değerleri yerlerine konulursa ağırlık ve bias değerlerinin belirtildiği En Küçük Ortalama Kare (Least Mean Square) algoritması bulunmuş olur.

$$_iw(k+1) = {}_iw(k) + 2ae_i(k)p(k) \tag{4.28}$$

$$_ib(k+1) = {}_ib(k) + 2ae_i(k) \tag{4.29}$$

Bu algoritma aynı zamanda Widrow-Hoff öğrenme algoritması veya Delta kuralı olarak da bilinir. Ayrıntılı bilgi için [109] numaralı kaynağa başvurulabilir.

4.13. Geri Yayılım Algoritmasının Dezavantajları ve Sezgisel Yaklaşımlar

En Küçük Ortalama Kare algoritması (EKOK), öğrenme oranının çok büyük olduğu durumlar hariç, ortalama karesel hatayı minimize eden sonuca doğru yakınsar. Bunun nedeni, ortalama karesel hatanın tek katmanlı doğrusal ağlar için 4. dereceden bir fonksiyon olması ve bu fonksiyonların sadece bir adet durağan noktasının bulunmasıdır. Ayrıca belirli bir yönde ilerlendiğinde kavisli olan taraf değişmez [109].

Geri yayılım algoritması, EKOK algoritmasının genelleştirilmiş bir biçimidir. Tek katmanlı doğrusal ağlarda bu iki algoritma birbirinin aynısıdır. Çok katmanlı ağlarda ise geri yayılımın karakteristiği oldukça değişiktir. Bunun nedeni, çok katmanlı ağların birden fazla yerel minimum noktasının bulunması ve buna bağlı olarak belirli bir yönde ilerlendiğinde kavis olan tarafın değişebileceği gerçeğidir. Bu durum geri yayılım algoritmasında farklı bölgelerde, farklı öğrenme oranlarına ve yerel/global minimum noktalarını bulmaya çalışırken yapılan salınımların yumuşatılmasına, ihtiyaç duyulmasına neden olabilir [109].

Bu tez çalışmasında, geri yayılım algoritmasının performansının iyileştirilmesi için kullanılan ve temellerinde standart optimizasyon algoritmaları olan iki metoda değinilecektir.

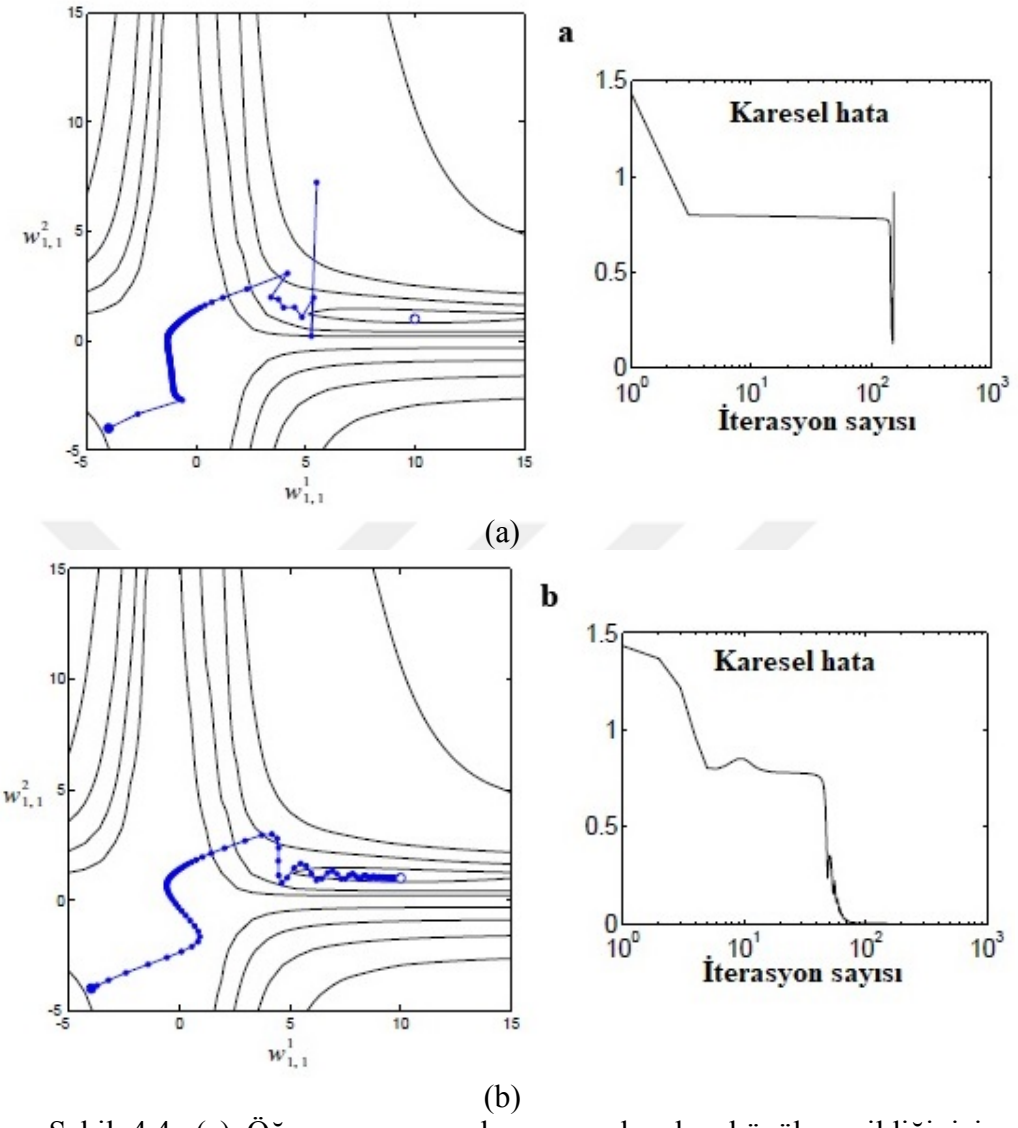
4.13.1. Momentum

Dinamik sistemlerdeki diferansiyel denklemler, akış (flow) olarak adlandırılır. Yörünge (trajectory) ise akışın izlediği yoldur [68]. Geri yayılım algoritmasında yörünge minimumdan (hedefinden) ıraksamaya başlarsa dar bir vadide ileri geri hareket eder. Yörüngenin izlediği yol, parametre güncellemelerinin ortalamasının alındığı alçak geçiren bir filtreden geçirilirse daha kararlı bir sonuç elde edilmiş olur. Bu işlem için algoritmaya momentum eklenmesi gerekir. Değeri 0 ile 1 arasında değişen momentum faktörü ile algoritmanın kararlılığı korunurken, öğrenme oranının artırılmasına olanak sağlanır. Momentumun artırılması ile yörüngenin aynı yönde devam etme eğilimi artar. Bununla birlikte bir yönde kararlı olarak ilerleme yaşanıyorsa momentum artışı yakınsamayı hızlandırır [109, 110].

Şekil 4.4'de görüldüğü gibi öğrenme oranı çok büyük olduğu için, hedefine ulaşamayan bir yörünge, momentum katsayısı eklenerek düzeltilmiştir [109].

4.13.2. Değişken Öğrenme Oranı

Yüzeyin düz olduğu bölgelerde öğrenme oranı artırılıp, eğimli olduğu yerlerde de öğrenme oranı azaltılırsa daha verimli bir yaklaşım elde edilebilir. Çok katmanlı ağlarda, yüzey şekli parametre uzayının farklı noktalarında çok farklı şekillerde olabilir. Bundan dolayı eğitim sırasında öğrenme oranı değiştirilerek yakınsama hızlandırılabilir. Bu oranın ne zaman ve ne kadar değiştirilmesi gerektiği ile ilgili çeşitli yaklaşımlar bulunmaktadır. Vogl ve arkadaşları [111] öğrenme oranını, algoritma performansına göre değiştirmişlerdir. Hangi durumda ne yapılacağı şu şekilde verilebilir [110];



Şekil 4.4. (a) Öğrenme oranı olması gerekenden büyük seçildiği için hedefine ulaşamayan bir yörünge ve karesel hatası (b) Aynı başlangıç şartlarına ve öğrenme oranına sahip fakat momentum katsayısı eklenmiş yörünge ve karesel hatası [109].

- Eğer hata kareler ağırlık güncellemesinden sonra belirli bir ζ yüzdesinden fazla artıyorsa; (yani öğrenme yanlış tarafa doğru gidiyorsa) ağırlık güncellemesi iptal edilip, öğrenme oranı 0 ile 1 arasında değişen bir ρ faktörüyle çarpıldıktan sonra, momentum sıfıra çekilir.
- Eğer hata kareler ağırlık güncellemesinden sonra azalıyorsa; ağırlık güncellemesi kabul edilip öğrenme oranı 1'den büyük bir η değeri ile çarpılır. Eğer momentum daha önce sıfıra çekilmiş ise tekrar baştaki değerini alır.

- Eğer hata kareler ağırlık güncellemesinden sonra belirli bir ζ yüzdesinden daha az artıyorsa; ağırlık güncellemesi kabul edilir fakat bu sefer öğrenme oranı değiştirilmez ve yine benzer şekilde momentum daha önce sıfıra çekilmiş ise baştaki değerine güncellenir [109].

5. BÖLÜM
BULGULAR

Kaotik sistemler, hem başlangıç şartlarındaki hemde sistem parametrelerindeki değişimler karşısında farklı davranışlar sergilemektedir. Bu davranışların kaotikliği konusunda en büyük Lyapunov üstelinin pozitif veya negatif olması önemli bir ipucudur. Ancak en büyük Lyapunov üstelinin hesaplanması zaman almaktadır. Bu zaman kaybını önlemek için YSA ile eğitimler yapılmış ve YSA modellerinin tahmin performansını ölçmek için test verileri kullanılmıştır.

Bu tez çalışmasında Lorenz, Chua ve Rössler sistemlerinin en büyük Lyapunov üstellerinin (EBLÜ) (Largest Lyapunov exponent - LLE) hesaplanması için kullanılan yörünge ayırma metodu işlemleri, 10 rastgele başlangıç değeri için hesaplanıp ortalaması alınmış, bu 10 değerin her birinin hesaplanması için gereken işlemler 100.000 zaman adımı için tekrarlanmıştır. Bu 100.000 zaman adımının ilk 2000'i hesaba katılmamıştır. Bu 3 sistemin en büyük Lyapunov üstelleri, her sistem için 1200'er farklı parametre seti için ayrı ayrı hesaplanmıştır.

En büyük Lyapunov üstellerini hesaplarken rastgele başlangıç değeri sayısı ne kadar artırılırsa gerçeğe o kadar yakın bir değer elde edilmiştir. Bu tez çalışmasında çok sayıda hesaplama yapıldığı için işlem yükünü daha da artırmamak adına rastgele başlangıç değeri sayısı 10 olarak belirlenmiştir.

Lorenz sistemi için kullanılan 1200 farklı parametre seti Tablo 5.1'de gösterilen değerler arasında 400 eşit aralığa bölünerek elde edilmiştir.

Chua sisteminde her parametre 300 eşit aralığa bölünmüştür. Chua sistemi için kullanılan parametre aralıkları Tablo 5.2'de, aynı şekilde Rössler sistemi için kullanılan parametre aralıkları Tablo 5.3'de gösterilmiştir.

Tablo 5.1. Lorenz sistemi parametre setlerinin oluşturulması için kullanılan parametre aralıkları

	Sigma	**Beta**	**Rho**
Lorenz	0 - 20	0 - (16/3)	0 - 56

Tablo 5.2. Chua sistemi parametre setlerinin oluşturulması için kullanılan parametre aralıkları

	Alfa	**Beta**	**a**	**b**
Chua	-7,32 – 10,68	11,66 – 40,48	-2,66 – 0,35	-0,99 – 0,43

Tablo 5.3. Rössler sistemi parametre setlerinin oluşturulması için kullanılan parametre aralıkları

	a	**b**	**c**
Rössler	-0,017–0,32	0 –0,4	0,48–11,88

Tablo 5.4'de Lorenz, Chua ve Rössler sistemleri için standart değerler verilmiştir. Parametre aralıkları bu literatürde sıklıkla kullanılan değerleri ortalayacak şekilde seçilmeye çalışılmış fakat bazı aralıklarda Lyapunov üsteli tanımlı olmadığından (hesaplanamadığından) aralıklar üstelin hesaplanabileceği şekilde kaydırılmıştır.

Tablo 5.4. Lorenz, Chua ve Rössler sistemleri için standart parametreler [70, 75, 81].

Lorenz	Sigma = 10	Beta = 8/3	Rho = 28	
Chua	Alfa = 9	Beta = 100/7	a= -8/7	b = -5/7
Rössler	a = 0,2	b = 0,2	c = 5,7	

Tahmin uygulaması için Çok Katmanlı İleri Beslemeli (MLP) bir YSA kullanılan bu çalışmada Lorenz sistemi için; sigma, beta, rho, Chua sistemi için; alfa, beta, a, b, ve Rössler sistemi için ise a, b, c, parametreleri yapay sinir ağlarının girişini, bu parametrelere karşılık gelen en büyük Lyapunov üstelleri ise yapay sinir ağlarının

çıkışını oluşturacak şekilde düzenlenmiş, her sistem için 1200 farklı değerden (parametreler-EBLÜ) oluşan veri setinin 960'ı (%80) eğitim verisi olarak, 240'ı (%20) ise test verisi olarak kullanılmıştır.

En uygun YSA modelini bulabilmek adına ilk olarak; 2 gizli katmana sahip bir ağda her bir gizli katman için 10 farklı nöron sayısı (1' den 10' a kadar) ve 2 farklı aktivasyon fonksiyonu (hiperbolik tanjant sigmoid, tansig ve logaritmik sigmoid, logsig) kullanılmış ve bu kombinasyon Tablo 5.5'de gösterilen 15 farklı Matlab eğitim algoritması için tekrarlanmıştır. Yani sonuç olarak Lorenz, Chua ve Rössler sistemlerinin her biri için 6000' er (10x10x2x2x15) adet farklı YSA modeli eğitilmiştir. Bu kombinasyon Şekil 5.1'de blok şekil olarak gösterilmiştir.

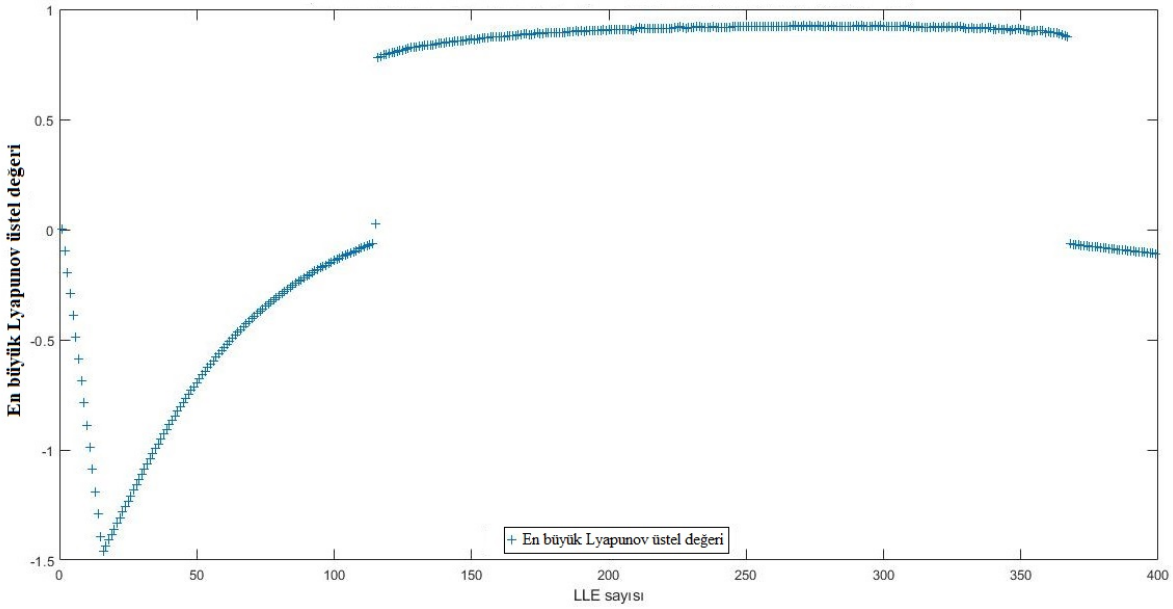
Lyapunov üstellerinin, parametre değerlerinin değiştirilmesine göre nasıl farklı değerler aldığını göstermek amacıyla Lorenz sisteminin Sigma (Prandtl sayısı - σ) parametresi 0,05 aralıklar ile değiştirilmiş ve elde edilen 400 farklı LLE değeri Şekil 5.2'de gösterilmiştir. Kaotik sistemlerin parametre değişikliğine verdiği tepkiye örnek olarak ise referans Sigmaya göre Sigma değerinin 0,1 artırıldığı Lorenz sisteminin zaman domenindeki cevabı Şekil 5.3'de gösterilmiştir. Sigma değerindeki 0,1'lik değişimin sistem davranışı üzerindeki etkisini bir başka şekilde göstermek için Şekil 5.4 verilmiştir.



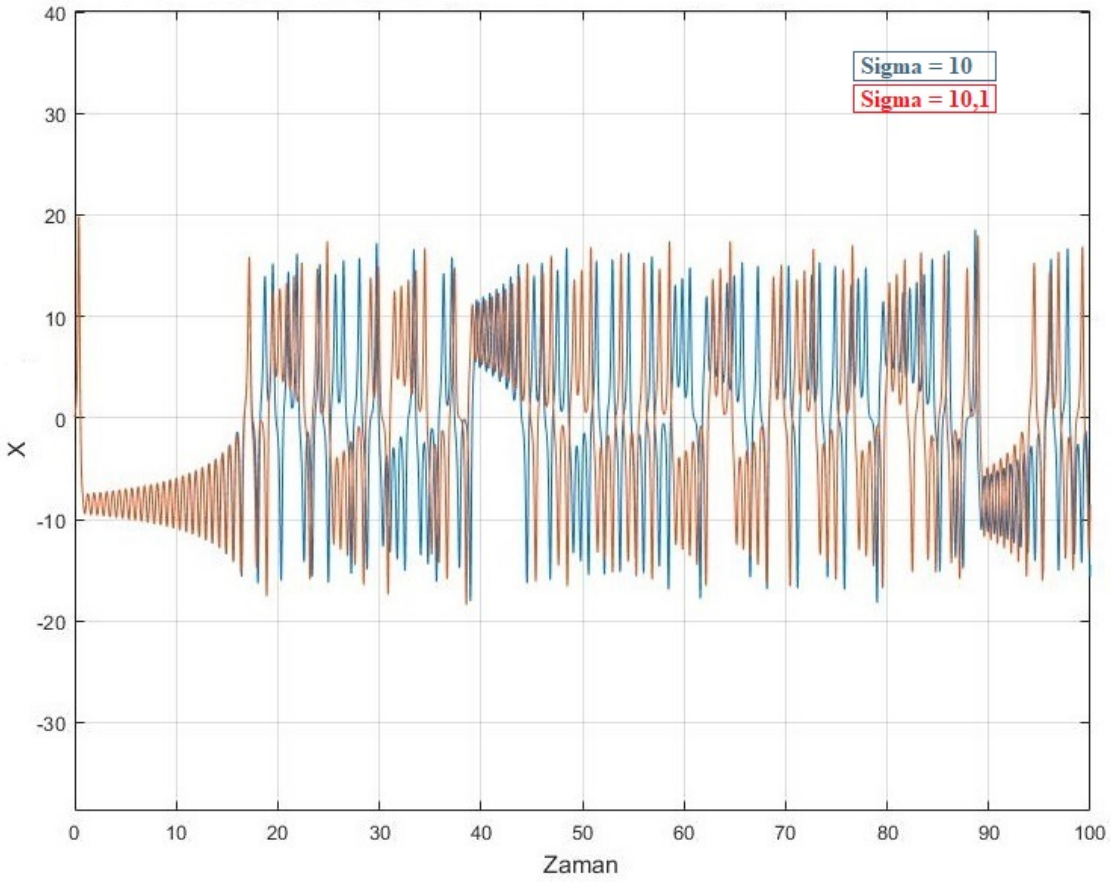
Şekil 5.1. Lorenz, Chua ve Rössler sistemlerinin her biri için ayrı ayrı uygulanan YSA kombinasyonları.

Tablo 5.5. Kullanılan Matlab eğitim algoritmaları.

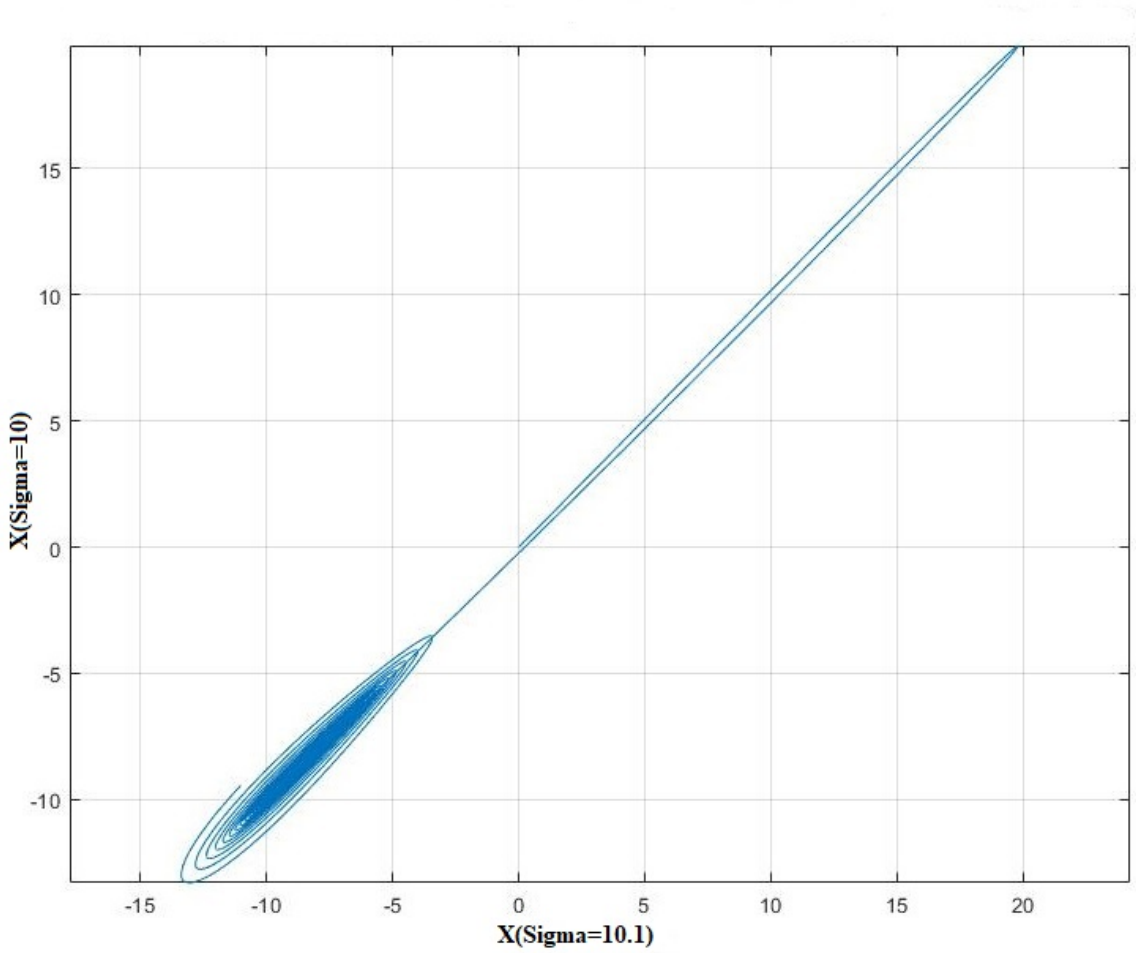
Sıra	Matlab Eğitim Algoritması	Açıklama	Tez içindeki kısaltması
1	trainb	Yığın (batch) Eğitim Algoritması	B
2	trainbfg	Broyden, Fletcher, Goldfarb ve Shanno Algoritması	BFGS
3	trainbr	Bayesyen Regülarizasyon Algoritması	BR
4	traincgb	Eşlenik Eğim Algoritması (Powell-Beale)	CGB
5	traincgf	Eşlenik Eğim Algoritması (Fletcher-Reeves)	CGF
6	traincgp	Eşlenik Eğim Algoritması (Polak-Ribiere)	CGP
7	traingd	Gradyent İniş (Gradient descent) Algoritması	GD
8	traingda	Gradyent İniş (Değişken (adaptive) öğrenme oranı geri yayılımı ile) Algoritması	GDA
9	traingdm	Gradyent İniş (Momentum ile) Algoritması	GDM
10	traingdx	Gradyent İniş (Değişken öğrenme oranı geri yayılımı ve momentum ile) Algoritması	GDX
11	trainlm	Levenberg-Marquardt Geri Yayılım Algoritması	LM
12	trainoss	Tek Adım Sekant Algoritması	OSS
13	trainrp	Esnek Yayılım Algoritması	RP
14	trains	Ardışık Sıra (Sequential order) Algoritması	S
15	trainscg	Ölçeklendirilmiş Eşlenik Eğim Algoritması	SCG



Şekil 5.2. Farklı Sigma değerleri için Lorenz sisteminin sahip olduğu LLE değerleri.



Şekil 5.3. $x_0 = 0$, $y_0 = 0$ ve $z_0 = 0$ başlangıç değerleri ve Beta = 8/3, Rho = 28 parametre değerlerine karşılık Sigma =10 (Mavi) ve Sigma =10,1 (Kırmızı) değerleri için Lorenz sisteminin ürettiği x(t) grafiği.



Şekil 5.4. Lorenz sisteminin parametre değişikliği halinde birbirlerine göre grafiği (Başlangıç değerleri: 0 , 1,01 , 0 Diferansiyel denklem çözüm hassasiyeti: 0,000001 Zaman adımı =15) X ekseni : Sigma = 10 Y ekseni Sigma = 10,1

Chua, Lorenz ve Rössler sistemlerinin bu tez çalışmasında ele alınan 15 Matlab eğitim algoritması ile elde edilen MSE cinsinden en iyi sonuçları sırası ile Tablo 5.6, Tablo 5.7 ve Tablo 5.8'de gösterilmiştir. Ayrıca Lorenz, Chua ve Rössler sistemleri için en iyi sonuç veren algoritma olan Levenberg-Marquardt Geri Yayılım Algoritması kullanılanarak elde edilen en iyi gerçek LLE/ağ çıkışı karşılaştırma grafikleri aşağıda verilmiştir.

Tablo 5.6. Chua sistemi için elde edilen en iyi sonuçlar (MSE).

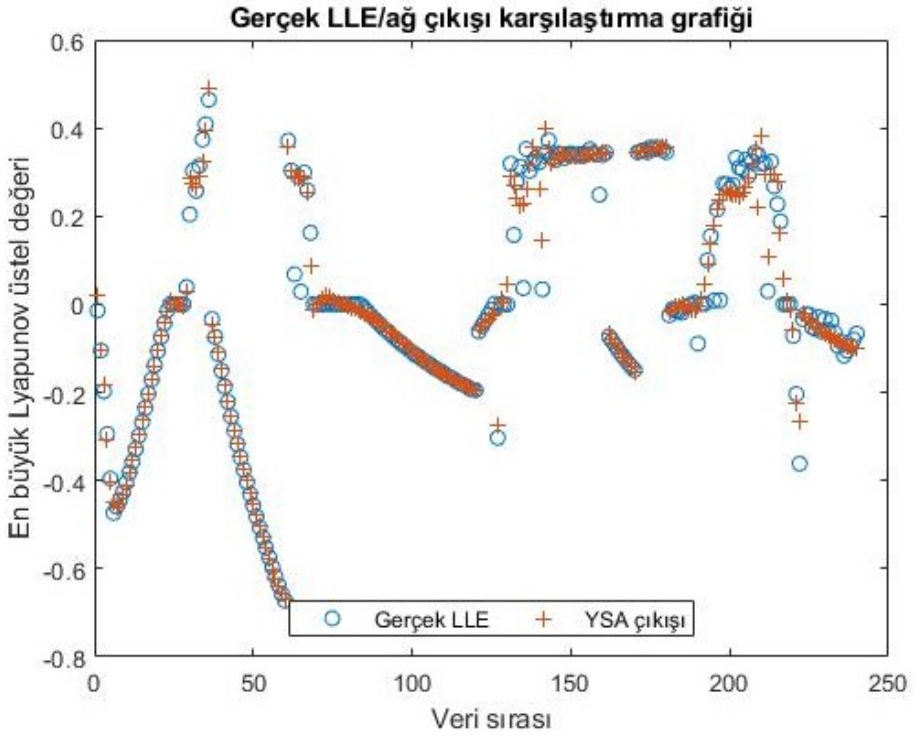
Algoritma	Test Hata Oranı	Eğitim Hata Oranı	Katmanlardaki Nöron Sayısı	Katmanlardaki Aktivasyon Fonksiyonları
LM	0,001669	0,002956	6x8	Tansig x Logsig
BR	0,001702	0,003107	9x7	Tansig x Logsig
BFGS	0,002377	0,005352	7x10	Tansig x Logsig
RP	0,002536	0,005815	7x9	Tansig x Logsig
SCG	0,002600	0,006172	5x8	Tansig x Logsig
CGB	0,002626	0,006148	8x9	Tansig x Tansig
CGP	0,002655	0,006283	8x4	Tansig x Logsig
CGF	0,002718	0,006646	9x5	Tansig x Tansig
OSS	0,002992	0,007354	8x7	Tansig x Tansig
GDX	0,005063	0,012614	4x7	Tansig x Tansig
GDA	0,007229	0,022221	10x6	Logsig x Logsig
GDM	0,010337	0,028083	8x6	Tansig x Tansig
GD	0,011254	0,030452	10x7	Tansig x Tansig
S	0,011598	0,030929	6x8	Tansig x Tansig
B	0,012231	0,030928	6x8	Tansig x Tansig

Tablo 5.7. Lorenz sistemi için elde edilen en iyi sonuçlar (MSE).

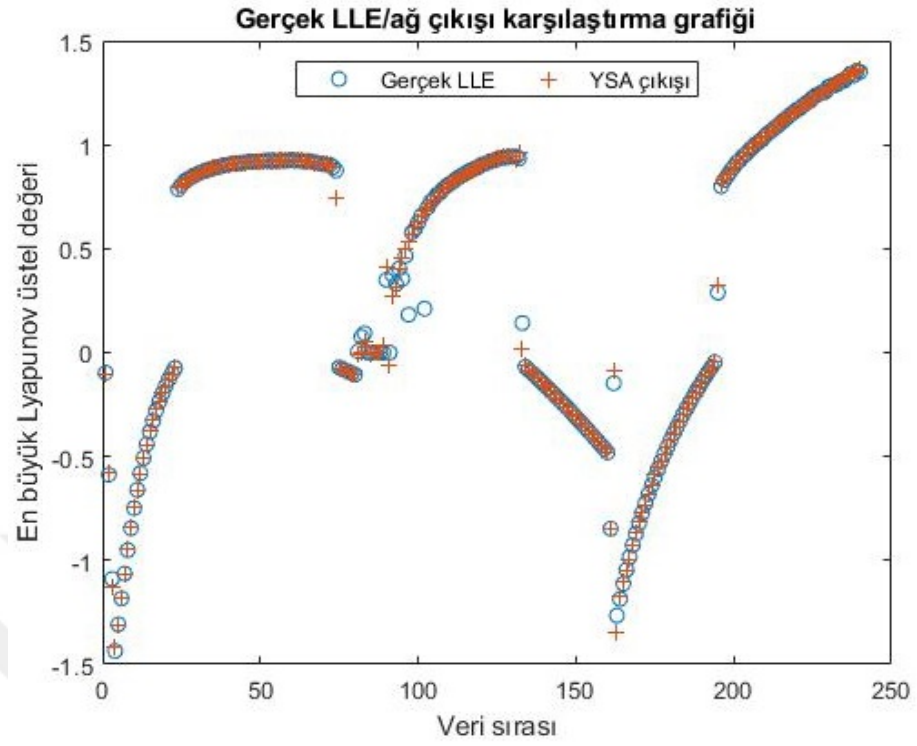
Algoritma	Test Hata Oranı	Eğitim Hata Oranı	Katmanlardaki Nöron Sayısı	Katmanlardaki Aktivasyon Fonksiyonları
LM	0,001809	0,000098	10x7	Tansig x Logsig
BR	0,002343	0,000252	6x7	Tansig x Tansig
BFGS	0,004052	0,000928	10x10	Tansig x Logsig
RP	0,005399	0,000899	6x9	Tansig x Logsig
CGB	0,007289	0,002623	6x5	Tansig x Tansig
CGF	0,008172	0,003208	10x5	Tansig x Logsig
SCG	0,008249	0,003340	9x8	Tansig x Logsig
CGP	0,008273	0,003161	10x9	Tansig x Tansig
OSS	0,011325	0,004919	7x10	Tansig x Tansig
GDX	0,029353	0,014150	10x8	Tansig x Logsig
GDA	0,034500	0,016576	5x10	Tansig x Tansig
GD	0,054298	0,026309	9x9	Tansig x Tansig
S	0,056229	0,026887	6x8	Tansig x Tansig
GDM	0,056681	0,027185	10x9	Tansig x Tansig
B	0,073386	0,036019	7x10	Logsig x Tansig

Tablo 5.8. Rössler sistemi için elde edilen en iyi sonuçlar (MSE).

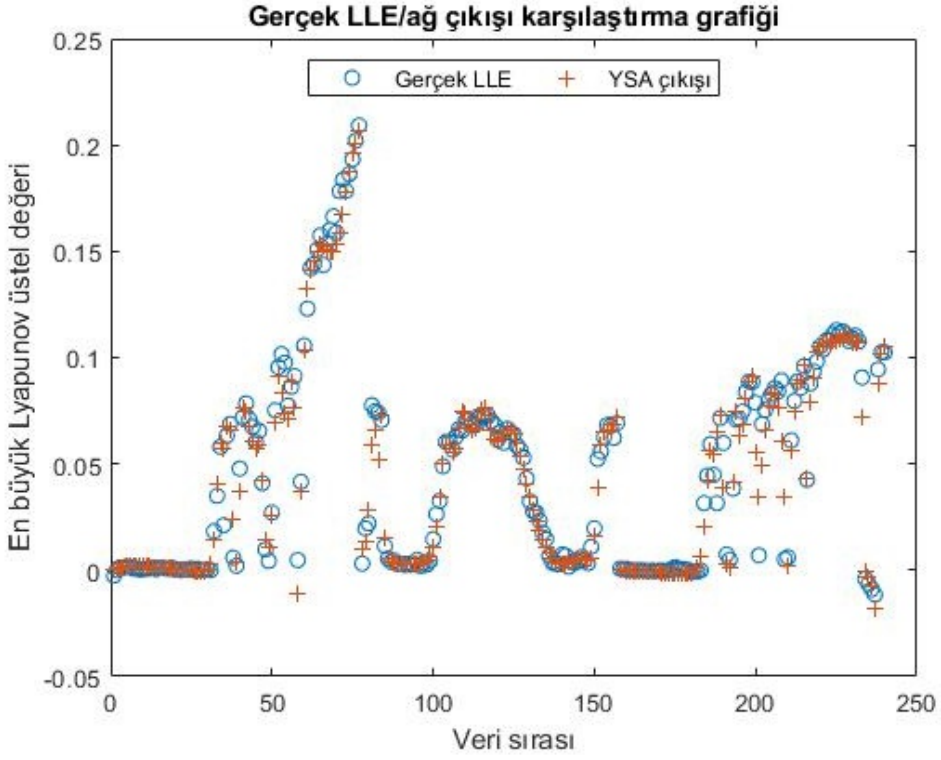
Algoritma	Test Hata Oranı	Eğitim Hata Oranı	Katmanlardaki Nöron Sayısı	Katmanlardaki Aktivasyon Fonksiyonları
LM	0,000054	0,007704	7x8	Tansig x Tansig
BR	0,000057	0,006463	9x9	Tansig x Logsig
BFGS	0,000116	0,013676	7x9	Tansig x Tansig
RP	0,000200	0,021444	6x8	Tansig x Logsig
CGF	0,000203	0,023024	7x6	Tansig x Tansig
CGB	0,000206	0,021111	5x9	Tansig x Tansig
CGP	0,000214	0,022585	8x9	Tansig x Tansig
SCG	0,000215	0,020766	8x10	Tansig x Tansig
OSS	0,000274	0,028366	10x7	Tansig x Tansig
GDX	0,000341	0,034140	7x8	Tansig x Tansig
GDA	0,000417	0,041716	10x2	Tansig x Tansig
GDM	0,000470	0,045965	9x9	Tansig x Tansig
S	0,000473	0,045001	9x10	Tansig x Tansig
GD	0,000488	0,046955	9x9	Tansig x Tansig
B	0,000606	0,055080	10x8	Logsig x Tansig



Şekil 5.5. Chua sistemi için en iyi sonucun gerçek LLE/ağ çıkışı karşılaştırma grafiği.



Şekil 5.6. Lorenz sistemi için en iyi sonucun gerçek LLE/ağ çıkışı karşılaştırma grafiği.



Şekil 5.7. Rössler sistemi için en iyi sonucun gerçek LLE/ağ çıkışı karşılaştırma grafiği.

3 sistem içinde en iyi algoritmalara bakıldığında ilk 4 sıranın aynı Matlab eğitim algoritmalarından (LM, BR, BFGS ve RP) oluştuğu görülmektedir. 5. sıranın ise değiştiği fakat hep eşlenik eğim (conjugate gradient) algoritmasına dayalı eğitim algoritmalarının bu sıraya yerleştiği gözlemlenmektedir. Zaten elde edilen en iyi sonuç tabloları incelendiğinde eşlenik eğime dayalı eğitim algoritmaları ile (SCG, CGB, CGF ve CGP) 3 sistem için de birbirine yakın sonuçlar elde edildiği görülmektedir.

Chua, Lorenz ve Rössler sistemleri için elde edilen en iyi 5 algoritma için en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranlarıda hesaplanmış ve ilk-ikinci gizli katman nöron sayıları ile birlikte aşağıda verilmiştir.

Tablo 5.9. Chua sistemi için elde edilen en iyi 5 algoritmada en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranları.

Algoritma	Yüzde bağıl hata oranı	Katmanlardaki Nöron Sayısı	Katmanlardaki Aktivasyon Fonksiyonları
LM	4,073017	7x8	Tansig x Logsig
BR	4,349564	10x6	Tansig x Logsig
BFGS	5,468888	4x10	Tansig x Logsig
SCG	6,536056	9x7	Tansig x Logsig
RP	6,611726	8x6	Tansig x Logsig

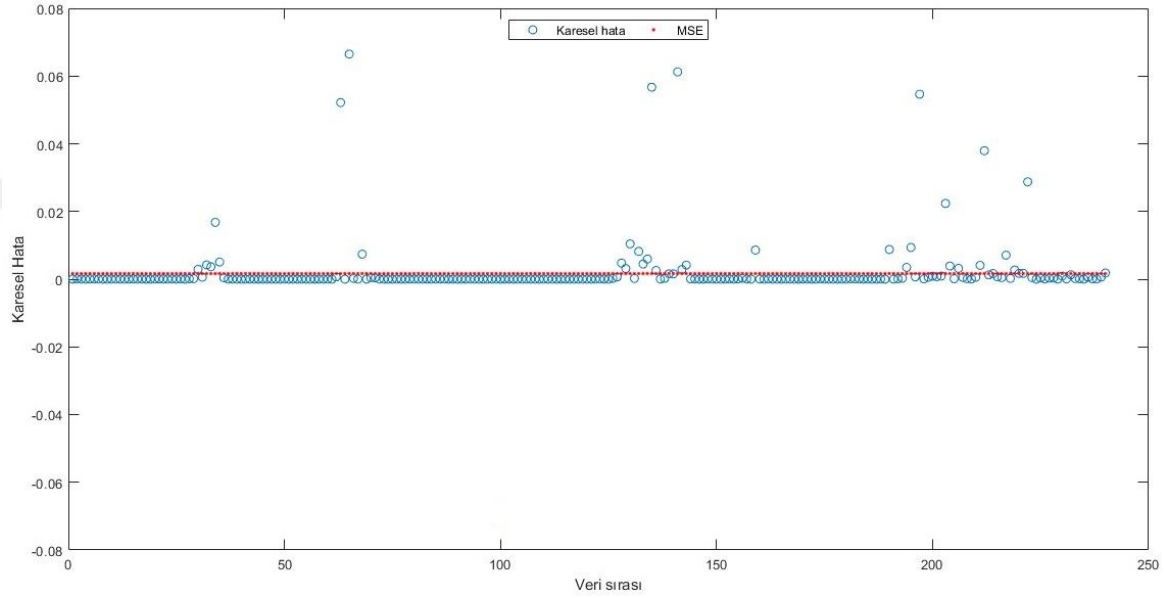
Tablo 5.10. Lorenz sistemi için elde edilen en iyi 5 algoritmada en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranları.

Algoritma	Yüzde bağıl hata oranı	Katmanlardaki Nöron Sayısı	Katmanlardaki Aktivasyon Fonksiyonları
LM	0,880957	8x8	Tansig x Logsig
BR	0,940397	10x7	Tansig x Tansig
BFGS	2,700301	9x10	Tansig x Tansig
RP	3,075843	8x9	Tansig x Logsig
CGB	3,568222	10x9	Tansig x Tansig

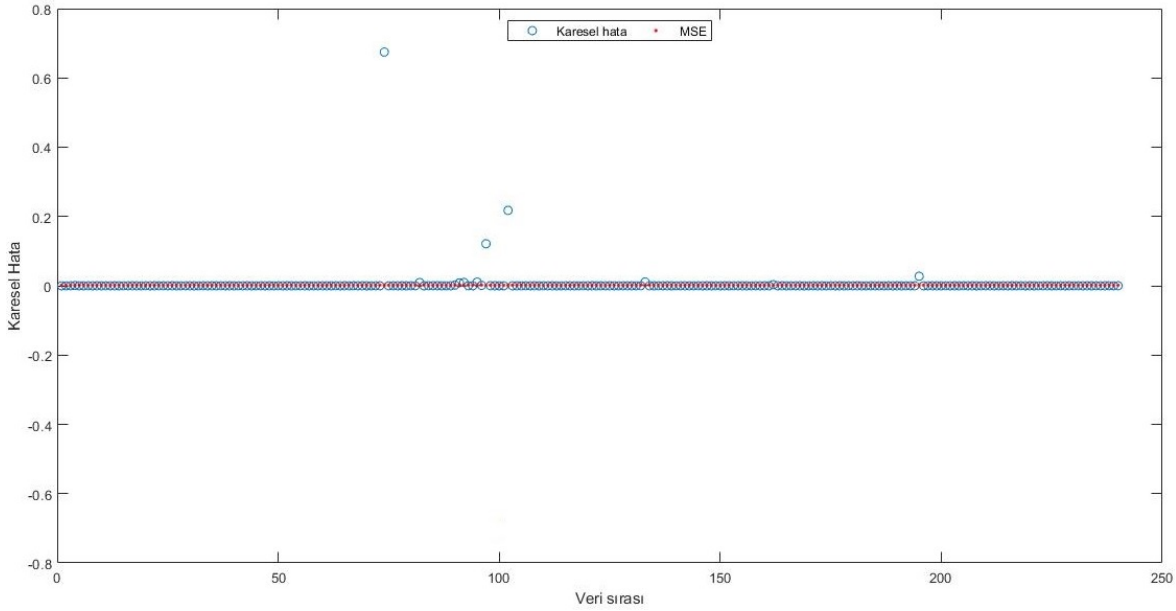
Tablo 5.11. Rössler sistemi için elde edilen en iyi 5 algoritmada en iyi algoritma-aktivasyon fonksiyonu çiftlerinin yüzde bağıl hata oranları.

Algoritma	Yüzde bağıl hata oranı	Katmanlardaki Nöron Sayısı	Katmanlardaki Aktivasyon Fonksiyonları
LM	5,747008	8x6	Tansig x Tansig
BR	6,169931	10x9	Tansig x Logsig
BFGS	8,992635	9x5	Tansig x Tansig
RP	10,198195	9x8	Tansig x Logsig
CGF	10,422349	7x7	Tansig x Tansig

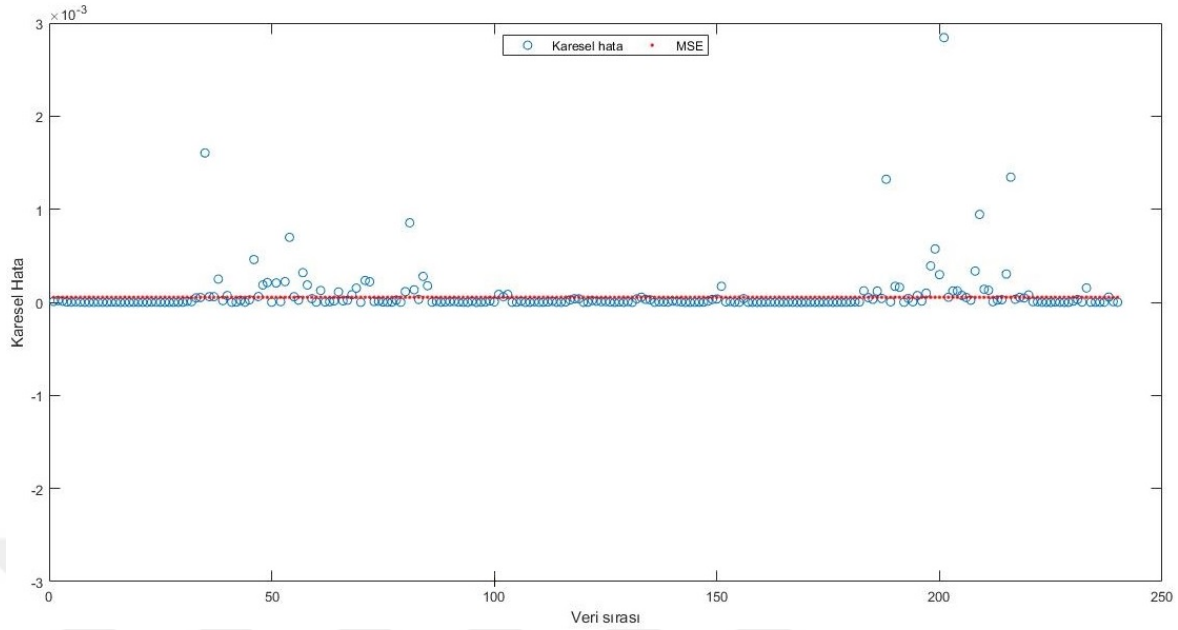
Chua, Lorenz ve Rössler sistemleri için gerçek en büyük Lyapunov üstelleri ile YSA tarafından bulunan en büyük Lyapunov üstelleri arasındaki farkın Ortalama Karesel Hata (MSE) cinsinden elde edilen iyi sonuçları alınmış ve bu sonuçlar tekil veri başına karesel hatalar ile karşılaştırılmıştır. Elde edilen grafikler Chua sistemi için Şekil 5.8'de, Lorenz sistemi için Şekil 5.9'da, Rössler sistemi için ise Şekil 5.10'da gösterilmiştir.



Şekil 5.8. Chua sistemi için elde edilen en iyi sonucun veri başına karesel hata - MSE karşılaştırması.



Şekil 5.9. Lorenz sistemi için elde edilen en iyi sonucun veri başına karesel hata - MSE karşılaştırması.



Şekil 5.10. Rössler sistemi için elde edilen en iyi sonucun veri başına karesel hata - MSE karşılaştırması.

Chua, Lorenz ve Rössler sistemlerin 4.dereceden Runge-Kutta (RK4) ve 8. Dereceden Runge-Kutta (RK8) metodları kullanılarak 10 iterasyon ile ve RK4 metodu kullanılarak 100 iterasyon ile 1 adet LLE'lerinin hesaplanması için gereken süreler saniye cinsinden Tablo 5.12'de verilmiştir.

Tablo 5.12. 1 adet en büyük Lyapunov üstelinin hesaplanması için gereken süre (saniye).

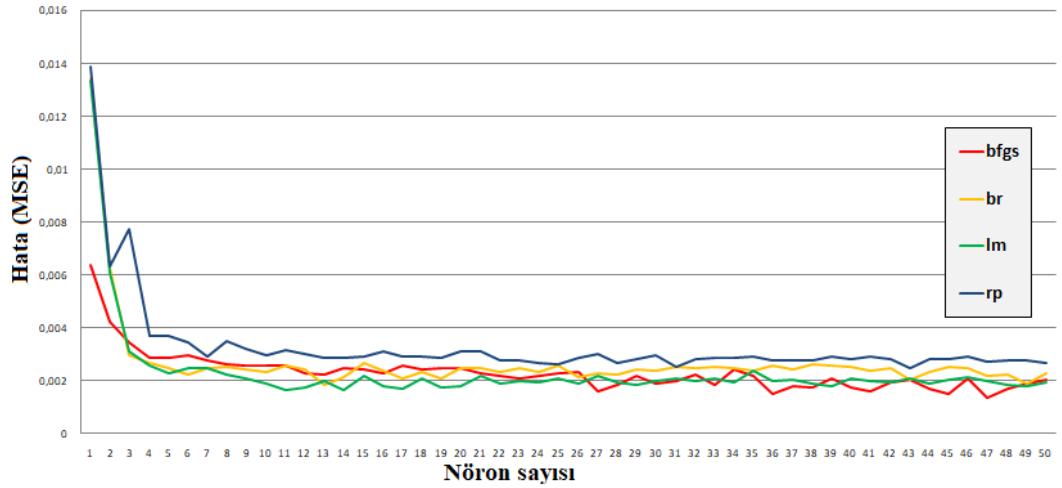
	RK4 (10 iterasyon)	**RK8 (10 iterasyon)**	**RK4 (100 iterasyon)**
Chua	28,618293s	394,261541s	281,557319s
Lorenz	29,484072s	474,694206s	304,539105s
Rössler	29,528852s	414,604868s	295,901611s

Çeşitli eğitim algoritmaları ile 4 adet eğitim verisi kullanılarak 1 adet LLE tahmini yapılabilmesi için geçen süreler saniye cinsinden Tablo 5.13'de verilmiştir.

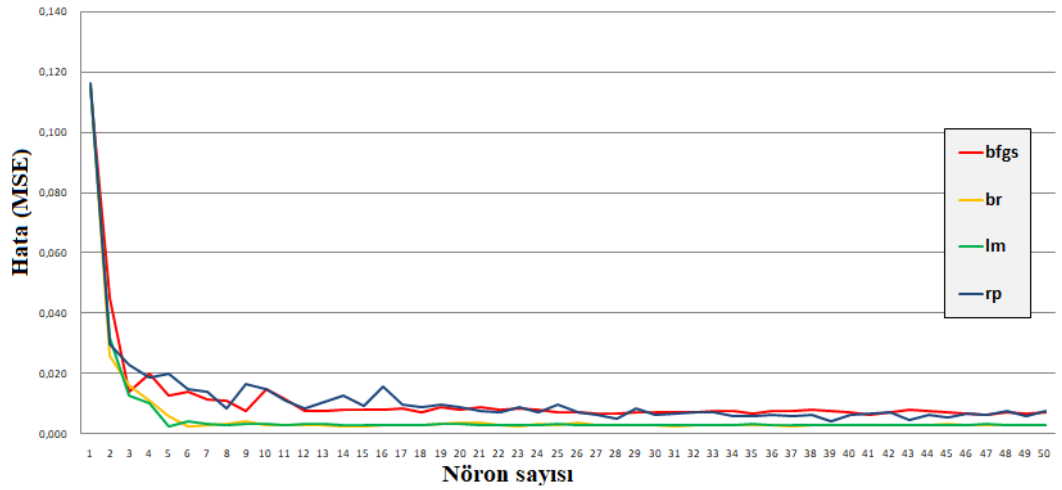
Tablo 5.13. Eğitim algoritmalarına göre 1 adet LLE tahmini yapılabilmesi için gereken saniye cinsinden eğitim ve test süreleri (Epoch:500, İlk katman nöron sayısı:5, İkinci katman nöron sayısı:5, İlk katman aktivasyon fonksiyonu: logsig, İkinci katman aktivasyon fonksiyonu: logsig).

Eğitim algoritması	**Eğitim süresi**	**Test Süresi**
GDX	2,633073s	0,174045s
RP	2,437990s	0,181679s
SCG	2,284476s	0,176142s
B	3,877936s	0,457317s
BFGS	2,427593s	0,174500s
BR	2,926481s	0,171247s
CGB	2,666431s	0,165939s
CGF	2,470813s	0,167488s
CGP	2,597657s	0,164333s
GD	2,921764s	0,219455s
GDA	2,983511s	0,166441s
GDM	2,880170s	0,233031s
LM	2,258660s	0,166016s
OSS	2,418030s	0,164673s
S	3,906590s	0,403503s

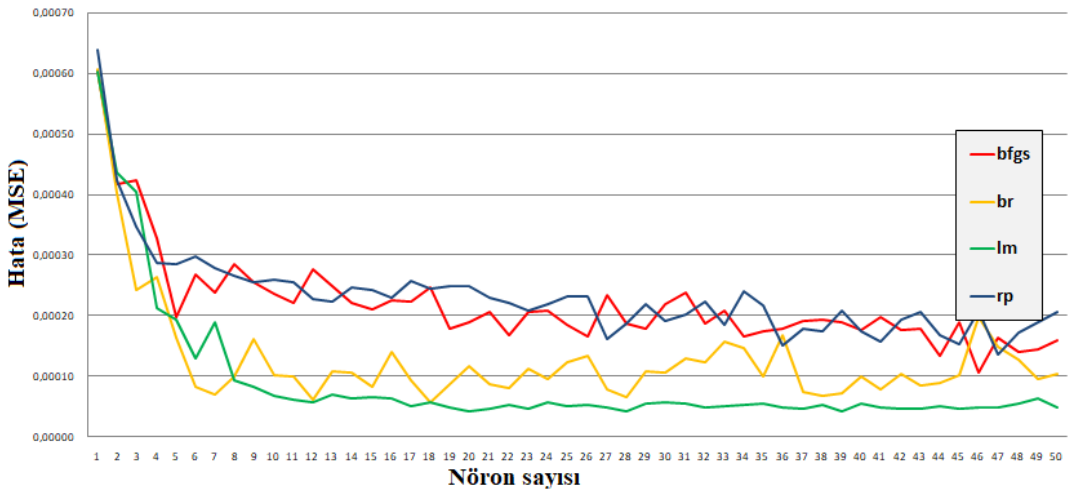
Gizli katmanlardaki nöron sayısının gerçek LLE ile YSA çıkışı LLE arasındaki MSE cinsinden hataya etkisini gözlemlemek amacı ile 3 sistem içinde en iyi 4 Matlab eğitim algoritması (LM, BR, BFGS ve RP) kullanarak ve geri kalan ağ yapısı sabit tutularak katmanlardaki nöron sayısı 1' den 50' ye kadar değiştirilmiş (katmanlardaki nöron sayıları birbirine eşit olacak şekilde: 1-1, 2-2, 3-3… 49-49, 50-50) ve sonuçlar gözlemlenmiştir. İlk gizli katman ve ikinci gizli katman aktivasyon fonksiyonları olarak logaritmik sigmoid aktivasyon fonksiyonu seçilmiş olup, nöron sayısının Chua sisteminden elde edilen sonuçlara etkisi için Şekil 5.11'e, aynı şekilde Lorenz sistemi için Şekil 5.12'ye ve Rössler sistemi için ise Şekil 5.13'e bakılabilir.



Şekil 5.11. Chua sistemi için nöron sayısı artırımının MSE cinsinden hataya etkisi.



Şekil 5.12. Lorenz sistemi için nöron sayısı artırımının MSE cinsinden hataya etkisi.



Şekil 5.13. Rössler sistemi için nöron sayısı artırımının MSE cinsinden hataya etkisi.

6. BÖLÜM
TARTIŞMA-SONUÇ ve ÖNERİLER

Bu tez çalışmasında önemli bir kaotiklik göstergesi olan en büyük Lyapunov üstelleri çok katmanlı ileri beslemeli geri yayılım algoritması kullanılan yapay sinir ağları ile tahmin edilmeye çalışılmıştır. Bu bağlamda literatürde önemli yeri olan Lorenz, Chua ve Rössler sistemleri için en büyük Lyapunov üstelleri 1200'er farklı parametre seti için hesaplatılmış ve elde edilen sonuçlar, çeşitli YSA modelleri için eğitim ve test verisi olarak kullanılmıştır. Tüm eğitimlerde 2 ara katmana sahip YSA modelleri kullanılmış olup bu ağların ara katmanlarındaki nöron sayıları, aktivasyon fonksiyonları ve Matlab eğitim algoritmaları değiştirilerek en iyi model bulunmaya çalışılmıştır.

Tablo 5.22'de görüldüğü gibi Chua sistemi için eğitilen YSA modelleri içinde MSE cinsinden en küçük test hatasına sahip olan model, ilk katman-ikinci katman 6x8 nöron sayılarına (Tablo 5.9) sahip ve trainlm eğitim algoritmasıyla eğitilen modeldir. Bu modelin test hatası 0,001669 olarak bulunmuştur. Chua sistemi için en iyi ikinci performans ise 9x7 nöron sayılarına (Tablo 5.10) sahip, trainbr eğitim algoritmasıyla eğitilen modele aittir. Bu model, 0,001702 test hatasına sahiptir.

Lorenz sistemi içinse Tablo 5.23'de verildiği gibi en iyi test hatası oranı 0,001810 bulunmuş olup bu sonuç 10x7 nöron sayıları (Tablo 5.14) ve trainlm eğitim algoritmasıyla elde edilmiştir. Lorenz sistemi için elde edilen en iyi ikinci sonuç, 6x7 nöron sayıları (Tablo 5.13) ve trainbr eğitim algoritmasıyla elde edilen 0,002343 değeridir.

Tablo 5.24'de görüldüğü gibi Rössler sistemi için en iyi test hata değeri olan 0,000054 değeri, 7x8 nöron sayıları (Tablo 5.20) ve trainlm eğitim algoritması ile, en iyi ikinci test hata değeri olan 0,000057 değeri ise 9x9 nöron sayıları (Tablo 5.18) ve trainbr

eğitim algoritması ile elde edilmiştir. Rössler sistemi için elde edilen 0,000054'lük test hata değeri, aynı zamanda bu tez çalışmasında elde edilen değerler arasındaki en iyi sonuçtur.

Matlab eğitim algoritmaları karşılaştırıldığında ise en iyi sonucun LM, daha sonra ise sırasıyla BR, BFGS ve RP algoritmaları ile alındığı gözlemlenmiştir.

Tablo 5.40 ve Tablo 5.41 dikkate alındığında eğitim için harcanan süreler hariç tutulma kaydıyla ele alınan sistemlerde LLE'yi tahmin etmek için gereken süre tüm algoritmaların ve sistemlerin (Chua, Lorenz, Rössler) ortalamaları alınarak kıyaslandığında RK4-10 iterasyon için yaklaşık **139 kat**, RK8-10 iterasyon için yaklaşık **2034 kat** ve RK4-100 iterasyon için yaklaşık **1396 kat,** test süresi en kısa olan CGP algoritması ile sistemlerin ortalaması alındığında ise RK4-10 iterasyon için yaklaşık **178 kat**, RK8-10 iterasyon için yaklaşık **2608 kat** ve RK4-100 iterasyon için yaklaşık **1792 kat,** fark olduğu görülmüş ve böylelikle hesaplaması uzun süre alabilen en büyük Lyapunov üstellerinin YSA'lar ile çok daha kısa sürede, küçük bir hatayla tahmin edilebileceği gözler önüne serilmiştir.

Chua sistemi için Şekil 5.25'de, Lorenz sistemi için Şekil 5.26'da ve Rössler sistemi için Şekil 5.27'de görüldüğü gibi gizli katmanlardaki nöron sayılarının test hatasına etkisini gözlemlemek amacı ile, diğer tüm parametreler sabit tutularak sadece katmanlardaki nöron sayısı artırıldığında, ilk başta test hatalarının yaklaşık 6-7 kat düştüğü fakat beşinci-altıncı nöron sayısından sonra nöron artırımının etkisinin azaldığı hatta duraksadığı gözlemlenmiştir.

Gelecek çalışmalarda Lyapunov üstellerinin YSA'lar kullanılarak tahmin edilmesi konusunda aşağıdaki iyileştirmeler yapılabilir;

- En büyük Lyapunov üstellerinin hesaplanması için kulanılan yörünge ayırma metodu işlemleri sırasında rastgele seçilen başlangıç değeri sayısı 10'dan fazla olabilir. Bu değer ne kadar büyük seçilirse, tahmin edilen Lyapunov üsteli gerçek değerine o kadar yakınsanır.

- Momentum, değişken öğrenme oranı ve benzerleri gibi değerler de değiştirilerek çeşitli kombinasyonlarda sonuçlar gözlemlenebilir.
- Lorenz, Chua ve Rössler sistemleri haricindeki diğer kaotik sistemler üzerinde de çalışma yapılabilir.
- Logaritmik Sigmoid ve Hiperbolik Tanjant Sigmoid dışındaki diğer aktivasyon fonksiyonları da kullanılabilir.

KAYNAKLAR

1. Dalkıran, İ., 2010. Yapay Sinir Ağı Tabanlı Kaotik İşaret Üreteci Tasarımı ve Uygulamaları. Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kayseri, 116 s.
2. Williams, G. P., 1997. Chaos Theory Tamed. Joseph Henry Press, Washington, 520pp.
3. Yılmaz, D., Güler, N. F., 2006. Kaotik Zaman Serileri Üzerine Bir Araştırma. **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, **21** (4): 759 779.
4. Sprott, J. C., 2000. Strange Attractors: Creating Patterns in Chaos. M & T Books, New York, 350 pp.
5. Wikipedia, 2014. Hadamard's Dynamical System. (Web page: **https://en.wikipedia.org/wiki/Hadamard%27s_dynamical_system**), (Erişim Tarihi: Nisan 2016).
6. Motter, A. E., Campbell, D. K., Chaos at fifty. **Phys. Today, 66** (5): 27-33.
7. Strogatz, S. H., 1994. Nonlinear Dynamics and Chaos. Perseus Books, Reading, 498 pp.
8. Wikipedia, 2016. Chaos Theory. (Web page: **https://en.wikipedia.org/wiki/Chaos_theory**), (Erişim Tarihi: Nisan 2016).
9. Matsumoto, T., 1984. A Chaotic Attractor from Chua's Circuit. **IEEE Transactions of Circuits and Systems, CAS-31** (12): 1055-1058.
10. Oseledec, V. I., 1968. A Multiplicative Ergodic Theorem. Lyapunov Characteristic Numbers for Dynamical Systems. **Trans. Moscow Math. Soc, 19** (2), 197-231.
11. Haken, H., Haken, H., Haken, H., Hénon, M., Heiles, C., 1964. The Applicability of the Third Integral of the Motion: **Some Numerical Results.Astron. J, 69**, 73.
12. Dabrowski, A., 2012. Estimation of the Largest Lyapunov Exponent from the Perturbation Vector and Its Derivative Dot Product. **Nonlinear Dynamics, 67** (1), 283-291.
13. Benettin, G., Galgani, L., Strelcyn, J. M., 1976. Kolmogorov Entropy and Numerical Experiments. **Physical Review A, 14** (6) , 2338.
14. Shimada, I., Nagashima, T., 1979. A Numerical Approach to Ergodic Problem of Dissipative Dynamical Systems. **Progress of Theoretical Physics, 61** (6), 1605 1616.
15. Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J. M., 1980. Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems - A

Method for Computing all of them. I-Theory. II Numericalapplication. **Meccanica, 15**, 9-30.

16. Wolf, A., 1986. Quantifying Chaos with Lyapunov Exponents. **Chaos**, 273-290.
17. Takens, F., 1981. Detecting Strange Attractors in Turbulence (pp. 366-381). Springer Berlin Heidelberg.
18. Wolf, A., Swift, J. B., Swinney, H. L., Vastano, J. A., 1985. Determining Lyapunov Exponents from a Time Series. **Physica D: Nonlinear Phenomena, 16** (3), 285-317.
19. Sano, M., Sawada, Y., 1985. Measurement of the Lyapunov Spectrum from a Chaotic Time Series. **Physical Review Letters, 55**(10), 1082.
20. Eckmann, J. P., Kamphorst, S. O., Ruelle, D., Ciliberto, S., 1986. Liapunov Exponents from Time Series. **Physical Review A, 34** (6), 4971.
21. Rosenstein, M. T., Collins, J. J., De Luca, C. J., 1993. A Practical Method for Calculating Largest Lyapunov Exponents from Small Data Sets. **Physica D: Nonlinear Phenomena, 65** (1), 117-134.
22. Parlitz, U., 1992. Identification of True and Spurious Lyapunov Exponents from Time Series. **International Journal of Bifurcation and Chaos, 2** (01), 155 165.
23. Kantz, H., 1994. A Robust Method to Estimate the Maximal Lyapunov Exponent of a Time Series. **Physics letters A, 185** (1), 77-87.
24. Parlitz, U,. 2016. Estimating Lyapunov Exponents from Time Series. In Chaos Detection and Predictability (pp. 1-34). Springer Berlin Heidelberg.
25. Yang, C., Wu, C. Q., & Zhang, P. 2012. Estimation of Lyapunov Exponents from a Time Series for n-dimensional State Space using Nonlinear Mapping. **Nonlinear Dynamics, 69** (4), 1493-1507.
26. Liao, H., 2016. Novel Gradient Calculation Method for the Largest Lyapunov Exponent of Chaotic Systems. **Nonlinear Dynamics,** 1-16.
27. Stefanski, A., 2000. Estimation of the Largest Lyapunov Exponent in Systems with Impacts. **Chaos, Solitons & Fractals, 11** (15), 2443-2451.
28. Stefanski, A., Dabrowski, A., & Kapitaniak, T. 2005. Evaluation of the Largest Lyapunov Exponent in Dynamical Systems with Time Delay. **Chaos, Solitons & Fractals, 23** (5), 1651-1659.

29. Kuznetsov, N. V., 2016. The Lyapunov dimension and its estimation via the Leonov method. **Physics Letters A, 380** (25-26), 2142-2149.
30. Hanbay, D., Türkoğlu, İ., Demir, Y., 2007. Chua Devresinin Yapay Sinir Ağı ile Modellenmesi. **Fırat Üniv. Fen ve Müh. Bil. Dergisi, 19** (1), 67-72.
31. Andrejević, M. V., Litovski, V. B., 2003. Electronic Circuits Modeling Using Artificial Neural Networks. **Journal of Automatic Control, 13** (1), 31-37.
32. Dalkiran, I., Danışman, K., 2010. Artificial Neural Network Based Chaotic Generator for Cryptology. **Turkish Journal of Electrical Engineering & Computer Sciences, 18** (2), 225-240.
33. Dalkiran, İ., Dalkiran, F. Y., Danişman, K., & Kiliç, R. 2007. Prediction of Dynamics of Chaotic Chua's Circuit with Artificial Neural Network. Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th, 1 4.
34. Maus, A. 2008. Using Artificial Neural Networks and Support Vector Regression to Model the Lyapunov Exponent. (Web page: **https://pdfs.semanticscholar.org/6773/c7e51567699c6aa43dc3fa9ce3fe17b797ee.pdf**), (Erişim Tarihi: Nisan 2016).
35. Golovko, V. 2003. Estimation of the Lyapunov Spectrum from One-Dimensional Observations Using Neural Networks. In Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings of the Second IEEE International Workshop on (pp. 95-98). IEEE.
36. Maus, A., Sprott, J. C., 2013. Evaluating Lyapunov Exponent Spectra with Neural Networks. **Chaos, Solitons & Fractals, 51**, 13-21.
37. Golovko, V., Savitsky, Y., 2014. Computing of Lyapunov Exponents Techniques Using Neural Networks. **International Journal of Computing, 3** (1), 93-98.
38. Savitsky, Y., Savitsky, A., 2015. Technique of Learning Rate Initialization for Efficient Training of MLP: Using for Computing of Lyapunov Exponents. In Information and Digital Technologies (IDT), 2015 International Conference on (pp. 298-301). IEEE.
39. Shintani, M., Linton, O., 2004. Nonparametric Neural Network Estimation of Lyapunov Exponents and a Direct Test for Chaos. **Journal of Econometrics, 120** (1), 1-33.

40. Übeyli, E. D., Güler, İ., 2004. Detection of Electrocardiographic Changes in Partial Epileptic Patients Using Lyapunov Exponents with Multilayer Perceptron Neural Networks. **Engineering Applications of Artificial Intelligence, 17** (6), 567-576.
41. Golovko, V., Artsiomenka, S., Kisten, V., Evstigneev, V., (2014). Towards Automatic Epileptic Seizure Detecion In Eegs Based on Neural Networks and Largest Lyapunov Exponent. **International Journal of Computing, 14** (1), 36-47.
42. Acar, H., Bayram, M., 2012. Estimation of Epileptic Seizure by Using Lyapunov Exponent, Wavelet Entropy and Artificial Neural Networks. In Signal Processing and Communications Applications Conference (SIU), 2012 20th (pp. 1-4). IEEE.
43. Palmer, T. N., 2009. Edward Norton Lorenz. 23 May 1917 -- 16 April 2008. Biographical Memoirs of Fellows of the Royal Society, 55: 139–155.
44. Crutchfield, J. P., Farmer, J. D., Packard, N. H., Shaw, R. S., 1986. Chaos. (Web page: **http://csc.ucdavis.edu/~chaos/courses/ncaso/Readings/Chaos_SciAm 1986/Chaos_SciAm 1986 .html**), (Erişim Tarihi: Nisan 2016).
45. Knightplumbing.ca, 2019. Dripping faucet photo. (Web page: **https://www.knightplumbing.ca/blog/5-common-plumbing-problems-and-their solutions/attachment/dripping-faucet/**), (Erişim Tarihi: Ocak 2019)
46. Alligood, K. T., Sauer, T. D., Yorke, J. A., 1996. Chaos : An Introduction to Dynamical Systems. Springer, New York, 603 pp.
47. Cvitanovic, P., Artuso, R., Mainieri, R., Tanner, G., Vattay,G., 2015. Chaos:Classical and Quantum. (Web page: **http://chaosbook.org/chapters/ChaosBook.pdf**) , (Erişim Tarihi: Nisan 2016).
48. Binous, H., 2007. Forced Duffing Oscillator. (Web page: **http://www.mathworks.com/matlabcentral/fileexchange/16731-forced duffing-oscillator**) , (Erişim Tarihi: Nisan 2016).
49. Broer, H., Takens, F., 2009. Dynamical Systems and Chaos. Springer, New York, 313 pp.
50. Abarnel, H. D. I., Brown, R., Sidorowich, J. J., Tsimring, L. S., 1993. The Analysis of Observed Chaotic Data in Psysical Systems. **Reviews of Modern Physics, 65** (4): 1331-1392.

51. Doğan, R., Genç, M., 2006. Yaşamımızı Çevreleyen İlginç Geometrik Şekiller: Fraktallar. **Journal of Istanbul Kültür University, 3**: 95-104.
52. Karaçay, T., 2005. Yirminci Yüzyılda Matematiği Sarsan Temel Düşünceler. **Matematik Dünyası, 2005 Bahar,** 62-71.
53. Taylor, R., 2005. Attractors: Nonstrange to Chaotic. (Web page: **https://www.siam.org/students/siuro/vol4/S01079.pdf**) , (Erişim Tarihi: Nisan 2016).
54. Çoban, G., 2007. Zaman Serilerinde Faz Uzayının Oluşturulması ve Kaosun Tespiti: Döviz Kurları Üzerine Bir Uygulama. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 143 s.
55. Parlitz, U., 1998. Nonlinear Time Series Analysis, pp. 209-239. In: Nonlinear Modelling-Advanced Black-Box Techniques (Eds: J.A.K. Suykens, J. Vandewalle). Kluwer Academic Publishers, Boston.
56. Fraser, A. M., Swimney, H. L., 1985. Independent Coordinates for Strange Attractors from Mutual Information. **Physical Review A, 33** (2): 1134 1140.
57. Schreiber, T., 1997. Detecting and Analysing Nonstationary in Time Series with Nonlinear Cross-Prediction. **Phys. Rev. Lett., 78** (843): 1-4.
58. Kantz, H., Schreiber, T., 2004. Nonlinear Time Series Analysis. Cambridge University Press, Cambridge, 369 pp.
59. Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., Farmer, J. D., 1992. Testing for Nonlinearity in Time Series: The Method of Surrogate Data. **Physica D, 58**: 77-94.
60. Liu, Z., 2010. Chaotic Time Series Analysis. **Mathematical Problems in Engineering, Vol 2010,** 31 pp.
61. Uzel, S., 2008. Zaman Serisi Analizi Yöntemi Üzerine Bir Uygulama. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 44 s.
62. Theiler, J., 1990. Estimating Fractal Dimension. **J. Opt. Soc. Am. A., 7** (6): 1055 1073.
63. Ott, E., 1993. Chaos in Dynamical Systems. Cambridge University Press, Canada, 385 pp.
64. Farmer, J. D., Ott, E., Yorke, J. A., 1983. The Dimension of Chaotic Attractors. **Physica D, 1983**: 153-180.

65. Al-Shameri, W. F. H., 2012. Correlation Dimension of an Attractor Generated by an Orbit of General Two-Dimensional Iterated Quadratic Map. **Int. J. Contemp. Math. Sciences, 7** (9): 413-424.
66. Sun, Y., 2011. Fault Detection in Dynamic Systems Using the Largest Lyapunov Exponent, Texas A&M University, Master's Thesis, Texas, 99pp.
67. Yükselen, M. A., 2008. HM504 Uygulamalı Sayısal Yöntemler Ders Notları. (Web sayfası: **http://web.itu.edu.tr/~yukselen/HM504/04%20Adi%20diferansiyel % 20denklemler.pdf**) (Erişim tarihi: Nisan 2016).
68. Meador, C. E. E., 2011. Numerical Calculation of Lyapunov Exponent for Three Dimensional Systems of Ordinary Differential Equations, Marshall University, Master's Thesis, Huntington, 70pp
69. Kablan, A., 2003. Runge-Kutta Metotlarının Yakınsaklık Mertebelerinin Tespiti. Gaziantep Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Gaziantep, 89 s.
70. Lorenz, E., 1963. Deterministic nonperiodic flow. **Journal of Atmospheric Sciences, 20**: 130-141.
71. Tucker, W., 1999. The Lorenz Attractor Exist. **C. R. Acad. Sci. Serie I (328)**: 1197 1202.
72. Hirsch, M. W., Smale, S., Devaney, R. L., 2004. Differential Equations, Dynamical Systems, and An Introduction to Chaos. Elsevier, California, 417 pp.
73. Igor, M., 2011. Lorenz Attractor Plot. (Web page: **http://www.mathworks.com/matlabcentral/fileexchange/30066-lorenz attaractor-plot/content/lorenz.m**) , (Erişim Tarihi: Nisan 2016).
74. Falat L. 2012. Numerical analysis of learning algorithms and strategies of multilayer perceptron neural network. Information and Communication Technologies International Conference (2012):12-17.
75. Rössler, O. E., 1976. An Equation for Continuous Chaos. **Physics Letters 57A** (5): 397-398.
76. Levermore, D., 2009. Rossler Chaotic Attractor. (Web page: **http://www.terpconnect. umd.edu /~lvrmr/2008-2009/S/Classes/MATH246/EXTRA/OusborneT.html**), (Erişim Tarihi: Nisan 2016).
77. Sambas, A., Sanjaya, M.W.S., Mamat, M., Halimatussadiyah 2013. Design and Analysis Bidirectional Chaotic Synchronization of Rossler Circuit and Its

Application for Secure Communication. **Applied Mathematical Scinces, 7** (1): 11-21.

78. Zhong, G.Q., Ayrom, F., 1985. Experimental confirmation of chaos from Chua's circuit. **International Journal of Circuit Theory and Applications**, 13: 93 98
79. Wikipedia, 2017. Chua's circuit. (Web page: **https://en.wikipedia.org/wiki/ Chua's_circuit**), (Erişim Tarihi: Nisan 2016)
80. Van Der Steen, R., 2006. Numerical and Experimental Analysis of Multiple Chua Circuits. Eindhoven University of Technology, Master's Thesis, Eindhoven, 72 pp.
81. Chua, L. O., Matsumoto, T., Komuro, M., 1985. The Double Scroll. **IEEE Transactions on Circuits and Systems (33)**: 798-818.
82. Sidersky, V., 2016. Matlab simulation of Chua's circuit. (Web page: **http://www.chuacircuits.com/matlabsim.php**) , (Erişim Tarihi: Nisan 2016).
83. Bayraktar, H. K., Mutlu, Ö., Koçal, O. H., 2017. Lyapunov Üstelleri İle İris Örüntüsünün Kaotik Yapısının İncelenmesi. **MSU Fen Bilimleri Dergisi**, **5** (1): 405-408.
84. Sprott, J.C., 2003. Chaos and Time-Series Analysis. Oxford University Press, New York, 507 pp.
85. Al Khatib, A., 2011. Developing a Wind Speed Prediction Tool Using ArtificialNeural Networks and Designing Wind Park in Syria Using Wasp Software. Cairo University, Kassel University, Master's Thesis, Cairo, Kassel, 180pp.
86. Baş, N., 2006. Yapay Sinir Ağları Yaklaşımı ve Bir Uygulama. Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 114 s.
87. McCulloch, W. S., Pitts, W. H., 1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics (5)**: 155-133.
88. Hebb, D. O., 1949. The Organization of Behavior; A Neuropsychological Theory. John Wiley & Sons, New York, 335 pp.
89. Goldstein, A., 1997. Bernard Widrow Oral History. (Web page: **http://www.ieeeghn.org/wiki/index.php/Oral-History:Bernard_Widrow**) (Erişim Tarihi: Nisan 2016).
90. Minsky, M. L., Papert, S. A., 1987. Perceptrons: An Introduction to Computational Geometry. The MIT Press, Cambridge, 308 pp.

91. Tas, O., Yaka, E., Ugurlu, U. 2018. Using artificial neural network and a statistical method for the estimation of Euro/Turkish Lira Exchange rate. **PressAcademia Procedia (PAP), V.7**, p.414-417

92. Atienza, F., Alzamora, N. M., De Velasco, J. A., Dreiseitl, S., Machado, L. O., 2000. Risk Stratification in Heart Failure Using Artificial Neural Networks. AMIA Symp. 2000, 4-8 Kasım 2000, Los Angeles, ABD, 32-36.

93. Shrivastava, V., Sharma, N., 2012. Artificial Neural Network Based Optical Character Recognition. **Signal & Image Processing: An International Journal, 3** (5): 73-80.

94. Uncini, A., 2003. Audio Signal Processing by Neural Networks. **Neurocomputing 55**: 593 – 625.

95. Karaatlı, M., Helvacıoğlu, Ö. C., Ömürbek, N., Tokgöz, G., 2017. Yapay Sinir Ağları Yöntemi ile Otomobil Satış Tahmini. **Uluslararası Yönetim ve İşletme Dergisi, 8** (17): 87-100.

96. Aksel, F., 2000. Esnek Regresyon Analizi ve Yapay Sinir Ağı Yöntemleri ile Uzun Dönem Yük Tahmini. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 92 s.

97. Azeez, O. S., Pradhan, B., Shafri, H. Z., Shukla, N., Lee, C. W., & Rizeei, H. M. 2019. Modeling of CO emissions from traffic vehicles using artificial neural networks. **Applied Sciences, 9** (2), 313.

98. Fausett, L., 1993. Fundamentals of Neural Networks; Architectures, Algorithmsand Applications. Prentice Hall, New Jersey, 461 pp.

99. Zurada, J. M., 1992. Introduction to Artificial Neural Systems. West Publishing, St. Paul, 683 pp.

100. Haykin, S., 2005. Neural Networks; A Comprehensive Foundation. Pearson Education, Delhi, 823pp.

101. Demuth, H., Beale, M., 2002. Neural Network Toolbox for Use with Matlab. The Mathworks, Natick, 840pp.

102. Bozik, O., 2011. Ana Bileşenler Analizine Dayalı Yapay Sinir Ağı Yaklaşımı ile Radarla Uçak Sınıflandırma. Hava Harp Okulu Komutanlığı, Havacılık ve Uzay Teknolojileri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 134 s.

103. Sağıroğlu Ş., Beşdok, E., Erler, M., 2003. Mühendislikte Yapay Zeka Uygulamaları I: Yapay Sinir Ağları. Ufuk Kitap, Kayseri, 426 s.

104. Altun, A. A., 2007. Esnek Hesaplama Yöntemleri ile Otomatik Parmak İzi Tanıma. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Konya, 228 s.

105. Moller, M. F., 1993. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. **Neural Networks, 6**: 525-533.

106. Wang ,W. , Peter, H.A.J.M., Vrijling, J. K., 2007. Comparing Bayesian regularization and cross-validated early-stopping for stremflow forecasting with ANN models. **Methodology in Hydrology, (311)**: 216-221.

107. Aşkın, D., İskender, İ., Mazımadeh, A., 2011. Farklı yapay sinir ağları yöntemlerini kullanarak kuru tip transformatör sargısının termal analizi. **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 26** (4) :905 911.

108. Ranka, S., Mehrotra, K., Mohan, C. K., 1996. Elements of Artificial Neural Networks. A Bradford Book, Cambridge, 344 pp.

109. Hagan, M. T., Demuth, H. B., Beale, M. H., De Jesus, O., Neural Network Design. (Web Page: **http://hagan.okstate.edu/NNDesign.pdf**), (Erişim Tarihi: Nisan 2016).

110. Taş, E., 2008. Geliştirilmiş Yapay Sinir Ağı Algoritmaları ve Uygulamaları. Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Doktora tezi, Eskişehir, 94s.

111. Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., Alkon, D. L., 1998. Accelerating the Convergence of the Back Propagation Method. **Biological Cybernetics, 59**: 257-263.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Koray Fethi PARLAK

Uyruğu: Türkiye (TC)

Doğum Tarihi ve Yeri: 3 Haziran 1985, Kayseri

Medeni Durumu: Bekâr

E-mail: korayfethi@gmail.com

Yazışma Adresi: Alpaslan Mah. Bahar Cad. Kandil Sk. Bezciler Sitesi C blok No: 9 Melikgazi/KAYSERİ

EĞİTİM

Derece	**Kurum**	**Mezuniyet Tarihi**
Yüksek Lisans	EÜ Fen Bilimleri Enstitüsü	-
Lisans	EÜ Elektrik-Elektronik Müh.	2007
Lise	N. M. Küçükçalık A. L., Kayseri	2002

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görev**
2018-2019	Türk Hava Yolları Teknik	Tesis İşletme Mühendisi
2017-2018	Ht-Saae	Ekipman Mühendisi
2014-2016	Boyteks	Yardımcı İşletmeler Mühendisi
2009–2011	Aspilsan	Arge Mühendisi

YABANCI DİL

İngilizce