

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**ECHO STATE NETWORK İLE SİSTEMLERİN MODELLENMESİ**

**YÜKSEK LİSANS TEZİ**

**Selin YAMAN**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı**

**Kontrol ve Otomasyon Mühendisliği Programı**

**MAYIS 2014**



**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**ECHO STATE NETWORK İLE SİSTEMLERİN MODELLENMESİ**

**YÜKSEK LİSANS TEZİ**

**Selin YAMAN  
(504121128)**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı**

**Kontrol ve Otomasyon Mühendisliği Programı**

**Tez Danışmanı: Yrd. Doç. Dr. Engin YEŞİL**

**MAYIS 2014**



İTÜ, Fen Bilimleri Enstitüsü'nün 504121128 numaralı Yüksek Lisans Öğrencisi **Selin YAMAN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**ECHO STATE NETWORK İLE SİSTEMLERİN MODELLENMESİ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :** **Yrd. Doç. Dr. Engin YEŞİL** .....  
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :** **Yrd. Doç. Dr. Gülay Öke GÜNEL** .....  
İstanbul Teknik Üniversitesi

**Yrd. Doç. Dr. İlker ÜSTOĞLU** .....  
Yıldız Teknik Üniversitesi

**Teslim Tarihi :** **5 Mayıs 2014**  
**Savunma Tarihi :** **30 Mayıs 2014**



*Rüzgâr Yaman Aydınol'a,*





## ÖNSÖZ

Bu tez çalışması sırasında öncelikle, her zaman bilgi ve fikirlerini aktarmaya çalışan değerli tez danışmanım Yrd. Doç. Dr. Engin Yeşil'e, bu sürecin oluşturulmasındaki katkılarından ötürü Yrd. Doç. Dr. İlker Üstoğlu'na, bu süreçte bilgi ve tecrübelerini benimle paylaşan Aydoğın Ersöz'e teşekkürlerimi sunarım. Çalışma sürecinin başından sonuna kadar desteklerini sürekli hissettiren aileme ve bu sürede her türlü desteği ve gösterdiği sabır için Saeid Rostami'ye de sonsuz teşekkür ederim.

Mayıs 2014

Selin Yaman  
(Kontrol Mühendisi)



## İÇİNDEKİLER

### Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER .....	ix
KISALTMALAR .....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY .....	xix
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. YAPAY SİNİR AĞLARI.....</b>	<b>3</b>
2.1 Giriş.....	3
2.2 Aktivasyon Fonksiyonları .....	4
2.3 Ağ Yapıları .....	6
2.3.1 Tek katmanlı ileri beslemeli ağlar .....	6
2.3.2 Çok katmanlı ileri beslemeli ağlar .....	7
2.3.3 Tekrarlı yapay sinir ağları .....	7
2.4 Öğrenme Algoritmaları .....	8
2.4.1 Danışmanlı öğrenme .....	8
2.4.2 Danışmansız öğrenme .....	9
2.4.3 Takviyeli öğrenme .....	9
2.5 Modeller .....	9
2.5.1 Hebb modeli .....	9
2.5.2 Perceptron modeli .....	10
2.5.3 Geriye yayılım modeli.....	11
<b>3. ECHO STATE NETWORK .....</b>	<b>13</b>
3.1 Yapısal Özellikler.....	13
3.2 Hesaplama ve Eğitim .....	19
<b>4. BULANIK MANTIK .....</b>	<b>27</b>
4.1 Giriş.....	27
4.2 Bulanık Kümeler ve Üyelik Fonksiyonları .....	27
4.3 Bulanık Sistemler .....	30
4.3.1 Bulanıklaştırma .....	31
4.3.2 Kural tabanı.....	31
4.3.3 Çıkarım mekanizması .....	32
4.3.3.1 Mamdani tipi yapı için çıkarım mekanizması.....	32
4.3.3.2 Takagi - Sugeno tipi yapı için çıkarım mekanizması.....	33
4.3.3.3 Tekli (Singleton) tip yapı için çıkarım mekanizması.....	34
4.3.4 Durulaştırma.....	35
4.3.4.1 Ağırlık merkezi yöntemi .....	35
4.3.4.2 Ağırlıklar ortalaması yöntemi .....	35
4.4 Bulanık Modelleme .....	36
4.4.1 Beyaz kutu modelleme yöntemi.....	37

4.4.2 Kara kutu modelleme yöntemi .....	38
4.4.3 Gri kutu modelleme yöntemi.....	38
<b>5. BENZETİM ÇALIŞMALARI .....</b>	<b>45</b>
5.1 MATLAB PT326 Sisteminin Modellenmesi.....	46
5.1.1 ANFIS ile oluşturulan bulanık model .....	47
5.1.2 ESN ile oluşturulan sistem modeli .....	48
5.2 Zaman Serilerinin Modellenmesi .....	51
5.2.1 ANFIS ile oluşturulan bulanık model .....	51
5.2.2 ESN ile oluşturulan sistem modeli .....	53
5.3 Çok Giriş Tek Çıkış Sistemlerin Modellenmesi.....	55
5.3.1 ANFIS ile oluşturulan bulanık model .....	56
5.3.2 ESN ile oluşturulan sistem modeli .....	57
<b>6. AKTİF SÜSPANSİYON DENEY SETİ.....</b>	<b>59</b>
6.1 ANFIS ile Oluşturulan Bulanık Model.....	61
6.2 ESN ile Oluşturulan Sistem Modeli .....	63
<b>7. SONUÇLAR .....</b>	<b>65</b>
<b>KAYNAKLAR.....</b>	<b>67</b>
<b>ÖZGEÇMİŞ.....</b>	<b>69</b>

## **KISALTMALAR**

<b>ANFIS</b>	: Adaptive Neuro – Fuzzy Inference System
<b>YSA</b>	: Yapay Sinir Ağları
<b>ESN</b>	: Echo State Network
<b>RNN</b>	: Recurrent Neural Network
<b>ANN</b>	: Artificial Neural Network
<b>ART</b>	: Adaptive Resonance Theory
<b>SOM</b>	: Self Organizing Map
<b>GA</b>	: Genetik Algoritma
<b>RH</b>	: Rezervuar Hesaplama
<b>DR</b>	: Dinamik Rezervuar
<b>ÇGTC</b>	: Çok Giriş Tek Çıkış Sistem
<b>NARMA</b>	: Nonlinear Autoregressive Moving Average



## ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 4.1 : Yayılım ve yöntem tablosu.....	42
Çizelge 5.1 : PT326 sistemi için hata analizi.....	50
Çizelge 5.2 : Rossler zaman serisi için hata analizi.....	55
Çizelge 5.3 : ÇGTÇ sistem için hata analizi.....	58
Çizelge 6.1 : Aktif süspansiyon deney seti için hata analizi .....	64





## ŞEKİL LİSTESİ

### Sayfa

Şekil 2.1 :Yapay sinir ağı yapısı .....	4
Şekil 2.2 :Lineer fonksiyon.....	5
Şekil 2.3 :Rampa eşik fonksiyon .....	5
Şekil 2.4 :İşaret fonksiyonu. ....	6
Şekil 2.5 :Sigmoid fonksiyon. ....	6
Şekil 2.6 :Tek katmanlı ileri beslemeli ağ yapısı.....	7
Şekil 2.7 :Çok katmanlı ileri beslemeli ağ yapısı .....	8
Şekil 2.8 :Tekrarlı yapay sinir ağı yapısı. ....	8
Şekil 3.1 :Temel echo state network yapısı .....	15
Şekil 3.2 :Echo state network yapısı.....	21
Şekil 4.1 :Öğrenci başarısı için bulanık küme gösterimi .....	28
Şekil 4.2 :Üyelik fonksiyonu bileşenleri .....	29
Şekil 4.3 :Üyelik fonksiyonu türleri. ....	30
Şekil 4.4 :Bulanık sistem yapısı .....	31
Şekil 4.5 :İki giriş iki kural Mamdani tipi için çıkarım mekanizması.....	33
Şekil 4.6 :İki giriş iki kural Takagi – Sugeno tipi yapı için çıkarım mekanizması. ..	34
Şekil 4.7 :İki giriş iki kural tekli tip yapı için çıkarım mekanizması .....	34
Şekil 4.8 :Alanın ağırlık merkezi.....	35
Şekil 4.9 :Alanın ağırlık ortalaması .....	36
Şekil 4.10 :ANFIS kullanıcı ara yüzü.....	39
Şekil 4.11 :ANFIS adaptif sinir ağı yapısı.....	40
Şekil 5.1 :PT326 sisteminin çıkış verisi .....	47
Şekil 5.2 :PT326 sistemi için ANFIS eğitim verisi grafiği.....	47
Şekil 5.3 :PT326 sistemi için ANFIS test verisi grafiği .....	48
Şekil 5.4 :PT326 sistemi için ANFIS öğrenme grafiği.....	48
Şekil 5.5 :PT326 sistemi için ESN rezervuar durumları .....	49
Şekil 5.6 :PT326 sistemi için ESN eğitim verisi grafiği.....	49
Şekil 5.7 :PT326 sistemi için ESN test verisi grafiği .....	50
Şekil 5.8 :Rossler zaman serisi çıkış verisi.....	51
Şekil 5.9 :Rossler zaman serisi içinANFIS eğitim verisi grafiği.....	52
Şekil 5.10 :Rossler zaman serisi içinANFIS test verisi grafiği .....	52
Şekil 5.11 :Rossler zaman serisi içinANFIS öğrenme grafiği.....	53
Şekil 5.12 :Rossler zaman serisi içinESN rezervuar durumları.....	53
Şekil 5.13 :Rossler zaman serisi için ESN eğitim verisi grafiği.....	54
Şekil 5.14 :Rossler zaman serisi için ESN test verisi grafiği .....	54
Şekil 5.15 :ÇGTÇ sistemin çıkış verisi .....	55
Şekil 5.16 :ÇGTÇ sistem için ANFIS eğitim verisi grafiği.....	56
Şekil 5.17 :ÇGTÇ sistem için ANFIS test verisi grafiği .....	56
Şekil 5.18 :ÇGTÇ sistem için ESN rezervuar durumları.....	57

<b>Şekil 5.19</b> :ÇGTÇ sistem için ESN eğitim verisi grafiği.....	57
<b>Şekil 5.20</b> :ÇGTÇ sistem için ESN test verisi grafiği.....	58
<b>Şekil 6.1</b> :Aktif süspansiyon deney setinin görünümü .....	59
<b>Şekil 6.2</b> :Yay–damper–eyleyici+yay–damper yapısındaki çeyrek araba modeli.....	60
<b>Şekil 6.3</b> :Aktif süspansiyon sistemi için ANFIS öğrenme grafiği.....	61
<b>Şekil 6.4</b> :Aktif süspansiyon sistemi için ANFIS eğitim verisi grafiği .....	62
<b>Şekil 6.5</b> :Aktif süspansiyon sistemi için ANFIS test verisi grafiği .....	62
<b>Şekil 6.6</b> :Aktif süspansiyon sistemi için ESN rezervuar durumları .....	63
<b>Şekil 6.7</b> :Aktif süspansiyon sistemi için ESN eğitim verisi grafiği .....	63
<b>Şekil 6.8</b> :Aktif süspansiyon sistemi için ESN test verisi grafiği .....	64

## ECHO STATE NETWORK İLE SİSTEMLERİN MODELLENMESİ

### ÖZET

Modelleme genel anlamda, bir sistemin deęişen koşullar altındaki davranışlarını incelemek, benzetimini yapmak ve kontrol etmek amacıyla elemanları arasındaki bağlantıları matematiksel olarak ifade etmektir. Gerçek hayatta kullanılan çoęu sistem doğrusal deęildir ve bu sistemlerin iç yapıları epey karmaşıktır. Bu durum, sistemlerin matematiksel modellerinin çıkarılmasını zorlaştırmakta hatta bazen imkansız kılmaktadır. Modellemede karşılaşılan bu zorluklar, bulanık modelleme ve yapay sinir aęları ile modelleme yöntemlerinin doğmasına sebep olmuştur. Gerçek sistemden elde edilen giriş – çıkış verileri ile sistemin bulanık modelinin oluşturulması bulanık modelleme; yine gerçek sistem verileri ile sistemin yapay sinir aęı modelinin oluşturulması ise yapay sinir aęları ile modelleme olarak tanımlanabilir.

Yapay sinir aę modellerinden tekrarlı yapay sinir aęlarınınını eğitmek için kullanılan gradyal azaltma yöntemine alternatif olması amacıyla son yıllarda “rezervuar hesaplama” yöntemi ortaya atılmış ve bu konuda çalışmalar yapılmıştır. Rezervuar hesaplamanın anahtar yöntemlerinden olan echo state network, bu alandaki aktif çalışmalardan biridir. Pratik, kavramsal olarak basit ve kolay uygulanabilir olduęu için pek çok alanda başarılı bir performans göstermektedir.

Bu çalışmanın amacı, herhangi bir sistemden toplanan giriş – çıkış verilerini kullanarak sistemin echo state network kullanarak modellenmesinin yapılmasıdır. Bu amaçla dört farklı sistemden alınan giriş – çıkış verileri ile sistemlerin modelleri echo state network ile oluşturulmuş ve ardından en popüler bulanık modelleme yazılımı olan ANFIS ile başarıları karşılaştırılmıştır.



## **SYSTEM MODELING USING ECHO STATE NETWORK**

### **SUMMARY**

In general, modeling means expressing the connections between its elements mathematically to analyze the system behaviour under changing conditions, to simulate and to control it. Most of the systems used in real life are not linear and internal structures of these systems are quite complex so it is hard or sometimes even impossible to design mathematical models of these systems. The difficulties encountered in modeling has led to the creation of fuzzy modeling and artificial neural network modeling methods. The creation of fuzzy model by using input – output data that are collected from real system is fuzzy modeling; again the creation of a model with ANNs with actual system data can be defined as artificial neural network model of the system.

Moreover, fuzzy logic was first introduced by Lotfi Zadeh in 1965. Zadeh and his coworkers continued to develop fuzzy logic and implement at the time. In the beginning the idea of fuzzy sets and fuzzy logic were not accepted in academic areas, because some of the basis mathematics of it had not yet been explored. Applications of fuzzy logic develops slowly because of this, except in the east.

The idea of fuzzy logic is more closer to the culture and the history of the Eastern people than Western people. Looking at the development of fuzzy logic, it can easily be seen that the rate of development was more faster in the East in the compare of the West. For example in Japan fuzzy logic was fully accepted and applied in products simply because fuzzy logic worked, regardless of mathematicians agree or not. In the early 80s, the success of many of the products based on fuzzy logic in Japan gave a lead to an appearance of fuzzy logic in the U.S in the late 80s. Since that time, in the area of fuzzy logic, U.S. has been trying to catch up with this.

Zadeh introduced his idea with a paper “Fuzzy Sets”. A fuzzy set is a set without a crisp boundary. The transition from “belonging to a set” to “not belonging to a set” is gradual and smooth and this is characterized by membership functions. Determination of membership functions is subjective. Fuzzy sets have nothing to do with randomness. Subjectivity and randomness are two important differences between fuzzy sets and probability theory.

In using fuzzy logic there is a need to a fuzzy inference system which uses fuzzy set theory to map inputs to outputs. Mainly there are three types of fuzzy inference system; Mamdani type of fuzzy inference system, Sugeno type of fuzzy inference system and finally singleton type of fuzzy inference system. In Mamdani type of fuzzy inference system, it is necessary to determine a set of fuzzy rules. Fuzzifying the inputs using the input membership functions is the next step of this process. Then it is combined the fuzzified inputs according to the fuzzy rules to establish a rule strength. Continuely, it is suggested to find the consequence of the rule by combining the rule strength and the output membership function. Finally, combining the

consequences to get an output distribution, and defuzzifying the output distribution (this step is only if a crisp output (class) is needed).

In Takagi – Sugeno type of fuzzy inference system, the first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

Artificial neural network modeling is one of the most important modeling ways in academic areas. It has basically two steps; mathematical structure of neuron and activation part. In activation part, learning is quite important to have a descent system model. Learning is a process by which the free parameters of neural networks (generally the weights and biases) are adopted through a process of stimulation by the environment in which the network is embedded. The sequence of events can be written as follows; neural network is stimulated by the environment, neural network undergoes changes in its free parameters as a result of this stimulation and neural network responds in a new way to the environment because of the changes that have occurred in its' internal structure.

Learning has two main parts; learning with a teacher and learning without a teacher. Learning without a teacher also divided into two different subclasses as reinforcement learning and unsupervised learning. In learning with a teacher, teacher has the knowledge of the environment while the neural network does not have it. Teacher and the neural network are both exposed to training vector drawn from environment. Teacher provides the neural network with a desired response. This is the optimum action to be performed by neural network. In learning without a teacher in reinforcement learning the main idea is sourced by animal learning. The basic idea is awarding the learner from correct actions and punishing wrong actions. Reinforcement learning is a process of trial and error, combined with learning. In unsupervised learning neural network develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically. Moreover, in unsupervised learning neural network tries to capture statistical proper of the input. Provision is made for a task independent measure of the quality of representation that the network is required to learn and the free parameters of the neural network are optimized with respect to that measure.

One of the leading toolbox in fuzzy modeling that has been accepted by many scientists who work in this area is ANFIS (Artificial Neuro – Fuzzy Inference System) because of its user interface and flexible which is available in MATLAB software. ANFIS combines the fuzzy and artificial neural network ideas that make it popular in academic area that much.

Similar to many popular softwares, ANFIS has its own weakness. As instance it only can have a single output, obtained using weighted average defuzzification. Also all the output membership functions must be the same type and either be linear or constant.

In addition, ANFIS has no rule sharing which means different rules cannot share the same output membership function, namely the number of output membership functions must be equal to the number of rules. It also has the unity weight for each rule, which means each rule should have same weight. This situation can reduce the success of the modeling systems.

Reservoir computing has emerged in the last years as an alternative to gradient descent methods for training recurrent neural networks. Echo State Network is one of the key methods for reservoir computing. While being practical, conceptually simple, and easy to implement, ESNs require achieve the hailed good performance in many tasks.

Echo state network was introduced by Prof. Dr. Herbert Jaeger, a professor of Computational Science in Jacobs University, as an academic report in 2001. Corrected and full version of this report was published in 2010. This report introduces a constructive learning algorithm for recurrent neural networks, which modifies only the weights to output units in order to achieve the learning task and that is the main idea of echo state network. He also put the free toolbox of echo state network on internet.

The perspective taken is mainly that of mathematics and engineering, where a recurrent network is seen as a computational device for realizing a dynamical system. The RNNs considered here are mostly discrete-time, sigmoid-unit networks. The basic idea of echo states has been independently investigated in a complementary fashion. In his work, the emphasis is on biological modeling and continuous-time (spiking) networks.

Because it is easy to implement and use ESN has been very popular in academic area. There are many publications including echo state network so far. In general, recurrent neural networks are potentially powerful approximators of dynamics. There are many ways to make this statement more precise. For instance, RNNs can be casted as representations of the vector field of a differential system or they can be trained to embody some desired attractor dynamics.

In ESN there is a term for network which is called “reservoir”. Reservoir should be rich and should have a random structure. One simple method to prepare such a rich “reservoir” echo state network is to supply a network which is sparsely and randomly connected. Sparse connectivity provides for a relative decoupling of subnetworks, which encourages the development of individual dynamics.

In this work, three different systems, single input – single output system (SISO), Rossler time series and multiple input – single output system (MISO) have been modeled by ANFIS Toolbox in MATLAB software and ESN Toolbox. In final step of this study is a real nonlinear system “Active Suspension Plant” and all results have been compared with each other.

In Active Suspension Plant, system consists of two masses, each supported by a spring and damper. The sprung mass represents the mass of the vehicle body while the unsprung mass represents the tire in the quarter – car model. This system is fourth order because there are four independent storage elements, the two masses and the two springs. The spring and the damper support the body weight over the tire. The spring and the damper model the stiffness of the tire in contact with the road. To derive the dynamical model of this plant free body diagram method has been used and derived the general dynamical equations.

The purpose of this study to desing an ESN model by using the input - output data collected from any system. For this purpose, system models are developed with echo state network from the input – output data of four different systems and then the performance of the models are compared with ANFIS, the most popular fuzzy modeling tool.





## 1. GİRİŞ

Tekrarlı sinir ağlarını (Recurrent Neural Networks) eğitmek doğaları gereği zor bir işlemdir ve önemli bir problemdir. Akademik dünyada RNN eğitimi genellikle kaçınılan bir konu olsa da, büyük dinamik hafızaları ve oldukça uyumlu hesaplama yetenekleri sayesinde bu alanda kapsamlı ve güçlü bir araç oldukları söylenebilir. Makine öğreniminde doğal zekâ ve insan beynine yakın sonuçlar almaları nedeniyle kullanım alanları geniştir. Son yıllarda tekrarlı yapay sinir ağları konusunda yeni bir anlayış, eğitim ve kullanım açısı geliştirilmiş ve “Echo State Network” olarak sunulmuştur. Bu anlayışta tekrarlı bir yapay sinir ağında, tüm ağırlıkları değiştirmeye gerek duymadan uygun sonuçlar elde edilebileceği savunulmaktadır.

ESN yapısı kullanılarak optimizasyon, bulanık mantık, uyarlamalı kontrol, model tahmin yöntemleri, optimal kontrol gibi pek çok kontrol alanında çalışmaları bulunduğu gibi; uçuş sistemleri, biyolojik sistemler, elektrik ve enerji sistemleri, robotik sistemler gibi geniş yelpazede çalışmalar mevcuttur. Literatürde bu şekilde ESN kaynaklı pekçok çalışma ile karşılaşmak mümkündür. Seri kompanzasyonlu iletim hatlarında mesafe korunması işleminde ESN yapısından faydalanılmış, parametre ayarlanması için Büyük Patlama Büyük Çöküş Algoritması kullanılmıştır [2]. Uyarlamalı kontrol tasarımında ESN ve Genetik Algoritma tabanlı çalışmalar mevcuttur [26]. Benzer şekilde lineer olmayan dinamik sistemlerde model öngörülü kontrol tasarımı için ESN kaynaklı bir çalışma yapılmıştır [17]. Pnömatik yapay kas modelleme ve kontrolündeki karşılaşılan sürecin belirsizliği ve matematik modelin karmaşıklığı gibi problemleri çözmek için ESN tabanlı modelleme ve kontrol anlayışı ile çalışılmıştır [25]. Uçuş kontrol sistemlerinde uçuş kalitesini arttırmak amacıyla, ESN tabanlı bir öngörülü kontrol tasarlanmıştır ve başarılı sonuçlar elde edilmiştir [15].

Yapılan çalışmalar son yıllarda özellikle hız kazanmış, lineer olmayan sistemlerde ESN yapıdaki modellemenin başarısı gözlenmiştir. Kullanışımı ve başarısını ispatlamış, modelleme alanının güçlü araçlarından olan ANFIS yazılımı ile bu alanda yine başarılı sonuçlar veren ESN yapısını karşılaştırmak, süreçleri sınıflandırmak ve

kullanılacak aracı doğru şekilde seçmek için gerekli olacaktır. Tez kapsamında yapılan çalışmalarda farklı sistemler üzerinde ANFIS yazılımı ile ESN yapısının başarımlar oranları hakkında kıyaslama yapıp başarımlar analiz edilmiştir.

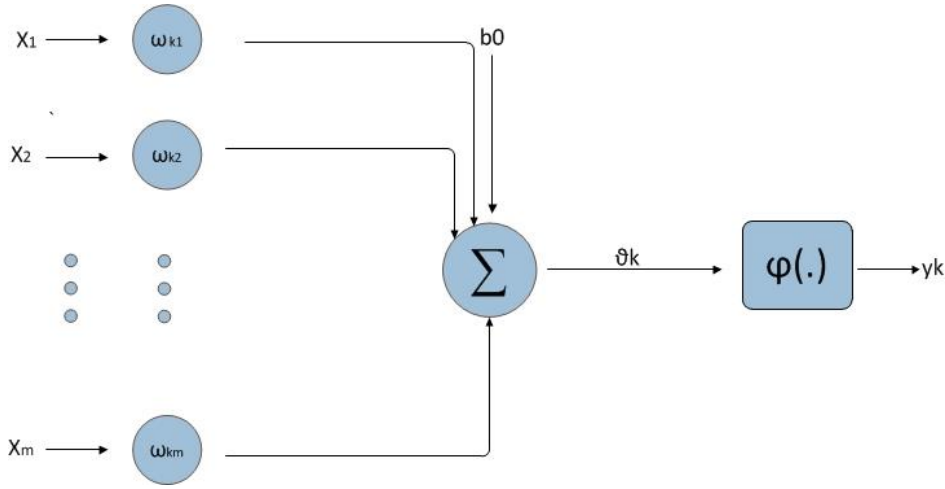
Tezde, giriş bölümünde genel tez kapsamında yapılan işin gerekliliği ve akademik dünyada kullanılabilirliği tartışılmıştır. İkinci bölümde yapay sinir ağları hakkında kısaca bilgi verilmiştir. Üçüncü bölümde tez çalışmasının başlangıcını oluşturan ESN yapısının matematiği anlatılmış ve bu modelleme ile ilgili ayrıntılı bilgi verilmiştir. Dördüncü bölümde bulanık mantık konusundan bahsedilmiş, konu ile ilgili temel bilgiler verilmiş, bulanık sistemler ve bulanık modelleme konusu kısaca anlatılmıştır. Beşinci bölümde, ESN yapısı ile akademik dünyada çok aktif olan ANFIS yazılımının üç farklı sistemden toplanan giriş – çıkış verileri kullanılarak model oluşturma performansları karşılaştırılmıştır. Altıncı bölümde, aktif süspansiyon deney seti üzerinde yapılan deneysel çalışmaların verileri ile elde edilen modelleme sonuçları, ESN yapısı ve ANFIS kullanılarak kıyaslanmıştır. Yedinci ve son bölümde ise beşinci ve altıncı bölümde yapılan işlerin değerlendirilmesi yapılmış, gelecekteki çalışmalar için hedefler belirtilmiştir.

## 2. YAPAY SİNİR AĞLARI

### 2.1 Giriş

Yapay sinir ağı (YSA), insan beyninin çalışma sistemi, bilgiyi işleme yöntemi esas alınarak geliştirilmiş bir teknolojidir. Bu teknoloji tek yönlü işaret kanalları ile birbirine bağlanan işlem elemanlarından oluşur. Çıkış işareti bir ya da birden fazla olabilir. İnsan beyninin bir modeli diyebileceğimiz yapay sinir ağlarının, çevre şartlarına göre davranışları değişebilir; verilen girişlere göre istenen çıkışları elde edebilmek için kendisini ayarlayabilir. Kompleks yapısı nedeniyle bugüne kadar gerçekleştirilen yapay sinir ağı, biyolojik sinir hücresinin işleyişini örnek olarak işlem yapan elemanlardan oluşmaktadır [6].

Gerçekte en karmaşık sinir ağı insan beynidir ve bilgi işleme merkezidir. Sinir sisteminin en basit yapısı olan sinir hücreleri (nöronlar), sinyal iletiminde görevli dendrit denilen kıvrımlı uzantılar, sinyallerin toplanma merkezi olan hücre gövdesi soma ve sinyali bir sonraki sinir hücresine aktarma görevinde olan aksonlardan oluşur. Beyinde her nöronun bir karşılığı vardır. Bir nöronun çıkışı ona bağlı olan bütün nöronlara iletilir fakat beyin işin yapılabilmesi için hangi nöron harekete geçirilecekse, sadece ona komut gönderir. Nöronlar arasındaki bağlantılar hücre gövdesinde veya "sinaps" adı verilen dendritlerdeki geçişlerde olur. Sinapslarla düzeltilen işaretleri taşıyan aksonlar ve dendritlerle içiçe geçmiş nöronlar bir sinir ağı oluştururlar [4].



**Şekil 2.1 :** Yapay sinir ağı yapısı.

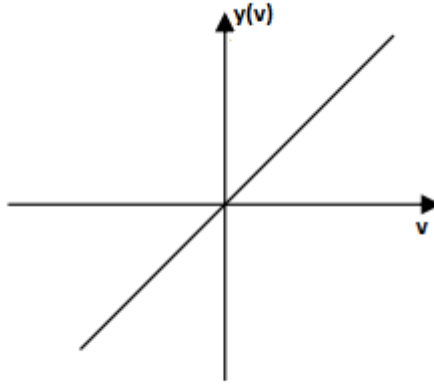
Matematiksel olarak Şekil 2.1'deki nöron yapısı, ifade (2.1)'de verilen denklemler ile tanımlanır.

$$\begin{aligned}
 u_k &= \sum_{j=1}^m w_{kj} x_j \\
 y_k &= \varphi(u_k + b_k) \\
 v_k &= u_k + b_k
 \end{aligned}
 \tag{2.1}$$

İfade (2.1)'de  $x_j$  olarak belirtilen değişkenler, yapay sinir ağının girişleri olup,  $w_{kj}$  olarak belirtilen katsayılar ise “ağırlıklandırma” olarak adlandırılmaktadır. Yine aynı ifadede verilen  $b_k$ , yapay sinir ağı için bias girişi olup, pozitif ya da negatif olması durumuna göre, aktivasyon fonksiyonunun girişi olan  $v_k$  değeri üzerinde arttırıcı ya da azaltıcı etkisi bulunmaktadır.

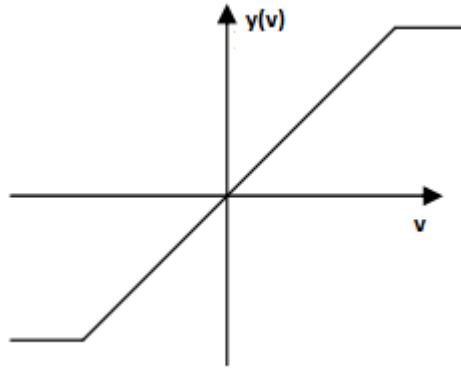
## 2.2 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları  $\varphi(v)$  ifadesi tarafından gösterilmektedir ve yerel ifade  $v$  için nöronun çıkışını sonuç olarak vermektedir. Temel halde türleri sıralanırsa:



**Şekil 2.2** : Lineer fonksiyon

$$y^{linear} = v \quad (2.2)$$



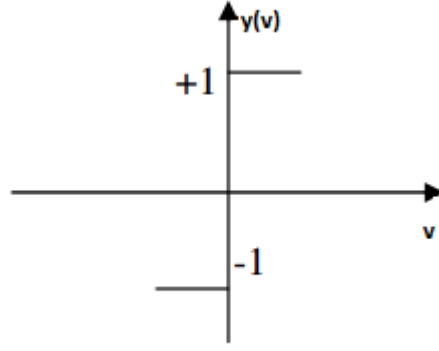
**Şekil 2.3** : Rampa eşik fonksiyon

Lineer fonksiyon,  $[\tau, -\tau]$  değerleri arasında kısıtlandığında Şekil 2.3'te verilen rampa eşik fonksiyonu olur ve denklemini ifade (2.3)'te verildiği gibidir.

$$y^{ramp} = v : |v| < \tau \quad (2.3)$$

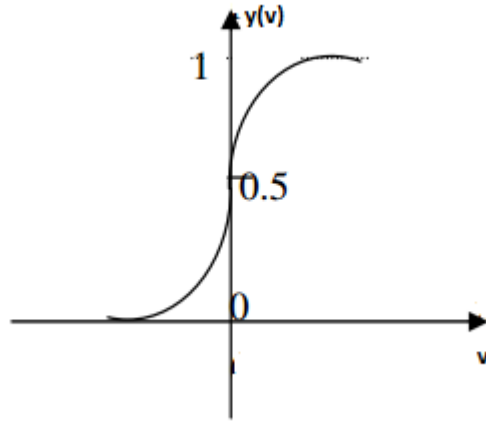
$$+\tau : v \geq \tau$$

$$-\tau : v \leq -\tau$$



**Şekil 2.4 : İşaret fonksiyonu**

$$y^{sign} = \begin{cases} +1: v \geq 0 \\ -1: v < 0 \end{cases} \quad (2.4)$$



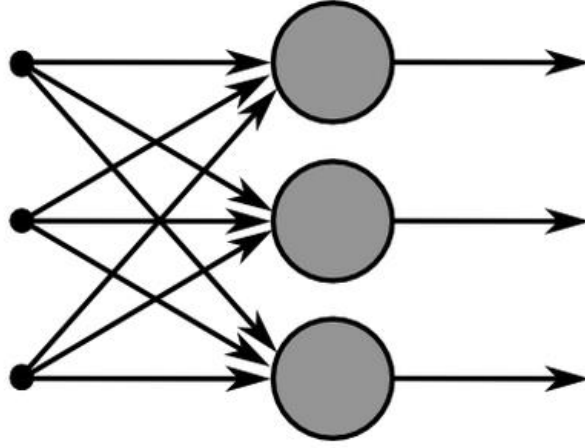
**Şekil 2.5 : Sigmoid fonksiyonu**

$$y^{sigmoid} = \frac{1}{1 + e^{-x}} \quad (2.5)$$

## 2.3 Ağ Yapıları

### 2.3.1 Tek katmanlı ileri beslemeli ağlar

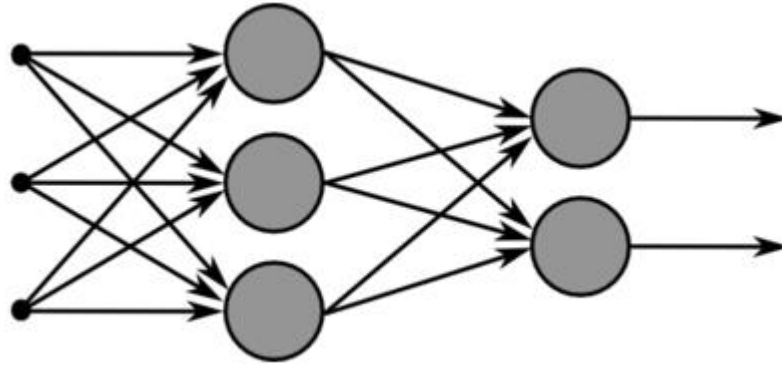
Katmanlı yapıdaki yapay sinir ağlarında nöronlar, katmanların biçimine göre düzenlenirler. Geri besleme yapısı içermeyen bu ağların, kaynak düğümlerinden oluşan bir giriş katmanı vardır ve nöronların çıkış katmanına etki eder, ancak tersi mümkün değildir [22].



Şekil 2.6 : Tek katmanlı ileri beslemeli ağ yapısı.

### 2.3.2 Çok katmanlı ileri beslemeli ağlar

Çok katmanlı ileri beslemeli ağ yapısı, her katmanın çıkışı bir sonraki katmanın girişi olacak şekilde düzenlenir ve ağın isimlendirmesi sırasıyla kaynak düğümleri sayısı, gizli nöronlar ve çıkış nöronları olarak yapılır. Şekil 2.7’de verilen ağ yapısı 3-3-2 ağ olarak isimlendirilir [22].

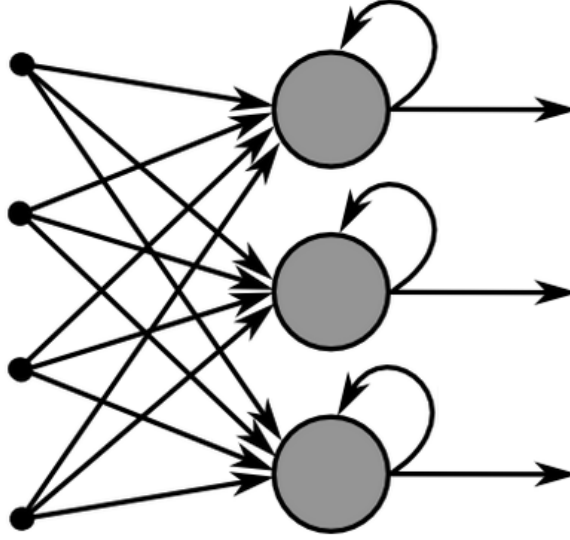


Şekil 2.7 : Çok katmanlı ileri beslemeli ağ yapısı.

### 2.3.3 Tekrarlı yapay sinir ağları

Bu ağ yapısı, ileri beslemeli yapay sinir ağı yapılarından en az bir geri besleme döngüsü içermesi ile ayrılır. Geri besleme döngüsü ağın performansı ve öğrenme kapasitesi üzerinde oldukça etkilidir. Ayrıca, geri besleme yolu üzerinde bulunan

birim geciktirme elemanları, ağına yapısına lineer olmayan bir karakteristik katmaktadır[6]. Şekil 2.8’de tekrarlı yapay sinir ağı yapısı için verilen örnek görülmektedir.



Şekil 2.8 : Tekrarlı yapay sinir ağı yapısı.

## 2.4 Öğrenme Algoritmaları

Yapay sinir ağlarının en önemli özelliklerinden sistemin çevreden öğrenmesi ve performansını öğrenme sırasında artırmasıdır. Bir yapay sinir ağında öğrenme sırasında, nöronlar arasında olan ağırlıklar ve bias değeri güncellenir. Her iterasyon sonunda sistem çevreden daha çok bilgi alır ve sistem performansını iyileştirir. Yapay sinir ağlarında, öğrenme sırasında çeşitli algoritmalar kullanılmaktadır, her algoritmanın kendine göre avantaj ve dezavantajı vardır.

Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Birtakım metot ve kurallar, gözlem ve eğitim ile ağdaki ağırlıkların değiştirilmesi sağlanmalıdır. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

### 2.4.1 Danışmanlı öğrenme

Bu tip öğrenmede, YSA’ya örnek olarak bir doğru çıkış verilir. Bu öğrenmede ağın ürettiği çıktılar ile hedef çıktılar arasındaki fark hata olarak ele alınır ve bu hata



minimize edilmeye çalışılır. Bunun için de bağlantıların ağırlıkları en uygun çıkışı verecek şekilde değiştirilir. Bu sebeple danışmanlı öğrenme algoritmasının bir “öğretmene” veya “danışmana” ihtiyacı vardır. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri besleme (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir [24],[20].

#### **2.4.2 Danışmansız öğrenme**

Bu tür öğrenmede ağa sadece girişler verilir. Ağın ulaşması gereken hedef çıktılar verilmez. Girişe verilen örnekten elde edilen çıkış bilgisine göre ağ sınıflandırma kurallarını kendi kendine geliştirir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler (patterns) oluşturmak üzere ayarlar. Grossberg tarafından geliştirilen ART öğrenme (Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir [5], [12].

#### **2.4.3 Takviyeli öğrenme**

Takviyeli öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz. Takviyeli öğrenme (reinforcement training) yöntemi öğreticili öğrenme yöntemine benzemekle birlikte, ağa hedef çıktılar yerine, ağın çıktılarının ne ölçüde doğru olduğunu belirten bir skor veya derece bildirilir. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski'nin geliştirdiği Boltzmann kuralı veya genetik algoritma takviyeli öğrenmeye örnek olarak verilebilirler[7].

### **2.5 Modeller**

#### **2.5.1 Hebb modeli**

Yapay sinir ağı modellerinin ilk bilinen tipidir. Öğrenme metodu olarak Hebb kuralını kullandığı için bu adı almıştır. Hebb'in koymuş olduğu kural çok basittir: Eğer iki hücre aynı aktivasyonu gösteriyorsa arasındaki bağ kuvvetlenmesi gerekir. Eğer ikisinin aktivasyonu farklı ise aralarındaki bağ değişmez ya da azalır. Yapısı itibariyle ileri beslemeli olan bu model, tek katmanlı bir yapı ile oluşturulmuştur. Bu yapı giriş hücreleri ve onların bağlı olduğu çıkış hücresinden ibarettir. Giriş

hücrelerinin birbirileri arasında bağlantıları söz konusu değildir [6]. Öğrenme süreci daha önce de belirtildiği gibi hücreler arasındaki ilişkinin kuvvetlendirilip, azaltılmasıyla sağlanmaktadır;  $w_{kj}$  ağırlıkları bağlı olduğu iki hücrede aynı aktiflikte ise arttırılır, farklı ise azaltılır ya da giriş örüntü tipine göre değiştirilmez. Burada giriş örüntüsünün olarak kullanılan veri tipi de önemlidir. Eğer giriş örüntüsü olarak ikili (binary, 0 ve 1) tipi kullanılıyorsa, o zaman  $w_{kj}$  ağırlıklandırmaları farklı aktiflikteyken değişime (azaltılmaya) uğratılmazlar. Ancak kullanılan örüntü iki kutuplu (bipolar) ise o zaman  $w_{kj}$  değerleri farklı aktifliklerde azaltılırlar. Buna göre öğrenme formülü ifade (2.8)'de verilmiştir.

Bilinen ilk model olması nedeniyle bu alandaki çalışmaların öncüsü olarak sayılan bu model basitliği ve kullanım kolaylığı nedeniyle de akademik alanda sarsılmayacak bir yer edinmiştir.

$$w_{kj}(\text{yeni}) = w_{kj}(\text{eski}) + x_j y_k \quad (2.8)$$

### 2.5.2 Perceptron modeli

Hebb modelinden sonra ortaya konan perceptron modeli, bugünkü yapay sinir ağları için önemli bir temel oluşturmaktadır. Çünkü Hebb'in öğrenme kavramı yerine şimdi aktivasyon fonksiyonundan geçirilme kavramı gelmektedir. Bu model yardımıyla, öğrenmede aktivasyonun rolü kabul edilmektedir.

Perceptron modelinde en önemli faktör aktivasyon değeridir. Saptanacak olan aktivasyon değeri problemin karakteristiğine göre belirlenebilir. Aktivasyon değeri yardımıyla perceptron modeli başarılı bir sınıflandırma yapabilmektedir. Hebb modelinde yer almayan bir kavram olan "kararsız (geçiş) bölge" diye adlandırılan çözüm bölgesini de önemseyen perceptron, Hebb yapısına göre sınıflandırmayı daha dikkatli yerine getirmektedir. Bilindiği gibi Hebb yapısı çözüm uzayını iki parçaya ayırmakta ve bu ayrımı sağlayan doğrunun (ya da düzlemin) her iki yanına ayrı bir aktivasyon yapmaktadır. Oysaki perceptron yine çözüm uzayını iki ayrı parçaya

ayırmakla beraber, her iki kısma da ait özelliklerin karma şeklini üzerinde taşıyan nötr sahayı da öğrenebilme yeteneğine sahiptir [16].

### 2.5.3 Geriye yayılım modeli

Geriye yayılım modeli, yapay sinir ağları içerisinde hemen hemen her probleme rahatlıkla uyarlanabilir bir yapıyı oluşturmaktadır. Geriye yayılım geribeslemeli bir öğrenme mekanizması kullanır. Burada yapay sinir ağının yapısı ileri yönde olmasına karşın, hatanın geriye doğru ket vurmasından kaynaklanan bir öğrenme olduğu için geribesleme söz konusu olmaktadır ve bu yapay sinir ağı modeli danışmanlı öğrenme stratejisini (supervised learning) kullanmaktadır.

Geriye yayılımmodeli sürekli (continuous) giriş tipini kullanır. Aktivasyon için ise türevi alınabilecek bir fonksiyonu işleme sokmaktadır. Genellikle sigmoid fonksiyonunu kullanan geriye yayılım öğrenme fonksiyonu olarak da Delta kuralını kullanır.

$$w_{kj}(yeni) = w_{kj}(eski) + \mu[t - f(y_{kj})]f'(y_{kj}) \quad (2.9)$$

İfade (2.9)'a göre türevi alınabilinen bir fonksiyon kullanılması, geriye yayılım tipi bir yapay sinir ağı modeli için önem taşımaktadır. Çünkü türev bir eğri üzerinde değişim olarak tarif edilmektedir. Bir başka deyişle; hatanın minimize edilmesi demek, hatanın türevinin sıfır olması anlamına gelmektedir. Bu yüzden geriye yayılım modeli, hatanın türevini  $w_{kj}$  ağırlıkları üzerinde öğrenmekte bu sayede hatalar her bir döngü sonunda sıfıra doğru yaklaşmaktadır. Geriye yayılım bu sebepten dolayı başarılı bir kullanım sahasına sahiptir [4].

$$w_{kj}(yeni) = w_{kj}(eski) + x_j y_k \quad (2.8)$$



### 3. ECHO STATE NETWORK

Echo state ağ yapısı, tekrarlı sinir ağları yapısında olup, ara katmandaki nöronların tüm giriş nöronlarına bağlantılı olmadığı bir yapı oluşturmaktadır ve özel giriş modellerini eşleme ve yeniden oluşturma için epey kullanışlı bir yapıdır. Çünkü öğrenme süresince, yalnızca ara katmandan çıkış katmanına doğru olan ağırlıklar değişebilir. Bu yüzden eğitim süresi gayet hızlıdır ve hesaplama bağlamında diğer seyrek bağlantılı ağ yapılarına göre daha etkilidir.

Tekrarlı yapay sinir ağlarına farklı bir bakış getiren echo state network yapısında, karmaşık dinamikleri temsil eden “rezervuar” yapısı bulunmaktadır. Ana fikir, herhangi bir kurala bağlı olmayan (rastgele), büyük ve sabit bir tekrarlı yapay sinir ağı oluşturmak ve rezervuar içindeki her nöronu giriş sinyali ile aktif hale getirmek, çıkış aşamasında ise her nöronun gelen sinyallerin lineer bir birleşimi ile istenen çıkış değerine ulaşmaktır [8].

Prof. Dr. Herbert Jaeger tarafından 2001 yılında yalnızca bir teknik rapor olarak ortaya atılan bu fikir, yıllar içinde Jaeger’in önderliğindeki MINDS (Making Intelligent Dynamical Systems) araştırma grubu ile ilerletilmiş ve en son 2010 yılında düzeltilmiş versiyonu ile sunulmuştur ve çalışmalar bu düzenleme ile epey hızlanmıştır [11]. Nitekim son yıllarda ESN içerikli yayınların fazlalığı açıkça görülmektedir. Prof. Dr. Herbert Jaeger hesaplamalı bilimler bölümünde, ekibi ile birlikte makine öğrenimi, lineer olmayan sistemlerin matematiksel ve algoritmik modellenmesi gibi konular üzerinde araştırmalarına devam etmektedir.

#### 3.1 Yapısal Özellikler

İlk adım olarak herhangi bir nöron modeli kullanarak, rastgele bir tekrarlı ağ yapısı oluşturulur. Burada rezervuar boyutu, yapılan işe göre değişmektedir. Giriş birimi, rastgele bir şekilde rezervuar elemanlarına bağlanır. Son olarak, çıkış birimi oluşturulur. Bu kısımda eğer yapılan iş çıkış geri besleme yapısı gerektiriyorsa, çıkıştan rezervuara doğru olacak şekilde yine rastgele oluşturulmuş bir bağlantı

kurulur. Eğer yapılan iş çıkış geri besleme yapısı gerektirmiyorsa, ayrıca bir bağlantı yapısı eklemeye gerek yoktur [27].

Sonraki adımda ise, eğitim verileri ile rezervuar çalıştırılır. Benzer şekilde, yapılan iş çıkış geri besleme yapısı gerektirmiyorsa rezervuar, yalnızca giriş sinyali ile çalıştırılır. Eğer yapılan iş çıkış geri besleme yapısı gerektiriyorsa rezervuar, hem giriş sinyali hem de eğitim sonucu elde edilen sonuç ile çalışır.

Son adımda çıkış ağırlıkları, eğitim sonucu elde edilen çıkışlarının rezervuar durumları üzerindeki lineer regresyon ağırlıkları olarak hesaplanır. Bu ağırlıklar rezervuardan çıkışa doğru olan bağlantıların yapılmasında kullanılır. Bu adımda eğitim tamamlanmış ve echo state network kullanım için hazır durumdadır.

$K$  giriş birimli,  $N$  iç katmanlı ve  $L$  çıkış birimli ayrık zamanlı bir ağ yapısı ele alındığında:

Giriş biriminin her örnekleme değeri  $n$  için etkinleştirme birimleri ifade (3.1)'de verildiği gibidir.

$$u(n) = u_1(n), \dots, u_k(n) \quad (3.1)$$

Benzer şekilde iç katman için ifade (3.2)'deki denklem yazılır.

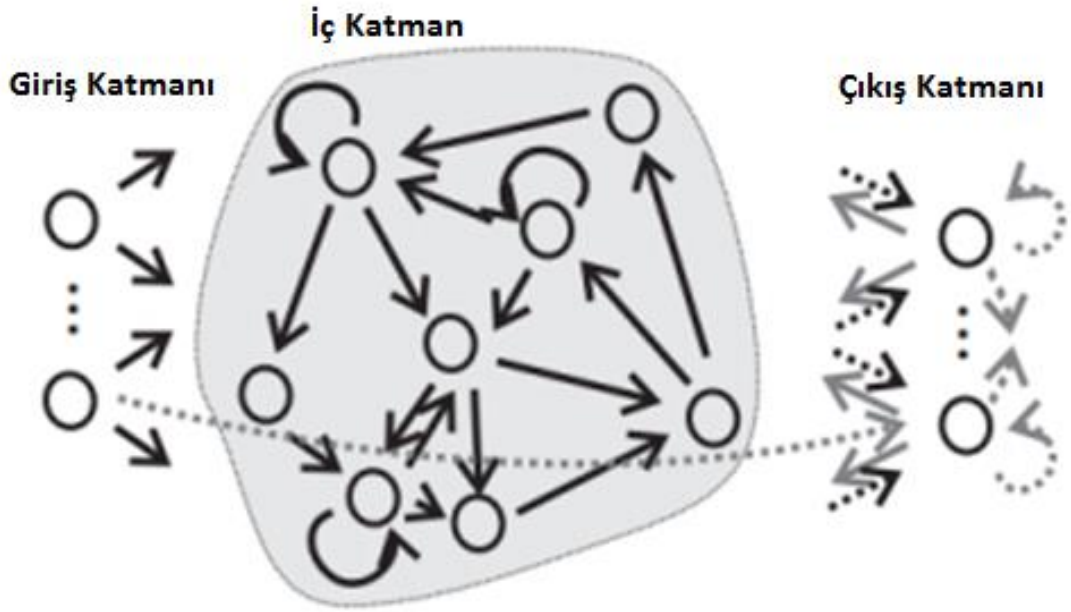
$$x(n) = x_1(n), \dots, x_N(n) \quad (3.2)$$

Son olarak da çıkış birimi için de ifade (3.3)'deki denklem yazılır.

$$y(n) = y_1(n), \dots, y_L(n) \quad (3.3)$$

Gerçek değerli bağlantı ağırlıkları, giriş ağırlıklandırmaları için  $N \times K$  boyutlu bir ağırlık matrisi olan  $W^{in} = (w_{ij}^{in})$  matrisinde, iç katmandaki ağırlıklandırmalar için  $N \times N$  boyutlu bir ağırlık matrisi olan  $W = (w_{ij})$  matrisinde, çıkış ağırlıklandırmaları için  $L \times (K + N + L)$  boyutlu bir ağırlık matrisi olan  $W^{out} = (w_{ij}^{out})$  matrisinde ve son olarak da geri besleme bağlantısındaki ağırlıklandırmalar için  $N \times L$  boyutlu bir matris olan  $W^{back} = (w_{ij}^{back})$  matrisinde saklanır. Burada dikkat edilmesi gereken hususlardan biri, giriş biriminden direk olarak çıkış ünitesine olan bağlantılara ve çıkış biriminin kendi içinde olan bağlantılarına izin verildiğidir. Zorunlu olmamakla

birlikte, iç katmanda tekrarlı bağlantı yapısı bulunabilir. Şekil 3.1’de temel echo state network yapıları görülmektedir.



**Şekil 3.1:** Temel echo state network yapısı.

İç katmandaki aktivasyonu sağlamak için ifade (3.4)’te verilen güncellenme kuralı uygulanır.

$$x(n+1) = f(W^{in}u(n+1) + Wx(n) + W^{back}y(n)) \quad (3.4)$$

Burada  $f = (f_1, \dots, f_N)$  ifadesi iç katmanın çıkış fonksiyonları olup, genellikle sigmoid fonksiyondur. Echo state network çıkışı ifade (3.5)’te verildiği gibi hesaplanır.

$$y(n+1) = f^{out}(W^{out}(u(n+1), x(n+1), y(n))) \quad (3.5)$$

Burada  $f^{out} = (f_1^{out}, \dots, f_L^{out})$  ifadesi çıkış biriminin çıkış fonksiyonları olup,  $(u(n+1), x(n+1), y(n))$  ifadesi ise; giriş biriminin, iç katmanın ve çıkışın bir önceki değerinin aktivasyon vektörlerini birbirine bağlayan ifadedir.

Giriş dizileri için ifade (3.6)’da verilen tanımlama ele alınmaktadır ve burada U uzayının kompakt uzay olması gereklidir.

$$(u(n))_{n \in J} \in U^J \quad (3.6)$$

Giriş dizilerinin soldan ve sağdan sonsuz olma durumu, sağdan sonsuz olma durumu, soldan sonsuz olma durumu ve belirli bir uzunlukta sonlu olma durumları için kısa ifadeler sırasıyla ifade (3.7)'de verilmiştir. Benzer isimlendirme çıkış ve iç katman için de kullanılmaktadır.

$$u_{-\infty}, u_{+\infty}, u_{-\infty}, u_{-h} \quad (3.7)$$

İç katman için için bir güncelleme operatörü tanımı ifade (3.8)'de verildiği gibidir.

$$x(n+h) = T(x(n), y(n), u_{-h}) \quad (3.8)$$

Geri besleme içermeyen yapılarda ifade (3.9)'da verilen tanım kullanılmaktadır.

$$x(n+h) = T(x(n), u_{-h}) \quad (3.9)$$

Bu aşamada yapılan analizlerde,

- Girişler kapalı bir uzaydan alınmaktadır.
- Network sınırlı bir aralık içinde tanımlanmaktadır.

Bu iki durum “standart sınır koşulları” olarak tanımlanmaktadır.

Standart sınır koşullarını sağlayan ve geri besleme yapısı içermeyen bir ağ yapısında eğer ağ durumları  $x(n)$ , sadece soldan sonsuz  $u_{-\infty}$  girişler ile belirlenebiliyorsa, bu yapıya echo state özelliği vardır denilir ve bu durum echo state fonksiyonlarının (ifade (3.10)) varlığını gerektirir. Başka bir deyişle bu özellik, sezgisel olarak, bir ağ çok uzun bir süredir aktif haldeyse, o anlık ağ durumunun benzersiz olarak giriş ve danışman sonucu elde edilen çıkış bilgileri ile elde edilebilir olduğunu söylemektedir.



$$\begin{aligned}
E &= (e_1, \dots, e_N) \\
e_i &: U^{-N} \rightarrow R \\
x(n) &= E(\dots, u(n-1), u(n))
\end{aligned} \tag{3.10}$$

İfade (3.11)'de verilen şartları sağlayan soldan sonsuz bir durum dizisi için bu hal, “giriş uyumlu” olarak adlandırılır. Giriş dizisi de yine soldan sonsuzdur.

$$\begin{aligned}
x_{-\infty} &= \dots, x(n-1), x(n) \in A^{-N} \\
u_{-\infty} &= \dots, u(n-1), u(n) \\
\forall i < n : x_{-\infty} &: T(x(i), u(i+1)) = x(i+1)
\end{aligned} \tag{3.11}$$

İfade (3.12)'de verilen şartları sağlayan sağdan ve soldan sonsuz bir durum dizisi için bu hal, “giriş uyumlu” olarak adlandırılır. Giriş dizisi sağdan ve soldan sonsuzdur.

$$\begin{aligned}
x_{-\infty} &= \dots, x(n-1), x(n) \in A^{-N} \\
u_{-\infty} &= \dots, u(n-1), u(n) \\
\forall i : T(x(i), u(i+1)) &= x(i+1)
\end{aligned} \tag{3.12}$$

İfade (3.13)'te verilen şartları sağlayan bir durum dizisi için bu hal, “son uyumlu” olarak adlandırılır. Giriş dizisi soldan sonsuzdur.

$$\begin{aligned}
u_{-\infty} &= \dots, u(n-1), u(n) \\
\dots, x(n-1), x(n) & \\
T(x(i), u(i+1)) &= x(i+1) \\
x &= x(n)
\end{aligned} \tag{3.13}$$

İfade (3.14)'deki şartları sağlayan bir durum dizisi için bu hal, “sınırlı giriş için son uyumlu” olarak adlandırılır. Giriş dizisi sonlu bir dizidir.

$$\begin{aligned}
& x \in A \\
& \bar{u}_h \\
& u \\
& x(n-h), \dots, x(n) \in A^{h+1} \\
& T(x(i), u(i+1)) = x(i+1) \\
& x = x(n)
\end{aligned} \tag{3.14}$$

Tüm sağdan sonsuz girişler için ifade (3.15)'te gösterilen tanımı sağlayan bir sıfır dizisi  $((\delta_h)_{h \geq 0})$  var ise, network için bu durum “durum kısıt özelliği” olarak adlandırılır. Burada  $d$ , tek boyutlu öklit uzaklığıdır.

$$\begin{aligned}
& (\delta_h)_{h \geq 0} \\
& x, x' \in A, h \geq 0 \\
& \bar{u}_h = u(n), \dots, u(n+h) \\
& d(T(x, \bar{u}_h), T(x', \bar{u}_h)) < \delta_h
\end{aligned} \tag{3.15}$$

Tüm soldan sonsuz girişler için ifade (3.16)'da gösterilen tanımı sağlayan bir sıfır dizisi var ise, network için bu durum “durum unutmama özelliği” olarak adlandırılır. Durum unutmama özelliği, sezgisel olarak, giriş katmandaki network durumlarının zamanla etkisini yitirdiğini söylemektedir.

$$\begin{aligned}
& (\delta_h)_{h \geq 0} \\
& x, x' \in A, h \geq 0 \\
& \bar{u}_h = u(n-h), \dots, u(n) \\
& d(T(x, \bar{u}_h), T(x', \bar{u}_h)) < \delta_h
\end{aligned} \tag{3.16}$$

Tüm soldan sonsuz girişler için ifade (3.17)'de gösterilen tanımı sağlayan bir sıfır dizisi var ise, network için bu durum “giriş unutmama özelliği” olarak adlandırılır.

$$\begin{aligned}
& (\delta_h)_{h \geq 0} \\
& u_h = u(n-h), \dots, u(n), h \geq 0 \\
& x \rightarrow w \quad u_h \\
& x' \rightarrow v \quad u_h \\
& d(x, x') < \delta_h
\end{aligned} \tag{3.17}$$

### 3.2 Hesaplama ve Eğitim

Genel olarak, tekrarlı yapay sinir ağları sistem dinamiklerini tahmin etme konusunda epey başarılıdır. Bu bağlamda giriş odaklı sistemler ele alındığı için, giriş-çıkış bilgileri verildiğinde bir sonraki çıkışı veren bir  $G$  fonksiyonu dinamik sistemi tanımlamak için kullanılır.

Echo state network yapısının giriş unutmama özelliği ile iç katmanların ilk geçici aktivasyonları ifade (3.18)'de verildiği gibi yazılabilir. Burada  $e_i$ ,  $i$  dereceden echo fonksiyonlarını göstermektedir.

$$x_i(n) \approx e_i(\dots, u(n), u(n+1)) \tag{3.18}$$

İfade (3.19) ve ifade (3.20)'de  $G$  fonksiyonu çıkış ilişkisi ve çıkışa göre öğrenme ilişkisi görülmektedir.

$$y(n+1) = G(\dots, u(n), u(n+1); \dots, y(n-1), y(n)) \tag{3.19}$$

$$y_{teach}(n+1) = G_{teach}(\dots, u(n), u(n+1)) \tag{3.20}$$

Buradaki önemli nokta ise,  $G_{teach}$  fonksiyonunu, “karesel hatanın ortalaması” yöntemi ile oluşturulan echo fonksiyonlarından elde etmektir. Çıkış geri besleme içermeyen network yapıları için tek kanal yapı versiyonu ifade (3.21)'de verildiği gibidir.

$$y(n) = f^{out} \left( \sum_{i=1, \dots, N} w_i^{out} x_i(n) \right) \tag{3.21}$$

Burada  $w_i^{out}$ ,  $i$  dereceden çıkış bağlantısının ağırlığı ifadesidir. Ters alınabilen bir  $f^{out}$  fonksiyonu için ifade (3.22) yazılabilir.

$$(f^{out})^{-1} y(n) = \sum_{i=1, \dots, N} w_i^{out} x_i(n) \quad (3.22)$$

Echo fonksiyonlarının eklenmesi ile denklemler ifade (3.23)'de verilen hali alır.

$$(f^{out})^{-1} y(n) = \sum_{i=1, \dots, N} w_i^{out} e_i(\dots, u(n-1), u(n)) \quad (3.23)$$

Hata, ortalama karesel hatada minimize olacak şekilde ağırlıklar belirlenirse sırasıyla ifade (3.24)'te verilen denklemler yazılabilir.

$$\begin{aligned} \varepsilon_{train}(n) &= (f^{out})^{-1} y_{teach}(n) - (f^{out})^{-1} y(n) \\ \varepsilon_{train}(n) &= (f^{out})^{-1} y_{teach}(n) - \sum_{i=1, \dots, N} w_i^{out} e_i(\dots, u(n-1), u(n)) \\ mse_{train} &= \frac{1}{n_{max} - n_{min}} \sum_{i=n_{min}, \dots, n_{max}} (\varepsilon_{train}(n))^2 \end{aligned} \quad (3.24)$$

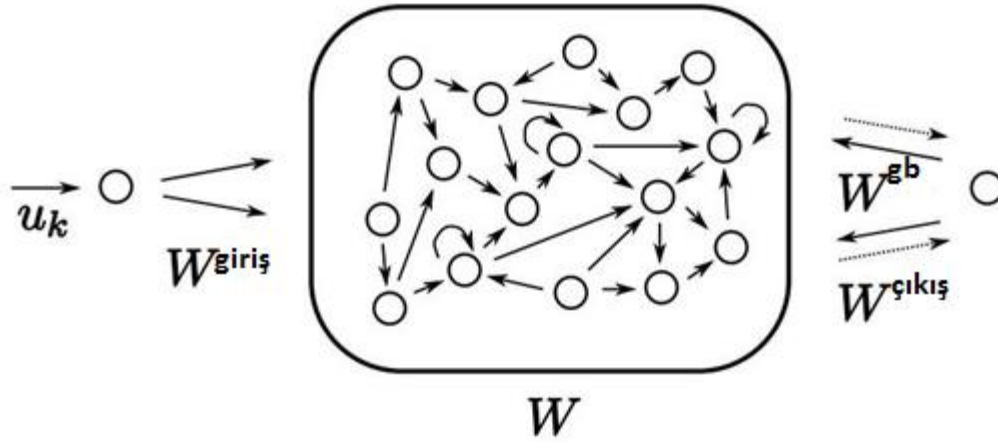
Burada  $n_{min}$ , eğitim dizisinin baştaki geçici durumun çıkarılması ile elde edilen noktayı temsil ederken,  $n_{max}$  ise son eğitim noktasıdır ve  $mse_{train}$  eğitim hatasıdır. Yapılan bu çalışmada daha titiz bir inceleme olması açısından “normalize edilmiş ortalama karesel hata” için (ifade (3.25)) optimizasyon yapılmıştır.

$$nrmse = \frac{\sqrt{mse}}{n_{max} - n_{min}} \quad (3.25)$$

Bu adımda biraz belirsiz ama daha sezgisel bir değişiklik yapılırsa ifade (3.26)'da gösterilen değişiklik yapılabilir. Bu değişiklik ile  $G'$  için yapılan ortalama karesel hata yaklaşımı, ağırlıkların ve echo fonksiyonların birleşimi haline gelir.

$$\begin{aligned} (f^{out})^{-1} G_{teach} &\rightarrow G' \\ G' &\approx \sum w_i e_i \end{aligned} \quad (3.26)$$

Bu işlemde ifade (3.26)'ya göre istenen durum ise,  $u(n)$  girişleri ile eğitilmiş, çıkışları  $(y(n))$ , danışman çıkışlarına  $(d(n))$  yakınsayan bir ESN  $(W^{giris}, W, W^{gb}, W^{cikis})$  yapısı oluşturmak. Burada  $W^{gb}$  geri besleme üzerindeki ağırlıklandırma için isimlendirilmiştir.



Şekil 3.2: Echo state network yapısı.

#### Adım 1:

Echo state özelliği içeren ve uyarıldığı zaman birbirinden çok farklı dinamikler sergileyen iç katmanlara sahip bir dinamik rezervuarın  $(W^{in}, W, W^{back})$  oluşturulması gerekir [9].

Bu adım pekçok sezgisel yöntem içermektedir, en çok kullanılan haliyle;

- Rastgele bir iç ağırlık matrisi  $(W_0)$  oluşturmak.
- $W_0$  matrisini normalize ederek bir  $W_1$  matrisi oluşturmak. Burada  $|\lambda_{max}|$ ,

$$W_0 \text{ 'ın spektral yarıçapıdır. } \left( W_1 = \frac{1}{|\lambda_{max}|} W_0 \right)$$

- $W = \alpha W_1, \alpha < 1$  şeklinde bir ölçeklendirme yaparak spektral yarıçapı belirlemek.

- Rastgele bir giriş ağırlık matrisi ( $W^{in}$ ) ve çıkış geri yayılım (backpropagation) ağırlık matrisi ( $W^{back}$ ) oluşturmak şeklinde özetlenebilir.

### Adım 2:

Ağın eğitim dinamiklerinden örnekler almak gereklidir. Bu adımda sezgisel yöntem bulunmamakta, sırası ile aşağıda verilen adımlar izlenmektedir.

- Sırasıyla ağ durumlarını başlatmak. Örneğin  $x(0) = 0$ .
- $x(n+1) = f(W^{in}u(n+1) + Wx(n) + W^{back}d(n))$  hesabı ile ağı eğitim verisi ile  $n=0, \dots, T$  kadar çalıştırmak.
- $n=0$  anında  $d(n)$  tanımlı olmadığından  $d(n) = 0$  kullanmak.
- Her ilk silerek geçme zamanından ( $T_0$ ) büyük ya da eşit zamanlar için ağ durumlarını  $x(n)$  durum toplama matrisinde ( $M$ ) toplamak. Tüm işlem sonunda bu matrisin ebatı  $(T - T_0 + 1) \times (K + N + L)$  olacaktır.
- Benzer şekilde her ilk silerek geçme zamanından ( $T_0$ ) büyük ya da eşit zamanlar için danışman çıkışlarını ( $sigm^{-1}d(n)$ ) danışman çıkış matrisinde ( $T$ ) toplamak. Tüm işlem sonunda bu matrisin ebatı  $(T - T_0 + 1) \times L$  olacaktır.

### Adım 3:

Çıkış ağırlıklarını hesaplamak gereklidir.

- Pseudo - ters işlemi uygulayarak çıkış ağırlık matrisini oluşturmak.  

$$\left( (W^{out})^T = M^{-1}T \right)$$
- $(W^{out})^T$  matrisini tekrar transpoze ederek  $W^{out}$  matrisini elde etmek.

**Adım 4:** Ağ  $(W^{in}, W, W^{back}, W^{out})$  artık işleme hazır haldedir ve istenen giriş serisi ile çalıştırılabilir.

$$\begin{aligned}x(n+1) &= f(W^{in}u(n+1) + Wx(n) + W^{back}y(n)) \\y(n+1) &= f^{out}(W^{out}(u(n+1), x(n+1), y(n)))\end{aligned}\tag{3.28}$$

Çalışmada önemli olan kavram rezervuarın oluşturulmasıdır ki bu noktada da ağırlıklandırmalar önem kazanır. Ağırlık matrisi  $W$  için spektral yarıçap  $|\lambda_{\max}| > 1$  olsun. Burada  $\lambda_{\max}$ ,  $W$  matrisinin en büyük mutlak değere sahip özdeğeridir. Böyle bir durumda network, asimptotik olarak kararsız bir sıfır durumuna sahip olacaktır ki bu da herhangi bir kabul edilebilir durum için, ağın echo state özelliğine sahip olmadığı anlamına gelir.

Rezervuar oluşturulurken, echo state özelliğine bağlı kalmak adına, spektral yarıçapın birden küçük olmasına dikkat edilmelidir. Genelleme yapmak gerekirse, hızlı ve kısa süreli hafızaya sahip işlemler için küçük değerlerde bir spektral yarıçap; yavaş ve uzun süreli hafızaya sahip işlemlerde ise büyük değerlerde bir spektral yarıçap kullanmak mantıklı olacaktır [13]. Yine de pekçok işlemde manuel olarak ayarlama yapmak durumu ile karşılaşılmaktadır. Optimal değer aralığı da çalışılan bölgeye yakın olacaktır. Örnek vermek gerekirse; 0.8 spektral yarıçapta başarılı sonuçlar alınan bir süreçte, 0.75 ve 0.85 değerli spektral yarıçaplarda da başarılı sonuçlar alınabildiği görülür. Spektral yarıçap bire ne kadar yakınsa optimal çözüm için çalışılan aralık da o kadar dar olacaktır.

Zengin bir rezervuar içeren bir ESN yapısı oluşturmanın bir yolu da, ağ yapısına seyrek ve rastgele bağlantı yapısı sağlamaktır. Seyrek bağlantı, alt ağların görece daha iyi ayrışmasını, böylece alt ağlardaki her birimdeki dinamiklerin daha iyi gelişmesini sağlar.

Spektral yarıçap kadar uygun bir model boyutu bulmak da önemlidir. Genel olarak dinamik rezervuarlar, karmaşık sistemleri öğrenmede ya da yüksek derecede doğruluk ile öğrenmede çok başarılıdır. Yine de çok parametrelili bir modele uyumlu olmak gerektiğinde, eğitim verisinde başarılı sonuçlar alınsa da, test verisinde zayıf

kalınacaktır. Bu durumda test verisinde kabul edilebilir sonuçlar alıncaya kadar model boyutunu büyütmede fayda vardır [13].

Unutulmamalıdır ki, ESN yapısı bir siyah kutu modelleme örneğidir. Bunun anlamı, ağı eğitimi sırasında eğitim verisinin hiç etki etmediği test verisinden güzel sonuçlar bekleyemeyeceğinizdir. Bu durum bazen doğrusal olmayan sistemlerde epey sorun oluşturmaktadır. Örnek vermek gerekirse, en çok kullanılan eğitim verisi beyaz gürültü verisidir. Doğrusal sistemlerde beyaz gürültü ile çok başarılı sonuçlar alınırken, doğrusal olmayan sistemlerde beyaz gürültü nedeniyle çalışılan aralığın küçük bir kısmında sınırlı kalmak gibi sorunlar yaşanmaktadır. Modelleme sonrasında sistem başka veri ile çalıştırıldığında, modellendiğinden neredeyse tamamen farklı bir aralıkta çalışmak durumunda kalacak ve sonuçlar başarısız olacaktır. Sonuç olarak söylenebilir ki, eğitim verisini daha geniş tutarak, test verilerini ve eğitim verilerini benzer tarzda kullanmak faydalı olacaktır [10].

Yaklaşımın doğru çalışması için “DR yapısı echo state özelliği içermeli” ifadesini daha açıklamak gerekirse, sistem mühendisliği bakışı yardımcı olacaktır. Bilinmeyen bir sistemin dinamikleri ifade (3.29)’deki güncelleme denklemi ile yönetilebilir.

$$d(n) = e(u(n), u(n-1), \dots, d(n-1), d(n-2)) \quad (3.29)$$

Burada  $e$ , sistem çıkışlarının ve geçmiş girişlerin fonksiyonudur (muhtemelen karmaşık ve doğrusal olmayan bir fonksiyon).

Siyah kutu modelleme yönteminde asıl iş, bilinmeyen bir sistem için sistem fonksiyonu  $e$ ’ye başarılı bir yaklaşımda bulunmaktır. Yazımı kolaylaştırmak için ESN yapısı doğrusal çıkış birimli alınır ve eğitilmiş ağ çıkışı ağ durumlarının doğrusal bir birleşimi haline gelir.

İfade (3.30)’dan, istenen sistem fonksiyonu yaklaşımının, echo state fonksiyonlarının doğrusal bir birleşimi olarak nasıl gösterileceği daha net bir şekilde anlaşılmaktadır.



$$\begin{aligned}
& e(u(n), u(n-1), \dots, d(n-1), d(n-2)) \\
& d(n) \\
& \approx y(n) \tag{3.30} \\
& \sum w_i^{out} x_i(n) \\
& \sum w_i^{out} e_i(u(n), u(n-1), \dots, y(n-1), y(n-2))
\end{aligned}$$

Buradaki şeffaf sistem yaklaşımı yorumu direk olarak ağ durumlarının echo state olarak yorumlanmasına dayanır. Echo state özelliği olmadan orijinal sistem çıkışı ile ağ çıkışı arasındaki matematiksel ilişkiyi de eğitim algoritmasının çalışma şeklini de anlamak mümkün değildir.



## 4. BULANIK MANTIK

### 4.1 Giriş

Matematiksel olarak bulanıklık, çok-değerlilik demektir ve kökenleri, kuantum mekaniğindeki Heisenberg'in konum-momentum belirsizliği ilkesine dayanır. Üç değerli bulanıklık, “doğruluk”, “yanlışlık” ve “belirlenemezlik”e ya da “varlık”, “yokluk” ve “belirsizlik”e karşılık gelir. Çok değerli bulanıklık, belirlenemezlik ya da belirsizliğin derecelerine, olay ya da ilişkilerin kısmi oluşlarına karşılık gelir. Mantık, bir şeyin doğru ya da yanlış olup olmadığı ile ilgilenir, tanımlamayla uğraşmaz. Bulanık mantıkta ise, doğruluğun ve yanlışlığın dereceleri vardır. Bu nedenle de bulanık mantık, “iki değerli mantık” ya da “çok değerli mantık” olarak ortaya çıkmaktadır.

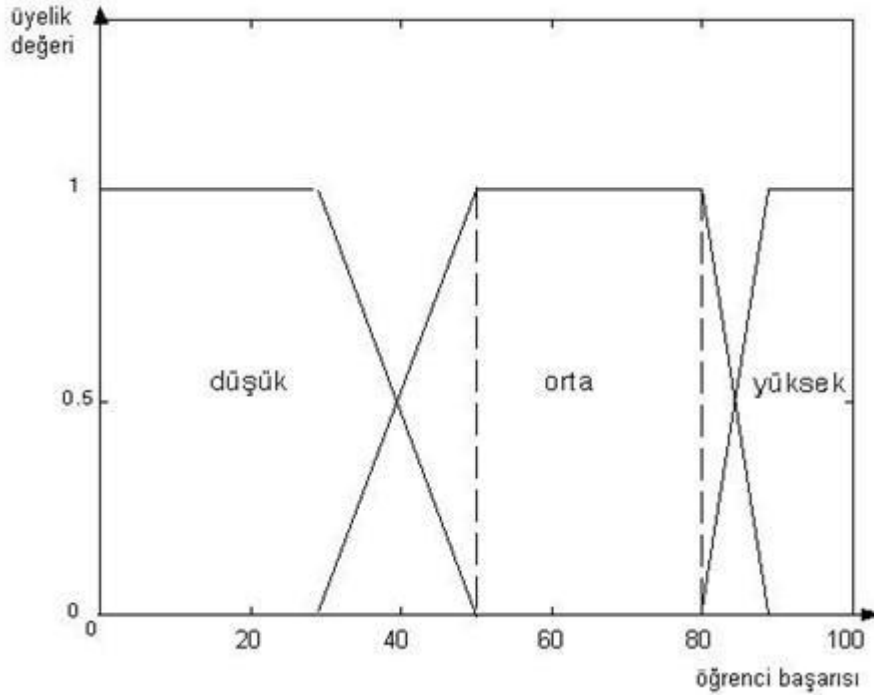
Belirlilik ortamının analizi matematik ve istatistiksel teknikler kullanılarak yapılabilir. Belirsiz sistemler ise bilgi yetersizliği sebebiyle yaklaşık değer teknikleri kullanılarak analiz edilir. Bulanık mantık belirsizlik analizinde etkin bir araçtır. İkili hesaplama yerine çok seviyeli hesaplama tekniğini kullanarak sistemin yaklaşık olarak analizini yapar. Temel yaklaşım, kesin yanlış ve kesin doğru ifadelerinin arasına sonsuz sayıda doğruluk değerini içeren fonksiyon yerleştirmektir. Bu fonksiyona “üyelik fonksiyonu” adı verilir[1].

### 4.2 Bulanık Kümeler ve Üyelik Fonksiyonları

Bulanık kümeler kuramının temel yapısında; belirsizlik ifade eden tanımlanması güç veya anlamı zor kavramlara üyelik derecesi atayarak onlara belirlilik getirmek vardır. Belirlilik getirme yaklaşımı, iki değerli kümeler kuramının, çok değerli kümeler kuramına dönüşümünden doğar.

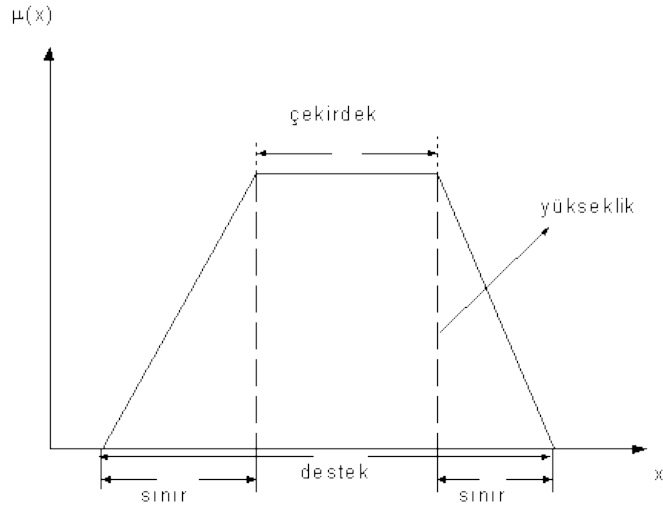
Bulanık küme, değişik üyelik derecesinde öğeleri olan bir topluluktur. Klasik küme kuramındaki kesin ayırım bulanık kümelerde yer almaz. Bulanık kümelerde eleman, bir bölümüyle (örneğin: 0.3) kümeye ait iken bir bölümüyle (örneğin: 0.7) de kümenin dışındadır. Bulanık kümelerde, klasik kümelerdeki üyeliği tanımlayan

karakteristik fonksiyon  $\mu_A : E \rightarrow \{0,1\}$ , yerini üyelik fonksiyonuna  $\mu_A : E \rightarrow [0,1]$  bırakır. Şekil 4.1’de yamuk biçimindeki üyelik fonksiyonları kullanılarak bir uygulama gerçekleştirilmiş ve öğrenci başarısı için örnek bulanık küme gösterimi verilmiştir. Orta-düşük ve orta-yüksek geçişlerinde paylaşım bölgesi söz konusu olup katı bir ayırım geçerli değildir.



**Şekil 4.1:** Öğrenci başarısı için bulanık küme gösterimi.

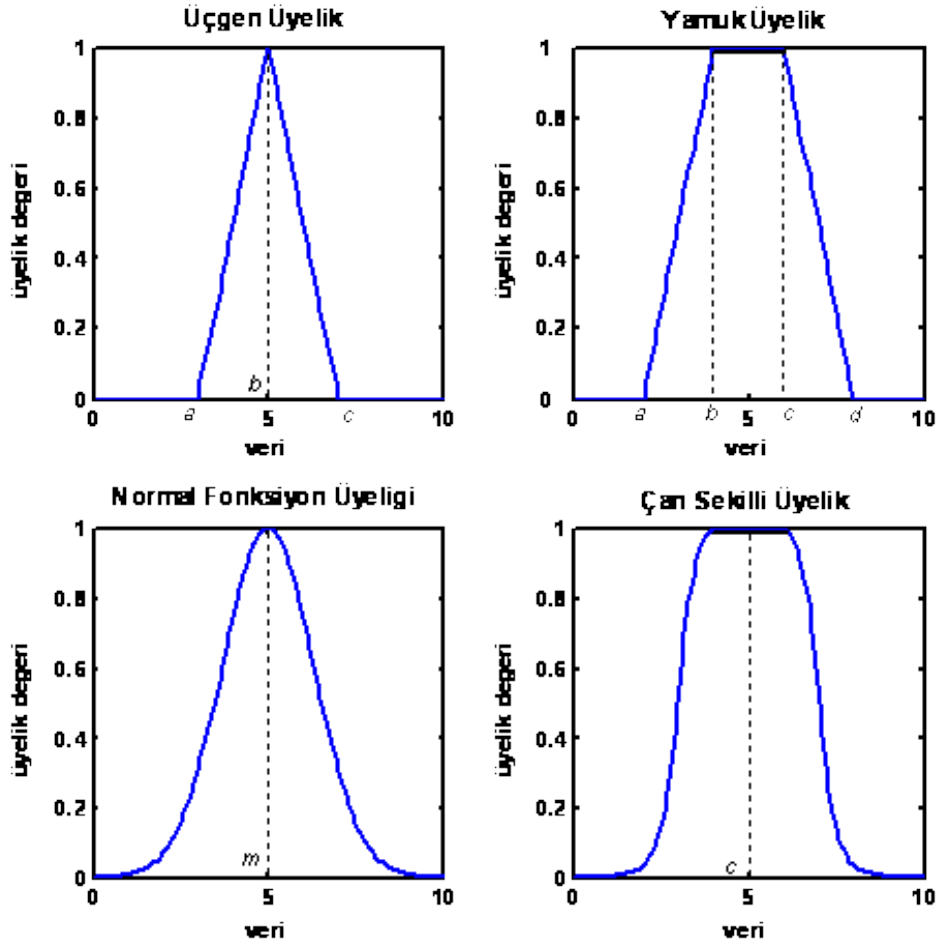
Bulanık kümeyi tanımlayan bilginin üyelik fonksiyonu üzerinde gösterilecek olması, fonksiyonunun önemini artırmaktadır. Bulanık küme işlemlerinde, problem yapısına uygun, bilgiyi temsil edecek fonksiyonun seçilmesi gerekmektedir. Üyelik fonksiyonu seçiminde “basitlik ve amaca uygunluk” özelliklerinin en önemli parametreler olduğu belirtilmekte ve sürekli ve kesikli fonksiyonlarda değişmeyen yapısal parçalar; çekirdek (core), destek (support), yükseklik (height) ve sınır (boundary) olarak tanımlanmaktadır[1].



**Şekil 4.2:** Üyelik fonksiyonu bileşenleri.

Çekirdek, üyelik fonksiyonun 1'e eşit olduğu bölgeyi (  $\mu_A(x) = 1$  ) ifade eder. Bu bölgede fonksiyon tam üyeliğe (full membership) sahiptir. Destek, fonksiyonun 0'dan büyük olan (  $\mu_A(x) > 0$  ) bölümüdür. Fonksiyonun iki yanında yer alan sınırlar ise, 0 ile 1 arasında üyelik değeri alan (  $0 < \mu_A(x) < 1$  ) tam üyeliğe ulaşamamış kısımları tanımlar.

Çok sayıda üyelik fonksiyonu bulunmasına rağmen uygulamada yaygın olarak dört tip fonksiyondan yararlanılmaktadır. Bunlar; üçgen (triangular), yamuk (trapezoidal), normal dağılım (Gaussian) ve çan şekilli (bell-shaped) fonksiyonlarıdır. Ayrıca, sigmoidal ve S-tipi üyelik fonksiyonları da kullanım amacına bağlı olarak sınırlı oranda kullanılabilir[1].



Şekil 4.3:Üyelik fonksiyonu türleri.

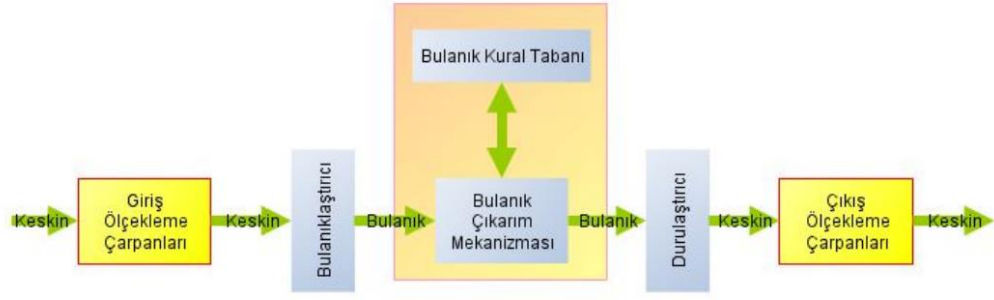
### 4.3 Bulanık Sistemler

Bulanık sistemlerin büyük bir kısmı “kural tabanlı bulanık sistemler” olarak isimlendirilir ve “eđer – o halde” kuralları ile tanımlanır.

Bir bulanık sistem en genel anlamda dört aşamadan oluşur :

- Bulandırma,
- Kural tabanı,
- Bulanık çıkarım,
- Durulama.

Bu aşamalardan önce ve sonra kullanılacak deđerleri kontrole uygun hale getirmek için ölçeklendirme işlemi uygulanır [1].



**Şekil 4.4:**Bulanık sistem yapısı.

Bir bulanık sistem yapısı tasarımında, bilinmesi gereken temel faktörler şunlardır:

- Gerçek giriş ve çıkışlar ve bunların evrensel kümeleri, yani her değişkenin alması muhtemel değerler aralıklarının belirlenmesi gerekir.
- Giriş ve çıkış değişkenlerinin ölçekleme faktörleri belirlenmelidir.
- Her giriş ve çıkış değişkenleri için bulanık değerlerin kurulmasında kullanılacak bulanık üyelik fonksiyonları belirlenmelidir.
- Bulanık kontrol kural tabanları oluşturulmalıdır.

#### 4.3.1 Bulanıklaştırma

Bulanıklaştırma, sisteme giren her bir giriş değerini bulanık küme değerine dönüştürmek amacıyla yapılır. Bunun için, giriş değişkenlerine dilsel değişkenler (büyük, çok büyük, küçük vb.) atanır. Bulanıklaştırma işleminde üyelik fonksiyonları kullanılır. Üyelik fonksiyonları giriş değerlerinin, fonksiyon tipine göre 0-1 arasında alacağı değeri gösterir. Üyelik fonksiyonlarının sayısı arttıkça işlemin hassasiyeti de artar.

Modellemenin doğru ve hassas yapılabilmesi için, giriş değerleri yeterli sayıda üyelik fonksiyonu ile belirlenmelidir. Üyelik fonksiyonu sayısı, sistemin karakteristiğine ve hassasiyetine göre değişiklik gösterir.

#### 4.3.2 Kural Tabanı

Bir bulanık kontrol sisteminin temel yapısı kural tabanıdır. Gözlemlere dayanılarak oluşturulan kural tabanı, bulanık “eğer – o halde” kural dizisinden oluşur.  $x_1$  ve  $x_2$  girişli y çıkışlı bir bulanık sistem için örnek bir kural, ifade (4.1)’de gösterilmektedir.

$$\text{Eğer } x_1 = A_1 \text{ ve } x_2 = A_2 \text{ ise o halde } y = B \quad (4.1)$$

İfade (4.1)'de  $x_1$  ve  $x_2$  terimleri skaler girişleri  $A_1$  ve  $A_2$  terimleri bulanık kümeler ile ifade edilen giriş dilsel değişkenlerini,  $B$ 'de yine bulanık kümeler ile ifade edilen çıkış dilsel değişkenini ifade etmektedir. İlk bölüm giriş değerlerinin niteliklerinin belirlendiği ve bağlaçlarla birleştirildiği bölümdür. Son bölüm ise çıkış değeri niteliğinin, giriş niteliklerine göre belirlendiği bölümdür.

### 4.3.3 Çıkarım mekanizması

Giriş işaretlerinin bulanık değerleri elde edildikten sonra, “eğer – o halde” ifadelerinden oluşan kural tabanı ile çıkış bulanık değerleri elde edilir. Bulanık giriş işaretlerinden bulanık çıkış işaretlerinin elde edilmesi bulanık çıkarım işlemidir. Öncelikle çıkarım işleminde seçilen methoda göre giriş işareti değerlendirilir. Daha sonra elde edilen değer kuralın “hareket etme derecesi” olarak tanımlanır. Aynı çıkış derecesine sahip iki ya da daha fazla kural varsa seçilen methoda göre belirlenen değer alınır ve çıkış işaretinin değeri belirlenir.

Bulanık çıkarım metodları çıkış değişkenine göre ayrılır.

- Mamdani (Max-Min) tipi bulanık çıkarım [14].
- Sugeno (Max-Product) tipi bulanık çıkarım [23].
- Tekli (Singleton) tipi bulanık çıkarım.

Uygulamada genellikle Mamdani ve Sugeno tipi bulanık çıkarım yöntemleri kullanılır. Mamdani tipinde kuralları formüle etmek daha kolaydır ve uzman kişilerce kolaylıkla anlaşılabilir. Sugeno tipi ise hesaplama olarak daha etkilidir ve matematik analizinde daha kullanışlıdır.

#### 4.3.3.1 Mamdani tipi yapı için çıkarım mekanizması

1975 yılında Mamdani ve Assilian tarafından bulunan bu yapının çıkarım mekanizmasında, kural tabanında bulunan kurallardaki giriş ve çıkışlar dilsel değişkenler ile ifade 4.2'deki gibi tanımlanır.

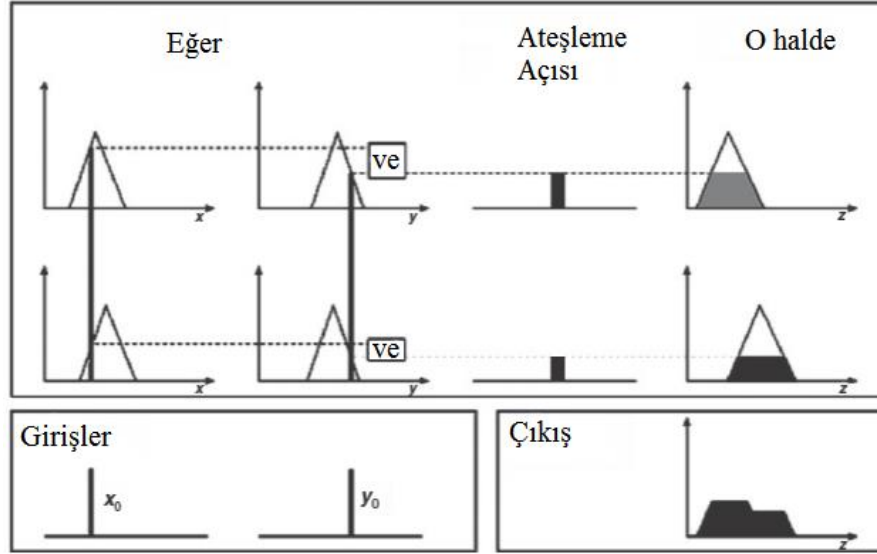
$$K_i: \text{Eğer } x = A_i \text{ ise o halde } y = B_i \quad (4.2)$$

İfade (4.2)'de,  $x$  skaler girişi,  $A_i$ 'ler giriş  $B_i$ 'ler de çıkış dilsel değişkenlerini ifade etmektedir.

Mamdani tipi yapı için çıkarım mekanizmasında, çıkış üyelik fonksiyonları bulanık kümelere dönüşmektedir. Çıkarım mekanizmasında ve operasyonu çalıştırdıktan sonra



her bir kural için durulaştırma gerektiren bulanık kümeler oluşur. Şekil 4.5'teki sistemde iki girişli ve iki kurallı Mamdani tipi yapı için çıkarım mekanizması gösterilmektedir.



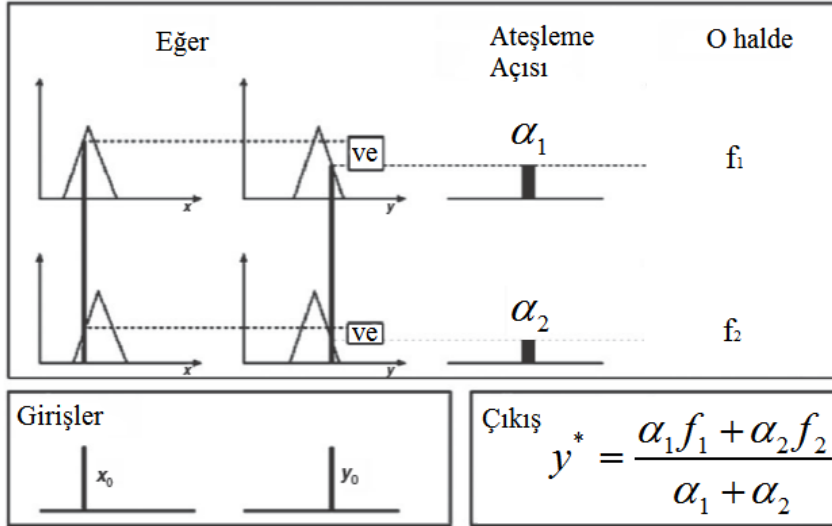
Şekil 4.5: İki girişli iki kurallı Mamdani tipi yapı için çıkarım mekanizması.

#### 4.3.3.2 Takagi – Sugeno tipi yapı için çıkarım mekanizması

1985 yılında Takagi ve Sugeno tarafından bulunan bu yapının kural tabanında bulunan kurallardaki girişleri dilsel değişkenler, çıkışları ise keskin fonksiyonlar ile ifade (4.3)'teki gibi tanımlamıştır.

$$K_i: \text{Eğer } x = A_i \text{ ise o halde } y = f_i(x) \quad (4.3)$$

Şekil 4.6'daki sistemde iki girişli ve iki kurallı Takagi - Sugeno tipi yapı için çıkarım mekanizması gösterilmektedir.



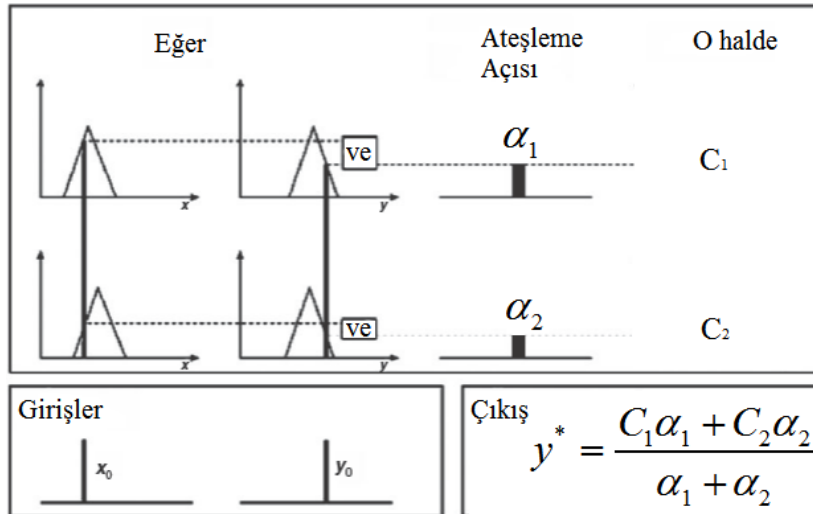
Şekil 4.6: İki giriş iki kural Takagi – Sugeno tipi yapı için çıkarım mekanizması.

#### 4.3.3.3 Tekli (Singleton) tip yapı için çıkarım mekanizması

Mamdani ve Takagi – Sugeno tip yapılar için çıkarım mekanizmalarının özel bir hali olan bu yapının çıkarım mekanizmasında, kural tabanında bulunan kurallardaki girişler dilsel değişkenler, çıkışlar ise sabit keskin değerler ile ifade (4.4)'teki gibi tanımlanır.

$$K_i: \text{Eğer } x = A_i \text{ ise o halde } y = b_i \quad (4.4)$$

Şekil 4.7'deki sistemde iki girişli ve iki kurallı tekli tip yapı için çıkarım mekanizması gösterilmektedir.



Şekil 4.7: İki giriş iki kural tekli tip yapı için çıkarım mekanizması.

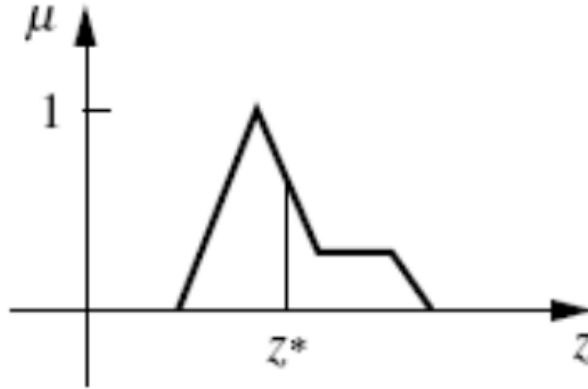
#### 4.3.4 Durulaştırma

Durulaştırma işlemi, bulanık çıkarım sonrasında elde edilen bulanık çıkış değerinin gerçek çıkış işaretine dönüştürülmesi işlemidir. Literatürde birçok farklı durulaştırma yöntemi bulunur. En çok kullanılan durulaştırma yöntemleri, ağırlık merkezi yöntemine ağırlık ortalaması yöntemidir [3].

##### 4.3.4.1 Ağırlık merkezi yöntemi

Bu yöntemde çıkarım mekanizması sonucunda oluşan alanların ağırlık merkezi bulunur ve durulaştırma işlemi ile keskin bir değer olarak hesaplanır. İfade (4.5)'te ağırlık merkezi yönteminin formülasyonu gösterilmektedir. Şekil 4.8'de alanın ağırlık merkezi yöntemi için verilen grafik görülmektedir.

$$z^* = \frac{\int \mu_z z dz}{\int \mu_z dz} \quad (4.5)$$



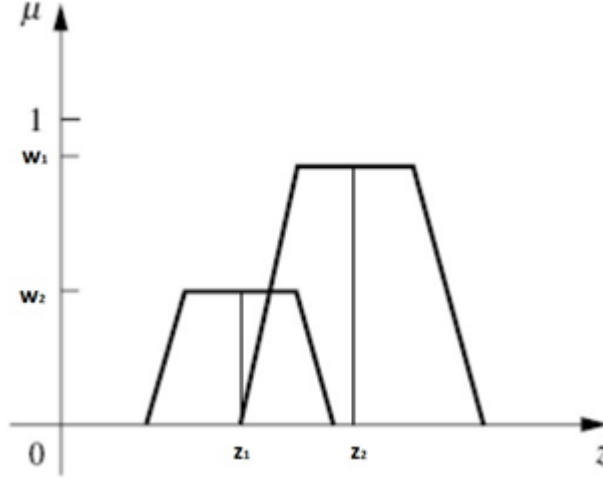
Şekil 4.8: Alanın ağırlık merkezi.

##### 4.3.4.2 Ağırlık ortalaması yöntemi

Bu yöntem genelde simetrik çıkış üyelik fonksiyonları oluşturduğunda kullanılır. Her bir üyelik fonksiyonu maksimum değeri ile ağırlıklandırılarak çıkış değeri hesaplanır. İfade (4.6)'da ağırlık ortalaması yönteminin formülasyonu gösterilmektedir.

Ağırlık merkezi yönteminin formülasyonu gösterilmektedir.

$$z^* = \frac{w_1 z_1 + w_2 z_2}{w_1 + w_2} \quad (4.6)$$



Şekil 4.9: Alanın ağırlık ortalaması.

#### 4.4 Bulanık Modelleme

Bu bölümde karmaşık, doğrusal olmayan veya kısmi belirsiz sistemlerin bulanık küme teorisine ve bulanık mantığa dayalı teknikler ile modellenmesi üzerinde durulacaktır. Bilim ve mühendislikteki çoğu disiplinde gerçek sistemlerin matematiksel modellerinin oluşturulması önemli bir konuteşkil etmektedir. Sistemlerin benzetiminde, analizinde ve kontrolünde sistemlerin matematiksel modelleri kullanıldığından; modelinin sistemin gerçekleri ile örtüşmesi elzemdir. Eğer model yeteri ve gerektiği kadar doğru olmazsa analiz, kestirim ve kontrol işlemlerinin başarılı olması imkânsızdır. Sistemin karmaşıklığı ile modelinin doğruluğu arasında ters orantı vardır. Dolayısıyla bir sistem karmaşıktıkça, modelin doğruluğu azalır. Oluşturulan modeller sistemin gereksiz ayrıntılarından arındırılmış olmalı ve sistemi mümkün olduğunca ifade edebilmelidir. Eğer sistem modeli çok basitse sistem tam olarak ifade edilemez ve model amacını yerine getiremez. Öte yandan model çok karmaşıksa kullanılabilirliğini yitirir [21].

Gelişen üretim teknikleri ile süreçlerin modellenmesi önem kazanmıştır. Endüstride çoğu sistem, çok iyi analiz edilememelerinden ötürü geleneksel teknikler ile iyi modellenememektedirler. Bunun nedeni gerçek sistemlerin lineer olmayan

özelliklerinin ve zamana bağılıklarının fazla oluşunun yanı sıra çokça bilinmezlik içermeleridir.

Bir sistemi modellemek için kullanılan yöntemler üç grupta incelenebilir:

- “Beyaz – kutu” modelleme yöntemi
- “Kara – kutu” modelleme yöntemi
- “Gri – kutu” modelleme yöntemi

Geleneksel olarak, modelleme, sistemin doğasını ve davranışını anlamak ve bunu matematiksel olarak ifade etmek olarak görülür.

#### **4.4.1 Beyaz kutu modelleme yöntemi**

Modellemenin geleneksel tanımına uygun olan yaklaşım “beyaz-kutu” (fiziksel, mekanik) modelleme olarak adlandırılır. Oysaki karmaşık, anlaşılması zor ve içyapısı bilinmeyen, belirsiz sistem parametreleri olan bir sistem için böyle bir model oluşturmak pratikte güçtür.

Diğer yandan böyle bir model elde edilse bile bu modelin anlaşılması da güç olabilir. Buna ek olarak ele alınan sistemin içyapısını anlamak pratikte zaman kaybına yol açabilir ya da çok pahalıya mal olabilir. Hatta bazen sisteme ait böyle bir modelin elde edilmesi imkânsız bile olabilir. Bazı sistemler için sistem yapısı belirlense bile parametrelerin tam olarak belirlenmesi bir başka problemdir. Bu noktada sistem tanıma yöntemlerine ihtiyaç duyulur.

Sistem tanıma yöntemlerinin büyük bir çoğunluğu da doğrusal sistemler için geliştirilmiştir. Oysaki gerçek fiziksel sistemlerin neredeyse tamamı doğrusal olmayan yapıdadır ve ancak yerel olarak yaklaşık modelleri elde edilebilir. Bu durum modelleme hataları oluşturmaktadır.

#### **4.4.2 Kara kutu modelleme yöntemi**

Modelleme için bir başka yaklaşım ise çalışılan sisteme ait genel fonksiyon yaklaşımçı olarak kullanılacak bir kara-kutu yapısı oluşturmaktır. Böylece modelleme problemi, sistemin doğrusal olmayan dinamiğini ifade edecek bir

yaklaşımçı yapısını oluşturma problemine dönüşür. Bu modelleme yönteminde, modelin yapısı, gerçek sistemin yapısıyla çok ilişkili değildir; sistem tanıma problemi de yalnızca modele ait parametrelerin kestirimiyle ilgilidir. Sistem verileri elde edildiği takdirde, kara-kutu model, sisteme ait belirgin bir bilgi gerektirmeden rahatlıkla elde edilebilir.

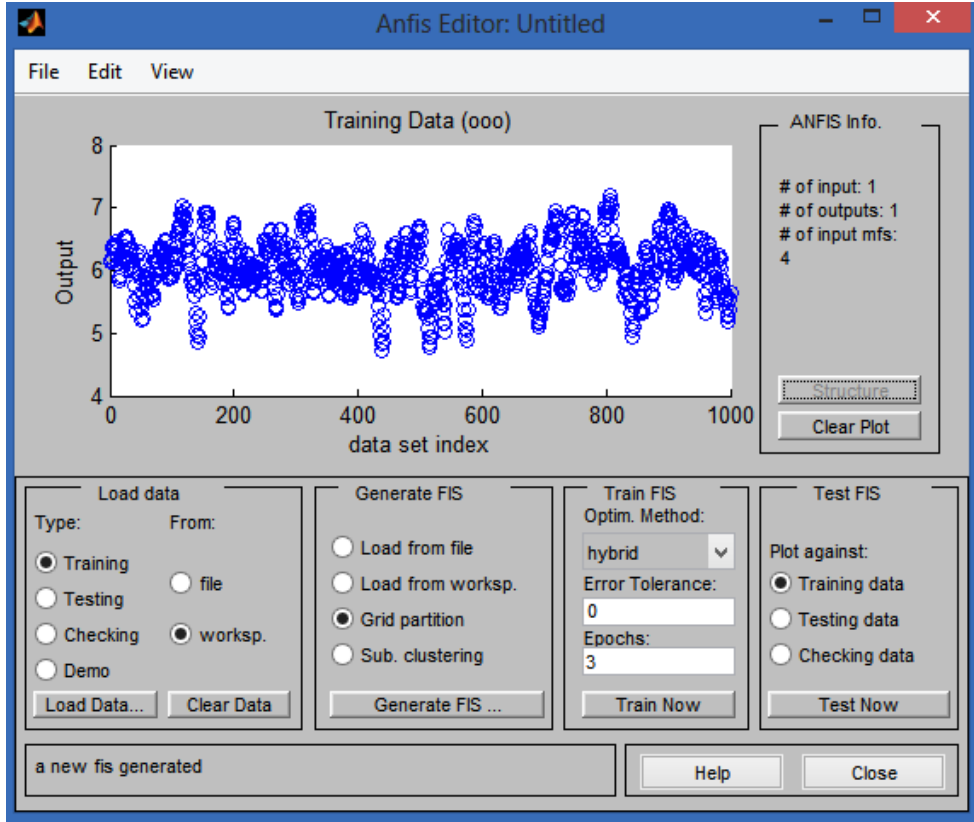
Bu yöntemin en önemli dezavantajı, elde edilen modelin yapısının ve parametrelerinin fiziksel bir anlamının olmamasıdır. Kara-kutu modeller yalnızca sayısal benzetim için kullanılır. Sistem davranışını anlamak, daha büyük ya da küçük ölçekteki sistemlerdeki sistemleri modellemek için kullanılamaz. Dolayısıyla endüstriyel uygulamalar için pek kullanışlı değildir.

#### **4.4.3 Gri kutu modelleme yöntemi**

Beyaz-kutu ve kara-kutu modelleme yöntemlerinin avantajlarını bir araya getirmek amacıyla birçok modelleme yöntemi geliştirilmiştir. Gri-kutu (yarı fiziksel) olarak adlandırılan bu yöntemler, sisteme ait bilinen kısımlarının beyaz kutu modelleme yöntemiyle modellenmesi, bilinmeyen veya daha az kesin olan kısımların ise sistem verilerini, kara-kutu modelleme yapılarını kullanarak kara-kutu yaklaşımı ile modellenmesini ve bu iki modelin birleştirilmesi fikrine dayanır. Geliştirilen modelleme yöntemlerinin çoğunun dezavantajı mühendislerin veya sistem operatörlerinin kesin olmayan ve nitel yapıya sahip olan bilgilerinin modelleme esnasında etkin kullanamamasıdır.

İnsanın daha önceden belirlenen belirsizlikler altında karmaşık işlerle başa çıkabilme gerçeği, araştırmacıları yeni modelleme ve kontrol yöntemleri geliştirmeye yöneltmiştir. “Akıllı” modelleme ve kontrol olarak adlandırılan yöntemler, biyolojik sistemler veya insan zekâsından esinlenerek dinamik sistemlerin modellenmesi veya kontrolüne yönelik geliştirilen yeni fikirlerin ürünüdür.

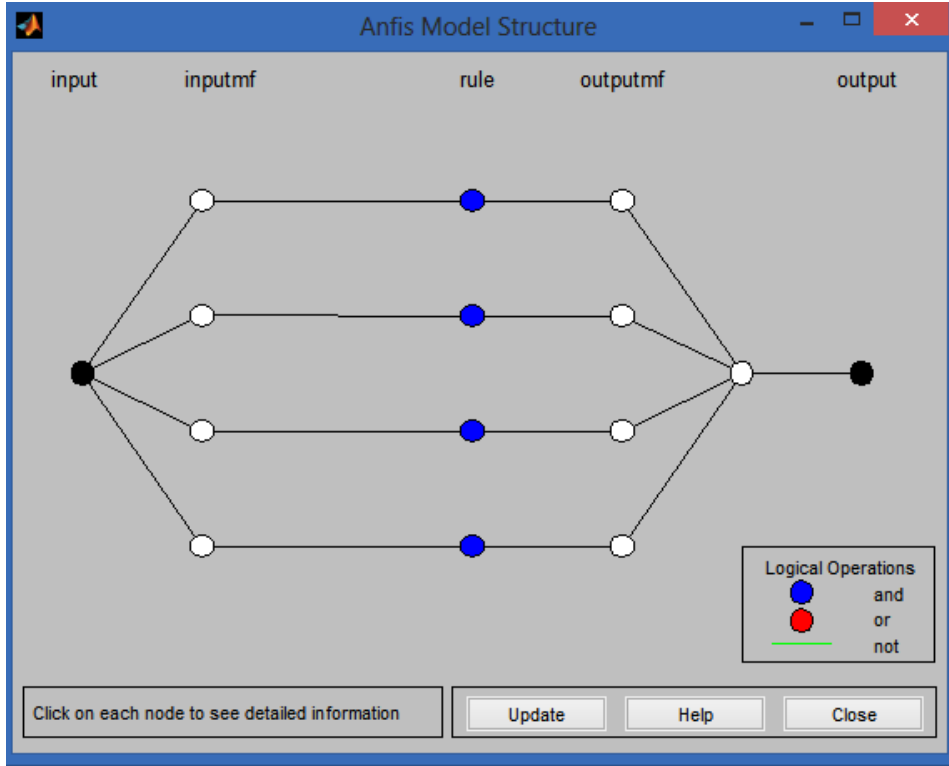
Yapılan tez çalışmasında elde edilen modelleme sonuçları ile karşılaştırma yapılması açısından en çok kullanılan bulanık modelleme yazılımı olan ANFIS seçilmiştir. MATLAB programının bulanık modelleme üzerine bir yazılımı olan ANFIS, 1993 yılında Roger Jang tarafından duyurulmuştur. ANFIS, bulanık modellemede çok önemli bir konuma sahiptir ve bunun getirdiği saygınlık ile en çok yatırım yapılan ve kullanılan araçtır. Şekil 4.10’da ANFIS’in kullanıcı arayüzü gösterilmektedir.



**Şekil 4.10:** ANFIS kullanıcı ara yüzü.

ANFIS, önceki bölümlerde bahsedilen Takagi – Sugeno yapısında bir bulanık model oluşturur. Yani, kısaca hatırlamak gerekirse, sisteme ait Mamdani (üçgen, trapezoid, gauss, vs.) yapısındaki girişlerden ve fonksiyon yapısındaki çıkışlardan bulanık model elde edilir.

ANFIS, çok girişli tek çıkışlı (ÇGTÇ) sistemler için bulanık model oluşturabilir. ANFIS'te giriş ve çıkış üyelik fonksiyonu parametreleri adaptif sinir ağı yapısı ile tanınır veya kestirilir. Şekil 4.11'de bir sistemden toplanan giriş – çıkış verileri ile oluşturulması beklenen bulanık model için ANFIS'in oluşturduğu adaptif sinir ağı yapısı gösterilmektedir.



Şekil 4.11: ANFIS adaptif sinir ağı yapısı.

ANFIS'te, bulanık model oluşturma işlemi için beş katmanlı bir adaptif sinir ağı yapısı yaratılmıştır. Bu katmanlardan kısaca bahsetmek gerekirse:

- Birinci katmanda giriş verileri bulanıklaştırılır.

$l$  katman sayısını,  $i$  düğüm sayısını göstermek üzere her katmandaki düğüm çıkışları  $O_{l,i}$  ile ifade edilirse;

$$O_{1,i} = \mu_{A_i}(x) \text{ for } i = 1, 2 \quad (4.7)$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \text{ for } i = 3, 4$$

Burada  $x$  ve  $y$  olarak verilen ifadeler girişleri  $A_i$  ve  $B_i$  ile gösterilen ifadeler dilsel değişkenleri göstermektedir. Üyelik fonksiyonları için  $\mu(x)$  ifadesi kullanılmaktadır. Bu düğümler uyarlanabilir olup, parametrelerine “öncül parametreler” denilmektedir.



- İkinci katmanda kurallar kullanılarak girişler için t-normu işlemleri yapılır.

Bu katmandaki her düğüm çıkışı aşağıda verilen ifade ile tanımlanır;

$$O_{2,i} = w_i = \mu_{A_i}(x)T\mu_{B_i}(y) \quad (4.8)$$

Burada T, T normunu ifade etmektedir. Genel olarak, “bulanık ve” mantığında çalışan herhangi bir T normu, “minimum alma” ve “çarpım” işlemlerini ifade eder. Ayrıca bu düğümler, sabit (değişmez) düğümlerdir.

- Üçüncü katmanda kurallardaki ateşleme açıları normalize edilir.

Burada  $i$  dereceli düğüm,  $i$  dereceli kuralın ateşleme açısının tüm kuralların ateşleme açılarının toplamına oranını hesaplar.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \text{ for } i = 1, 2 \quad (4.9)$$

Bu katmanın çıkışları “normalize edilmiş ateşleme açıları”denir. Yine benzer şekilde bu düğümler, sabit (değişmez) düğümlerdir.

- Dördüncü katmanda her bir lineer fonksiyon için ağırlıklandırılmış çıkışlar hesaplanır.

Bu katmandaki her düğüm, uyarlanabilir düğümlerdir ve aşağıda verilen ifade ile hesaplanır.

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \text{ for } i = 1, 2 \quad (4.10)$$

Burada  $p_i, q_i$  ve  $r_i$  ile verilen ifadeler, bu düğümün parametre takımındadır ve “sonuç parametreleri” olarak isimlendirilir.

- Beşinci katmanda kestirilmiş çıkış değeri hesaplanır.

Bu katmandaki tek düğüm, gelen tüm sinyallerin toplamı olarak, tüm ağın çıkışını hesaplar.

$$O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (4.11)$$

İfadede kullanılan  $f$  fonksiyonu aşağıdaki gibi tanımlanabilir:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \quad (4.12)$$

$$f = \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2)$$

ANFIS, şekilde (Şekil 4.11) gösterilen yapıyı kullanarak geriye yayılım veya hem geriye yayılım hem de en küçük kareler yöntemlerini beraber kullanarak bir hibrit öğrenme algoritması ile bulanık model oluşturma özelliğine sahiptir.

Biraz ayrıntıya inilirse, bu ileri yönlü hibrit öğrenmede, fonksiyonel sinyalleri dördüncü katman ve sonuç parametreleri en küçük kareler yöntemi ile tanımlanıncaya kadar ileri yönde hareket eder.

Geri yönlü öğrenmede ise, hata sinyali geriye yayılım gösterir ve öncül parametreler gradyal azaltma yöntemi ile hesaplanır.

**Çizelge 4.1:** Yayılım ve yöntem tablosu.

	İleri Yönlü	Geri Yönlü
Öncül Parametreler	Sabit	Gradyal azaltma
Sonuç Parametreleri	En küçük kareler	Sabit
Sinyaller	Düğüm çıkışları	Hata sinyalleri

Özetle, bulanık model oluşturma işleminde adaptif sinir ağı yapısı kullanıldığından giriş ve çıkış parametreleri için bir eğitmeden bahsedilebilir. Eğitim işleminde bir optimizasyon yöntemi olan gradyan vektörü kullanılır. Gradyan vektörü, bulanık modelin gerçek sisteme ne kadar benzediğini ölçer [18]. Gradyan vektörü oluşturulduktan sonra, ANFIS, güncel ve beklenen çıkış arasında RMS ölçütü üzerinden kestirim hatasının minimizasyonu veya giriş – çıkış parametrelerinin eğitimi için geriye yayılım yöntemini kullanarak belli bir iterasyon sayısı sonrasında veri toplanan sistem için bulanık model oluşturur.

ANFIS'in kısıtlarından bahsedilirse sırası ile verilen adımlar konuşulabilir:

- Bu modelleme yalnızca birinci ya da sıfırıncı dereceden Sugeno tipi modelleme yapar.
- Yalnızca bir çıkışı vardır ve ağırlıklar ortalaması durulama yöntemi ile elde edilir. Tüm çıkış üyelik fonksiyonları sabit ya da doğrusal türlerinden biri olmak üzere aynı tipte olmalıdır.
- Kural paylaşımı yoktur; farklı kurallar aynı çıkış üyelik fonksiyonunu kullanamazlar. Bir başka deyişle, çıkış üyelik fonksiyonlarının sayısı kural sayısına eşit olmalıdır.
- Her kuralın ağırlıklandırması aynıdır.



## 5. BENZETİM ÇALIŞMALARI

Bu kısımda ESN yapısı için oluşturulmuş ESN TOOLBOX ile ANFIS yazılımı kullanılarak belli sistemlerden alınan giriş – çıkış verileri yardımıyla modeller oluşturulacaktır. Bu bölümdeki amaç, ANFIS yazılımının ve ESN TOOLBOX yazılımının model oluşturma işlevlerinin karşılaştırılmasıdır.

ESN TOOLBOX Jaeger tarafından geliştirilmiş ve kullanıma açık olan bir çalışmadır. İçeriğindeki dosyaları kısaca açıklamak gerekirse, çoğu işlemin yapıldığı ana dosya “demoScript” dosyasından başlanabilir.

**demoScript:** Veri setini yüklemek, ağ oluşturmak, ESN eğitmek, yeni veri setleri ile denemek ve ağ çıkışını çizdirmek işlemlerini yürütür.

Alfabetik olarak açıklamaya devam edilirse:

**compute\_NRMSE:** Verilen ESN çıkışı ve danışman bilgilerine göre ESN için normalize edilmiş hatanın karesel ortalamasını hesaplar.

**compute\_statematrix:** Girişlerin ESN boyunca iletimini sağlar ve rezervuar durumlarını saklar.

**compute\_teacher:** Aktivasyon fonksiyonunun tersini danışman üzerinde ölçekler, kaydırır ve uygular.

**generate\_esn:** ESN rezervuarını oluşturur, giriş ve çıkışlara göre ağırlık matrislerini belirler.

**generate\_internal\_weights:** Rezervuar ağırlık matrisini oluşturur.

**generate\_NARMA\_sequence:** NARMA denklemine göre eğitim ve test dizileri oluşturur.

**load\_esn:** Bir dosyadan eğitilmiş ya da eğitilmemiş ESN yükler.

**plot\_sequence:** Danışman ve ağ çıkışını aynı grafikte çizer.

**plot\_states:** Rezervuar durumlarını istenen sayıda çizer.

**pseudoinverse:** Çıkış ağırlıklarını pseudo-ters işlemine göre hesaplar.

**save\_esn:** Oluşturulan ESN yapısını kaydeder.

**split\_train\_test:** Verileri eğitim ve test olmak üzere ayırır.

**test\_esn:** Eğitilmiş bir ESN yapısını giriş verisi üzerinde test eder.

**train\_esn:**Seçilen methoda göre ESN yapısını eğitir (ağırlık matrislerini hesaplar vs).

Benzetim çalışması öncelikle sistemin seçilip veri sayısı – çıkış değeri bilgilerinin verilmesi ile başlar. Bundan sonra belirlenen ölçütler altında iki araç da kullanılarak sistemlerin modelleri oluşturulur. Modelleme işlemi bittikten sonra her bir aracın ulaşabildiği hatanın NRMSE değerleri kaydedilir.

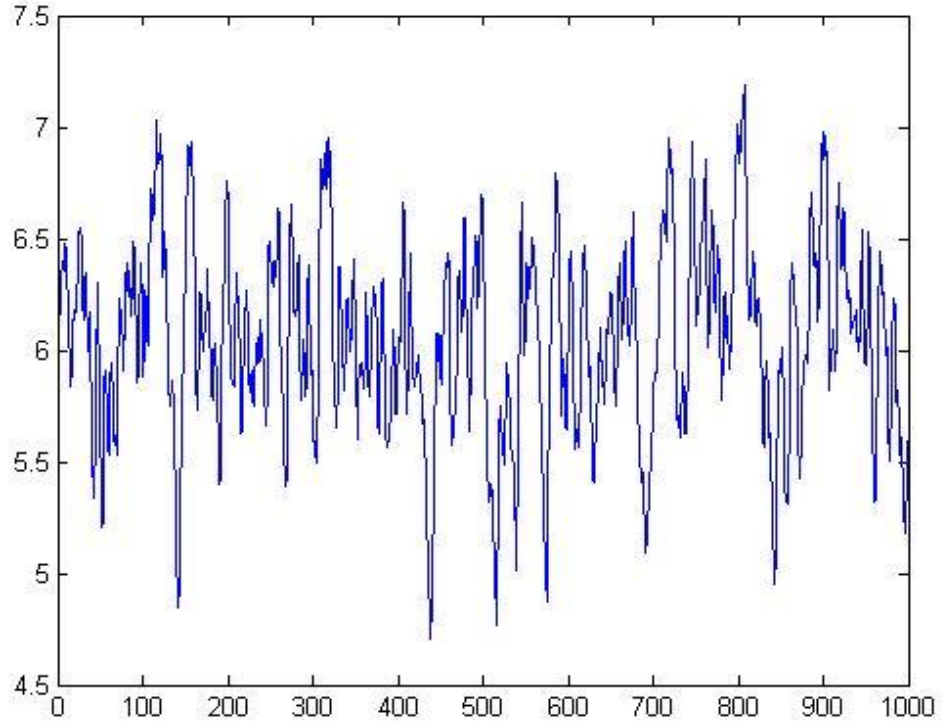
İlk benzetim çalışmasında tek giriş tek çıkış sistemin (MATLAB *PT326*), ikinci benzetim çalışmasında zaman serilerinden *Rosler* serisinin ve son benzetim çalışmasında da çok giriş tek çıkış bir sistemin modellenmesi yapılmaktadır.

Benzetim çalışmalarının yapıldığı bilgisayarın ve MATLAB programının teknik özellikleri şöyledir:

- İşlemci: Intel (R) Core (TM) i5-3317U CPU @ 1.70 GHz
- RAM: 4,00 GB
- Sistemtürü: 64 bit işletim sistemi
- İşletimsistemi: Windows 8
- MATLAB sürümü: MATLAB R2013a

## 5.1 MATLAB PT326 Sisteminin Modellenmesi

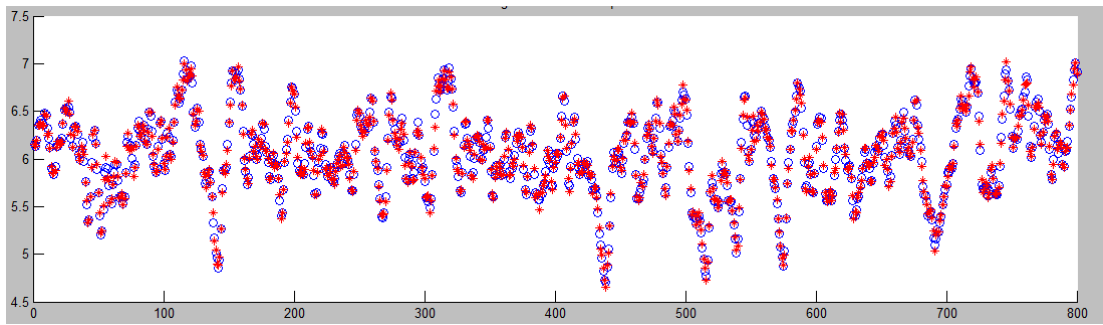
Kullanılan PT326 ısıtma süreci sisteminde, pervane yardımıyla dışortamdan çekilen hava ısıtıcı ızgara ve tüp içinden geçirilerek tekrar dış ortama gönderilir. Süreç, tüpün içinden geçen havanın sıcaklığını, kullanıcı tarafından ayarlanan referans değeri ile karşılaştırarak ısıtıcıelemana uygulanacak elektriksel gücün miktarını belirleyen bir kontrol işareti üretmektir. Bulanık modeli çıkarılacak olan PT326 sistemi için giriş – çıkış verileri MATLAB ile birlikte gelmektedir ve elde edilen veriler MATLAB komut satırına “*load dryer2.mat*” komutunun yazılması ile çalışma alanına yüklenir.Çalışma alanına iki sütun vektör yüklendiği görülür. Bunlardan  $u_k$  vektörü sisteme uygulanan giriş verilerini,  $y_k$  vektörü de sistemden alınan çıkış verilerini ifade etmektedir. PT326 sisteminden toplanan veri bilgileri altında, sistemden alınan veri sayısı – çıkış değeri grafiği Şekil 5.1’de verilmektedir.



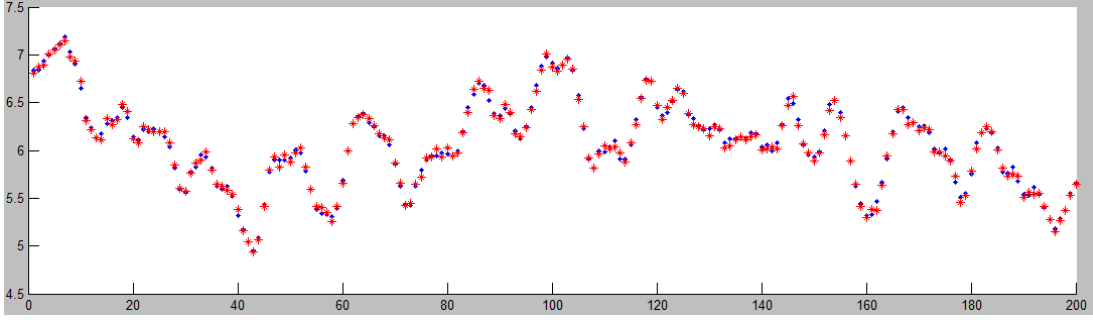
**Şekil 5.1:** PT326 sisteminin çıkış verisi.

### 5.1.1 ANFIS ile oluşturulan bulanık model

Oluşturulan bulanık model eğitim verisi ve test verisi ile sistemden toplanan giriş – çıkış verileri ile üst üste çizdirilirse Şekil 5.2 ve Şekil 5.3’te verilen grafikler elde edilir. Bu grafiklerde modellemedeki başarı göze çarpmaktadır.



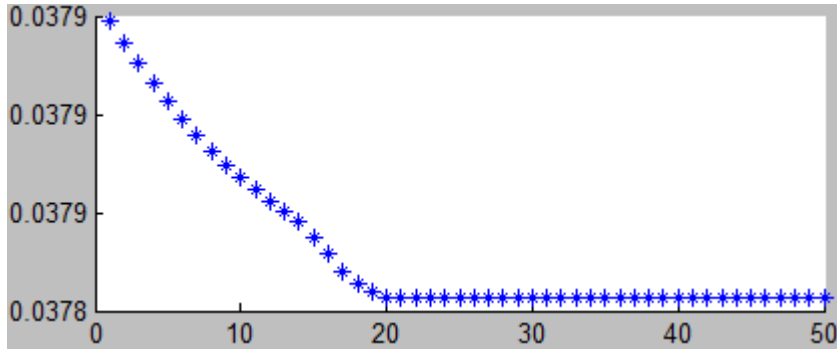
**Şekil 5.2:** PT326 sistemi için ANFIS eğitim verisi grafiği.



**Şekil 5.3:** PT326 sistemi için ANFIS test verisi grafiği

Modellemenin başarılı olması açısından giriş vektörü,  $y_k$  vektörü kullanılarak  $y_{k-1}, y_{k-2}, u_{k-3}, u_{k-4}$  vektörlerinden oluşturulmuştur. ANFIS yazılımında modelleme yapılırken devir sayısı 50, giriş üyelik fonksiyonlarının sayısı üç olarak alınmıştır. Sugeno tipi çıkarım mekanizması ile giriş üyelik fonksiyonları üçgen, çıkış üyelik fonksiyonları ise sabittir.

Bulanık model çıkışı ile gerçek sistem çıkışı arasındaki hatanın devir sayısı ölçütüne göre değişim grafiği Şekil 5.4'te gösterilmektedir. Benzetim bittiğinde hatanın değeri 0.037843 olarak ölçülmüştür. ANFIS yazılımında toplam veri, eğitim verileri %80 oranında, test verileri %20 oranında olacak şekilde paylaşılmıştır.

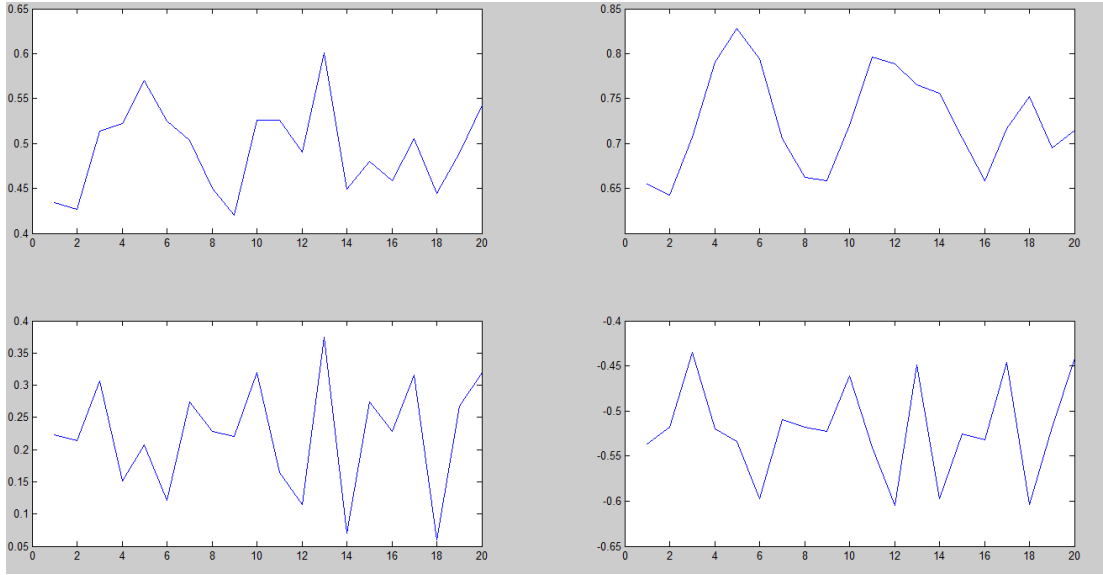


**Şekil 5.4:** PT326 sistemi için ANFIS öğrenme grafiği.

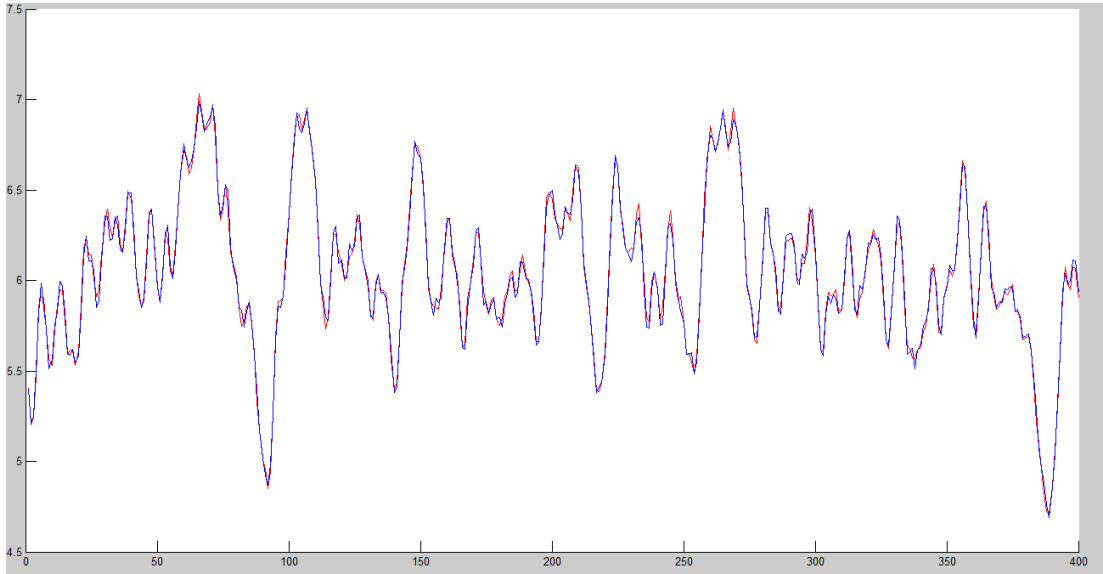
### 5.1.2 ESN ile oluşturulan sistem modeli

ESN ile oluşturulan modelde ilk dört rezervuar biriminin durumları şekil 5.5'te verildiği gibidir. Eğitim verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik şekil 5.6'da verilmektedir. Test verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik şekil 5.7'de görülmektedir.

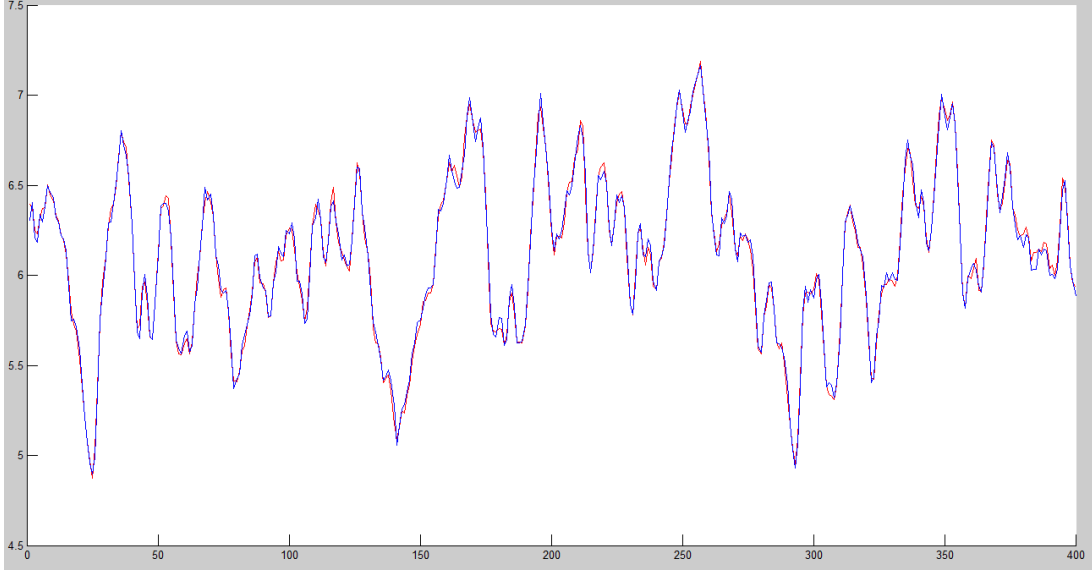




**Şekil 5.5:** PT326 sistemi için ESN rezervuar durumları.



**Şekil 5.6:** PT326 sistemi için ESN eğitim verisi grafiği.



**Şekil 5.7:** PT326 sistemi için ESN test verisi grafiği.

Grafiklerde ESN yapısının değerleri yakalama ve takip etmedeki başarısı görülmektedir. İç katmanda 30 birimlik bir rezervuar yapısı kullanılmış olup, spektral yarıçapı 0.6 birimdir. Eğitim ve test verileri %50 oranı ile paylaşılmıştır.

**Çizelge 5.1:** PT326 sistemi için hata analizi.

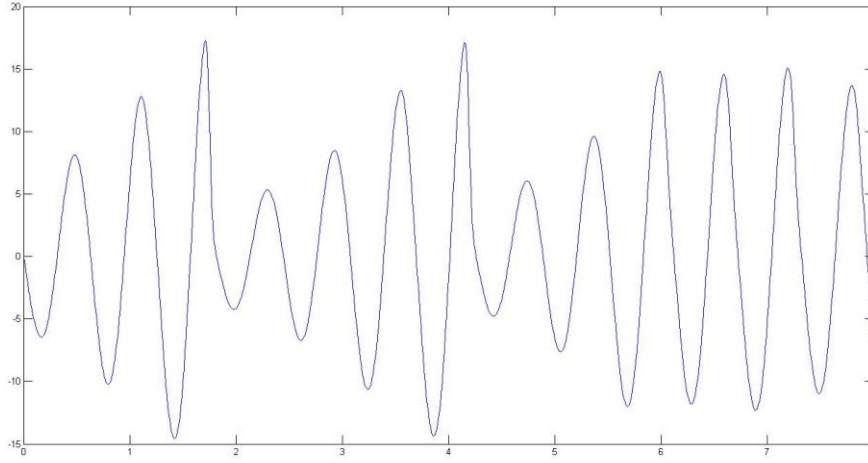
	Eğitim Hatası	Test Hatası
ESN	0.084319	0.082188
ANFIS	0.037843	0.038423

ESN yapısı ile modellemede, ANFIS ile yapılan modellemeye yakın sonuçlar alındığı gözlenmektedir. Hatta eğitim hatası ile test hataları kıyaslandığında ESN yapısının daha başarılı olduğu gözlenmektedir. ESN yapısında yalnızca çıkış katmanındaki ağırlıklar hesaplandığı için ANFIS yazılımına göre sonuçlar çok daha hızlı elde edilmiştir.

## 5.2 Zaman Serilerinin Modellenme

Zaman serilerinden örnek olarak Rossler zaman serisi incelenmiştir [19]. İfade (5.1)'de Rossler serisi oluşturmada kullanılan denklemler görülmektedir. Burada standart değerler ( $a = 0.15, b = 0.20, c = 10.0$ ) kullanılmaktadır. Şekil 5.8'de Rössler zaman serisi için çıkış verisi gösterilmektedir.

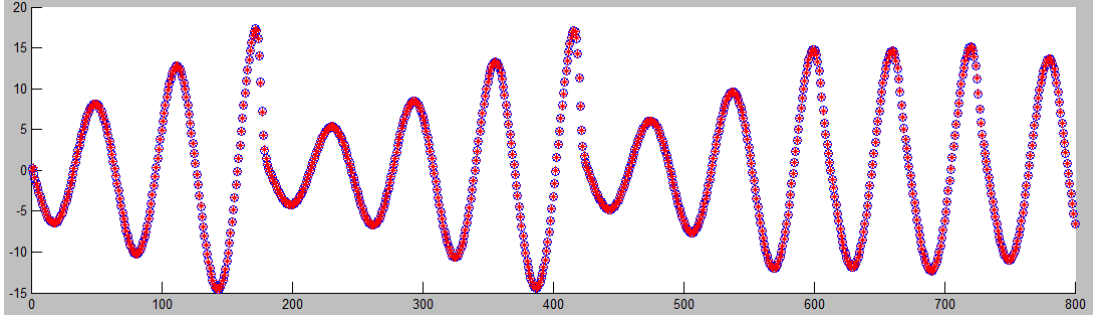
$$\begin{aligned}\dot{x} &= -z - y \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c)\end{aligned}\tag{5.1}$$



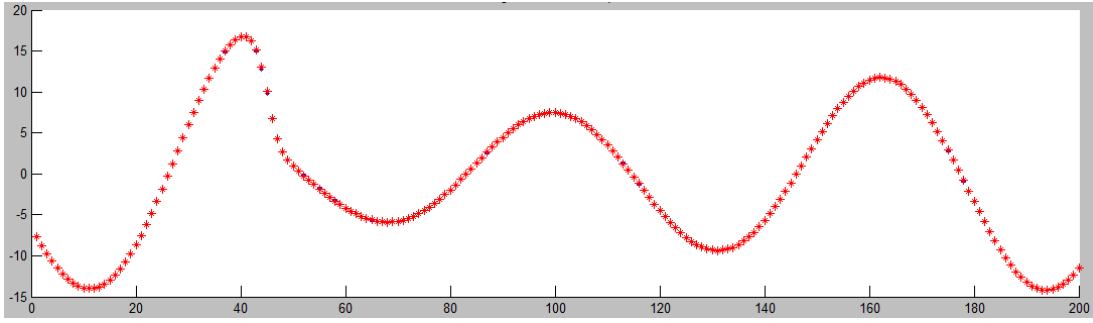
**Şekil 5.8:** Rossler zaman serisi çıkış verisi.

### 5.2.1 ANFIS ile oluşturulan bulanık model

Oluşturulan bulanık model eğitim verisi ve test verisi ile sistemden toplanan giriş – çıkış verileri ile üst üste çizdirilirse Şekil 5.9 ve Şekil 5.10'da verilen grafikler elde edilir. Bu grafiklerde modellemedeki başarı göze çarpmaktadır. Şekil 5.11'de öğrenme grafiği görülmektedir.



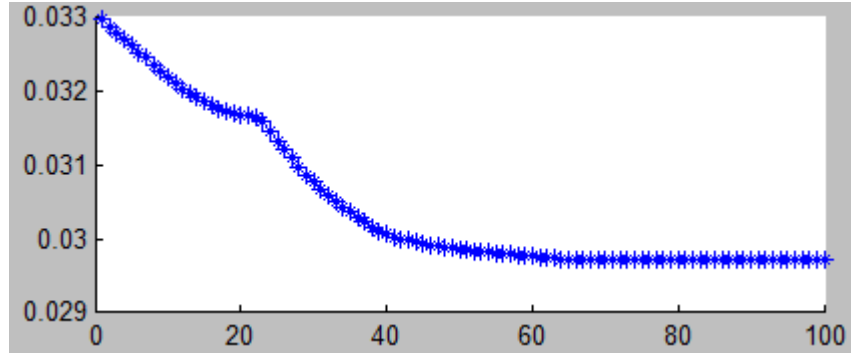
**Şekil 5.9:** Rossler zaman serisi için ANFIS eğitim verisi grafiği.



**Şekil 5.10:** Rossler zaman serisi için ANFIS test verisi grafiği.

Modellemenin başarılı olması açısından giriş vektörü,  $y_k$  vektörü kullanılarak  $y_{k-1}, y_{k-2}, y_{k-3}, y_{k-4}$  vektörlerinden oluşturulmuştur. ANFIS yazılımında modelleme yapılırken devir sayısı 100, giriş üyelik fonksiyonlarının sayısı üç olarak alınmıştır. Sugeno yapısına bağlı kalınmıştır; giriş üyelik fonksiyonları üçgen, çıkış üyelik fonksiyonları ise sabittir.

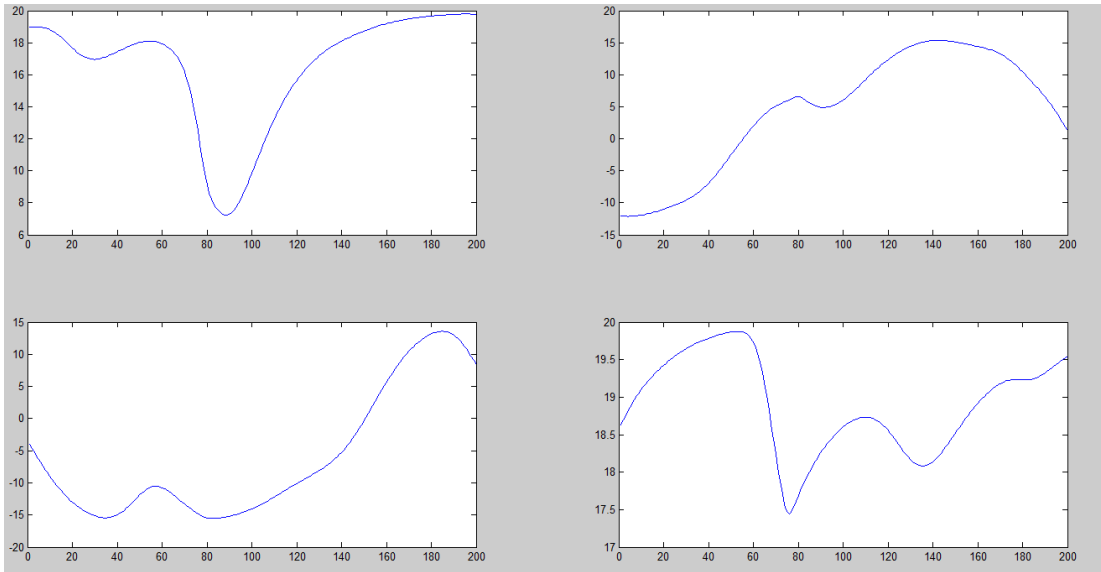
Bulanık model çıkışı ile gerçek sistem çıkışı arasındaki hatanın devir sayısı ölçütüne göre değişim grafiği Şekil 5.11'de gösterilmektedir. Benzetim bittiğinde hatanın değeri 0.029002 olarak ölçülmüştür. ANFIS yazılımında toplam veri, eğitim verileri %80 oranında, test verileri %20 oranında olacak şekilde paylaşılmıştır.



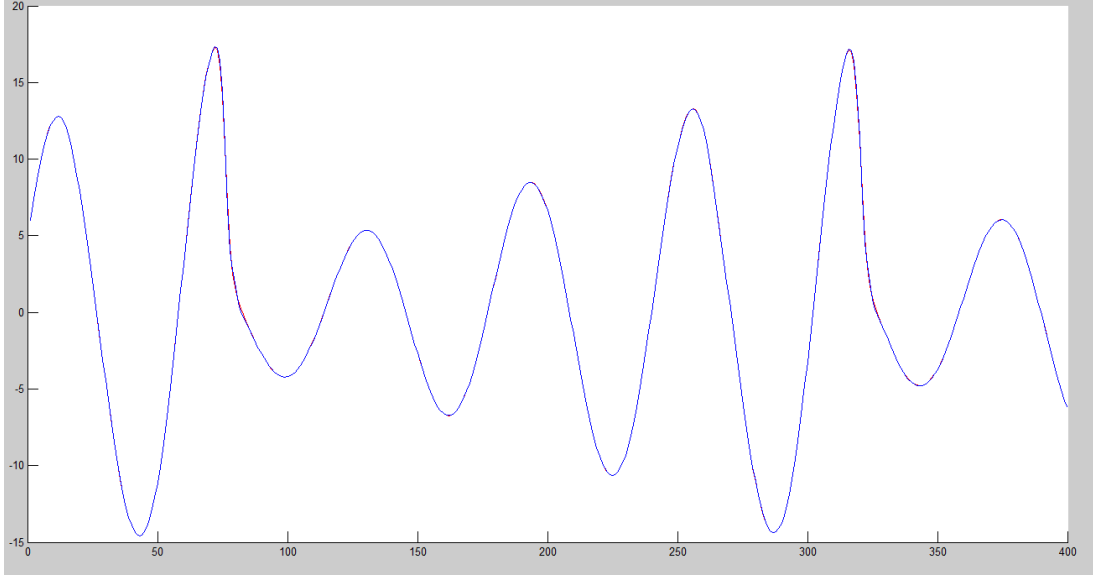
**Şekil 5.11:** Rossler zaman serisi için ANFIS öğrenme grafiği.

### 5.2.2 ESN ile oluşturulan sistem modeli

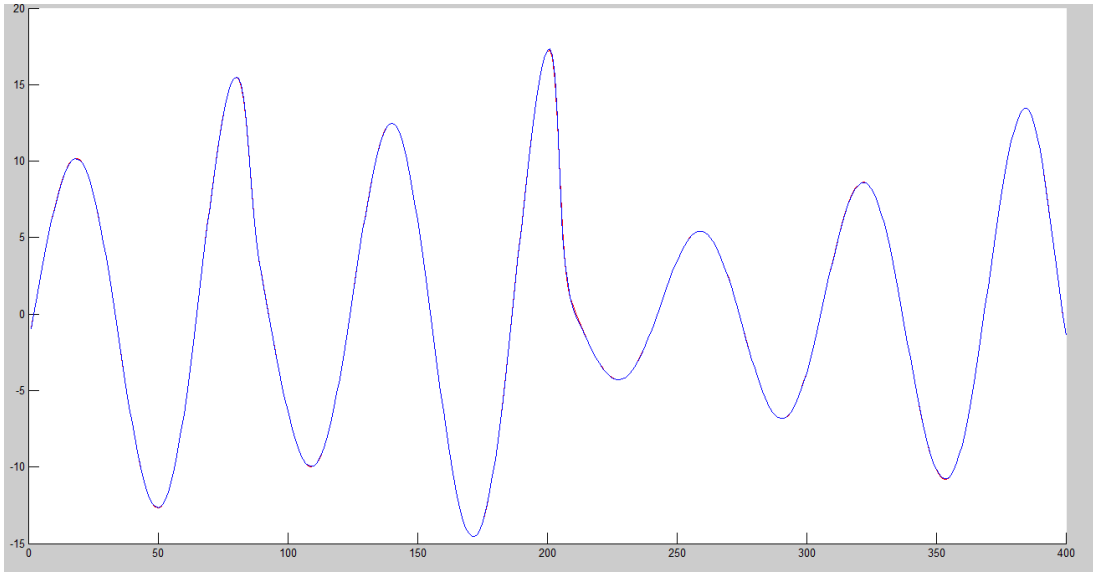
ESN ile oluşturulan modelde ilk dört rezervuar biriminin durumları Şekil 5.12’de verildiği gibidir. Eğitim verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik Şekil 5.13’te verilmektedir. Test verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik şekil 5.14’te görülmektedir.



**Şekil 5.12:** Rossler zaman serisi için ESN rezervuar durumları.



**Şekil 5.13:** Rossler zaman serisi için ESN eğitim verisi grafiği.



**Şekil 5.14:** Rossler zaman serisi için ESN test verisi grafiği.

Grafiklerde ESN yapısının değerleri yakalama ve takip etmedeki başarısı görülmektedir. İç katmanda 30 birimlik bir rezervuar yapısı kullanılmış olup, spektral yarıçapı 0.13 birimdir. Eğitim verileri %80 oranı ile test verileri %20 oranı ile paylaşılmıştır.

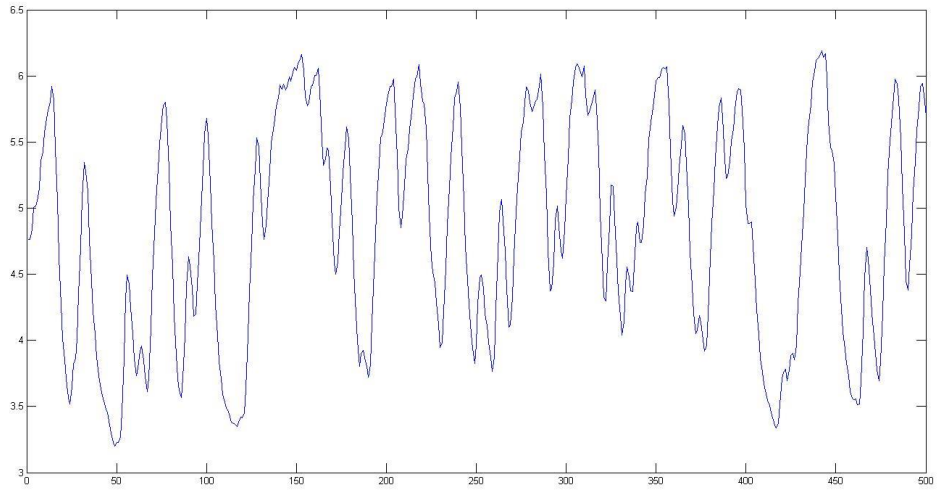
**Çizelge 5.2:** Rossler zaman serisi için hata analizi.

	Eğitim Hatası	Test Hatası
ESN	0.0052473	0.0055466
ANFIS	0.037843	0.0297214

ESN yapısı ile modellemede, ANFIS ile yapılan modelden daha başarılı sonuçlar alındığı gözlenmektedir. ANFIS yazılımında üyelik fonksiyonlarındaki artış ile meydana gelen yavaşlama ve öğretimdeki yavaşlık burada görülmektedir.

### 5.3 Çok Giriş Tek Çıkış Sistemlerin Modellenmesi

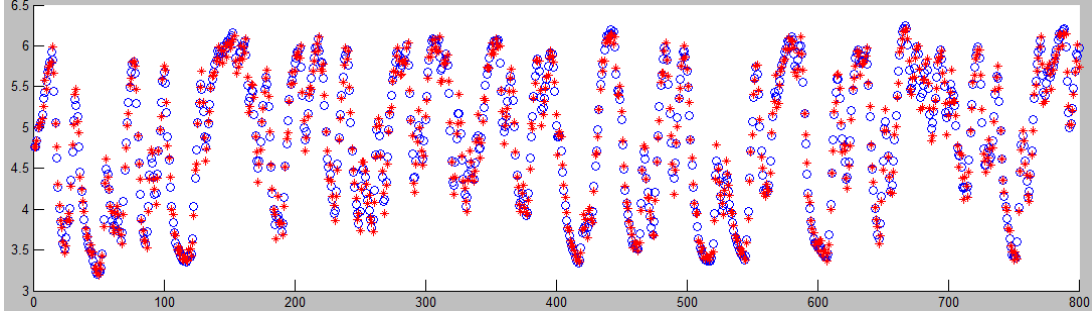
MATLAB üzerinde “*load iddata8*” komutu ile elde edilen çok giriş çok çıkış data seti ile benzetim çalışmaları gerçekleştirilmiştir. Çalışma alanına üç giriş tek çıkış sistem verilerinin yüklendiği görülür. Devam eden çalışmalarda diğer çalışmalara benzer şekilde ANFIS yazılımı ile modelleme ve ESN yapısı ile modelleme arasında karşılaştırma yapılacaktır. Şekil 5.15’te kullanılan ÇGTÇ sistemin çıkış verisi gösterilmektedir.



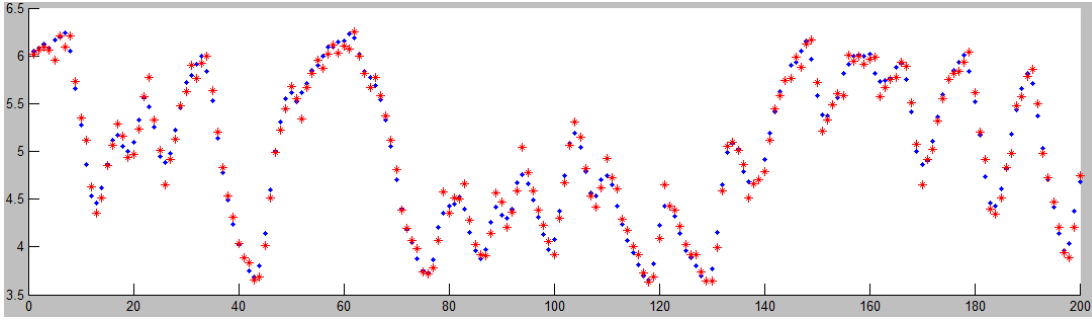
**Şekil 5.15:** ÇGTÇ sistemin çıkış verisi.

### 5.3.1 ANFIS ile oluşturulan bulanık model

Oluşturulan bulanık model eğitim verisi ve test verisi ile sistemden toplanan giriş – çıkış verileri ile üst üste çizdirilirse Şekil 5.16 ve Şekil 5.17’de verilen grafikler elde edilir.



Şekil 5.16: ÇGTÇ sistem için ANFIS eğitim verisi grafiği.



Şekil 5.17: ÇGTÇ sistem için ANFIS’te test verisi grafiği.

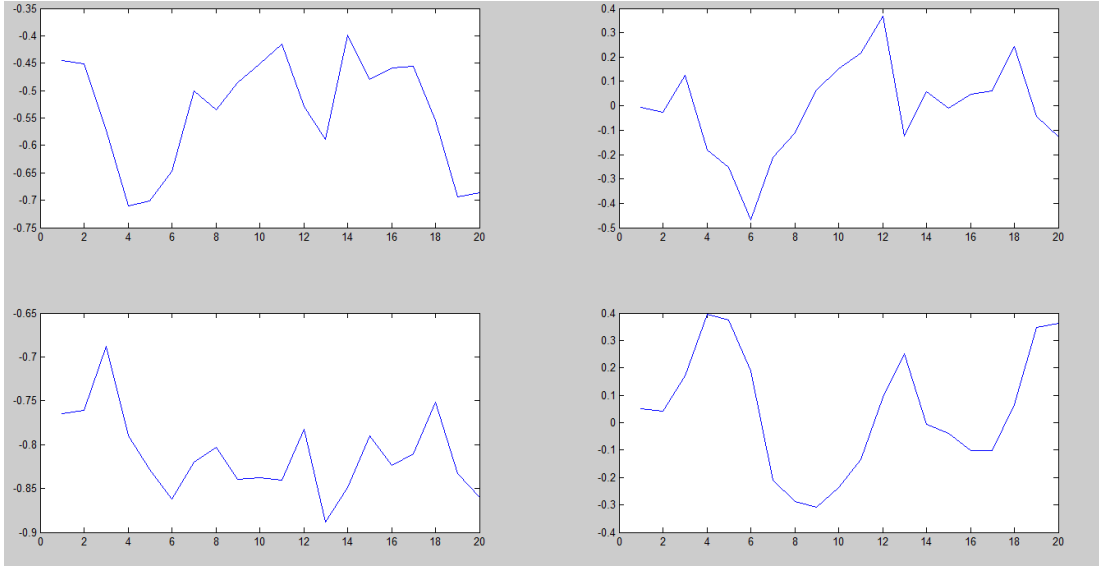
Modellemenin başarılı olması açısından giriş vektörü,  $y_k$  vektörü kullanılarak  $y_{k-1}, y_{k-2}, y_{k-3}, u_{k-5}$  vektörlerinden oluşturulmuştur. ANFIS yazılımında modelleme yapılırken devir sayısı 10, giriş üyelik fonksiyonlarının sayısı üç olarak alınmıştır. Sugeno yapısına bağlı kalınmıştır; giriş üyelik fonksiyonları üçgen, çıkış üyelik fonksiyonları ise sabittir. Benzetim bittiğinde hatanın değeri 0.093384 olarak ölçülmüştür. ANFIS yazılımında toplam veri, eğitim verileri %80 oranında, test verileri %20 oranında olacak şekilde paylaşılmıştır.

Bu adımda ANFIS yazılımında karşılaşılan sorunlardan biri, modelleme yapılırken giriş sayısının fazla olmasından ve bu yüzden üyelik fonksiyonları ile modelleme oluşturulurken 10 devir için dahi yavaş ve kötü sonuçlar vermiş olmasıdır.

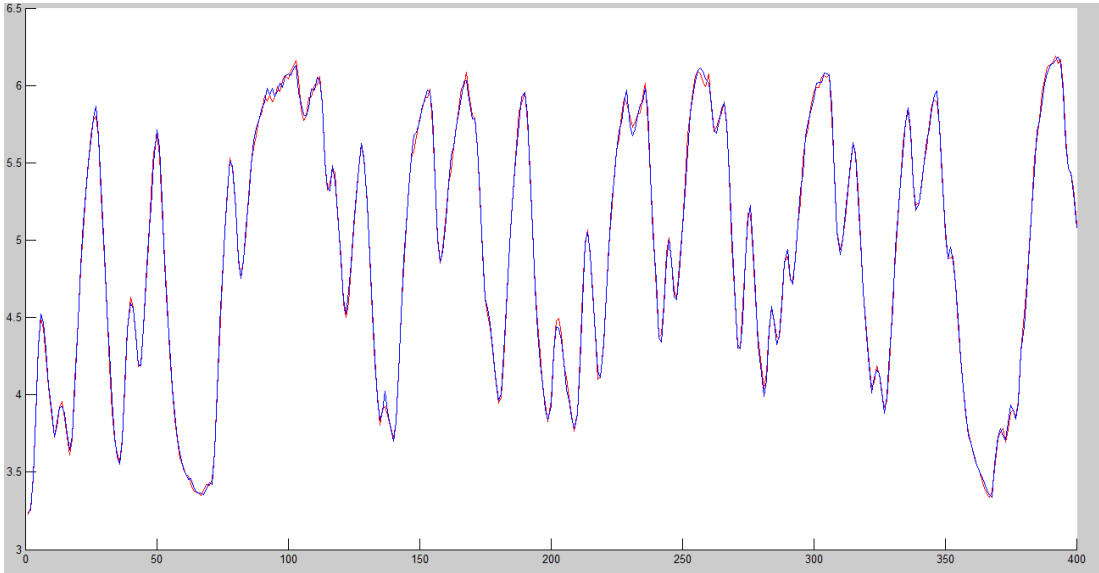


### 5.3.2 ESN ile oluşturulan sistem modeli

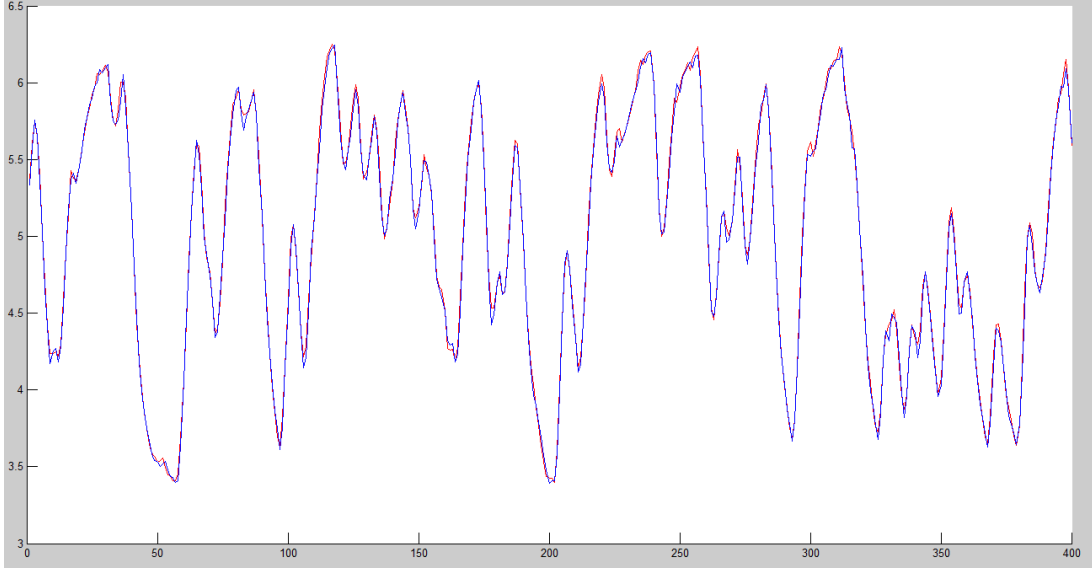
ESN ile oluşturulan modelde ilk dört rezervuar biriminin durumları Şekil 5.18’de verildiği gibidir. Eğitim verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik Şekil 5.19’da verilmektedir. Test verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik şekil 5.20’de görülmektedir.



Şekil 5.18: ÇGTÇ sistem için ESN rezervuar durumları.



Şekil 5.19: ÇGTÇ sistem için ESN eğitim verisi grafiği.



**Şekil 5.20:** ÇGTÇ sistem için ESN test verisi grafiği.

İç katmanda 30 birimlik bir rezervuar yapısı kullanılmış olup, spektral yarıçapı 0.29 birimdir. Eğitim verileri %80 oranı ile test verileri %20 oranı ile paylaşılmıştır.

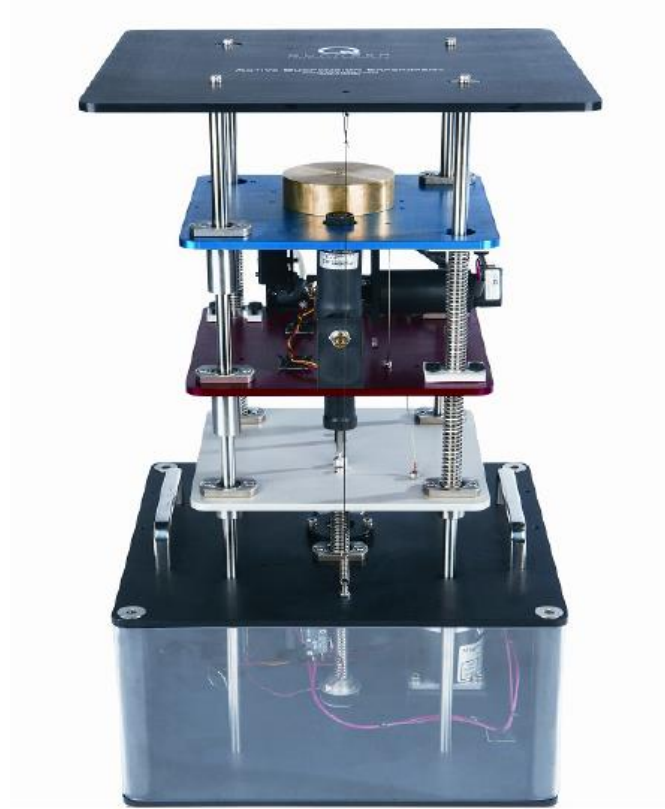
**Çizelge 5.3:** ÇGTÇ sistem için hata analizi

	Eğitim Hatası	Test Hatası
ESN	0.045242	0.05242
ANFIS	0.0933843	0.0933741

Yapılan çalışmalarda görülmüştür ki ESN yapısı ile oluşturulan modeller ANFIS yazılımı ile oluşturulan modellere göre daha başarılı sonuçlar vermektedir. Çok giriş tek çıkış sistemlerde ANFIS modeli donanım sıkıntısı yaratırken, ESN yapısında daha hızlı ve başarılı sonuçlar alınmaktadır. Ayrıca çok sayıda üyelik fonksiyonu kullanmak istenildiğinde ANFIS yazılımında sorun yaşanırken, ESN yapısı ile istenildiği kadar büyük bir ağ oluşturulabildiği ve sorunsuz kullanılabildiği görülmüştür.

## 6. AKTİF SÜSPANSİYON DENEY SETİ

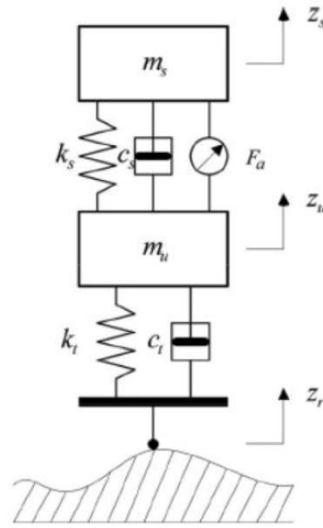
Yapılan deneysel çalışmalarda, İ.T.Ü. Kontrol Mühendisliği Bölümü bünyesindeki Kontrol Laboratuvarı'nda bulunan, Quanser firmasının “Aktif Süspansiyon Tesisi” isimli, çeyrek araba modeline dayanan bir aktif süspansiyon sistemi deney seti kullanılmıştır. Bu bölümde bu deney seti kısaca tanıtılacaktır. Deneysel çalışmalarda kullanılan aktif süspansiyon deney setinin genel görünüşü, Şekil 6.1 ile verilmiştir.



**Şekil 6.1:** Aktif süspansiyon deney setinin görünümü.

Aktif süspansiyon deney seti, özellikle çeyrek araba matematik modeli kullanarak yapılan aktif süspansiyon kontrolü amaçlı kontrol tasarımlarında, uygun bir deneysel platform tesis etmesi açısından avantajlıdır. Deney seti, yol bozucusunu üreten ve set için sabit bir platform oluşturan bir taban kısmı ile yapısı çeyrek araba modeliyle eşdeğer şekilde arabanın çeyrek gövdesinin bir benzetimini yapan çeyrek araba gövdesi ve süspansiyon kısmından oluşur. Burada yapılan açıklamalar, Şekil 6.2’de modeli verilen, ifade (6.1) ve ifade (6.2)’de denklemleri verilen yay-damper-eyleyici

+ yay-damper yapısındaki çeyrek araba matematik modeline dayanmaktadır. Taban kısmının içinde, yol bozucusunu üreten bir elektrik motoru ve bu motorun hareketini, yolun konumunu temsil eden, Şekil 6.2’de taban kısmının üzerindeki ilk plaka olarak gözüken kaideye aktaran kayışlar ve vidalı mil bulunur. Bu kaidenin düşey konumu  $z_r$  ile verilmiştir. Bu kaidenin üzerindeki kaide, tekerlek, lastik ve süspansiyonun ortak kütesini temsil eden, konumu  $z_u$  ile verilen kısımdır. Bu iki kaidenin arasında yay ve damper elemanları bulunur. En üstteki kaide ise, düşey konumu  $z_s$  ile verilen kısımdır ve araba gövdesinin çeyrek kütesini temsil etmektedir. Bu kaide ile tekerlek ve süspansiyonun ortak kütesini temsil eden kaide arasında ise yay ve damper elemanlarının yanısıra aktif eleman olan eyleyici olarak bir elektrik motoru bulunur.



**Şekil 6.2:** Yay-damper-eyleyici + yay-damper yapısındaki çeyrek araba modeli.

Şekil 6.2 ile verilen yay-damper-eyleyici + yay-damper yapısındaki çeyrek-araba modelinde,  $k_s$  ve  $c_s$  parametreleri pasif süspansiyon elemanları olan yay ve damper elemanlarını,  $F_a$  ise aktif süspansiyon elemanı olan eyleyiciyi temsil etmektedir.

Modeldeki  $k_t$  ve  $c_t$  parametreleri, tekerleğin yay sabitini ve sönümünü ifade eder. Bu model için hareket denklemleri şu şekilde yazılabilir:

$$m_s \ddot{z}_s + c_s (\dot{z}_s - \dot{z}_u) + k_s (z_s - z_u) = F_a \quad (6.1)$$

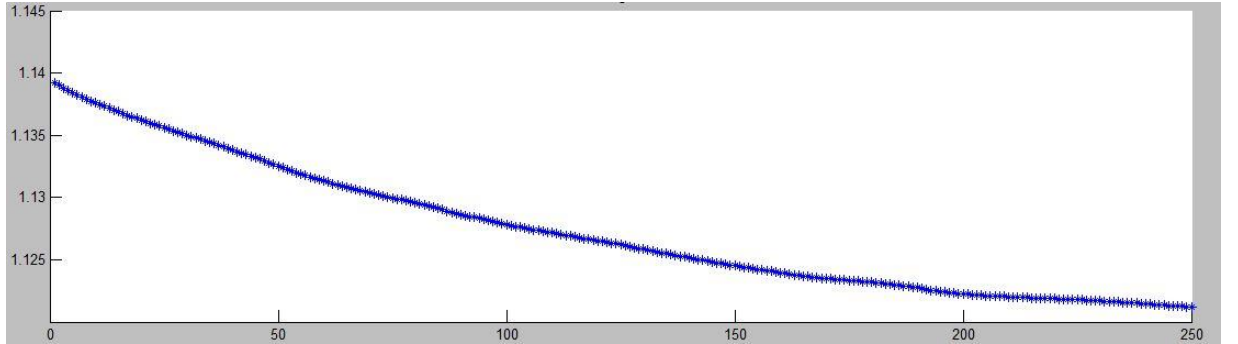
$$m_u \ddot{z}_u - k_s (z_s - z_u) + k_t (z_u - z_r) - c_s (\dot{z}_s - \dot{z}_u) + c_t (\dot{z}_u - \dot{z}_r) = -F_a \quad (6.2)$$

Algılayıcı olarak ise, aktif süspansiyon sisteminin tüm durumlarını ölçmeye olanak sağlayacak şekilde, dört adet algılayıcı ile deney seti donatılmıştır. Araba gövdesinin düşey ivmesini ölçmek için, en üstteki kaidenin üzerinde bir adet ivmeölçer

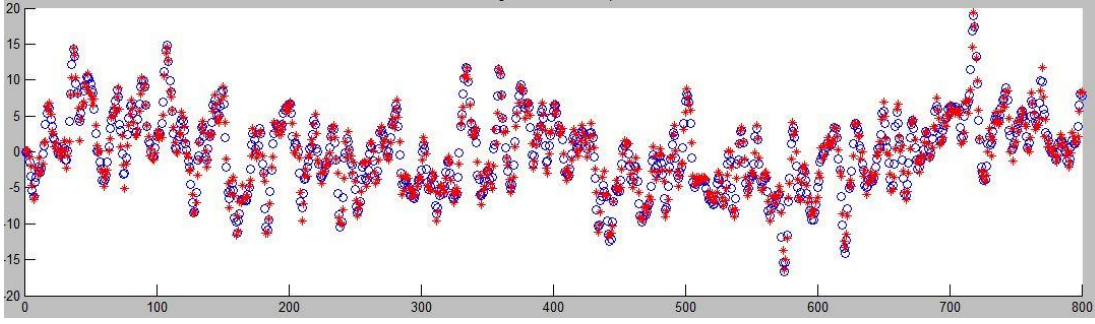
mevcuttur. Süspansiyon sıkışmasını ölçmek içinse, içindeki plastik ip en üst ve orta kaidelere sabitlenmiş bir doğrusal bocurgat mekanizması ile birlikte bir optik kodlayıcı bulunmaktadır. Araba gövdesinin mutlak düşey konumunu ölçmek için, en üst kaidede ayrıca bir optik kodlayıcı bulunur. Yol konumunu ölçmek içinse, alttaki kaideye de bir optik kodlayıcı konmuştur. Güvenlik için, yol bozucusunu üreten mekanizmanın düşey konumu, bir limit anahtarı yardımıyla sınırlandırılmıştır; herhangi bir tehlikeli çalışma durumunda bu limit anahtarı sınırlarına ulaşırsa deney seti otomatik olarak gücü kesmektedir, tekrar çalışma için limit anahtarını güvenli bölgeye getirip taban kısmındaki reset tuşuna basmak gerekir.

### 6.1 ANFIS ile Oluşturulan Bulanık Model

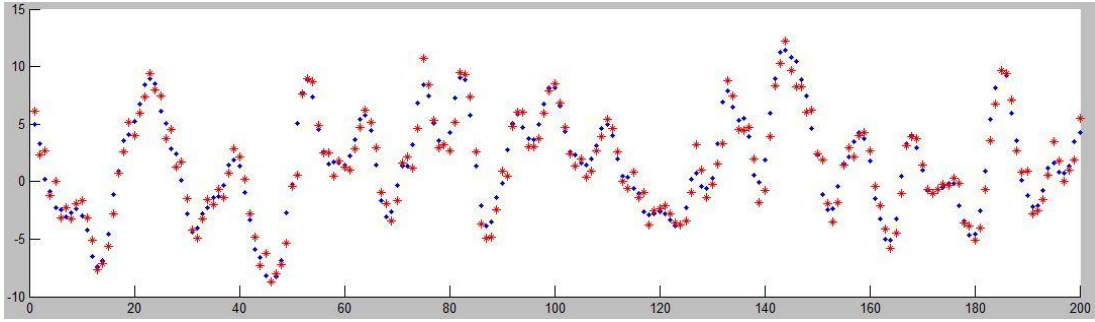
Oluşturulan bulanık model eğitim verisi ve test verisi ile sistemden toplanan giriş – çıkış verileri ile üst üste çizdirilirse Şekil 6.3 ve Şekil 6.4'te verilen grafikler elde edilir.



**Şekil 6.3:** Aktif süspansiyon deney seti için ANFIS öğrenme grafiği.



**Şekil 6.4:** Aktif süspansiyon deney seti için ANFIS eğitim verisi grafiği.



**Şekil 6.5:** Aktif süspansiyon deney seti için ANFIS test verisi grafiği.

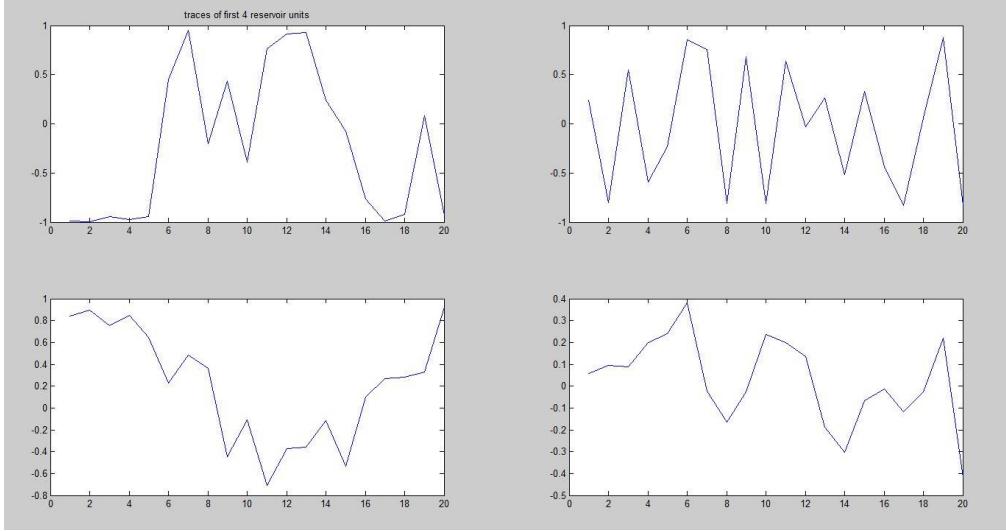
ANFIS yazılımında modelleme yapılırken devir sayısı 250, giriş üyelik fonksiyonlarının sayısı üç olarak alınmıştır. Sugeno yapısına bağlı kalınmıştır; giriş üyelik fonksiyonları üçgen, çıkış üyelik fonksiyonları ise sabittir.

Benzetim sonunda hatanın değeri 0.8122 olarak ölçülmüştür. . ANFIS yazılımında toplam veri, eğitim verileri %80 oranında, test verileri %20 oranında olacak şekilde paylaşılmıştır.

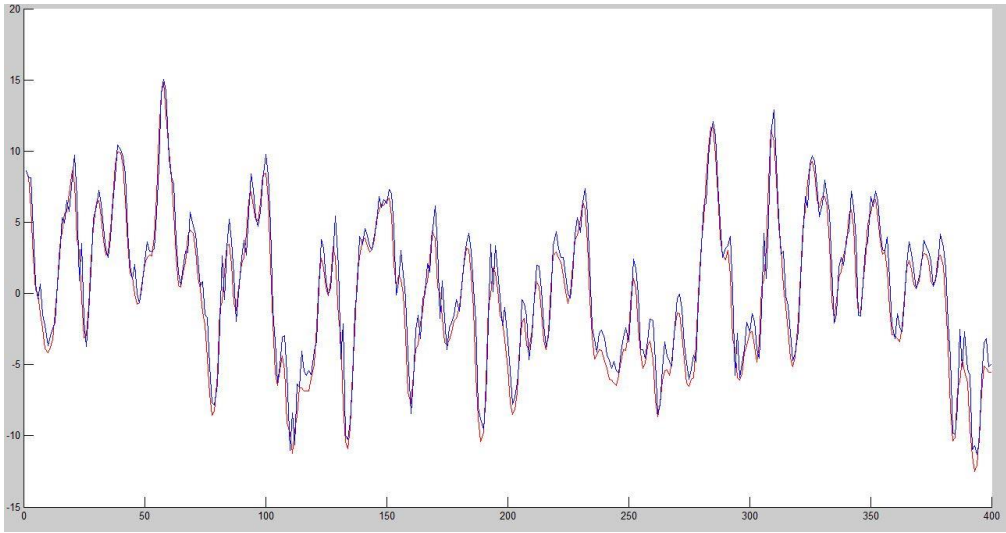
Bu adımda ANFIS yazılımında modelleme yapılırken, devir sayısı ne kadar fazla olursa olsun hata anlamında başarılı sonuçlar alınmadığı görülmüştür. ANFIS, modellemede kullanılan bulanık parametreleri iyileştirmeyip, bu parametreleri ezberlemektedir.

## 6.2 ESN ile Oluşturulan Bulanık Model

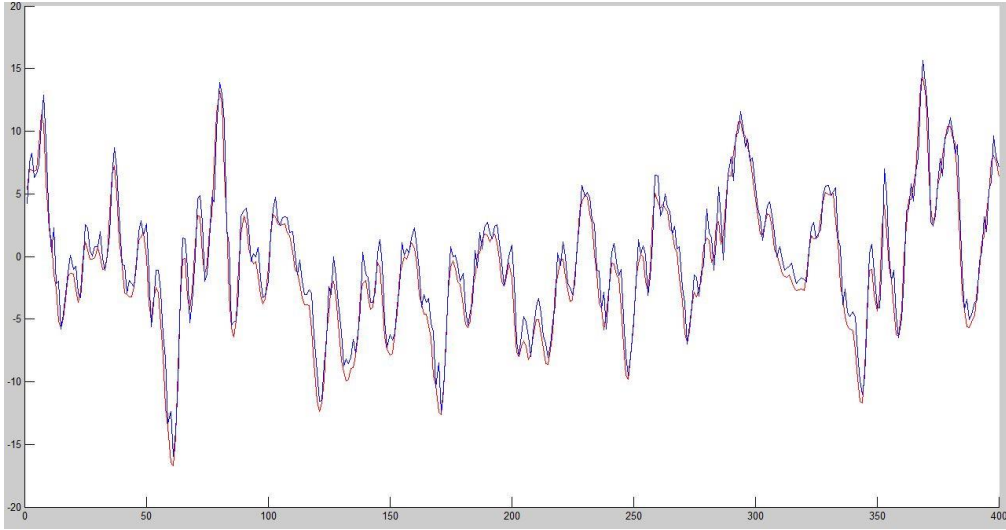
ESN ile oluşturulan modelde ilk dört rezervuar biriminin durumları Şekil 6.6'da verildiği gibidir. Eğitim verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik Şekil 6.7'de verilmektedir. Test verisi ile sistem çıkışının test edilmesi sonucu elde edilen grafik Şekil 6.8'de görülmektedir.



Şekil 6.6: Aktif süspansiyon deney seti için ESN rezervuar durumları.



Şekil 6.7: Aktif süspansiyon deney seti için ESN eğitim verisi grafiği.



**Şekil 6.8:** Aktif süspansiyon deney seti için ESN test verisi grafiği.

İç katmanda 30 birimlik bir rezervuar yapısı kullanılmış olup, spektral yarıçapı 0.6 birimdir. Eğitim verileri %50 oranı ile test verileri %50 oranı ile paylaşılmıştır. Yapılan çalışmalarda görülmüştür ki ESN yapısı ile oluşturulan modeller ANFIS yazılımı ile oluşturulan modellere göre daha başarılı sonuçlar vermektedir. ANFIS yazılımına kıyasla benzetim çalışmalarında ESN yapısında daha hızlı ve başarılı sonuçlar alınmaktadır. Ayrıca çok sayıda üyelik fonksiyonu kullanmak istenildiğinde ANFIS yazılımında sorun yaşanırken, ESN yapısı ile istenildiği kadar büyük bir ağ oluşturulabildiği ve sorunsuz kullanılabilirdiği görülmüştür.

**Çizelge 6.1:** Aktif süspansiyon deney seti için hata analizi.

	Eğitim Hatası	Test Hatası
ESN	0.26716	0.25264
ANFIS	0.81224	0.81218



## 7. SONUÇLAR

Biyolojik – tekrarlı yapay sinir ağları, tekrarlı yapay sinir ağları ile çok yönlü ve etkili bir öğrenmenin mümkün olduğuna bir kanıt oluşturmaktadır. Tekrarlı yapay sinir ağlarının özelleşmiş bir hali olan echo state network yapısı, pek çok farklı görev için aynı kolaylık ve hassasiyet ile eğitilirler. Her görev için neredeyse aynı yapı çalışır.

Echo state yapısı ile ilgili en faydalı temel olgulardan biri ise, bellek kapasitesi için nicel bir ölçünün tanımlanmış olması ve ağ birimleri ile sınırlandırılmış olmasıdır.

Eğitim için temel hesaplar, “normalize edilmiş hatanın karesel ortalaması” ile çevrimdışı olarak yapılmıştır. Adaptif sinyal işleme gibi alanlarda benzer hesaplamalar için çok sayıda çevrim içi yöntem geliştirilmiştir fakat bu alanda da echo state network yapısının verimli olarak kullanılması mümkündür.

Bu yapıya karşıt görüş olarak “birim israfı” olduğu söylenebilir. Çok sayıda birim kullanılarak oluşturulmuş olan bir echo state ile çözülen bir problem, çok daha az sayıda birim içeren standart bir tekrarlı yapay sinir ağı yapısı ile de çözülebilir. Bu sebeple bu görüşe kısmen onay vermek mümkündür. Fakat bu karşıt görüşe gösterilecek en uygun kanıt, bir görev için oluşturulmuş bir echo state yapısının çok farklı alanlardaki farklı görevler için de iş görür olduğudur. Her görev için sadece istenen sayıda giriş ve çıkış birimi eklenmesi ve uygun olan önemli küresel ölçekleme değerlerini belirlemek gereklidir. Bu durum aslında biyolojik açıdan çok uygundur çünkü beynin belirli bir bölgesi birden fazla görev yapabilmektedir.

Yapılan tez çalışmasında popüler modelleme yazılımı olan ANFIS ile ESN yapısı ile yapılan modellemelerin başarımları analizleri yapılmıştır. ESN ile modellemede başarılı sonuçlar alındığı ve ANFIS ile karşılaştırılan devir sayısı fazlalığında veya giriş üyelik fonksiyonlarının sayısı arttırıldığında hata oluşması gibi istenmeyen durumların kesinlikle oluşmadığı görülmektedir.

Echo state network yapısını eğitmek, hem çok yönlü hem de kesin sonuçlar vermektedir. İstenen çıkış ve verilen giriş değerleri gerçek sistemi oluşturur, yani yapılan işlem, çözüme ulaşmak için kullandığı aracı otomatik olarak düzenler.

Burada esas güzel olan şey kusursuz öğrenme sonuçları değil, bu sonuçların kararlılığıdır.

Bu alanda hala cevapsız kalmış olan bazı sorular (hangi koşullar altında kararlı kalınacağı, gürültü eklemenin kararlılığı nasıl etkilediği vb.) vardır. Yaklaşımın sadeliği nedeniyle bu alanda epey hızlı bir ilerlemenin olacağı öngörülebilir.

## KAYNAKLAR

- [1] **Babuska R.**, (1998). *Fuzzy Modeling and Control*, Kluwer Academic Publishers, Massachusetts, ABD.
- [2] **Deihimi, A., Solat, A.**, (2014). Optimized echo state networks using a big bang–big crunch algorithm for distance protection of series-compensated transmission lines, *Electrical Power and Energy Systems*, **54**, 408 – 424.
- [3] **Espinosa J., Vandewalle J., Wertz V.**, (2005). *Fuzzy Logic, Identification and Predictive Control*, Springer, ABD.
- [4] **Elmas, Ç.**, (2003). *Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)*, Seçkin Yayınları, Ankara.
- [5] **Grossberg, S.** (2013). Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing World, *Neural Networks*, **37**, 1- 47.
- [6] **Haykin, S.** (1994). *Neural networks*. New York: MacMillan.
- [7] **Hinton, G. E., Sejnowski, T. J.** (1983). Optimal Perceptual Inference, *Proceedings of the IEEE*.
- [8] **Jaeger, H.**, Advanced Machine Learning Lecture Notes.
- [9] **Jaeger, H.**, (2005). A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, *German National Research Center for Information Technology*.
- [10] **Jaeger, H.**, (2002). Short Term Memory in Echo State Networks, *GMD Report*.
- [11] **Jaeger, H.**, (2001). The “echo state” approach to analysing and training recurrent neural networks, GMD Report 148, *German National Research Center for Information Technology*, 43.
- [12] **Kohonen, T.**, (1990). The Self Organizing Map, *Proceedings of IEEE*, **78**, 9, 1464 – 1480.
- [13] **Lukosevicius, M.**, (2012). A Practical Guide to Applying Echo State Networks, *Neural Networks: Tricks of the Trade, Reloaded*, Springer.
- [14] **Mamdani E. H.**, (1974). Applications of fuzzy algorithms for control of simple dynamic plant, *Proceedings of IEEE*, **121**, 12, 1585 – 1588.
- [15] **Ni, S., Liu, L., Wang, Z., Wang, Z.**, (2011). Predictive Control of Vehicle Based on Echo State Network, *International Conference on Intelligence Science and Information Engineering*.
- [16] **Oke, G.**, (2012). *Intelligent Control Systems Lecture Notes*, İstanbul.

- [17] **Pan, Y., Wang, J.**, (2012). Model Predictive Control of Unknown Nonlinear Dynamical Systems Based on Recurrent Neural Networks, *IEEE Transactions on Industrial Electronics*, **59**, 8.
- [18] **Ross T. J.**, (1994). *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, Cornwall, Büyük Britanya.
- [19] **Rössler, O. E.** (1976), An Equation for Continuous Chaos, *Physics Letters*,**57**, 397–398.
- [20] **Rumelhart, D. E., McClelland, J. L.**,(1980) An Interactive Activation Model of the Effect of Context in Perception. Part II. Report No. 8003.
- [21] **Self K.**, (1990). Designing with fuzzy logic, *IEEE Spectrum*, **11**, 42-44.
- [22] **Şen, Z.**, (2004). Yapay Sinir Ağları İlkeleri, Su Vakfı Yayınları, İstanbul.
- [23] **Takagi T., Sugeno M.**, (1985). Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics*, **15**, s. 116-132.
- [24] **Widrow, B., Stearns,S.D.**(1985).Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey.
- [25] **Xing, K., Wang, Y., Zhu, Q., Zhou, H.**, (2012). Modeling and Control of McKibben Artificial Muscle Enhanced with Echo State Network, *Control Engineering Practice*, **20**, 477 – 488.
- [26] **Xu, D., Lan, J., Principe, J.**, (2005). Direct Adaptive Control: An Echo State Network and Genetic Algorithm Approach, *International Joint Conference on Neural Networks*.
- [27] **Yıldız, I., Jaeger, H., Kiebel, S.**, (2012). Re-visiting Echo State Property, *Neural Networks*, **35**, 1-9.
- [28] **Zhang H., Liu D.**, (2006). *Fuzzy Modeling and Fuzzy Control*, Birkhäuser, Boston, ABD.

**Url-1** <<http://reservoir-computing.org/node/129>>, alındığı tarih: 11.03.2014.

## ÖZGEÇMİŞ

**Ad Soyad:** Selin Yaman  
**Doğum Yeri ve Tarihi:** Bursa / 30 Ağustos 1990  
**Adres:** Kağıthane / İSTANBUL  
**E-Posta:** yamansel@itu.edu.tr  
**Lisans:** İstanbul Teknik Üniversitesi Elektrik Elektronik  
Fakültesi  
Kontrol ve Otomasyon Mühendisliği Bölümü  
Kontrol ve Otomasyon Programı