T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY ve BULANIK SİNİR AĞLARI İLE SİSTEMLERİN MODELLENMESİ

Tezi Hazırlayan
Ali DURMUŞ

Tez Yöneticisi
Yrd. Doç. Dr. Nurhan KARABOĞA

Elektrik-Elektronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Temmuz 2005
KAYSERİ

Yrd. Doç. Dr. Nurhan KARABOĞA danışmanlığında, **Ali DURMUŞ** tarafından hazırlanan **"Yapay ve Bulamık Sinir Ağı ile Sistemlerin Modellenmesi"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

…../…/……

JÜRİ:

Başkan :……

Üye :……

Üye :……

ONAY

Bu tezin kabulü, Enstitü Yönetim Kurulunun ....…….. tarih ve ……

sayılı kararı ile onaylanmıştır.

…../…/……

Enstitü Müdürü

TEŞEKKÜR

Tez çalışmasının danışmanlığını yürüten ve yüksek lisans çalışmalarım boyunca, benden bilgilerini ve yardımlarını esirgemeyen saygıdeğer hocam, Yrd. Doç. Dr. Nurhan KARABOĞA'ya teşekkürü bir borç bilirim. Öğrenim hayatım boyunca bana maddi ve manevi destek olan çok kıymetli aileme ve bana her türlü yardımı esirgemeyen arkadaşlarıma sonsuz teşekkürlerimi sunuyorum.

BULANIK SİNİR AĞI İLE SİSTEMLERİN MODELLENMESİ

Ali DURMUŞ

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Temmuz 2005
Tez Danışmanı: Yrd.Doç. Dr. Nurhan KARABOĞA

ÖZET

Bu tez çalışmasında, yapay sinir ağları ve bulanık mantığın birleşiminden oluşan bulanık sinir ağı sistem modelleme amacıyla doğrusal sistemlerle, doğrusal olmayan sistemlere uygulanmıştır.

İkinci bölümde sistem tanımı, model tanımı, modelleme tipleri, matematiksel model yapıları hakkında genel bilgi verilmiştir. Doğrusal ile doğrusal olmayan sistemler incelenmiştir. Sistemlerin modellenmesinde kullanılan Least Mean Square (LMS) ve Recursive Least Square (RLS) yöntemleri hakkında bilgi verilmiştir.

Üçüncü bölümde sistemlerin modellenmesinde kullanılan yapay zeka tekniklerinden yapay sinir ağları konusunda bilgi verilmiş. Temel ağ yapıları ve öğrenme yöntemlerinin tanıtımı yapılmıştır. Ayrıca bu bölümde YSA'nın sistem modellemeye nasıl uygulanacağı anlatılmıştır.

Dördüncü bölümde, bulanık mantık ile bulanık sinir ağları hakkında bilgi verilmiş, bulanık küme, üyelik fonksiyonları, bulanık çıkarım sistemlerinin tanıtımı yapılmıştır. Ayrıca bulanık sinir ağı yapısı ve bu yapının sistem modellemeye uygulanması anlatılmıştır.

Beşinci bölümde, sistemler yapay sinir ağları ve bulanık sinir ağı ile simüle edilmiş ve bunların performanslarının karşılaştırılması incelenmiştir.

Son bölümde sonuçlar ve değerlendirmeye yer verilmiştir.

Anahtar Kelimeler: Sistem Modelleme, yapay sinir ağları, bulanık mantık, bulanık sinir ağı.

SYSTEM MODELLING WİTH NEURAL NETWORK AND NEURO-FUZZY METHODS

Ali DURMUŞ
Erciyes University, Graduate School of Natural and Applied Sciences
M.S. Thesis, July 2005
Thesis Supervisor: Assist. Prof. Nurhan KARABOĞA

ABSTRACT

In this thesis, the neuro-fuzzy network which is the combination of neural network and fuzzy logic, has been applied to the linear systems and nonlinear systems for system modelling.

In Section 2, general information is given about system definition, model identification, modelling types and mathematical model structures. Linear and nonlinear systems are examined. Basic information is given for the Least Mean Square (LMS) and Recursive Least Square (RLS), methods used for system modelling.

In Section 3, the artificial nueral network (ANN), which is one of the artificial intelligence techniques and used system modelling is explained. Basic network and training structures are given. Also in this section, the application of the ANN to the system modelling has been presented.

In Section 4, fuzzy logic and neuro-fuzzy networks are explained and fuzzy set, membership function and fuzzy inference systems are introduced. Also, the structure of Neuro-fuzzy systems and application of this structure to the system modelling has been given.

In Section 5, linear and nonlinear systems have been simulated by using ANN and Neuro-fuzzy network and their performances have been compared.

The last section, simulation results are given.

Keywords: System modeling, artificial neural networks, fuzzy logic, Neuro-fuzzy network.

İÇİNDEKİLER

KABUL ve ONAY….....I
TEŞEKKÜR.....II
ÖZET…….…….III
ABSTRACT…….…..….IV
KISALTMALAR ve SİMGELER……..... IX
TABLOLAR LİSTESİ…….....…….X
ŞEKİLLER LİSTESİ…….....…...XI

BÖLÜM 1

GİRİŞ…….....1

BÖLÜM 2

SİSTEM MODELLEME YAPILARI ……..……..…….....3
2.1. Giriş……..…….....3
2.2. Sistemlerin Yapısı……..3
2.3. Modelleme İşleminin Yöntemleri……..4
2.3.1. Sistem Modellemenin Temel Basamakları….....5
2.4. Model Çeşitleri…….....…….6
2.4.1. Modellerin Sınıflandırılması ….....……..6
2.5. Adaptif Sistemler ……..…….....7
2.6. Doğrusal Sistem Modelleme…….....8
2.6.1. AR Modelleme.....……..8
2.6.2. MA modelleme……..….....….....9
2.6.3. ARMA Modelleme……..9
2.7. Doğrusal Olmayan Sistem Modelleme..…….....……...10
2.7.1. Bilineer Model……..…….....10
2.8. Sistem Modelleme için Kullanılan Yöntemler….....11
2.8.1. LMS Algoritması…….....…….....…….....11
2.8.2. RLS.....…….....……..13

BÖLÜM 3

YAPAY SİNİR AĞLARI…….....16
3.1. Yapay Sinir Ağlarının Genel Özellikleri…….....…….....19
3.2. Transfer Fonksiyonları……..…….....…….....20
3.3. YSA'ların Özellikleri…….....…….....22
3.3.1. Doğrusal Olmama.…….....……..22

3.3.2. Öğrenme……..…….....22
3.3.3. Genelleme Yapabilme….....…….22
3.3.4. Adaptasyon……..…….....22
3.3.5. Donanım Olarak Gerçekleştirme….....….....23
3.3.6. Hataya Karşı Duyarlılık…….....23
3.4. YSA'ların Sınıflandırılması…….....…….23
3.4.1. YSA'ların Ağ Yapılarına Göre Sınıflandırılması.....……23
3.4.1.1. İleri Beslemeli Ağ Yapısı.....……...24
3.4.1.2. Geri Beslemeli Ağ Yapısı.....…….....24
3.4.2. YSA Yapıları …….....25
3.4.2.1. ÇKP YSA Yapısı……... …….....…….25
3.4.3. YSA'ların Öğrenme Algoritmalarına……..……..26
3.4.3.1. Geri Yayılım (BP)…….....…….26
3.4.3.2. Momentulu Geri Yayılım (BPM)..…….....……..27
3.4.3.3. Esnek Geri Yayılım (BPR) …….....……..27
3.4.3.4. Azaltmalı Geri Yayılım……..28
3.4.3.5. Quasi Newton…….28
3.4.3.6. Delta-Bar-Delta……..29
3.4.3.7. Genişletilmiş Delta-Bar-Delta…….....30
3.4.3.8. Levenberg Marquardt (LM)…….....30
3.4.3.9. Bayessian Düzenleştirme (BD)……...31
3.5. Sistem Modellemede YSA Kullanımı……..…….....31

BÖLÜM 4

BULANIK SİNİR AĞLARI……..…33
4.1. Bulanık Mantık……..…….....…….....33
4.1.1. Bulanık Mantık Gelişim Süreci…….....…….....33
4.1.2. Klasik ve Bulanık Mantık Farkı …….....34
4.1.3. Bulanık Mantık Uygulama Alanları …….....35
4.2. Bulanık Küme Teorisi…….....35
4.2.1. Klasik ve Bulanık Kümeler.....…….....38
4.2.2. Üyelik Fonksiyonu…….....38
4.2.3. Bulanık Kümelerde Temel İşlemler.....…….....40
4.2.3.1. Kapsama…….41
4.2.3.2. Denklik…….....41
4.2.3.3. Kesişim…….....41
4.2.3.4. Birleşim…….....41
4.2.3.5. Tümleyen……42
4.2.3.6. Cebirsel Toplam…….....42

4.2.3.7. Cebirsel Çarpım…….....42
4.2.3.8. Fark…….....42
4.3. Bulanık Sistemler için Bulanık Kurallar.....…….....43
4.4. Bulanık Çıkarım Sistemi…….....…….……..45
4.4.1. Sugeno Bulanık Çıkarım Metodu…….....45
4.4.1.1. Sugeno Tipi Modelin Avantajları……...47
4.5. Bulanık Sistem Geliştirme Aşamaları…….....47
4.6. Bulanık Sinir Ağı…….....48
4.7. Bulanık Sinir Ağı İle Sistem Modelleme…….....50

BÖLÜM 5

UYGULAMALAR…….52
5.1. YSA ile Sistem Modelleme…….52
5.1.1. YSA ile Doğrusal Sistem Modelleme……...52
5.1.2. YSA ile Doğrusal Olmayan Sistem Modelleme……...57
5.2. Bulanık Sinir Ağı ile Sistem Modelleme…….....60
5.2.1. Bulanık Sinir Ağı ile Doğrusal Sistem Modelleme…….....60
5.2.2. Bulanık Sinir Ağı ile Doğrusal Olmayan Sistem Modelleme……..68

BÖLÜM 6

6. SONUÇ ve ÖNERİLER…….....…….....72

KAYNAKLAR…….....…….74

EKLER……...79
EK-1……....79
EK-2……...82

ÖZGEÇMİŞ……...85

KISALTMALAR VE SİMGELER

AR	Autoregressive
MA	Moving Average
ARMA	Autoregressive Moving Average
LMS	Least Mean Square-Anlamlı En küçük Kareler
RLS	Recursive Least Square-Öziyneli En Küçük Kareler
PAR	Polynomial Auoregressive
LM	Levenberg-Marquart
MLP	Muifi Layer Perceptron
YSA	Yapay Sinir Ağları
BFGS	Broyden, Fletcher, Goldfarb ve Shanno algoritması
GD	Geri-yayılım
MSE	Mean Square Error
RMS	Root Mean Square
MGS	Mackey-Glass Serisi
ANFIS	Adaptive Neuro Fuzzy Inference System

TABLOLAR LİSTESİ

Sayfa No

Tablo 3.1. Biyolojik sinir sistemi ile YSA'ların karşılaştırılması 18

Tablo 3.2. Bazı transfer fonksiyonlarının matematik ifadeleri 21

Tablo 4.1. Bulanık mantık ve YSA açısından sistem modelleme adımları 50

Tablo 5.1. Gürültüsüz durumda farklı öğrenme algoritmaları ile eğitilmiş YSA'ların modellemedeki performansları 53

Tablo 5.2. Gürültülü durumda farklı öğrenme algoritmaları ile eğitilmiş YSA'ların modellemedeki performansları 54

Tablo 5.3. 1. Sistem için farklı öğrenme algoritmalarının hata değerleri 57

Tablo 5.4. Değişik Gürültü Değerlerine Göre Sistemlerin MSE ve RMS Hataları 59

Tablo 5.5. Doğrusal Sistemlerin Farklı Üyelik Fonksiyonlarına Verdiği MSE Hata Değeri 65

Tablo 5.6. AR Sistemin YSA ve Bulanık Sinir Ağına Göre Hata değerleri 66

Tablo 5.7. ARMA Sistemin YSA ve Bulanık Sinir Ağına Göre Hata değerleri 67

Tablo 5.8. Bilineer Sistemin YSA ve Bulanık Sinir Ağına Göre Hata değerleri 68

ŞEKİLLER LİSTESİ

Şekil 2.l.	Bir dinamik sistem seması	3
Şekil 2.2.	Genel sistem modelleme yapısı	4
Şekil 2.3.	Adaptif sistem modelleme	8
Şekil 3.l.	Biyolojik nöron yapısı	17
Şekil 3.2.	Yapay nöronun işlem yapısı	20
Şekil 3.3.	YSA'larda en çok kullanılmakta olan transfer fonksiyonları	21
Şekil 3.4.	İleri beslemeli YSA yapısı	24
Şekil 3.5.	Geri beslemeli YSA yapısı	25
Şekil 3.6.	Genel bir ÇKP yapısı	26
Şekil 4.1.	Klasik ve bulanık mantık farkı	34
Şekil 4.2.	*A* kümesinin üyelik fonksiyonu	39
Şekil 4.3.	Çeşitli üyelik fonksiyonu biçimleri	40
Şekil 4.4.	Bulanık çıkarım diyagramı	45
Şekil 4.5.	Sugeno bulanık çıkarım metodu	47
Şekil 4.6.	Basit bir ANFIS yapısı	50
Şekil 5.1.	AR ve ARMA sistemler için LM eğitim algoritması ile eğitim esnasında oluşan hata değerleri	54
Şekil 5.2.	AR ve ARMA sistemlerin sistem çıkışı ile model çıkışı arasındaki fark	55
Şekil 5.3.	AR sisteme ait çıkış ve hata şekilleri	55
Şekil 5.4.	ARMA sisteme ait çıkış ve hata şekilleri	56
Şekil 5.5.	YSA ile modellenmiş Bilineer sisteme ait çıkış şekilleri	58
Şekil 5.6.	Bulanık sinir ağı ile modellenmiş AR sistemin çıkış şekilleri	61
Şekil 5.7.	Bulanık sinir ağı ile modellenmiş MA sistemin çıkış şekilleri	62
Şekil 5.8.	Bulanık sinir ağı ile modellenmiş ARMA sistemin çıkış şekilleri	64
Şekil 5.9.	2. AR sisteme ait çıkış ve hata şekilleri	66
Şekil 5.10.	2. ARMA sisteme ait çıkış ve hata şekilleri	67
Şekil 5.11.	Bulanık sinir ağı ile modellenmiş Bilineer sistemin çıkış şekilleri	68
Şekil 5.12.	Bulanık sinir ağı ile modellenmiş MG sistemin çıkış şekilleri	71

BÖLÜM 1

GİRİŞ

Sistemlerin matematik modellerinin önemi bugün tüm alanlarda hızla artmaktadır. Özellikle üretim alanında sistemin verimli, hızlı ve kaliteli olabilmesi için bu sistemin modellenmesi gerekmektedir. Böylece sistemin ileriki dinamik davranışı tahmin edilebilir ve üretimin istenilen seviyede olması sağlanır. Sistem modellemede amaç bilinmeyen bir sistemin transfer fonksiyonu adı verilen geçiş eğrisinin belirlenmesidir. Modelleme işlemi, biyoloji, bankacılık ve ekonomi gibi teknik olmayan alanlarda da kullanılmaktadır. Matematik modeli verilen fiziksel bir sistemin analizi yapılabilir veya değişik girişlere karşı çıkışları tahmin edilebilir. Sistem, girişine uygulanan işareti değiştirerek, çıkış elde eden elemanlar topluluğu olarak tanımlanır [1].

Eğer bir sistemin fiziksel kuralları biliniyorsa, bu kuralları kullanarak elde edilen modele "white-box" (beyaz kutu) modeli denir. Bu durumun dışında kalan modellemeye ise "black-box"(kara kutu) modelleme denir [2]. Sistem modelleme kavramı ilk kez 1962'de Lotfi Zadeh tarafından kullanılmıştır. Zadeh bu kavramı "Bir sistemi giriş ve çıkış ölçümlerinden belirlemek" şeklinde tanımlamıştır [3].

Doğrusal sistemlerin modellenmesinde genellikle, azaltılmış dereceli modeller kullanılarak yaklaşım sağlanmaktadır. Bununla beraber, bu modeller, belirli özel çalışma bölgelerinde geçerli olmaktadır. Çalışma durumu değiştiğinde, başka bir modele ihtiyaç duyulmaktadır veya modelin parametrelerinin değişmesi gerekmektedir. Sistemlerin modellenmesi için kullanılan klasik algoritmalar LMS- En Küçük Ortalama Kareler (Least Mean Sqaure) ve RLS- Özyineli En Küçük Kareler (Recursive Least Sqaure) yöntemleri olarak iki ana sınıfta gruplandırılar. Bu modelleme teknikleri yapay zeka tekniklerine göre daha karmaşıktır ve matematik işlemler daha fazladır [3,4].

Endüstriyel uygulamaların büyük çoğunluğu, doğrusal olmayan dinamik sistemlerdir. Bu nedenle, böyle sistemlerin modellenmesi fazla karmaşık olmaktadır. Bazı araştırmacılar, doğrusal olmayan dinamik sistemlerin modellenmesi işleminde, yapay sinir ağlarını (YSA) kullanarak iyi bir yaklaşım sağlamışlardır. Fakat, yapay sinir ağının eğitilmesi ile elde edilen kara-kutu modeli sonuçlarının yorumlanması çok zor olmaktadır. Buna ek olarak, sınırlı veya gürültülü eğitim bilgisi kullanıldığında sistem çıkışında tutarsız veya anlamsız bilgi alınmaktadır [5].

YSA bir çok alanda kullanılmaktadır. Özellikle matematik modelleri oluşturulamayan veya matematik modelleri çok zor tanımlanabilen problemlerin çözümü için kullanılmaktadır [6-31]. Giriş ve çıkış eğitim çiftleri verilen bir fonksiyonu, yapay sinir ağı öğrenebilmektedir. Öğrenme işlemi, ağ içerisindeki ağırlıkların belirlenmesiyle gerçekleşmekte ve verilen fonksiyona optimal yaklaşım sağlanmaktadır. Böylece yapay sinir ağları sistemi öğrenmektedir.

Bu tez çalışmasında bulanık mantığın belirsiz bilgileri işleme ve yapay sinir ağının öğrenme yeteneğinden yararlanılarak sistem modelleme gerçekleştirilmiştir. Aynı zamanda bulanık mantık ve yapay sinir ağının birleştirilmesi, birbirlerinin dezavantajlarını da örtmektedir. Bulanık sinir ağı değişik birçok sistemin modellenmesine uygulanmıştır [31-60].

Bu çalışmada hem doğrusal hemde doğrusal olmayan sistemler incelenmiştir. İlk olarak doğrusal sistemlerden olan zamanla değişen AR ve ARMA sistemler yapay sinir ağı ile modellenmiştir. Modellenme işlemi için farklı öğrenme algoritmaları kullanılmış ve bu algoritmaların performansları birbiriyle mukayese edilmiştir. İkinci olarak da doğrusal olmayan sistemlerden olan bilineer (çift doğrusal) sistem YSA ile modellenmiş ve farklı öğrenme algoritmaları ile eğitilmiştir. Son olarak da her iki sisteme de gürültü eklenmiş ve bunun sonucu incelenmiştir.

Uygulamaların ikinci bölümünde doğrusal sistemler olan AR, MA ve ARMA sistemleri ve doğrusal olmayan sistemler olan bilineer ve Mackey-Glass zaman gecikmeli sistemin bulanık sinir ağı ile modellenmesi sonucu elde edilmiş simülasyon sonuçları verilmiştir.

BÖLÜM 2

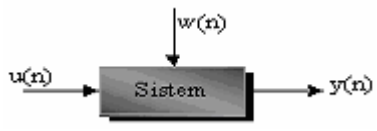
SİSTEM MODELLEME

2.1. Giriş

Dış dünya ve kendi iç elemanları arasındaki ilişkilerin belirlenebildiği yapılara sistem denir. Sistemin basitleştirilmiş, genellikle matematik veya hesaplanabilir yapıdaki gösterimine model adı verilir. Model, herhangi bir olayı zaman ve mekan kısıtlaması olmaksızın incelemek için kurulan bir düzendir. Fiziksel olayların modelini kurmak ise gerçekte, bu olayları, bir takım matematik ifadelere dönüştürmek demektir. Sistem modelleme, deney yoluyla elde edilmiş verilerden faydalanarak dinamik sistemlerin modelinin elde edilmesidir. Bu model yardımıyla, sistemin ileri aşamalardaki çıkış işareti kestirilebilir, farklı alanlarda karşılaşılan problemlerin çözülmesinde oldukça yararlıdır. Bir sistemin matematik veya bilgisayar tarafından modelinin oluşturulması ve bu modelleri kullanarak sistemin özellikleri konusunda çalışılmasına simülasyon adı verilir [1-8].

2.2. Sistemlerin Yapısı

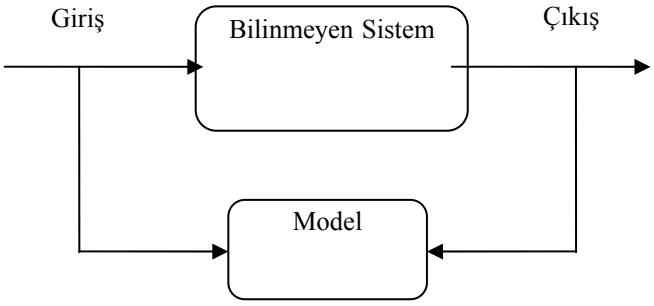
Şekil 2.1 'de bir dinamik sistemin temel şeması gösterilmiştir. Sistem, u(n) giriş işareti ve w(n) bozucu işaretinden etkilenerek y(n) çıkış işareti üretir.



Şekil 2.1. Dinamik Sistem Şeması

Herhangi bir sistemin modelinin elde edilmesi işlemi, u(n) ve y(n) işaretlerinden faydalanılarak, model çıkış işareti elde etmektir. Genel sistem modelleme yapısı Şekil 2.2' de verilmiştir. Eğer sistemlerde giriş işareti bilinmiyorsa, çıkış işareti kullanılarak modelleme yapılabilir. Bir çok endüstriyel alanda, örneğin kağıt, demir veya cam üretiminde, sistemin

güvenli, verimli ve istenilen kalitede üretim yapabilmesi için kontrol düzeneğinin kurulması gerekir. Kontrol edilecek sistemin matematik modelinin bilinmesi zorunludur. Böylece sistemin modeli kullanılarak sistemin çıkışı, arzu edilen şekilde tasarlanabilir.



Şekil 2.2. Genel Sistem Modelleme Yapısı

İşaret işlemenin bir çok uygulamasında, tahmin, veri haberleşmesi, ses işaretleri işleme, radar, sonar ve elektrokardiyogram analizi gibi, ele alınan veriler bazı yollarla ve yüksek geçiren, alçak geçiren, rezonans frekansının durumu gibi işaretlerin özelliklerini yansıtabilen iyi tasarlanmış süzgeç yapıları ile filtrelenir. Spektral özelliklerinin belirlenmesi için işaretin modeline ihtiyaç duyulur. Bir çok durumda, modellemenin başlıca amacı, tasarıma yardımcı olabilmektir. Ayrıca birçok durumda modellemenin amacı sistem kazancı ve sistemin dinamik davranışlarını bilmektir [8].

2.3. Modelleme İşleminin Yöntemleri

Dinamik sistemlerin modelleri birçok farklı ve uygulamalarda kullanılmaktadır. Model oluşturmanın iki yolu vardır:

l. Matematik Modelleme: analitik bir yaklaşımdır. Bir olayın veya bir işlemin davranışını tanımlamada örneğin Newton kanunları fiziğin temel kurallarını kullanırlar.

2. Sistem Modelleme: deneye dayalı bir yaklaşımdır. Sisteme birçok deney uygulanır, sonrasında model, kaydedilen verilerle parametreler yardımıyla uygunlaştırılır [8].

Bu iki model oluşturma metodu karşılaştırılacak olursa, sadece fiziksel kavramlar kullanılarak uygun model elde etmenin mümkün olmadığı birçok durumda bu işlemler çok karmaşıktır. Bazı durumlarda, matematik modellemenin modelleme amaçlı kullanımı zordur. Bir çok model yapısı, fiziksel kanunlardan elde edilmiş dahi olsa, bilinmeyen birçok parametre değerine bağlıdır. Modelleme de amaç bilinmeyen

parametreleri tahmin etme uygulamasındır. Sistem modelleme ile elde edilen modeller, matematik modellemeden farklı olarak aşağıda sıralanan özelliklere sahiptir;

• Sistem modelleme metodu matematik modellemeye göre daha kolay uygulanır ve daha kolay kullanılır.

• Sistem modelleme metodunda çok küçük fiziksel kavramları ifade ederler, çünkü bir çok durumda model parametreleri fiziksel bir anlam ifade etmemektedir. Parametreler, sistemin davranışını iyi bir şekilde tanımlamak için bir araç olarak kullanılır.

Modelleme, modelleme yapan kişi tarafından çok fazla etkilenmektedir. Bunun modelleme yapısına etkileri aşağıda sıralanmıştır.

• Modelleme yapılırken, uygun bir model yapısı bulunmak zorundadır. Eğer sistem dinamikleri doğrusal olmayan yapıda ise bu durum oldukça zorlaşır.

• Gerçek hayattaki veriler kesinlikle mükemmel değildir. Gürültü tarafından bozulan veriler dikkate alınmak zorundadır.

• İşlemler zamanla değişmektedir. Bunun sonucu olarak da, modelleri zamanla değişmeyen yapıda tanımlamak gerekmektedir.

• Modelin karakteristiğini ihtiva eden bazı işaretleri/değişkenleri ölçmek zor hatta imkansızdır [8,9].

2.3.1. Sistem Modellemenin Temel Basamakları

Aşağıda genel olarak bir sistemin modellemesinin temel basamakları verilmektedir:

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

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

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

d. Parametreleri belirlenen modelin girişine, sisteme uygulanmış olan giriş işareti uygulanıp, modelden alınan çıkış işareti ile sistemin gerçek çıkışı arasındaki fark bulunur. Eğer fark büyükse, başka bir model yapısı veya yeni bir parametre tespit

yöntemi belirlenmesi için b. basamaktaki işleme geri dönülür. Eğer fark çok küçükse bu model sistemi tanımlamak ve kontrol etmek için kullanılabilir [9].

2.4. Model Çeşitleri

Literatürde bir çok model çeşitleri mevcuttur. Dinamik ve Statik model bunlara örnek olarak verilebilir. Zaman dinamik modellerin temel elemanıdır. Dinamik modellerde, statik olanların aksine, zaman, modelin temel elemanı olarak tanımlanır. Bir dinamik model, belli bir zamandaki sistemin durumu, bu andaki sistemi etkileyen dış faktörleri ve bir sonraki duruma ulaşmak için geçiş ilişkilerini içerir.

Tanımlayıcı modeller, sistemin çalışmasını, sistemin temel prensipleri kullanarak ortaya çıkarma amacıyla kullanılırlar.

Açıklayıcı modeller, sistemin iç yapısının tanımlanması yerine, sistemin karakteristiğini elde etmek için kullanılırlar.

Model parametreleri, modelin çalışması esnasında sabit kabul edilen matematik yapıda kullanılan sayılardır. Model verisi ise modelin durum değiştirmesini sağlayan giriş zaman serileridir. Bu veriler belli çalışma şartlarında modelin davranışını belirlemede kullanılırlar [9-13].

2.4.1. Modellerin Sınıflandırılması

Giriş-çıkış durumuna göre;
a) SISO (Single Input Single Output) : Tek girişi ve tek çıkışı olan modellerdir.
b) MISO (Multi Input Single Output): Çok girişi tek çıkışı olan modellerdir.

Zamana göre;
a) Zamana bağlı modeller: Sistem zamanla değişiyorsa bu tip sistemlere zamana bağlı modeller denir.
b) Zamandan bağımsız modeller: Sistemin iç yapısının ve karakteristiğinin zamana göre değişmiyorsa bu tip modellere zamandan bağımsız olarak kullanılır.

Domene göre;

a) Zaman domeni: Diferansiyel veya fark denklemleri ile sistemlerin modellenmesinde kullanılır.
b) Frekans domeni: Spektral yoğunluk veya Bode eğrisi gibi sistemin karakteristiğini frekans domeninde belirleyen modellerdir.

Doğrusallık durumuna göre;
a) Doğrusal modeller: Çıkışla, giriş ve bozucu etkiler arasındaki ilişkinin doğrusal eşitliklerle ifade edilebildiği modellerdir.
b) Doğrusal olmayan modeller: Sistemin giriş-çıkış ilişkisinin, doğrusal olmayan matematik ifadelerle sağlanabildiği modellerdir.

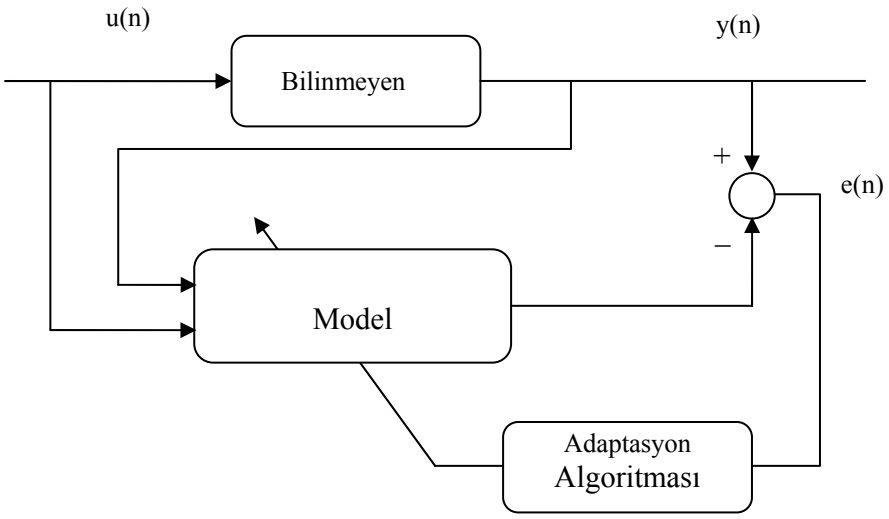
Bozucu etkilere göre;
a) Deterministik modeller: Deterministik modellerde giriş bilinirse çıkış tam olarak hesaplanabilir.
b) Stokastik modeller: Sistemlerde dış etkenlerin sonucunda oluşan rastgele terimlerin yer aldığı modellerdir [13].

2.5. Adaptif Sistemler

Adaptif sistemler değişen şartlara göre kendini en iyiye doğru kanalize eden sistemlerdir. Bu sistemler işaret işleme sahasında daha fazla kullanılmaktadır [14-16]. Adaptif sistemlerin en önemli özelliği zamanla değişen sistemlere rahatlıkla uygulanabilmesi ve yeni durumlara göre kendi kendini ayarlayabilmesidir. Doğrusal sistemlerde karşılaşılan, sadece belirli girişlere karşı çıkışın istenen şekilde olması, diğer tür girişlerin uygulandığı durumlarda veya kontrol edilen sistemin zamanla, çevre şartlarından etkilenerek özelliklerinin değişmesi durumunda sistemin kararsız davranması durumu adaptif sistemlerde daha az gözlenmektedir.

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

Adaptif sistemlerdeki temel yaklaşım, modellenecek sistem parametrelerinin elde edilmesi aşamasında, her bir iterasyon sonucunda oluşan hata değerinin minimize edilmesi için sistem parametrelerini belirli bir şekilde değiştirmektir. Hata değerinin minimuma indirilmesi için genellikle sistemin amaç fonksiyonunun (hatayı veren fonksiyonun) türevi kullanılır. Bu türev değerini her iterasyonda sıfır yapan parametre değerleri adaptif olarak bulunur. Sistemin çıkışının arzu edilen şekilde olması için gereken adaptif kontrol sistemi tasarımı Şekil 2.3'de gösterilmiştir.



Şekil 2.3 Adaptif Sistem Modelleme

Bu şekilde istenilen ile elde edilen çıkış arasındaki fark hata adını almakta ve adaptif sisteme giriş olarak verilmektedir. Bu farkın sıfır olması, sistemin arzu edilen şekilde çalışması anlamına gelmektedir. Adaptif modelleme, sistem parametrelerini, hatayı sıfır yapacak şekilde ayarlamak için kullanılır [9-13].

2. 6. Doğrusal Sistem Modelleme

Doğrusal sistem modellemede autoregressive (AR), moving average (MA) ve autoregressive moving average (ARMA) modelleri kullanılır.

2. 6.1. AR (Autoregressive-Özbağlanımlı) Modelleme

Bir çok ayrık zamanlı sistemlerde, veri olarak yalnızca çıkış değerlerinin yardımıyla sistemin modellenmesi gerekir. Bu tür sistemler özbağlanımlı olarak modellenebilir. Sadece kutuplara sahip olan AR modeli;

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) - .. - a_p y(n-p) + e(n) = -\sum_{k=1}^{p} a_k y(n-k) + e(n) \quad (2.1)$$

şeklindedir. Eşitlik 2.1'de y(n) çıkış dizisini, e(n) gürültü dizisini, a(k) ise AR parametrelerini belirtmektedir. AR modellerde, sistem parametrelerinin ayarlanması, sistemden elde edilen çıkış işaretinin eski değerlerini, bazı katsayılarla çarparak çıkışın istenilen özellikte olması işlemidir [7].

2.6.2. MA (Moving Average-Yürüyen Ortalamalı) Modelleme

MA model yapısı kullanılarak yapılan modellemede veri olarak yalnızca giriş işareti değerleri kullanılır. Sadece sıfırlara sahip olan bu yapı;

$$y(n) = b_0 x(n) + b_1 x(n-1) + ...b_q x(n-q) + e(n) = \sum_{k=1}^{q} b_k x(n-k) + e(n) \quad (2.2)$$

şeklindedir. Burada y(n) çıkış dizisini, x(n) giriş dizisini ve b_k ise MA parametrelerini belirtir. MA modellerde, sistem parametrelerinin ayarlanması, sisteme uygulanan giriş işaretinin o andaki ve geçmiş değerlerini bazı katsayılarla çarparak çıkışın istenilen özellikte olması işlemidir [7].

2.6.3. ARMA (Autoregressive Moving Average-Özbağlanımlı Yürüyen Ortalama) Modelleme

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

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

şeklinde doğrusal fark denklemi yazılabilir, p, AR model derecesini, q ise MA model derecesini belirtmektedir. Bir ARMA modelin transfer fonksiyonu ele alındığında bütün sıfırları ve kutupları z düzleminde birim dairenin içinde olduğu varsayılır. Eğer sistemin

bütün kutupları ve sıfırları birim dairenin içindeyse bu tür sistemlere kararlı ve minimum fazlı sistemler adı verilir. Hepsi birim dairenin dışında ise bu durumda maksimum fazlı sistemler olarak adlandırılırlar [9].

2.7. Doğrusal Olmayan Sistem Modelleme

Endüstriyel uygulamaların büyük çoğunluğundaki sistemler, doğrusal olmayan dinamik sistemlerdir. Böyle sistemlerin modellenmesi daha zordur. Fakat, analiz tekniklerinin büyük kısmı, doğrusal matematik model yaklaşımını kullanılır. Halbuki, birçok durumuda bu yaklaşım iyi sonuç vermemektedir [10]. Doğrusal olmayan sistemin, birim zamandaki değişim hızı küçükse ancak o zaman doğrusal olarak modellenebilir. Aksi takdirde doğrusal model, sistemin dinamik yapısını tam olarak belirleyememektedir. Genelde sistem değişkenleri çok fazla ise, sistemin doğrusal modellenmesi oldukça zordur.

Doğrusal olmayan modellerde sistemi modellemek için temel bir yapı bulunmamaktadır. Bir çok yapı bulunmasına rağmen literatürde en çok karşılaşılan güç serisine dayalı doğrusal olmayan modelleme yaklaşımı, Volterra, bilineer ve PAR (polinomiyal autoregressive-çok terimli özbağlanımlı) modelleridir. Bu yapılar bir çok sistemin modellenmesinde kullanılırlar.

2.7.1. Bilineer Model

Doğrusal olmayan diğer modellerde, modelinde çıkış işareti, sadece giriş işaretinin ağırlıklandırılması sonucu elde edilir. Bu durum çıkış işaretinin önceki değerlerine bağlı olan ve doğrusal olmayan sistemlerin modellenmesinde yetersizdir. Bu durumda bilineer modeller kullanılır. Bilineer modelde çıkış işareti;

$$y_n = \sum_{i=1}^{N} h_i y_{n-i} + \sum_{i=1}^{N}\sum_{j=o}^{N} b_{i.j} y_{n-i} x_{n-j} + \sum_{j=0}^{N} c_j x_{n-j} \qquad (2.4)$$

şeklinde tanımlanır. Model parametreleri h, b ve c ile ifade edilmiştir. Bu modelde çıkış işareti ile giriş işaretinin çarpımı modele doğrusal olmama özelliği kazandırmaktadır [11].

2.8 Sistem Modelleme için Kullanılan Yöntemler

Sistemlerin modellenmesi için kullanılan klasik algoritmalar LMS ve RLS yöntemleri olmak üzere iki ana sınıfta gruplandırılabilir [12]. En küçük kareler kontrol, iletişim ve jeofizik sinyal işleme gibi farklı uygulamalarda yaygın olarak kullanılır. Özyineli(recursive) ve adaptif filtreleme yeni deneysel kanıtların var olduğu, bilinmeyen süzgeç modelleri hakkında her zaman daha çok bilgi veren özel bir tasarımı kullanır.

Bir algoritma performansını aşağıdaki faktörler belirler [13,14]:

1. Beklenen teoriksel yapıyla ilgili olarak çözüm elde etme doğruluğu,
2. Yakınsama hızı,
3. İstatistiksel olarak zamanla değişim izleme yeteneği,
4. Hesaplama karmaşıklığı,
5. Toplam yuvarlama hatasına dayanıksızlık.

2.8.1 LMS Algoritması

Sistem tanımlama algoritmaları; sistem dinamikleri hakkında yeterli bilgiyi veren bilinmeyen bir işletme modelinin, bir dizi parametrelerinin tahmini ile ilgilenir. Doğrusal sistem parametreleme; uygulamaların geniş bir alanıyla sistem modellemenin önemli bir sınıfıdır. Doğrusal model sınıfları arasında en yaygını FIR (Finite İmpuls Response-sonlu darbe cevaplı) yapılı olanıdır. Bu sınıflama ya fiziksel sistem modellemedeki esastır yada tahmini işi basitleştirmek için kabul edilir ve gerçek zaman uygulamalarında hesaplama yükünü azaltmak için kullanılır. Statik ve dinamik FIR modellerinin her ikisi de geçmişi göz önünde bulundurur ve algoritmaların büyük bir çoğunluğu model parametrelerinde verimlilik tahmini için önerilmiştir [7].

LMS yönteminde parametreler her iterasyonda hatayı en aza indirecek şekilde değişmektedir. Sistem modeli farklı eşitliklerle tanımlanabilir [7];

$$y(n) = \sum_{i=1}^{m} w(n)x(n-i+1) + \eta(n) \tag{2.5}$$

Burada x(n) giriş işareti, y(n) çıkış sinyalinde, w(n)'ler ise süzgeç katsayılarıdır. η(n) bozucu sinyaldir. (2.5) ile verilen eşitlik, eşitlik (2.6)'da verildiği gibi yazılabilir;

$$y(n) = x^T(n)w(n) + \eta(n) \tag{2.6}$$

x (n); boyutları M x l olan bilgi vektörüdür ve eşitlik (2.7)'da verildiği gibi gösterilir.

$$x(n) = [x(n)x(n-1)...x(n-M+1)]^T \tag{2.7}$$

w(n), Mxl boyutlu katsayı vektörüdür ve aşağıdaki gibi ifade edilir.

$$w = [w_1 w_2 ... w_M]^T \tag{2.8}$$

Denklem (2.5) ile gösterilen sistem için FIR kestirici eşitlik (2.9)'de verilmektedir.

$$\hat{y}(n) = x^T(n)w \tag{2.9}$$

Bununla birlikte tahmini hata; ölçülen ve önceden tahmin edilen sistem çıkışı arasındaki fark olarak gösterilir. Buda eşitlik (2.10)'de verilmiştir.

$$e(n)=y(n)-\hat{y}(n)=y(n)-x^T(n)w \tag{2.10}$$

Verilen bir dizi x(n) giriş sinyali ve ölçülen y(n) çıkış sinyali eşitlik (2.9)'de verilen FIR modelinin optimum paremetreleri; Eşitlik (2.11)'de verilen minimize edilmiş MSE (Mean Square Error- Ortalama Karesel Hata) kriteriyle tahmin edilir.

$$V(w) = \varepsilon[e^2(n)] = \varepsilon[(y(n) - \hat{y}(n))^2] \tag{2.11}$$

ε [.]; beklenti operatörüdür. Böylece;

$$w = \arg\min V(w) \arg\min \varepsilon[(y(n) - x^T(n)w)^2] \tag{2.12}$$

Değer fonksiyonu (2.11), aşağıda verilen V(w)'nin quadratik fonksiyonudur.

$$V(w) = \varepsilon[y^2(n)] + w^T Rw - 2d^T w \tag{2.13}$$

Burada R giriş sinyali x(n)'nin oto korelasyon matrisidir. D, giriş sinyali x(n) ve sistemin beklenen çıkış sinyali y(n) arasındaki çapraz ilişki vektörüdür.

$$R = \varepsilon[x(n)x^T(n)] \quad , \qquad d = \varepsilon[x(n)y(n)] \tag{2.14}$$

Denklem (2.11)'deki değer fonksiyonunun süzgeç katsayılar vektörüne göre minimizasyonu V(w)'nin gradyantının sıfıra eşitlemesiyle elde edilir.

$$\Delta V(w) = 0 \rightarrow Rw - d = 0 \tag{2.15}$$

$$Rw=d \tag{2.16}$$

Elde edilen minimum MSE, aşağıdaki gibi verilir.

$$E_{\min} = \varepsilon[y^2(n)] - d^T w \tag{2.17}$$

x(n) sinyalinin AR model eşitliği ile tanımlandığını varsayarsak,

$$\sum_{i=0}^{M} a_i x(n-i) = e(n) \tag{2.18}$$

eşitliği elde edilir. Burada a_0 1'e eşitlenir ve e(n) gürültü sinyalidir. Eşitlik (2.18)'deki x(n) işareti aşağıdaki gibi verilir.

$$\hat{x}(n) = -x^T(n)a \tag{2.19}$$

Her iterasyon için bir önceki hata ve çıkış değerlerinden faydalanarak hatayı minimize edecek şekilde yeniden hesaplanır. Böylelikle adaptif LMS yöntemi elde edilir. Bu yönteme birçok ilaveler yapılarak daha iyi sonuçlar vermesi sağlanmıştır.

2.8.2. RLS yöntemi

Adaptif transversal süzgeçlerin tasarımında özyinelemeli bir algoritma geliştirmek için LS (Least Square-En Küçük Kareler) metodunun kullanımı genişletilebilir. Öyle ki (n-1) anında süzgeçin ağırlık vektörünün LS tahmini verilmiştir. Yeni bilgi gelmesi üzerine bu vektörün yenilenen tahmini n zamanında hesaplanabilir. Bunun için RLS algoritma sonuçlarına başvurulur. RLS algoritması, Kalman süzgeç teorisinin deterministik bir kopyası gibi görünebilir. Kalman süzgeçleri; MS (Mean Square-Anlamlı Kareler) duyusu optimum olan bir özyinelemeli süzgeç sonucu izleme ortalamalardan elde edilir [15].

Öncelikle; LS metoduna ait olan bazı bağıntıları gözden geçirilerek RLS algoritmasının gelişimi aşağıdaki gibi açıklanır. Lemma tersleyen matrisi olarak bilinen matris cebrinin bağlantılarının açıklanmasıyla RLS algoritması geliştirilir. RLS algoritmasının en önemli özelliği, algoritma başlama zamanına kadar var olan tüm bilgilerin giriş bilgisinde gösterilmesidir. RLS algoritması, basit LMS algoritmasından daha hızlıdır. Performanstaki bu gelişim hesap karmaşıklığının çok fazla miktarda artmasıyla meydana gelir. Yani LMS' den karmaşık bir yapısı vardır.

Bu yöntemde hata fonksiyonu LMS'den farklı şekilde seçilmektedir. Hata fonksiyonu;

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

şeklinde verilmektedir. Hata fonksiyonunun parametre değerine göre türevi alınırsa normal denklemi eşitlik (2.21)'de verildiği gibi elde edilir.

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

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

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

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

Denklem 2.23' e dikkat edilecek olursa optimum parametre değerlerinin bulunması için korelasyon matrisinin tersine ihtiyaç duyulmaktadır. Bu matrisin tersini almak için Matris tersleyen lemma olarak bilinen matris kuralı kullanılır. Sonuç olarak RLS algoritması n=1,2,3,..., için zamanın her anında hesaplanır [9].

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

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

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

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

şeklinde elde edilir ve başlangıç şartları

$$P(o) = \delta^{-1} I \qquad \delta = \text{pozitif küçük bir sabit} \tag{2.28}$$

$\hat{w}(0) = 0$ şeklindedir. Bu yöntemde, adaptif kazanç, kovaryans matrisi, P yardımıyla her iterasyon için ayarlananır. Çoğunlukla RLS yöntemi, LMS yöntemlerine göre daha hızlı yakınsar. Fakat başlangıç değerleri ve yuvarlatma hataları açısından LMS'den daha hassastır [13].

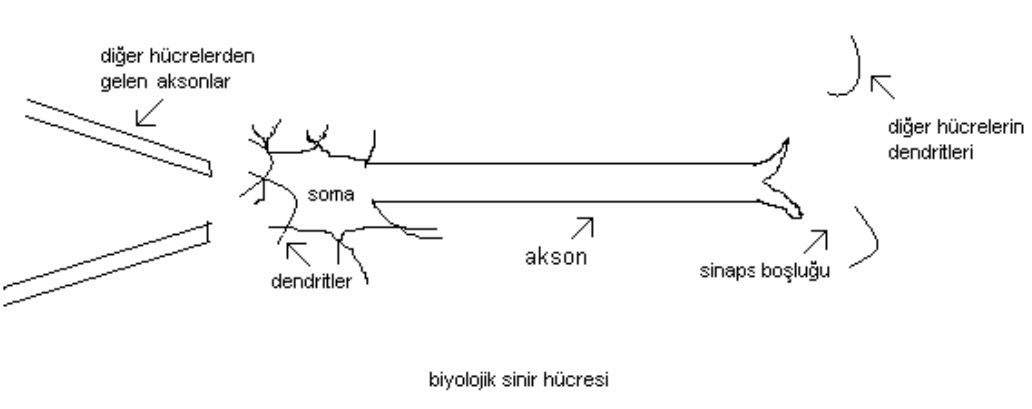
BÖLÜM 3

YAPAY SİNİR AĞLARI

Yapay sinir ağı, biyolojik sinir ağına benzer şekilde, belirli bir performans karakteristiğine sahip bilgi işleme sistemidir. Yapay sinir ağları, sinir biyolojisinin matematik modelleri genelleştirilerek aşağıdaki temeller üzerine geliştirilmiştir [17].

1.Bilgi işleme, hücre(neuron) adı verilen basit elemanlar üzerinde gerçekleşir.
2.Sinyaller, hücreler arası bağlantılar üzerinden, bir hücreden diğerine geçer.
3.Tipik bir sinir ağında, her bir bağlantı bir ağırlığa sahiptir ve bu ağırlık bağlantı üzerinden geçen sinyalin çarpanıdır.
4.Her bir hücre kendi çıkışını belirlemek için net girişlerine, genellikle doğrusal olmayan bir aktivasyon fonksiyonu uygular.

Beyin veya sinir hücresi ile işlem birimi (yapay hücre) yapıları arasında çok yakın bir benzetim vardır. İşlem birimi yapısının anlaşılabilmesi açısından, biyolojik hücre üç temel bileşene ayrılmıştır: dendritler, soma (çekirdek) ve akson. Biyolojik sinir hücresinin yapısı Şekil 3.1'de gösterilmiştir. Dendritler diğer hücrelerden gelen sinyalleri almaktadırlar. Sinyaller, sinaps (synaps) boşluğundan kimyasal işlemler aracılığı ile geçen elektriksel dürtülerdir (puls). Soma, dendritlerden gelen sinyalleri toplamaktadır. Yeterli giriş alındığında hücre etkinleşmekte, yani akson üzerinden diğer hücrelere sinyal göndermektedir. Genellikle, herhangi bir anda hücrenin etkin olduğu ya da olmadığı varsayılırsa, gönderilen sinyaller ikili (binary) gibi davranmış olurlar. Bununla birlikte, hücrenin etkinleşme frekansı değişkendir. Bir sinyalin belirli bir hücreden gönderilmesi, hücre akson zarının her iki tarafındaki iyon yoğunluğunun farklı olmasından kaynaklanan aksiyon potansiyeli aracılığı ile gerçekleşmektedir. Buradaki iyonlar; potasyum, kalsiyum ve klorid iyonlarıdır [18-23].



Şekil 3.1. Biyolojik nöron

Yapay sinir ağlarının bir başka önemli karakteristiği ise, tıpkı biyolojik sinir sisteminde olduğu gibi, hata tolerans özelliğine sahip olmasıdır. Biyolojik sinir sistemi iki çeşit hata toleransına sahiptir. Birincisi, daha önce karşılaşılmış olan sinyallerden farklı bir sinyalle karşılaşıldığında bunun farkedilmesidir [24].

İkincisi, sinir sisteminin kendisinde meydana gelen zararın tolere edilmesidir. Örneğin; insan doğduğunda yüz milyarlarca beyin hücresine sahiptir. Zamanla hücreler öldüğünde yenisi ile değiştirilememektedir. Hücre ölümü sürekli olmasına rağmen insanın öğrenmesi devam etmektedir. Böyle durumlarda, kaybedilen hücrelerin fonksiyonunu diğer hücreler üstlenmektedir. Aynı şekilde, yapay sinir ağları da küçük zararları karşılayabilecek şekilde tasarlanabilmekte, veya büyük zararlar karşısında yapay sinir ağı tekrar eğitilebilmektedir [25].

YSA'lar, biyolojik sinir sisteminin taklit edilmesiyle, oluşturulmuş yapay sinir sistemleridir. Biyolojik sinir sistemleri ile yapay sinir sisteminin benzerlikleri Tablo 3.1'de verilmiştir.

YSA'nın temel işlemci elemanının doğrusal bir yapıda olmaması, bu işlemci elemanların birleştirilmesi ile oluşturulan YSA yapısının doğrusal olmayan karmaşık problemlerin çözümünde de etkili sonuçlar üretebilmesini sağlayan bir özelliktir.

Tablo 3.1. Biyolojik sinir sistemi ile YSA'ların karşılaştırılması

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
➢ Nöron ➢ Dentrit ➢ Hücre Gövdesi ➢ Aksonlar ➢ Sinapslar	➢ İşlemci Eleman ➢ Toplama Fonksiyonu ➢ Transfer Fonksiyonu ➢ Yapay Nöron Çıkışı ➢ Ağırlıklar

YSA'ların arzu edilen davranışı sergileyebilmesi için problemin yapısına göre tanımlanması gerekmektedir. Temel fikrini aldıkları biyolojik sinir sistemine paralel olarak oluşturulan YSA'ların karmaşık yapısı, bağlantıların ve ağırlık değerlerinin önceden belirlenmesini engeller. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilenilen problemden alınan eğitim örneklerini kullanarak öğrenme işlemini gerçekleştirmek zorundadır. Bu öğrenme işlemi genel olarak tamamen bağlantı ve ağırlık değerlerinin ayarlanması şeklinde gerçekleşmektedir. Böylece YSA'ları farklı problemlere farklı çözümler sunabilen bir yapıya sahip olan ağ yapılarıdır [26].

Bu terminolojik benzerliklerle farklı yetenekleri de kazanan çok sayıda hücrenin çeşitli şekillerde birbirlerine bağlanması ile oluşan YSA modelleri paralel dağılmış bir yapıya sahip olurlar. Bu ağ yapısının sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağıtılmıştır. Böylece öğrenme işlemini gerçekleştirmiş olan bir YSA modeli biyolojik sinir sistemine benzer olarak herhangi bir bağlantının zarar görmesi hatta bazı bağlantı elemanlarının bozulması durumunda bile doğru bilgi üretebilecek bir yapıya sahip olur. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir [27].

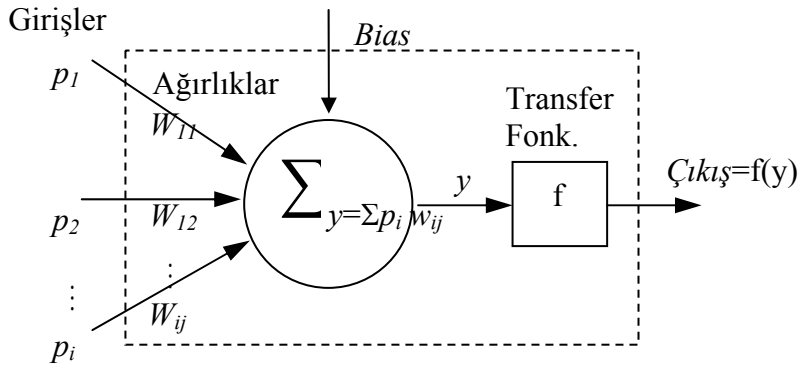
YSA teknolojisi havacılık ve uzay alanında yüksek performanslı otomatik pilot geliştirme çalışmalarında, uçuş kontrol sistemlerinde ve simülasyon cihazlarında kullanılmaktadır. YSA'ları ayrıca otomotiv sektöründe oto-rehberlik geliştirme uygulamaları geliştirmede de kullanılmaktadırlar. Bankacılık alanında ise YSA'ları, kredi risk analizlerinde, kontrol ve döküman okuma uygulamalarında kullanılmaktadır. Savunma alanında YSA'larından, hedef tanıma ve izleme, silah oryantasyonu, veri kodlama ve sıkıştırma, sayısal görüntü işleme gibi birçok alanda faydalanılmaktadır. Meteorolojik tahminlerde de yaygın olarak kullanılmaya başlanılan YSA'ları elektronikte bilgisayar destekli görme, ses anlama, görüntü ve işaret işleme, ve

filtreleme alanlarında da kullanılmaktadırlar. Ayrıca eğlence ve finans sektörünün yanı sıra sigortacılık, robotik, tıp, haberleşme ve diğer birçok alanda YSA'larından yaygın olarak faydalanılmaktadır [28].

3.1 Yapay Sinir Ağlarının Genel Özellikleri

Genel anlamda YSA'ları, biyolojik sinir sisteminin işlevi taklit edilerek insan beyninin modellenmesi için tasarlanmış sistemlerdir. Yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmalarından oluşan ağ yapısı genellikle katmanlar şeklinde düzenlenir. İnsan beyninin bilgi işleme yöntemine uygun olarak YSA'ları, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı şekilleri ve ağırlıkları ile bu bilgiyi saklama ve genelleme yapabilme yeteneğine sahip paralel dağılmış bir işlemcidir. YSA'ları, insan beyninin çalışma prensibi örnek alınarak geliştirilmeye çalışılmış ve aralarında Tablo 3.1'de de verilmekte olan yapısal benzerliklere sahip bir tekniktir. Bu benzerlikleri ile YSA'ları karmaşık problemleri çözebilme yeteneğine sahip olmaktadırlar. YSA'larının hesaplama özelliklerini, paralel dağılmış yapılarından ve öğrenme ve genelleme yapabilme özelliklerinden aldığı söylenebilir. Genelleme yapabilme özellikleri bilinen durumlardan bilinmeyen durumlar içinde çözümler sunabilme özelliği olarak tanımlanabilir. YSA'ları doğrusal olmayan yapıları, öğrenme ve genelleme yapabilme kabiliyetleri, hataya karşı toleranslı davranabilme yetenekleri, paralel işlem yapabilmelerinden kaynaklanan hızlı ve doğru çözümler sunabilmeleri, donanım olarak gerçekleştirilebilmeleri, adaptasyon kabiliyetleri ve piyasada hazır birçok yazılım bulunması nedeniyle çok farklı mühendislik alanında başarıyla uygulanmaktadır [17].

YSA'ların en küçük bilgi işleme elemanı olan yapay nöron, Şekil 3.2'de verilmektedir. Şekilden de görülebileceği gibi bir yapay nöron, girişler, ağırlıklar, birleştirme fonksiyonu, transfer fonksiyonu ve çıkış bileşenlerinden oluşmaktadır. Girişler, diğer sinir hücrelerinden yada dış bağlantılar üzerindeki ağırlıklardan hücreye giren verilerdir. Birleştirme fonksiyonu, yapay nörona gelen net girişin, girişlerin ilgili ağırlıklarla çarpılarak toplanmasıdır. Birleştirme fonksiyonu ile elde edilen net girişin bir işlemden geçirilerek nöron çıkışını belirleyen ve genellikle doğrusal olmayan fonksiyon ise Transfer fonksiyonu olarak bilinmektedir [29].



Şekil 3.2. Yapay nöron

YSA modeline verilen giriş değerlerine önce toplama fonksiyonları uygulanır ve her bir nöronun çıkış değeri;

$$y=\sum p_i w_{ij}\text{-}Bias \tag{3.1}$$

ile bulunur. Burada p_i, i'inci girişi, w_{ij}, j'inci elemana bağlantı ağırlığını ve *Bias* eşik değerini göstermektedir. Daha sonra bu y değerleri bir transfer fonksiyonuna uygulanır. Burada sigmoid fonksiyonu kullanıldığı kabul edilirse çıkış değeri *Ç*;

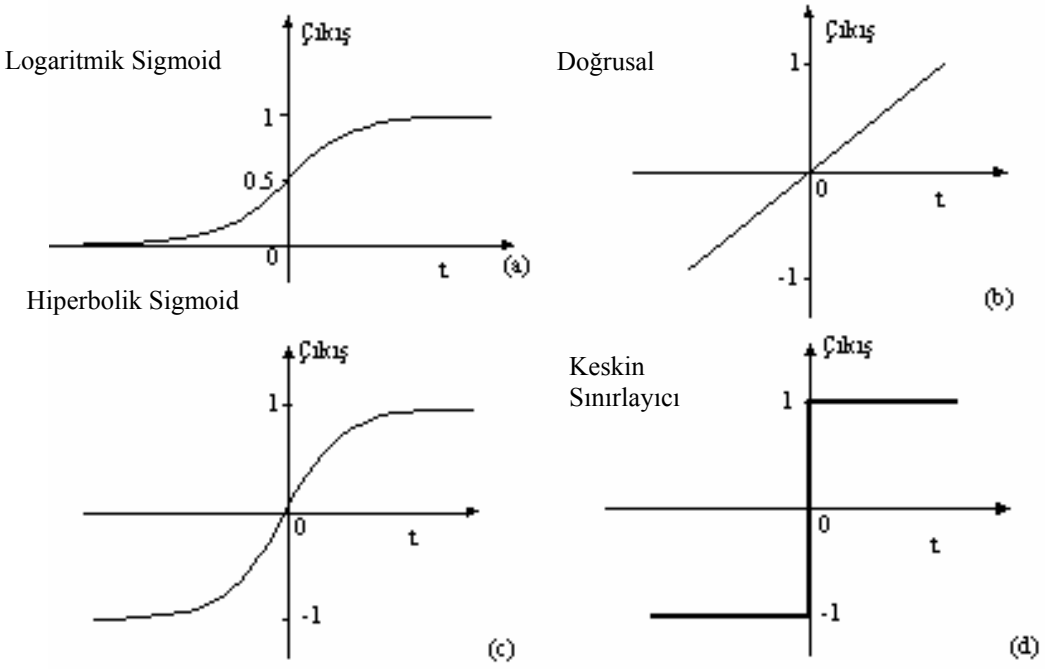
$$Ç = f(y) = \frac{1}{1+e^{-y}} \tag{3.2}$$

şeklinde elde edilir.

3.2 Transfer Fonksiyonları

YSA nöronları; transfer fonksiyonu adı verilen fonksiyonlar içerirler. Gradyan azaltımında kullanılan asıl fonksiyonlar transfer fonksiyonlarıdır. YSA'da en sık kullanılan transfer fonksiyonlarına ait grafikler Şekil 3.3'te gösterilmiştir.

Doğrusal olmayan transfer fonksiyonlarının da kullanılması ile yapay sinir ağları daha karmaşık problemlere uygulanmaya başlanmıştır [17-30]. Transfer fonksiyonları öğrenme eğrisi olarak da adlandırılmaktadır. Tablo 3.2'de YSA'da en çok kullanılmakta olan transfer fonksiyonlarının matematik ifadeleri verilmiştir [31].



Şekil 3.3. YSA'larda en çok kullanılmakta olan transfer fonksiyonları

Tablo 3.2. Bazı transfer fonksiyonlarının matematik ifadeleri

Fonksiyon tipi	**Matematik İfadesi**
Doğrusal	X
Logaritmik-Sigmoid	$\frac{1}{1+e^{-x}}$
RBF–Cauchy	$(1+x)^{-1}$
RBF–Ters Multiquadratik	$(1+x)^{0.5}$
RBF–Gaussian	$e^{\frac{-x^2}{\beta^2}}$
Hiperbolik Tanjant	$\frac{1-e^{-2x}}{1+e^{-2x}}$
Keskin Sınırlayıcı	$\begin{cases} 1 & x \geq x_0 \\ 0 & diğer \end{cases}$

Transfer fonksiyonları türevleri tanımlanabilir, sürekli fonksiyonlar olmalıdırlar.

3.3. YSA'ların Özellikleri

YSA'ların en önemli özellikleri; öğrenme, genelleme yapabilme, doğrusal olmama, gürültüyü yayabildiklerinden dolayı gürültüye karşı tolerans, paralel işlem yapabilme, donanım olarak gerçekleştirilebilme ve uyarlanabilirliktir [9,13].

3.3.1 Doğrusal Olmama

YSA'nın temel işlem elemanı olan nöron doğrusal olmak zorunda değildir. Bu durumda hücrelerin birleşmesinden meydana gelen YSA'da doğrusal olmayacaktır. Bu yüzden, YSA doğrusal olmayan karmaşık problemlere kolayca çözüm getirebilmektedir.

3.3.2 Öğrenme

Bir sistemin girişi ve çıkışı arasındaki ilişkiyi en iyi tanımlayacak olan YSA ağırlıklarının bulunması işlemine öğrenme denir. Bu işlemde YSA yapısı üzerindeki ağırlıkların bir öğrenme algoritması kullanılmasıyla, istenilen bağıntıyı öğrenmesi sağlanır. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilişkilendirmek istenen sistemin uygun örnek giriş ve çıkışlarından faydalanılarak gerçekleştirilir.

3.3.3 Genelleme Yapabilme

YSA'ları bir problemi öğrendikten sonra eğitimde kullanılmayan test verileri içinde beklenen veya istenilen tepkiyi üretme kabiliyetine sahiptir. Nöral hesaplamada hafızalar birleşiktir. Yani eğitilmiş ağa giriş örneğinin bir kısmı verilse bile, ağ hafızadan bu girişe en yakınını seçerek uygun çıkış değeri üretir. YSA'na bozuk veya eksik bir bilgi verilse bile ağ kabul edilebilir en uygun çıkışı verecektir. Burada temel özellik az veriyle sistem davranışının veya ilişkisinin öğrenilmesidir.

3.3.4 Uyarlanabilirlik

YSA'nın ağırlıkları, problemden probleme değişir. Yani bir problemde iyi çıkışlar veren ağırlıklar diğer bir problem için iyi çıkışlar vermeyebilir. YSA'ları problemlerdeki değişmelere göre farklı şekilde eğitilebilirler. Bu özellik YSA'larının hedef tanıma ve izleme, uyarlamalı örnek tanıma, sınıflandırma, görüntü ve sinyal işleme, otomatik sistem kimliklendirme ve kontrol gibi alanlarda kullanım kolaylığı sağlamaktadır.

3.3.5 Donanım Olarak Gerçekleştirilebilme

YSA'ları paralel işlem yapabilme yetenekleri nedeni ile büyük ölçekli entegre devre (VLSI-Very Large Scale Integration) teknolojisiyle tarafından desteklenmektedirler. Bu özellikleri ile de YSA'lar en ileri düzeydeki veri işleme teknolojileri arasındadır. Bu nedenle hızlı işlem yapabilme kabiliyetine sahiptirler.

3.3.6 Hataya Karşı Duyarlılık

YSA'ları çok sayıda işlemci elemandan oluşmuştur. Bu elemanlar arası bağlantılar paralel dağıtılmış bir yapıya sahiptir. Ağın sahip olduğu bilgi, ağdaki bütün bağlantılara dağıtılmıştır. Giriş veri setinde bulunabilecek herhangi bir gürültü, bütün ağırlıklar üzerine dağıtıldığından dolayı, gürültü etkisi tolere edilebilir. Geleneksel yöntemlere göre hatayı tolere etme yetenekleri daha fazladır.

3.3.7 Veri İşleme

Biyolojik sistemlerde veri dağınık yapıda saklanır. Yapay sinir ağlarında ise veriler ağırlıklar üzerine dağıtılmıştır. Ağırlıklar herhangi bir anda YSA'nın problemiyle ilgili olarak sahip bulunduğu bilgiyi ifade eder. Yapay sinir ağlarında veri ağırlıklar üzerine paralel olarak dağıtılmış şekilde temsil edilmekte, korunmakta ve işlenmektedir. YSA'larında dağıtılmış birleşik hafıza yapısı kullanılır [11].

3.4. YSA'ların Sınıflandırılması

YSA'ları işlemci eleman olarak adlandırılan nöronlardan ve ağ bağlantılarından oluşmaktadırlar. Bu nedenle ağ bağlantı şekillerine, öğrenme kuralları ve transfer fonksiyonlarına göre belirlenen bu YSA modellerini yapılarına ve öğrenme algoritmalarına bağlı olarak sınıflandırmak mümkündür [31].

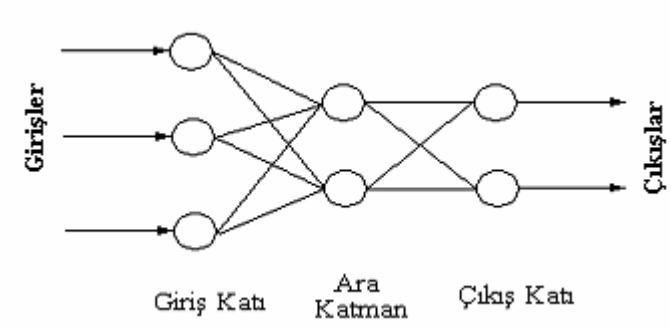
3.4.1 YSA'larının Ağ Yapılarına Göre Sınıflandırılması

YSA'ları mimari yapılarına göre ileri beslemeli (feed-forward) ve geri beslemeli (feed-back) ağlar olarak ikiye ayrılabilir [32]. Bu yapılar aşağıdaki kısımlarda açıklanmıştır.

3.4.1.1 İleri Beslemeli Ağ Yapısı

İleri beslemeli bir ağda işlemci elemanlar katmanlar üzerinde bulunur. Giriş katmanı, dış ortamdan aldığı bilgileri hiçbir değişiklik yapmadan ara katmandaki işlemci elemanlara iletir. Bilgi, ara katman veya katmanlarla çıkış katmanında işlenerek ağ çıkışına iletilir. Girişler tek yönlü olarak çıkışa doğru iletilirken, her bir katman çıkışı diğer katmana giriş olarak uygulanmaktadır. Herhangi bir andaki çıkış değeri, sadece o andaki girişin fonksiyonu şeklinde ortaya çıkmaktadır. Böylece bu ağ yapısı, statik yapılı bir hafızaya sahip olur.

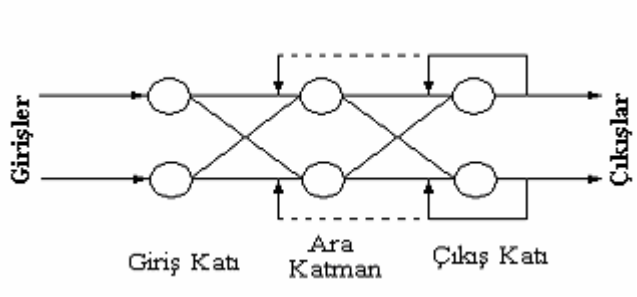
İleri beslemeli ağlar doğrusal olmayan bir yapıya sahip olup çok katmanlı perseptronlar (ÇKP), LVQ (Learning Vector Quantization-Nicemleme Öğrenme Vektörü) ağ yapıları bunlara örnek olarak verilebilir. ÇKP yapısının eğitilmesinde en çok kullanılan öğrenme algoritması geri yayılım algoritmasıdır. İleri beslemeli YSA yapısı Şekil 3.4'te verilmiştir.



Şekil 3.4. İleri beslemeli YSA yapısı

3.4.1.2 Geri Beslemeli Ağ Yapısı

Geri beslemeli bir sinir ağı, en az bir işlemci eleman çıkışının kendisine veya diğer işlemci elemanlara bir gecikme elemanı üzerinden giriş olarak uygulanması ile elde edilen ağ yapısıdır. Geri besleme, bir katmandaki işlemciler arasında olduğu gibi katmanlar arasındaki işlemciler arasında da gerçekleştirilebilmektedir. Geri beslemeli YSA yapıları, doğrusal olmayan dinamik bir davranış gösterirler ve herhangi andaki çıkış değeri hem o andaki hem de daha önceki giriş değerlerine sahiptirler. Dinamik yapılarından dolayı geri beslemeli YSA'lar tahmin uygulamalarında başarılı bir şekilde kullanılmaktadırlar. Şekil 3.5'te geri beslemeli bir YSA yapısı verilmiştir.



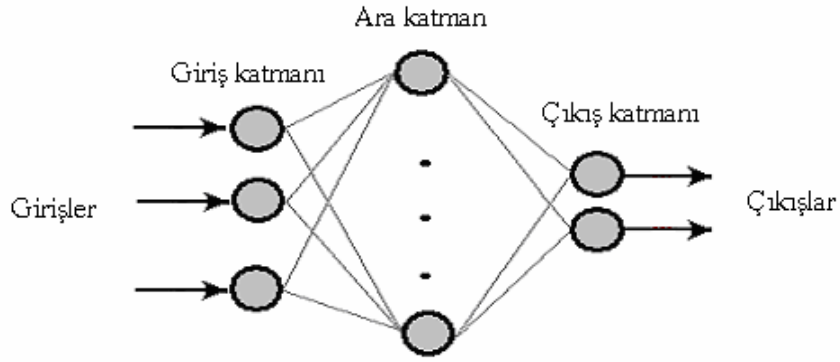
Şekil 3.5. Geri beslemeli YSA yapısı

3.4.2 Yapay Sinir Ağı Yapıları

Literatürde çok sayıda YSA ağ yapısı bulunmaktadır. Bunlara, ÇKP'ler, LVQ'lar, Elman ve Jordan Ağları, Radyal Tabanlı Yapay sinir ağı (RTYSA) yapıları örnek olarak verilebilir. Bu ağlardan en çok kullanılanı ÇKP anlatılmıştır.

3.4.2.1 Çok Katmanlı Perseptron YSA Yapısı

Çok katmanlı bir perseptron sinir ağı modeli, Şekil 3.6'da gösterilmiştir. Bu ağ modeli özellikle mühendislik uygulamalarında en çok kullanılan sinir ağı modeli olmuştur. Bu modelin yaygın kullanılmasının sebebi, bir çok öğrenme algoritmasının bu ağın eğitiminde kolaylıkla kullanılabilmesidir. Bu sebeplerden dolayı bu çalışmada ÇKP YSA kullanılmıştır. Bir ÇKP modeli, bir giriş, bir veya daha fazla ara kat ve birde çıkış katından oluşur. Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup geri besleme yoktur. Bu yüzden bu ağ yapısı ileri beslemeli sinir ağı modeli olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Ara katman sayısı ve ara katmanlardaki işlem elemanı sayısı genellikle deneme yanılma yoluyla bulunur. Çıkış katmanındaki eleman sayısı yine uygulanan probleme bağlı olarak belirlenebilir. ÇKP ağlarında, ağa bir örnek gösterilir ve örnek neticesinde nasıl bir sonuç türeteceği de bildirilir. Örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve arzu edilen çıkış arasındaki hata tekrar geri doğru ağırlıklar üzerine yayılarak hata minimuma düşünceye kadar ağırlıklar değiştirilir. ÇKP ağı ileri beslemeli ağ olup, en genel sonuç çıkış katmanından elde edilir. Kullanılan eğitme algoritmasına göre ağın çıkışı ile anlamıyla giriş ile çıkış uzayı arasında statik haritalama yaparlar, bir andaki çıkış sadece o andaki girişin bir fonksiyonudur [19].



Şekil 3.6. Genel bir ÇKP Yapısı

3.4.3 YSA'larının Öğrenme Algoritmalarına Göre Sınıflandırılması

Öğrenme; gözlem ve eğitim sonucu oluşan hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanabilir. Bu tanıma göre, YSA'larında öğrenme yani gözlem ve eğitim sonucu oluşan davranış değişiklikleri, bir takım metot ve kurallar yardımıyla ağırlıkların değiştirilmesiyle sağlanabilir. Öğrenme sırasında, ağın uyarılması sonucunda ağırlıklarında uyarılması ve ağın problemi çözmeye yönelik yeni tepkiler vermesi gerekir.

Literatürde, YSA yapılarına göre farklılık gösteren ve aşağıdaki bölümlerde kısaca anlatılan çok sayıda öğrenme algoritması mevcuttur [20-27].

3.4.3.1 Geri Yayılım (BP)

Bir çok uygulamada yaygın bir şekilde kullanılmakta olan bir öğrenme algoritmasıdır. Anlaşılması ve matematik olarak kolayca ispatlanabilir olmasından dolayı tercih edilmektedir. Bu algoritma, hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır.

Bu algoritmayla, *i* ve *j* katman nöronları arasındaki ağırlık ifadesi olan Δw_{ji}(k) değişikliği hesaplanır. Bu ifade,

$$\Delta w_{ji}(k) = \eta \delta_j x_i \tag{3.3}$$

olarak verilir. Bu eşitlik η öğrenme katsayısı, δ_j ara veya çıkış katmanındaki herhangi bir j nöronuna ait bir faktördür. Çıkış katmanı için bu faktör aşağıdaki şekilde verilir.

$$\delta_j = \frac{\partial f}{\partial net_j}\left(y_j^k - y_j\right) \tag{3.4}$$

Burada $net_j \equiv \Sigma x_i w_{ji}$ ve y_j^k ise j nöronunun hedef çıkışıdır. Ara katmanlardaki nöronlar için bu faktör,

$$\delta_j = \left(\frac{\partial f}{\partial net_j}\right)\Sigma w_{qj}\delta_q \tag{3.5}$$

olarak verilir. Ara katmanlardaki nöronlar için herhangi bir hedef çıkış olmadığından eşitlik (3.4) yerine eşitlik (3.5) kullanılır. Bu duruma bağlı olarak, çıkış katmanından başlayarak δ_j faktörü, bütün katmanlardaki nöronlar için hesaplanır. Tüm ağırlıklar eşitlik (3.3)'e bağlı olarak güncelleştirilir.

3.4.3.2 Momentumlu Geri Yayılım (BPM-Back Propagation Momentum)

Momentum ağın lokal gradyanleri aşmasını sağladığı gibi aynı zamanda hatanın hızlı düşmesine de yardımcı olur. Bir alçak geçiren süzgeç gibi davranan momentum, ağın hata yüzeyindeki küçük değerlerin ihmal edilebilmesini sağlar. Momentum olmaksızın ağ lokal bir minimuma takılarak salınımlar yapabilir. Momentum kullanılır ise ağ lokal minimumun diğer yüzüne sıçrama imkanı kazanır. Bu yöntemde öncelikle, öğrenilen her bir elemandan hesaplanan gradyanlerin toplamından toplam gradyan değeri hesaplanır ve tüm öğrenme kümesinin işlenmesi bittiğinde ağırlıklar ve bias'lar güncelleştirilir.

$$\Delta w_{ji}(k) = \eta\delta_j x_i + \mu\Delta w_{ji}(k-1) \tag{3.6}$$

3.4.3.3 Esnek Geri Yayılım (BPR-Back Resillient Propagation)

Sinir düğümü yapısında sıklıkla sigmoid tabanlı fonksiyonlar kullanılır. Bu fonksiyonlar sonsuz giriş aralığındaki veriyi sonlu aralığa transfer ederler ve giriş değeri büyüdükçe fonksiyonlardaki eğiklik sıfıra doğru gider. Bu durum sigmoid fonksiyonlarına sahip sinir ağlarında türevdeki değişikliklerin parametre üzerindeki etkisini azaltır. Bu yüzden türevin değeri yerine yalnızca işaretine göre parametrelerin yenilenmesini sağlamak amacıyla bu yöntem geliştirilmiştir. Bu yöntem önceki anlatılan yöntemden daha hızlı yakınsamaktadır.

3.4.3.4. Eşleştirmeli Gradyan Azaltma Geri Yayılım (CG-Conjugate Gradien)

Basit geri-yayılım algoritması ağırlık değerlerini gradyanın negatifi yönünde dengeler. Bu doğrultu gradyanın hızla düştüğü doğrultu olarak bilinir. Bu doğrultuda performans fonksiyonu da hızla düşer. Performans fonksiyonundaki hızlı düşüşe rağmen bu gelişme sonuca en hızlı yakınsama için gerekli olmayabilir. Eşleştirmeli gradyan algoritmalarında hızlı gradyan düşüş doğrultusundan genellikle daha çabuk sonuca yakınsayan eşleştirme doğrultusunda bir arama işlemi uygulanır. Bir çok eşleştirmeli gradyan algoritmasında adım boyutu her bir iterasyonda yeniden düzenlenir. Performans fonksiyonunun yakınsama doğrultusu boyunca minimize edilmesi için güncelleme adım boyutunun belirlenmesi amacıyla eşleştirmeli gradyan doğrultusu boyunca bir arama gerçekleştirilir.

3.4.3.5. Quasi-Newton (QN)

Newton yöntemleri hızlı optimizasyon için kullanılabilecek yöntemlerdir. Newton metodu temel olarak:

$$p_k = -A_k^{-1} g_k \tag{3.7}$$

ile ifade edilir. Burada A_k Hessian matrisidir ve ikinci derece türevlerde tanımlanır. Bu öğrenme algoritması Hessian matrisinin her iterasyonda yaklaşık değerinin hesaplanması esasına dayanır. Bir sonraki iterasyondaki Hessian matrisi ve ΔA_k :

$$A_{k+1} = A_k + \Delta A_k \tag{3.8}$$

$$\Delta A_k = \frac{g_k . g_k^T}{g_k^T . p_k} + \frac{\Delta g_k . \Delta g_k^T}{\Delta g_k^T \Delta X_k} \tag{3.9}$$

şeklindedir. Burada ΔX_k, k. iterasyondaki ağırlık değişimi olarak alınır. QN eşleştirmeli gradyan yöntemlerine göre çok daha fazla hafıza ve her iterasyonda çok daha fazla hesaplama gerektirse de genellikle birkaç iterasyonda sonuca erişir. Küçük boyutlu ağlarda kullanımı daha uygundur. n, ağdaki ağırlıkların ve bias'ların değerine eşit olmak üzere $n^2 \times n^2$ boyutlu tahmini Hessian matrisinin hafızada saklanması gerekir.

3.4.3.6.Delta-Bar-Delta (DBD)

Jacobs tarafından geliştirilmiş bir ÇKP öğrenme algoritması olan DBD'nin eğitme kuralı, ağırlıkların yakınsama hızını arttırmak için kullanılan sezgisel bir yaklaşımdır. DBD öğrenme algoritmasında, ağın her bir bağlantısına atanan öğrenme katsayılarının optimum değerlerinin belirlenmesi çok zor ve zaman alıcı olabilmektedir. Bu algoritmada, bir bağlantıya ait ağırlık değişim işaretinin ardışık birkaç adımı sırasıyla farklı değerler alması durumunda öğrenme oranı azaltılmalı aynı kalması durumda ise öğrenme oranı artırılmalıdır. Bu değişimi sağlayabilmek için ağırlıklar, hatanın derecesine bağlı olan kısmi türevlerine ve ağırlık değeri noktasının çevresindeki hata yüzeyinin eğrilik ölçüsüne göre güncellenmelidir. Güncelleme işlemi,

$$w(k+1) = w(k) + \alpha(k) \cdot \delta(k) \tag{3.10}$$

eşitliği ile gerçekleştirilir. Bu ifadedeki $\alpha(k)$ öğrenme katsayısı her bir ağırlık için farklı bir değerde atanmalıdır. Her bir ağırlığın öğrenme hızının artırılması için, $\delta(k)$ eğim bileşeninin ağırlıklı ortalaması kullanılarak bulunur ve ağırlıklı ortalama değeri,

$$\bar{\delta}(k) = (1-\theta) \cdot \delta(k) + \theta \cdot \delta(k-1) \tag{3.11}$$

eşitliği ile düzenlenebilir. Bu ifadede yer alan θ konveks ağırlık faktörüdür. Üstel olarak artan önceki eğim bileşeni ve mevcut eğim bileşeni aynı işaretli ise öğrenme katsayısı, κ sabiti ile artırılan ağırlıkla birleştirilir. Mevcut eğim bileşeni üstel ortalamasına göre zıt işaretli ise, öğrenme katsayısı mevcut değeri ile orantılı olarak azaltılır. Bu işlem,

$$\Delta\alpha(k) = \begin{cases} \kappa & \bar{\delta}(k-1) \cdot \delta(k) > 0 \\ -\varphi \cdot \kappa & \bar{\delta}(k-1) \cdot \delta(k) < 0 \\ 0 & diğer \end{cases} \tag{3.12}$$

eşitliği ile gerçekleştirilebilir. Burada, κ artma faktörünü, φ azaltma faktörünü, $\delta(k)$ ise k anındaki öğrenme katsayısını göstermektedir. Bu ifadeye göre DBD algoritması, öğrenme katsayısını doğrusal olarak artırmakta, fakat geometrik olarak azaltmaktadır denilebilir.

3.4.4.7 Genişletilmiş Delta-Bar-Delta (EDBD-Extended Delta-Bar-Delta)

Bu algoritma isminden de anlaşılabileceği gibi Jacobs tarafından sunulan DBD algoritmasının geliştirilmiş halidir. DBD algoritmasından farklılıkları; sezgisel momentum kullanması, ağırlık uzayında büyük artmaları ortadan kaldırması ve geometrik azaltımın büyük orandaki atlamalarını engellemede yeterince hızlı olmaları şeklinde sıralanabilir. EDBD'de ağırlık değişimleri,

$$\Delta w(k+1) = \alpha(k) \cdot \delta(k) + \mu(k) \cdot \Delta w(k) \tag{3.13}$$

eşitliği ile belirlenebilir. Burada $\delta(k)$, ağırlık değişiminin gradyan bileşeni, $\alpha(k)$ ve $\mu(k)$ sırasıyla öğrenme ve momentum katsayılarıdır. Ağırlıklardaki değişimin belirlenmesi ile yeni ağırlık değeri,

$$w(k+1) = w(k) + \Delta w(k) \tag{3.14}$$

şeklinde güncellenmelidir. Öğrenme katsayısındaki değişim ise,

$$\Delta\alpha(k) = \begin{cases} \kappa \cdot \exp\left(-\gamma_\alpha \cdot \left|\delta(k)\right|\right) & \bar{\delta}(k-1) \cdot \delta(k) > 0 \\ -\varphi_\alpha \cdot \kappa & \bar{\delta}(k-1) \cdot \delta(k) < 0 \\ 0 & diğer \end{cases} \tag{3.15}$$

ifadesi ile verilmektedir. Bu ifadedeki κ_α öğrenme katsayısının derecelendirme sabiti, γ_α öğrenme katsayısını üstel olarak artırmada kullanılan sabiti, φ_α öğrenme katsayısının azaltma sabitidir. Ağırlık uzayında osilasyonları ve aşırı atlamaları engellemede tüm bağlantılar için $\alpha(k) \leq \alpha_{max}$ ve $\mu(k) \leq \mu_{max}$ değerinin sağlanması gerekmektedir. Burada α_{max}, öğrenme katsayısının üst sınırı ve μ_{max}, momentum katsayının üst sınırıdır.

3.4.4.8 Levenberg Marquardt (LM)

Oldukça başarılı bir optimizasyon metodu olan LM algoritması, öğrenmede kullanılan geri yayılım algoritmasının farklı öğrenme tekniklerinden birisidir. Bu öğrenme algoritmasının seçilmesinin sebebi, hızlı öğrenme ve iyi yakınsayabilme özelliğinden kaynaklanmaktadır. Temel olarak LM algoritması, maksimum komşuluk fikri üzerine kurulmuş bir hesaplama metodu olup Gauss-Newton ve Steepest-Descent (adım azaltmalı) algoritmalarının en iyi özelliklerinden oluşmuştur ve bu iki metodun kısıtlamalarını ortadan kaldırmaktadır. Performans fonksiyonu kareler toplamı biçimine sahip olduğunda Hessian matrisine;

$$H=J^T J \tag{3.16}$$

şeklinde yaklaşıklık yapılabilir ve buradan gradyan;

$$g=J^T.e \tag{3.17}$$

olarak hesaplanabilir. Eşitliklerde , J ağda yer alan ağırlık ve biaslara (yanlılık) ait hataların ilk türevlerini içeren Jacobiyen matrisidir. e ise, ağ hatalarının vektörüdür. T matris transpozesini temsil eder. Bu metot ile performans fonksiyonu algoritmanın her iterasyonunda azalan bir eğim gösterecektir ve J matrisini Hessian matrisi yerine kullanır:

$$w_{k+1}=w_k-[J^T.J+\mu I]^{-1} J^T \times e \tag{3.18}$$

Eğer μ büyük ise, minimum yaklaşım adımı küçük olur. Bu yüzden, her başarılı adımda μ değeri azaltılmalıdır. Eğer performans fonksiyonu artıyorsa μ arttırılmalıdır.

3.4.4.9 Bayesian Düzenlileştirme (BD) Öğrenme Algoritması

Optimum düzenlileştirme parametrelerinin otomatik bulunması arzu edilir. Bu amaçla kullanılan yöntemlerden biri Bayesian düzenlileştirme yaklaşımıdır. Bayesian yaklaşımının ağda bulunan ağırlıklar ve bias değerleri, farklı dağılımlı değişkenler gibi düşünülür. Düzenlileştirme parametreleri bu dağılımların birleşiminin değeri bilinmeyen varyansı ile ilişkilendirilerek tahmin edilir. Bu amaçla istatistiksel yöntemler kullanılır. Burada LM'ye benzer şekilde, *J*'ler Jacobian matrisi, *E* hata, *I* birim matris ve μ momentum katsayısı olmak üzere;

$$dX=-(JX*JX+I*\mu)/(JX*E) \tag{3.19}$$

dir.

3.5. Sistem Modellemede Yapay Sinir Ağlarının Kullanımı

Adaptif modelleme düşük donanım maliyeti, yakınsak olması, hata analizinin yapılabilmesi gibi özelliklerinden dolayı en fazla kullanılan yöntemlerdendir. Fakat doğrusal olmayan özelliklere sahip ortamlarda ve zamanla değişen doğrusal sistemlerde, bu tür yöntemlerin performansları oldukça düşmektedir. Yapay zeka tekniklerinden olan yapay sinir ağları evrensel yaklaşım (universal approximation) yeteneği, bu ihtiyaca cevap verebilecek niteliktedir. Ayrıca sistem modellemede, model yapısının bilinmesine

ihtiyaç yoktur. YSA'nın öğrenme özelliğini kullanarak girişlere karşılık çıkışlar verilir ve bunun YSA tarafından öğrenilmesiyle sistem modellenir. YSA ile sistem modelleme farklı birçok alana uygulanmıştır [17-31].

YSA, beynin fizyolojisinden faydalanılarak oluşturulmaya çalışılan bilgi işleme modelleridir. YSA'lar insan beyninin çalışma prensibi örnek alınarak geliştirilmeye çalışılmıştır. Yapısal olarak benzerlikler gösterirler. YSA'larının öğrenme özelliği araştırmacıların dikkatini çeken en önemli özelliklerinden biridir. Çünkü bir olay hakkında giriş ve çıkışlar arasındaki ilişkiyi, doğrusal olsun veya olmasın, elde bulunan mevcut örneklerden öğrenerek daha önce hiç görülmemiş olayları, önceki örneklerden çağrışım yaparak ilgili olaya çözümler üretebilme özelliği, YSA'larındaki zeki davranışın temelidir [19-23].

Yapısı ve öğrenme yöntemleri dikkate alındığında sinir ağları ile adaptif sistemler arasında birçok ortak nokta vardır. Her iki yapıda da temel olarak adaptif doğrusal birleştirme özelliği yer alır. Aynı zamanda, sinir ağlarının temel öğrenme yöntemlerinden olan geri yayılma yöntemi, adaptif süzgeçlerde sıklıkla kullanılan LMS yöntemiyle aynı özellikleri taşır. Bu ortak noktalarının ve genel bir öğrenme yeteneğinin olması sinir ağlarının işaret işleme ve modellemede başarılı bir şekilde kullanılmasını sağlamıştır [30].

BÖLÜM 4

BULANIK SİNİR AĞI

4.1. Bulanık Mantık

Gerçek hayat karmaşıktır; yaşamdaki karmaşıklık genellikle belirsizliğin formunda olan kesinsizlikten ortaya çıkar. Karmaşık ve belirsiz özellikli problemler, insanların düşünebildiğinden beri, insanlar tarafından bilinçsizce çözülmektedir.

Dr. Zadeh' in birbirine uymayan önerilerinin prensibine göre, karmaşıklık ve belirsizlik ilişkisini, 'Bir kişi gerçek yaşamdaki bir problemi ne kadar yakından incelerse, problemin çözümü o kadar bulanıklaşır ' şeklinde belirtmiştir [32].

Klasik mantığın birçok alanda yetersiz hale gelmesi, insan zekasının işleyişine uygun olmaması sonucu bulanık mantık kavramı popüler hale gelmiştir.

Bulanık mantık, bulanıklığı açıklayan mantıktır. Daha geniş bir ifade ile, bulanık mantık, üyelik derecelerini temel almış, ikili mantığın aksine çok değerli, matematik bir disiplindir. Bulanık mantık, yanlış veya doğru olma durumlarından çok, doğru olma durumunun olasılıklarına dayanır. Ayrıca, bulanık mantık uygulaması, matematik modele ihtiyaç duymamaktadır [33].

4.1.1 Bulanık Mantık Gelişim Süreci

Bulanık mantık, 1965 yılında Lütfi Zadeh' in California Berkeley Üniversitesi'nde 'Bulanık Kümeler' adlı makalesini 'Information and Control' adlı dergide yayınlamasıyla başlamıştır. Bu çalışma basılmasından iki yıl önce yazılmasına rağmen radikal fikirlerinden dolayı hiçbir teknik dergi yayınlamayı kabul etmemiştir. Zadeh daha sonraları bulanık mantık ve düşünüşün temelini oluşturan Bulanık Algoritması'nı ileri sürmüştür. 1972 yılında Michio Sugeno bulanık ölçüm ve integral kavramlarıyla

bulanık konusuna yeni bakış açıları getirmiştir. 1974 yılında, Ebraham Mamdani bulanık mantığı ilk kez bir buhar makinesinin kontrol aşamasında kullanmıştır [34].

Bulanık mantık kuramının ilk önemli endüstriyel uygulaması, 1980 yılında Danimarka'daki bir çimento fabrikasında gerçekleştirilmiştir. Değirmen içinde çok hassas bir denge ile oranlanması gereken sıcaklık ve oksijen ayarı en uygun biçimde ayarlanmıştır [35].

1980 yılında F.H. Smidth Japon Fuji Elektrik firmasında su arıtma sistemlerinin kontrolünde bulanık mantığı kullanmıştır. Daha sonra yine bir Japon firması olan Hitachi (1987) otomatik tren bulanık kontrol sistemini geliştirmiştir. Bunlar 1990'lı yılların ilk yarısında Japonya'da yayılacak olan bulanık mantığın ilk öncüleri sayılmıştır [36].

4.1.2. Klasik ve Bulanık Mantık Arasındaki Fark

Karmaşık sistemler bulanık mantık teknikleri ile, '0' ve '1' değerlerinden oluşan ikili mantıktan daha iyi açıklanabilir.

Klasik mantık kavramında herhangi bir nesne bir kümeye ya aittir, ya da değildir. Yalnızca '0' veya '1' değerlerine sahiptir. Bunun ortası söz konusu değildir. İki değerli klasik mantığın aksine bulanık mantıkta elemanlar bulanık kümeye kısmen aittir ve '0' ile '1' arasındaki değerler de kullanılmaktadır [36].

Aşağıda Şekil 4.1.'de ikili (klasik) mantık ve bulanık mantık farkı şematik olarak gösterilmiştir. Görüldüğü gibi ikili mantıkta ara değer yoktur oysa bulanık mantıkta ara değerler söz konusudur.



Şekil 4.1. Klasik ve Bulanık Mantık Arasındaki Farkı

Bulanık kümelere dayalı olan bulanık mantık genelde, insan düşüncesine özdeş işlemlerin gerçekleşmesini sağlamakla, gerçek dünyada sık sık meydana gelen belirsiz ve kesin olmayan verileri modellemede yardımcı olur. Klasik küme teorisinde kesin

hatlar vardır ya doğru yada yanlış. Bulanık küme teorisi ise az, sık, orta , düşük, birçok, gibi dilbilimsel yapıları kullanarak dereceli veri modellemesi gerçekleştirmektedir. Kurallar, bulanık sistemin davranışını tanımladığından, bulanık kümeler kendi içerisinde öğrenmektedir. Bulanık küme, kesin geçişleri elimine ederek belirsizlik kavramının tanımını yeniden verir ve evrendeki bütün bireylere üyelik derecesi değerini atayarak matematik olarak tanımlar. Bulanık kümede her bir elemanın kümeye üyelik derecesini veren üyelik fonksiyonu vardır. Bulanık kümede üç tip üyelik fonksiyonu vardır; üçgen, yamuk ve parabolik . Üyelik derecesinin değeri sıfır ile bir arasındadır.

Bulanık mantık kullanılarak modelleme yapılırken konu hakkında tecrübeye sahip bir bilirkişi bulunmalıdır.

4.1.3. Bulanık Mantık Uygulama Alanları

Bulanık mantığın uygulama alanları denetim sistemlerinin de ötesine uzanmaktadır. Bulanık mantık, Fizik, Biyoloji, Mühendislik, İşletme gibi birçok alanda sürekli sistemleri modellemek üzere kullanılabilmektedir. Çoğu alanda, bulanık mantıklı modellerin matematik modellerden çok daha iyi sonuçlar verdiği görülmüştür.

Japonya' da bir metro sisteminde kullanılan bulanık mantık çok başarılı olmuştur. 1990'lı yıllarda bulanık mantık, fotoğraf makinesinden ev aletlerine kadar çok değişik alanlarda kullanılmıştır. Bulanık çamaşır makinesinin değişmez komutlarla çalışan 'aptal' bir makineden daha temiz yıkadığı görülmüştür. Araba fren-vites sistemlerinde, hatta borsa gibi birçok farklı sektörde bulanık mantık uygulamaları mevcuttur [33-44].

4.2. Bulanık Küme Teorisi

Bulanık mantık doğal lisandaki belirsizliği temsil etmek için kullanılmaktadır. Belirsizliği tanımlamak amacıyla, Zadeh bulanık küme kabulünü açıklamıştır. Bulanık kümeler, nesnelerin uzunluk, kısalık, soğukluk, parlaklık v.b. gibi göreceli sıfatlarını tanımlamakta kullanılmaktadır. Klasik kümelerde bir elemanın, belirli bir kümeye olan üyeliği kesin şekilde belirlenmektedir. Eleman, kümenin ya üyesidir ya da değildir. Oysa ki, bulanık kümelerde, bir elemanın kümeye olan üyeliği aşamalıdır. Bulanık küme sınırları, klasik küme sınırları gibi kesin bir çizgi ile ayrılmamaktadır .

$$f(l_i) = \begin{cases} 0, & l_i \le a \\ 1/(b-a), & a < l_i < b \\ 0, & b < l_i \end{cases} \quad (4.1)$$

Bir model, veriler arasındaki ilişkilerin sistemidir. Bu verilerin bazıları kesindir, bazıları rastgeledir, bazıları ise belirsizdir. Kesin veya/ve rastgele verilerin kullanıldığı modeller iyi bilinmektedir ve uzun zamandır kullanılmaktadır. Bilim adamları belirsiz veriyi buldukları zaman, Laplace kuralı veya bazen bir Bayes işlemi gibi olasılık yasalarını kullanarak, belirsizliği tesadüfiliğe dönüştürmeye çalışmışlardır. Tesadüfilik, belirsizlikten farklıdır. Örneğin, L_1 ve L_2 gibi rastgele iki değişkeni aynı ihtimal yasasında kullandığımızı varsayalım:

$L_1 , L_2 , a , b \in \mathrm{R}$; $i = 1, 2$ durumunda ve Eşitlik (4.2) ile, Eşitlik (4.3) elde edilmektedir.

$$L_1 \,(+)\, L_2 = L \quad (4.2)$$

$$f(l) = \begin{cases} 0, & l < 2a \\ (x-2a)/(b-a)^2, & 2a \le l \le a+b \\ (-x+2b)/(b-a)^2, & a+b \le l \le 2b \\ 0, & 2b \le b \end{cases} \quad (4.3)$$

Böylece $f(l)$ üçgen yoğunluk kuralı olarak hesaplanabilir. Şimdi, eşit olan mesafesini inceleyelim.

$A_1 = [a, b]$ ve $A_2 = [a, b]$ olsun.

Toplamları,

$$A_1 \,(+)\, A_2 = [a, b] \,(+)\, [a, b] = [2a, 2b] \quad (4.4)$$

şeklinde verilir. Eşitlik (4.4)'deki gibi kullanılan alt kümeler, kendi karakteristik fonksiyonları ile tanımlanabilir:

$$\mu_{A_i}(x_i) = \begin{cases} 0, & x_i < a \\ 1, & a \le x_i \le b \\ 0, & b \le x_i \end{cases}$$ Burada, (4.5)

$a, b \in R; i = 1,2$ olmaktadır ve toplamları aşağıda verilen Eşitlik (4.6) ile bulunur:

$$\mu_A = \mu_{A_1} (+) \mu_{A_2} = \begin{cases} 0, & x_i \leq 2a \\ 1, & 2a \leq x_i \leq 2b \\ 0, & 2b \leq x_i \end{cases} \tag{4.6}$$

Böylece 2*a* ve 2*b* ve 0 arasındaki bir *x* değeri için,

$$\mu_A (x) = 1 \tag{4.7}$$

olarak hesaplanmaktadır. Bu örnek, rastgelelik ve belirsizlik arasındaki asıl farkı açıklamaktadır. İlk durumda, Laplace'dan bir hipotezi kabul edilmektedir. İkinci durumda, olasılığın bütün kurallarını görülmez. Sonuçta, olası belirsizlik ve olası olmayan belirsizlik hakkında dikkatli olmamız gerektiğini görmekteyiz.

Bulanık küme teorisi, olası olmayan belirsizlik ile ilgilidir. Birçok durumda, modellemede, olasılık teorisinden çok bulanık küme teorisini kullanmak çok daha iyidir çünkü sonuçlar çok daha gerçekçi çıkmaktadır. Eğer bir modelde bir veri ölçülemiyorsa, bulanık küme teorisinin ürettiği değer tavsiye edilmektedir [38].

Bulanık küme düşüncesi, bilgisayar sistemleri içerisine belirsizlik, kesinsizlik, bulanıklık gibi konuları gidermek amacıyla Zadeh tarafından geliştirilmiştir. Küme sınırlarındaki bulanıklık, bulanık küme teorisinde 0 ve 1 arasında değerleri alabilen bulanık küme belirsizliğidir. Bu belirsizliğin miktarı üyelik fonksiyonu ile ifade edilir. Klasik matematikte yer alan kümeler karakteristik fonksiyonlarla tanımlanır. Bulanık kümeler ise üyelik fonksiyonlarıyla tanımlıdır.

Bulanık kümelerin esası, bir elemanın içinde bulunduğu kümeye ait olma veya üyelik derecesinin belirlenmesine dayanmaktadır. Zadeh, bulanık kümelerin tanımını "üyelikten üyeliğe derecelenmiş bir dizinin olduğu bir grup yada takım" olarak yapmıştır [39].

$A = \{x \mid x = y + 0.1 \; ; y \in [0\ 0.9]\}$ şeklinde bir *A* kümesi olsun. Görüldüğü gibi *A* kümesi [0 1] aralığındaki sayıların oluşturduğu bir kümedir. Eğer bir sayı [0 1]

aralığında ise *A* kümesinin elemanıdır, değilse *A* kümesinin elemanı değildir. Bu durum aşağıdaki şekilde ifade edilebilir:

A={ (0.1 1) , (0.2 1) , (0.3 1) ,..., (1 0) , (2 0) , (3 0) , (4 0), ...}

A kümesinde sıralanan ikililerden ikinci rakamlar 0 veya 1 olup, eleman olup olmadığının derecesini göstermektedir. 0.1 rakamının yanında bulunan (1), 0.1 sayısının *A* kümesinin elemanı olduğunu belirtmektedir. 1 rakamının yanında bulunan (0), 1 sayısının bu kümenin elemanı olmadığını belirtmektedir.

4.2.1. Klasik ve Bulanık Kümeler

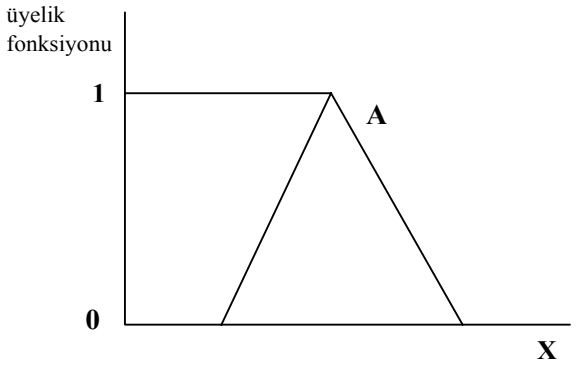
Klasik veya belirgin kümelerde, verilen bir kümede, evrendeki bir elemanın üyelik veya üyeliği olmayan arasındaki geçişi anidir ve iyi belirlenmiştir ki, buna belirgin denir. Evrende bulanık küme içeren bir eleman için geçiş kademelidir. Çeşitli üyelik değerlerine bağlı olan bu geçiş, bulanık kümelerin sınırının muğlak ve belirsiz olması nedenine bağlanarak düşünülebilir. Böylece, evrendeki bir kümeden bir elemanın belirsizliği ve bulanıklığı bir fonksiyon sayesinde ölçülür.

Bulanık bir küme, böylece, bir kümede değişik üyelik derecelerine sahip elemanlar içeren kümedir. Bu düşünce klasik veya belirgin kümelerle zıttır çünkü bir klasik kümenin üyeleri kendi üyelikleri tam olmadıkça klasik kümenin üyesi değildir. Bulanık bir küme elemanları, üyeliklerinin tam olmasına ihtiyaç duymadıkları için, aynı zamanda, aynı evrende başka bulanık kümelere üye olabilmektedirler [33].

4.2.2. Üyelik Fonksiyonu

Kuralları sisteme girmek için ihtiyaç duyduğumuz üyelik fonksiyonu, her girişin katılımının büyüklüğünün grafik temsilidir. Üyelik fonksiyonu, gerekli her giriş ile ağırlığı birleştirmekte, girişler arasındaki fonksiyonel hataları belirlemektedir ve son olarak bir çıkış cevabı oluşturmaktadır. Kurallar, son çıkış kararının bulanık çıktıdaki etkisini belirlemek için, faktörleri ağırlıklandırarak, giriş üyelik değerlerini kullanmaktadır [34].

Bir üyelik fonksiyonu, 0 ile 1 arasında bir üyelik değerinde giriş uzayındaki her noktanın nasıl belirleneceğinin ayrıntısıyla planlamasını sağlamaktadır. Bulanık kümenin her elemanı, küme içerisinde bir üyelik değerine sahiptir ve bulanık *A* kümesinin elemanlarının değerleri Şekil 4.2'de gösterildiği gibi 0 ile 1 arasındaki sayılardan oluşur.

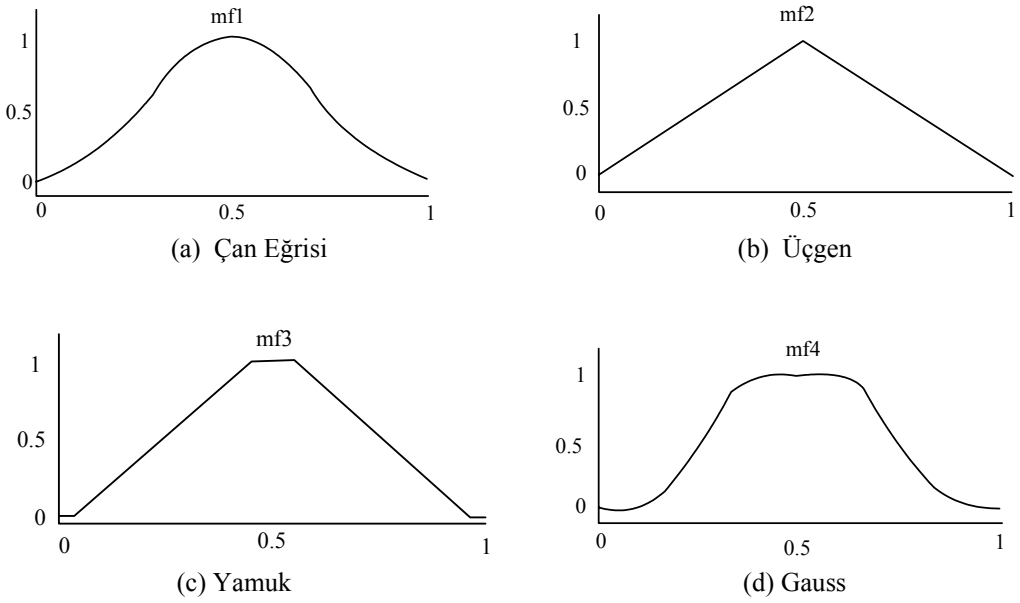


Şekil 4.2. *A* Kümesinin Üyelik Fonksiyonu

Bir bulanık küme, kendi aitlik fonksiyonu ile açık olarak temsil edilebilmektedir. Şekil 4.2.'de görüldüğü gibi, aitlik fonksiyonu 0 ile 1 arasındaki her değeri alabilir. Böyle bir aitlik fonksiyonu ile 'kesinlikle ait' (1) veya 'kesinlikle ait değil' (0) arasında istenilen incelikte ayarlama yapmak mümkündür [35].

Bir fiziksel değişkenin üyelik derecesini tanımlamak için kullanılan üyelik fonksiyonları, dilsel ifadelerden oluşan bir anlam grubudur. Bulanık küme teorisinin temelini oluşturan üyelik fonksiyonları 0 ile 1 arasında bir üyelik derecesine sahiptir. Üyelik derecesi, elemanın bulanık kümeyle temsil edilen kavrama ne derece uygun olduğu veya bulanık kümenin temsil ettiği özellikleri ne dereceye kadar taşıdığını gösterir. Üyelik fonksiyonu, biçimsel olarak denetlenen sürecin özelliklerine göre değişik şekillerde olabilir. Üyelik fonksiyonları, genellikle aşağıda Şekil 4.3.'de görülen, üçgen (triangular), yamuk (trapezoidal), gauss veya çan eğrisi (bell-shaped) biçimlerinde olmaktadır.

En yaygın olarak kullanılan üyelik fonksiyonu üçgen tipidir. Çok daha karmaşık üyelik fonksiyonları kullanılabilir fakat karmaşıklık arttıkça, daha fazla hesaplama gücü gerekmektedir.



Şekil 4.3. Çeşitli Üyelik Fonksiyonu Biçimleri

4.2.3. Bulanık Kümelerde Temel İşlemler

Klasik kümeler üzerinde tanımlanan üç temel işlem olan tümleyen alma, birleşim ve kesişim işlemlerinin bulanık kümeler üzerine genişletilmesi işlemini birden fazla yolla yapmak mümkündür. Bulanık kümeler ile, uygulama alanının getirdiği özelliklere göre yeni işlemler tanımlanabilir. *U* uzayından, *A* kümesinin elemanlarını oluşturmak için seçtiğimiz x objeleri [0,1] reel aralığında üyelik dereceleri alıyorlarsa, *A* kümesi bulanık bir kümedir. μ_A (x), elemanların *A* kümesine üyelik derecelerini göstermektedir [33]. Dolayısıyla bulanık kümeler sıralanmış ikililerden oluşur.

$$A=\{(x, \mu_A(x)) \mid x \in U\} \tag{4.8}$$

Bu denklemi Zadeh aşağıdaki şekilde ifade etmiştir.

$$A=\{ \mu_A(x_1) / x_1 + \mu_A(x_2) / x_2 + \ldots\}$$

$$= \{ i\Sigma (\mu_A(x_i) / x_i) \}$$

$$= \{ \int (\mu_A(x) / x) \} \tag{4.9}$$

4.2.3.1. Kapsama

A ve B' nin x' lerden oluşan bulanık kümeler olduğu varsayılmıştır. B' nin A' yı kapsama şartı;

$$\mu_A(x) \leq \mu_B(x) \quad x \in X \tag{4.10}$$

($A \subset B$) şeklinde sembolize edilir.

4.2.3.2. Denklik

A ve B' nin x' lerden oluşan bulanık kümeler olduğu varsayılmıştır. A'nın B'ye denk olma şartı:

$$\mu_A(x) = \mu_B(x) \tag{4.11}$$

$A = B$ şeklinde sembolize edilir.

4.2.3.3. Kesişim

A ve B' nin x' lerden oluşan bulanık kümeler olduğu varsayılmıştır. A ve B bulanık kümesinin kesişimi $A \cap B$ şeklinde sembolize edilir.

$$\begin{aligned} \mu_{A \cap B} &= \min \{ \mu_A(x), \mu_B(x) \} \quad x \in X \\ &= \mu_A \wedge \mu_B \end{aligned} \tag{4.12}$$

biçiminde tanımlanır. Kesişim "ve" birleştiricisine karşılık gelir.

4.2.3.4. Birleşim

A ve B' nin x' lerden oluşan bulanık kümeler olduğu varsayılmıştır. A ve B bulanık kümesinin birleşimi $A \cup B$ şeklinde sembolize edilir. $A \cup B$ bulanık kümesi, A ve B' yi kapsayan en küçük bulanık küme olarak tanımlanır. AUB kümesi oluşturulurken maksimum işlemcisi (V) dikkate alınır. Birleşim;

$$\mu_{AUB} = \max \{ \mu_A(x), \mu_B(x) \} \quad x \in X \quad = \mu_A \vee \mu_B \tag{4.13}$$

biçiminde tanımlanır.

4.2.3.5. Tümleyen

A ve *B*' nin *x*' lerden oluşan bulanık kümeler olduğu varsayılmıştır. *B*' nin *A*' nın tümleyeni olması şartı:

$\mu^{c}_{A}(x)= 1- \mu_B(x) \quad x \in X$

(4.14)

olarak ifade edilir. *A* bulanık kümesinin tümleyeni A^c, $\overline{A}$ şeklinde sembolize edilir ve tümleyen "değil" bağlacına karşılık gelir.

4.2.3.6. Cebirsel Toplam

A ve *B*' nin *x*'lerden oluşan bulanık kümeler olduğu varsayılmıştır. *A* ve *B*'nin cebirsel toplamları:

$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \ \mu_B(x) \qquad x \in X$ (4.15)

olarak ifade edilip, $(A \oplus B)$ şeklinde sembolize edilir.

4.2.3.7. Cebirsel Çarpım

A ve *B*' nin *x*' lerden oluşan bulanık kümeler olduğu varsayılmıştır. *A* ve *B*'nin cebirsel çarpımları:

$\mu_{AB}(x) = \mu_A(x).\ \mu_B(x) \qquad x \in X$ (4.16)

olarak ifade edilip, *AB* şeklinde sembolize edilir.

4.2.3.8. Fark

A ve *B*' nin *x*' lerden oluşan bulanık kümeler olduğu varsayılmıştır. *A*'nın *B'* den farkı;

$\mu_{A \cap B^c}(x) = \min \{ \mu_A(x), \mu_{B^c}(x) \} \quad x \in X$ (4.17)

$\mu_{A \cap B^c}(x) = (\mu_A(x) \wedge \mu_{B^c}(x)) \quad x \in X$ (4.18)

olarak sembolize edilir ve belirtilir.

4.3. Bulanık Sistemler için Bulanık Kurallar

İnsanlar kurallara dayanan kararlar oluştururlar. Örneğin oluşturduğumuz bütün kararlar EĞER-İSE durumları şeklinde bilgisayara girilmektedir. Eğer hava durumu bugün havanın kötü, yarın iyi olacağını söylüyorsa, dışarı çıkma kararını yarına erteleyebiliriz. Kurallar, bir olayın diğerleriyle ilgili olarak, fikirlerin bir araya gelmesi ile oluşmaktadır. Her zaman, insanın davranışlarını taklit etmeye çalışan bulanık makineler, aynı şekilde çalışmaktadırlar. Bununla beraber, karar ve kararı seçen anlamlar bulanık kümeler ile yer değiştirmektedir. *A* ve *B*, *x* ve *y*'nin kümeleri olmak koşulu ile, 'EĞER *x* *A* İSE y *B*'dir.' bulanık kurala örnek olarak gösterilebilmektedir [36].

Bulanık kurallar, bulanık mantıktaki anahtar fikir olan bulanık parçaları tanımlamaktadır. Eğer parçalar geniş ise, kurallar dikkatsizce yapılmış demektir. Eğer bulanık parçalar küçük ise, kurallar iyidir. Bulanık sistemdeki bu basitleştirme, tüm kuralların parçalar halinde görülmesidir ve bulanık makinenin çıkışı bu parçalarla birlikte birleşebilmektedir [37].

Bulanık bir kural, bulanık bir deyim ve teklifden oluşmaktadır. Bulanık kural formu : "EĞER *bulanık deyim* İSE *bulanık teklif*." şeklindedir. Bulanık bir teklif, *X* uzayında bulanık bir küme olan *A* için "*x*, *A*'dır." genel şeklinin "yükseklik uzundur." gibi verilen bazı dilsel değişkenler için bir değer öne süren bir durumdur. Bulanık bir deyim, bulanık teklifin cebirsel ikili ifadesidir.

Bulanık Uzman Sistemler' de kural formu aşağıdaki gibidir:
EĞER *x* iyi ve *y* zayıf İSE *z* orta'dır.

Burada x ve y değişkenleri giriş değerleridir ve z, çıkış değişkenidir . Bu giriş ve çıkış değişkenlerine sistemimizde istediğimiz isimleri atayabiliriz. Zayıf, orta ve iyi; sırasıyla *y, x* ve *z* için tanımlanmış üyelik fonksiyonlarıdır. Kural iki kısımdan oluşmaktadır; EĞER ve İSE. İkinci kısım, İSE' den sonra şartlı çıkış değeri tanımlanmıştır, bu, kural sonucunu bildirmektedir.

Tipik bir bulanık uzman sistem birden fazla kuraldan oluşmaktadır. Kuralların oluşturduğu yapı, kural tabanı veya bilgi tabanıdır. İstenen her kural ağırlıklarına göre sıralanarak sistem kural tabanına yerleştirilebilir yani her kural için farklı ağırlıklar, öncelikler tanımlanabilmektedir.

Günlük konuşmalarımızda kullanılan kelimelerin tanımında yatan belirsizlik, vurguların sayısal olmaması belli esneklikler sağlar. Bulanık mantık kuralların eksik veya "Bulanık" olarak tanımlandığı durumlarda kullanılan bir akıl yürütme yöntemidir ve etkin olarak kullanılabilmesinin en önemli sebebi ise dilsel belirsizlikten kaynaklanan durum ve olayların esnek kurallar ile temsil edilebilmesinden kaynaklanır. Temel iletişim aracımız olan dil kesin olmaktan çok uzaktır. Kullandığımız sıfatlar algılamayı daha yalın hale getirmeyi sağlarken, sınıflandırma yapma ifadeleri anlaşılır kılmayı sağlar. Burada sözü edilen sınıflandırma türlerinin kesin sınırları yoktur. Bu sınıflandırmalar genellikle belli bir aralık ve hassasiyet ile bağlantılıdır. O halde yaşadığımız dünyayı üyelik dereceleri ile sınıflandırma doğal düşünce tarzıdır. Genelde, bir dilsel değişkenin değeri birbirine bağlı anatomik terimlerden oluşan birleşik bir terimdir. Bu terimler dört gruba ayrılabilir [38]:

1. Birincil Terimler: Bu terimler tanım uzayının bulanık alt kümelerini tanımlayan etiketlerdir. Yani "sıcak", "soğuk", "zor", "düşük" v.b.,
2. Olumsuz DEĞİL (NOT) ve bağlaçlar VE (AND) ve VEYA (OR),
3. Pekiştirmeler, yani, "çok", "daha", "önemsiz", hemen hemen" v.b.,
4. İşaretler, yani parantezlerdir.

Birleşik terimler gibi birincil terimler de olasılık etiketli "muhtemel", "çok muhtemel", "muhtemel olmayan" gibi dilsel terimler olabilir. Ek olarak, birincil ve birleşik terimler "doğru", "oldukça doğru", "çok doğru", "yanlış", "oldukça yanlış" ve "çok yanlış" gibi doğru nicelikli yapılarla anlamlı olarak da düzeltilebilir. Olasılıklı etiketler olasılığa dayanırlar. Kurallar gibi birincil terimler de kesinlik etiketli "tanımsız", "bilinmeyen" ve "tanımlı" gibi dilsel değişkenlerle sınırlanabilir. Şart kuralları birleşik veya basit şekilde olabilir. Birleşik şart kuralları EĞER-İSE kurallar takımı formunda veya "değilse" veya "-medikçe" gibi dilsel terimli kurallar olabilir.

Dilsel değişken terimler, 'sıcak' veya 'soğuk' gibi kelimeler ve ifadelerle tanımlanabilen değişkenlerdir. Bir dilsel değişkenin değerleri bulanık kümeleri ile ifade

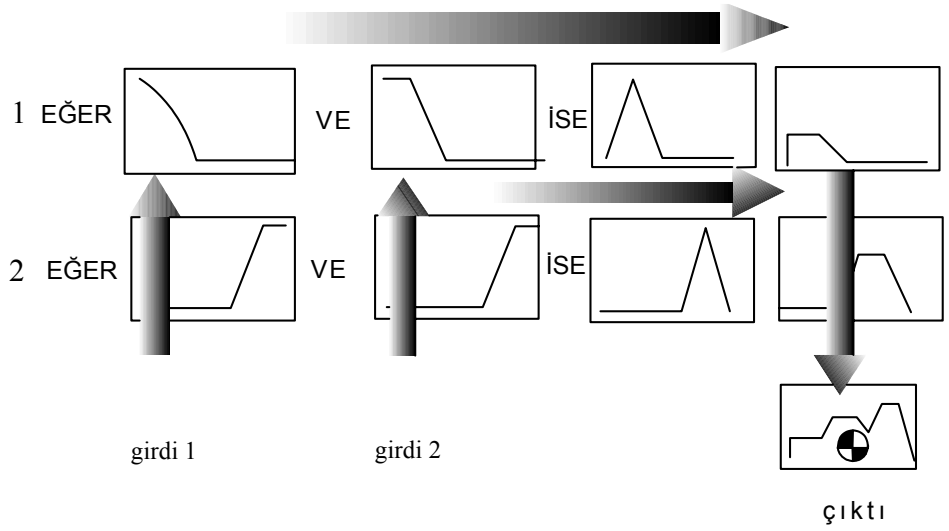
edilmektedir. Örneğin oda sıcaklığı dilsel değişken için ‘sıcak’, ‘soğuk’ ve ‘çok sıcak’ ifadelerini alabilmektedir. Bu üç ifadenin her biri ayrı ayrı bulanık kümeleri ile modellenmektedir [59].

4.4. Bulanık Çıkarım İşlemi

Bulanık mantık kullanan bir sistemde, girişe verilen bilgiye karşılık, kural tabanı kullanılarak bulanık çıkış elde edilmektedir. Bu işlem, sistemin karar verme yetkisine sahip çıkarım birimi tarafından yapılmaktadır [38].

Kurallar ve üyelik fonksiyonları oluşturulan bulanık uzman sistem, bu verileri bütünleştirmek, sistemden bir çıktı elde edebilmek için bulanık çıkarım mekanizmasına ihtiyaç duyar. Çıkarım mekanizması, bilgi tabanının değerlendirilmesi için kurulmuş bir uzman sistem aracıdır. Şekil 4.4.’de bulanık çıkarım diyagramı verilmiştir.

Bulanık çıkarım, bulanık mantık kullanarak ayrıntılı olarak verilen bir girişi çıkış haline dönüştüren işlemdir. Başlıca bulanık çıkarım sistemleri, Mamdani, Sugeno ve Tsukamoto tipi bulanık çıkarımlarıdır.



Şekil 4.4. Bulanık Çıkarım Diyagramı

Bu çalışmada Bulanık Sinir ağı olarak Sugeno Bulanık Çıkarım Metodu kullanıldığı için, bu metod hakkında bilgi verilecektir.

4.4.1. Sugeno Bulanık Çıkarım Metodu

Bulanık çıkarım sistemlerinde ilk olarak Mamdani'nin bulanık çıkarım metodu kullanılmıştır. Sugeno bulanık çıkarım metodu ise 1985 yılında verilen giriş ve çıkış kümelerinden bulanık kurallar üretmede sistematik bir yaklaşım geliştirmek için Takagi Sugeno ve Kang tarafından önerilmiştir. Bulanık çıkarım metodlarının ilk iki kısmı olan girişlerin bulanıklaştırılması ve bulanık operatörlere uygulanması tam olarak bir birinin aynısıdır. Mamdami ile Sugeno arasındaki en önemli fark Sugeno bulanık çıkarım metodunun çıkış üyelik fonksiyonunun doğrusal veya sabit olmasıdır [38-41]. Sugeno bulanık modelde tipik bir bulanık kural aşağıdaki gibi gösterilmektedir:

Eğer x A ise ve y B ise $z = f(x, y)$ (4.19)

$z = f(x, y)$ keskin fonksiyon iken A ve B bulanık kümelerdir. $f(x, y)$ x ve y giriş değişkenleri olan bir polinomdur. $f(x, y)$ birinci dereceden bir polinom olduğunda bulanık çıkarım sisteminin sonucu birinci dereceden Sugeno Bulanık Model olarak isimlendirilir. f sabit ise sıfırıncı dereceden Sugeno Bulanık Modeldir ve Mamdani bulanık çıkarım sisteminin veya Tsukamato bulanık modelinin özel bir durumudur. Tsukamoto da kuralların sonucu, adım fonksiyonunun merkezinin üyelik fonksiyonu ile belirlenir.

Her kurala ait çıkış seviyesi olan z_i, kuralların ağırlık ifadesi olan w_i ile ağırlıklandırılmıştır. Örneğin AND kuralı için 1. giriş= x ve 2. giriş= y iken w_i aşağıdaki şekilde ifade edilmektedir:

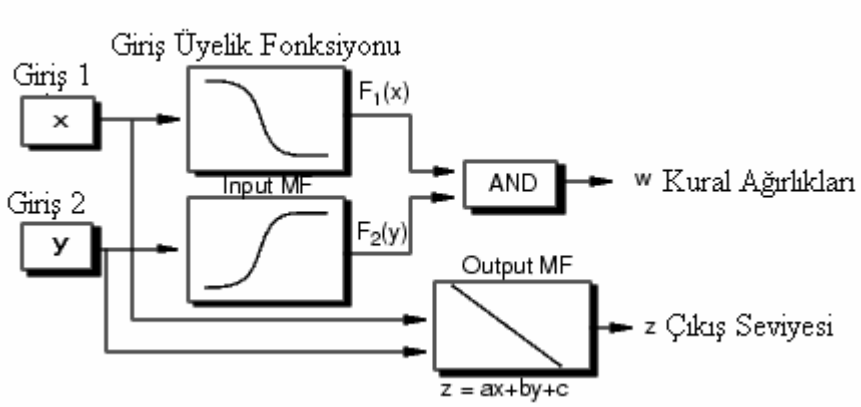
$w_i = AndMethod(F_1(x), F_2(y))$

(4.20)

Burada $F_{1,2}()$ 1. giriş ve 2. giriş'e ait üyelik fonksiyonlarını ifade etmektedir. Sistemin en son çıkış ifadesi tüm kuralların çıkışlarının ağırlıklandırılmış ortalamasıdır ve aşağıdaki şekilde ifade edilmektedir [41]:

$$\text{En son çıkış} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{4.21}$$

Sugeno, bulanık kuralları Şekil 4.5'deki gibi işlenmektedir.



Şekil 4.5. Sugeno Bulanık Çıkarım Metodu

Mamdami'ye göre daha kapsamlı olması ve işlemsel gücü açısından daha üstün olmasından dolayı Sugeno bulanık çıkarım modeli adaptif tekniklerle birlikte bulanık modellerin çözümünde büyük kullanım alanına sahiptir. Bu adaptif teknikler üyelik fonksiyonlarını amaca göre özelleştirebilmekte böylece bulanık sistemler verileri en iyi şekilde modelleyebilmektedir.

4.4.1.1 Sugeno Tipi Modelin Avantajları

- Etkili işlem gücüne sahiptir.
- Doğrusal tekniklerle çok iyi şekilde çalışmaktadır (PID kontrol).
- Optimizasyon ve adaptif tekniklerle iyi bir şekilde çalışmaktadır.
- Çıkış yüzeyinde sürekliliğe sahiptir.
- Matematik analizlerle uyumlu bir şekilde çalışabilmektedir.

4.5. Bulanık Sistem Geliştirme Aşamaları

Bulanık bir Uzman Sistem geliştirme işlemi beş aşamada gerçekleşmektedir. Bunlar aşağıdaki gibi sıralanır:

1. Problemin ve temel dilsel değişkenlerin belirlenmesi,

2. Bulanık kümelerin oluşturulması,
3. Bulanık kuralların belirlenmesi,
4. Bulanık çıkarımı oluşturmak için, bulanık kümeler ve bulanık kuralların uzman sistem haline dönüştürülmesi,
5. Sistemin değerlendirilmesi ve düzeltilmesidir.

4.6. Bulanık Sinir Ağı (Neuro-Fuzzy)

Yapay sinir ağı ve bulanık mantık tekniklerinin her ikisi de temel olarak insan beyninin çalışma mantığına dayanmaktadır; insan beyninin sonuç çıkarma mekanizmasını taklit ederken, YSA'ları beynin fiziksel yapısı temel alınarak oluşturulmuştur. Her iki teknik de kontrol edilecek sistemin matematik modeline ihtiyaç duymazlar, bu sayede karmaşık ve doğrusal olmayan sistemler ile doğrusal sistemleri modelleyebilirler [41-49].

Bulanık sinir ağı, yapay sinir ağları ve bulanık sistemlerin sentezlenmesinden meydana gelen yapay zeka tekniklerinden birisidir. Bulanık mantığın belirsiz bilgileri işleme yeteneğinden ve yapay sinir ağının öğrenme yeteneğinden yararlanabilmek için bu iki teknoloji değişik yöntemlerle birleştirilmektedir [50].

Yapay sinir ağı giriş ve çıkış eğitim çiftleri verilen bir statik fonksiyonu öğrenebilmektedir. Öğrenme işlemi, ağ içerisindeki ağırlıkların belirlenmesiyle gerçekleşmekte ve verilen fonksiyona optimal yaklaşım sağlanmaktadır.

Yapay sinir ağı bulanık sistemin parametrelerini belirlemek için kullanılmaktadır. Öğrenme aşamasından sonra bulanık sistem yapay sinir ağına ihtiyaç duymadan çalışmaktadır. Sistem eğitim yaparken yapay sinir ağını, karar verme işleminde de bulanık mantığı kullanmaktadır. Bulanık sinir ağı ile yapılan farklı bir çok uygulama literatürde mevcttur [35-60].

Bulanık sinir ağı, dört katmandan oluşmaktadır. Bulanıklaştırma katmanı, iki saklı katman, fonksiyon katmanı ve berraklaştırma katmanı [58].

Bulanıklaştırma katmanının girişleri, bulanık çalışma bölgelerini tanımlamak için kullanılan sistem değişkenleridir. Bu katmanda üç tip işlem birimi aktivasyon fonksiyonu kullanılmaktadır. Bunlar; sigmoid, gauss ve ters sigmoid fonksiyonlarıdır.

Kural katmanı bulanık çıkarım gerçekleştirmektedir. Bu katmandaki işlem birimleri sigmoid fonksiyonu kullanmaktadır. Kural katmanının ilk katmanındaki işlem birimleri girişlerinin her biri bir bulanık kümeye karşılık gelmektedir. Son katmandaki her bir işlem biriminin çıkışı ise çalışma bölgelerinin üyelik fonksiyonları olmaktadır.

Fonksiyon katmanındaki işlem birimleri, bulanık çalışma bölgeleri için azaltılmış dereceli modelleri gerçekleştirmektedir. Her bir işlem birimi bir çalışma bölgesine karşılık gelmektedir. İşlem birimlerinin çıkışları, ağırlıklandırılmış sistem değişkenlerinin toplamıdır. Fonksiyon katmanındaki ağırlıklar, çalışma bölgelerindeki doğrusal modellerin parametreleridir.

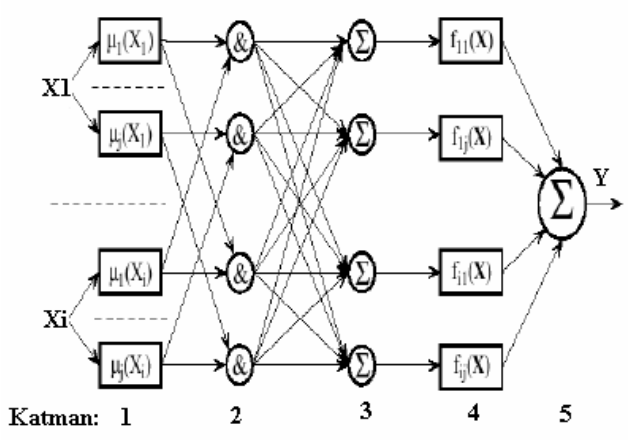
Berraklaştırma işlem biriminin girişi, çalışma bölgeleri ve bu bölgelere ait üyelik fonksiyonlarıdır. Berraklaştırma katmanı, ağırlık yöntemi ile berraklaştırma işlemini gerçekleştirmekte ve ağ çıkışını oluşturmaktadır.

Bulanık mantık kontrolü, sağduyuya dayalı kurallar kullanarak sistemleri tanımlar ve kontrol eder. Bu kurallar, "X İSE Y" – X giriş, Y çıkış – formunda kurallardır. Bulanık mantıkta matematik değişkenler yerine insanın karar verme mekanizmasını taklit edecek şekilde kelimelerle ifade edilen değişkenler kullanılır. Geleneksel mantıkta bir değişkenin gerçeklik değeri ya '1' ya da '0' olabilir. Ancak bulanık mantıkta bu değer 0 ile 1 aralığı içindeki tüm değerleri alabilir. Bu değerlere üyelik değeri adı verilir. Bir girdinin hangi üyelik değerini alacağını belirleyen eğriler, üyelik fonksiyonları olarak adlandırılır. Kullanılan matematiğin basit olması, doğrusal olmayan sistemleri modelleyebilme yeteneği, günlük dile dayalı olması, esnek olması (kesin sınırlardan oluşmaması), ve kesin olmayan bilgiye karşı toleranslı olması bulanık mantığa dayalı sistemlerin kullanımının getirdiği en önemli avantajları arasındadır.

YSA'ları ise beynin fizyolojik yapısı temel alınarak tasarlanan sistemler olup birbirleriyle paralel çalışan çok sayıda basit işlemciden oluşurlar. Bir sinir ağının davranış biçimini büyük oranda nöronlar arasındaki bağlantılar belirler; öğrenilen tüm bilgi bu bağlantılarda saklıdır. YSA'ları belirli bir işlevi yerine getirmek için eğitilebilirler ve bu eğitim bağlantıların değerlerini değiştirmekle sağlanır. YSA'larını geleneksel sistemlerden ayıran en önemli özellikleri öğrenme, genelleme yapabilme ve

paralel çalışma özellikleridir. Bu özellikler yapay sinir ağlarına hız, hataya karşı tolerans ve verimlilik gibi avantajlar kazandırmaktadır.

Şekil 4.6'da birçok uygulamada kullanılan ANFIS (Adaptive Neuro Fuzzy Inference System-Adaptif Bulanık Sinir Ağı Çıkarım Sistem) yapısına ait şekil verilmiştir.



Şekil 4.6. Basit Bir ANFIS Yapısı

Giriş ve çıkış eğitim çiftleri verilen bir fonksiyonu, yapay sinir ağı öğrenebilmektedir. Öğrenme işlemi, ağ içerisindeki ağırlıkların belirlenmesiyle gerçekleşmekte ve verilen fonksiyona optimal yaklaşım sağlanmaktadır. Böylece yapay sinir ağları sistemi öğrenmektedir.

Bulanık mantığın belirsiz bilgileri işleme ve yapay sinir ağının öğrenme yeteneği sistem modelleme de kullanılmıştır [44]. Aynı zamanda bulanık mantık ve yapay sinir ağının birleştirilmesi, birbirlerinin dezavantajlarını da örtmektedir. Sistem modellemeye, hem bulanık mantık hem de yapay sinir ağı açısından ayrı ayrı bakıldığında, modelleme aşamaları Tablo 4.1' deki gibidir:

Tablo 4.1. Bulanık Mantık ve YSA Açısından Sistem Modelleme Adımları

Adım	**Bulanık Mantık açısından**	**Yapay Sinir Ağı açısından**
1	giriş değişkenlerinin belirlenmesi	giriş değişkenlerinin belirlenmesi
2	gerekli olan üyelik fonksiyonu ve kural sayısının belirlenmesi	ağ yapısının ve başlangıç ağırlıklarının belirlenmesi
3	Üyelik fonksiyonlarının ayarlanması	ağın eğitilmesi

4	gereksiz kuralların belirlenmesi	gereksiz hücrelerin iptal edilmesi

4.7. Bulanık Sinir Ağı ile Sistem Modelleme

Matematik modeli verilen fiziksel bir sistemin analizi yapılabilir veya değişik girişlere karşılık çıkışları kestirilebilir. Bununla beraber, sistemin matematik modelinin belirlenmesinde zorluklar çıkabilmektedir. Basit bir sistem için, Newton yöntemi kullanılarak bir model üretilebilir. Fakat karmaşık sistemerde bu yöntem yeterli olmamaktadır [48].

Giriş ve çıkış eğitim çiftleri verilen bir statik fonksiyonu, yapay sinir ağı öğrenebilmektedir. Öğrenme işlemi, ağ içerisindeki ağırlıkların belirlenmesiyle gerçekleşmekte ve verilen fonksiyona optimal yaklaşım sağlanmaktadır. Ancak yapay sinir ağı ile yapılan modelleme işleminde fazla sayıda ağırlık kullanılmaktadır. Bu da geliştirilen model üzerinde yorum yapılmasını güçleştirmektedir. Ayrıca sınırlı veya gürültülü eğitim bilgisi kullanıldığında sistem çıkışından tutarsız veya anlamsız bilgi alınmaktadır.

Bulanık mantığın belirsiz bilgileri işleme yeteneğinden ve yapay sinir ağının öğrenme yeteneğinden yararlanabilmek için, bu iki teknoloji birleştirilebilmektedir. Aynı zamanda bulanık mantık ve yapay sinir ağının birleştirilmesi, birbirlerinin dezavantajlarını da örtmektedir.

BÖLÜM 5

SİSTEM MODELLEME UYGULAMALARI

5.1 Yapay Sinir Ağı İle Sistem Modelleme

Bu bölümde, Yapay sinir ağları ile doğrusal ve doğrusal olmayan sistemler modellenmiştir.

5.1.1 YSA İle Doğrusal Sistem Modelleme

1. Sistem

Doğrusal sistemlerden olan AR farklı öğrenme algoritmalarına göre eğitilmiştir. Bu çalışmada zamanla değişen bir transfer fonksiyonuna giriş işareti olarak 5.1 eşitliğinde verilen X işareti uygulanmıştır.

$$X = \sin(\sin(2t) \times 4t) \tag{5.1}$$

Peryod 6 saniye alınarak, her saniyede 200 adet örnek alınmıştır. Transfer fonksiyonu;

$H(z) = \dfrac{1}{1.1 - 0.5 * z^{-1}}$ olarak verilmiştir. Bu fonksiyon 4 saniye sonra $H(z) = \dfrac{1}{5 - z^{-1}}$

formuna dönüşmüştür.

Bu sisteme gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 2400 adet giriş dizisi uygulanmış ve istenen çıkış dizisi elde edilmiştir. Sistemler gürültülü ve gürültüsüz durumlar için ayrı ayrı test edilmiştir. Buradaki YSA yapısı 3 giriş, 10 ve 20 nörondan oluşan iki adet gizli katman ve bir adet nöronun yer aldığı çıkış katmanından oluşmaktadır. YSA yapıları için ara katmanlarda hiperbolik tanjat sigmoid, çıkış katmanında ise doğrusal transfer fonksiyonları kullanılmıştır. Kullanılan ağ yapıları deneme yanılma yoluyla en iyi sonucu verecek şekilde elde edilmiştir. Model çıkışı ile sistem çıkışı arasındaki farkın RMS ve MSE değerleri ve hesaplama süreleri gürültüsüz

durumlar için Tablo 5.1'de verilmiştir. Bu tabloda test işlemi sonucunda meydana gelen hataların RMS ve MSE değerleri verilmiştir.

2. Sistem

İkinci olarak ARMA sistem incelenmiştir. ARMA sisteme ait çıkış fonksiyonu;

$$y(n)=x(n)+0.433*x(n-1)+0.490*x(n-2)+2.595*y(n-1)-3.339*y(n-2)+2.2*y(n-3)-0.731*y(n-4); \quad (5.2)$$

olarak alınmıştır. EK-2'de bu sisteme ait program verilmiştir.

Tablo 5.1. Gürültüsüz Durumda 1. Sistem ve 2. Sistem için YSA Eğitim Algoritmalarının Modellemedeki Performansları

Sistemler	**YSA Algoritmaları**	**Gürültüsüz Durum**				
		RMS		**MSE**		**Eğitim Süresi(s)**
		Eğitim	**Test**	**Eğitim**	**Test**	
1. Sistem	Trainbfg	0.0299	0.1168	$9.0105x10^{-4}$	0.0137	133.2580
	Trainb	0.2735	0.2858	0.0748	0.0817	74.8220
	Trainbr	0.0147	0.2484	$2.6416x10^{-4}$	0.0617	134.8040
	Traingd	0.2614	0.2737	0.0683	0.0749	16.0030
	Traincgf	0.0361	0.1315	0.0013	0.0173	32.4870
	Traincgb	0.0370	0.1356	0.0014	0.0184	32.7070
	Trainlm	0.0037	0.3641	$1.3389x10^{-5}$	0.1326	123.3770
2. Sistem	Trainbfg	0.0021	2.2324	$4.4359x10^{-6}$	4.9837	165.3840
	Trainb	6.0570	9.5014	36.6113	90.2764	111.6770
	Trainbr	$1.8716x10^{-2}$	0.8522	$3.7182x10^{-4}$	0.4553	260.2381
	Traingd	16.4857	20.0910	271.7792	403.6466	118.8270
	Traincgf	3.2861	6.5369	10.7987	42.7315	114.2123
	Traincgb	0.2882	1.6782	0.0830	2.8163	121.6710
	Trainlm	$9.7987x10^{-5}$	0.5960	$9.6014x10^{-9}$	0.3552	140.7120

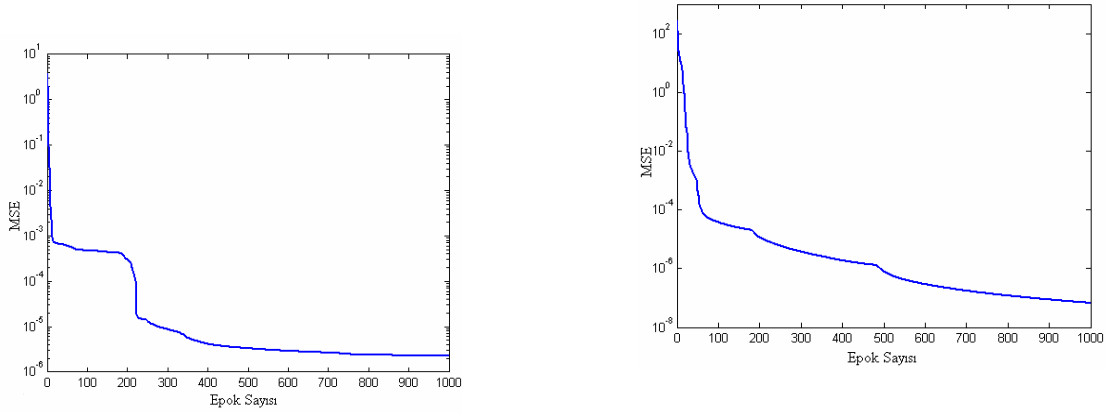
Buradaki değişkenlerin başlangıç değerleri sıfır olarak alınmıştır. Bu sisteme gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 500 adet giriş dizisi uygulanmış ve istenen çıkış dizisi elde edilmiştir. Sistemler gürültülü ve gürültüsüz durumlar için ayrı ayrı test edilmiştir. Buradaki YSA yapısı 7 giriş, 8 ve 10 nörondan oluşan iki adet gizli katman ve bir adet nöronun yer aldığı çıkış katmanından oluşmaktadır. Eğitim için 300 adet, test için 200 adet veri kullanılmıştır. YSA yapıları için ara katmanlarda hiperbolik tanjat sigmoid, çıkış katmanında ise doğrusal transfer fonksiyonları kullanılmıştır. Kullanılan ağ yapıları deneme yanılma yoluyla en iyi sonucu verecek

şekilde elde edilmiştir. Model çıkışı ile sistem çıkışı arasındaki farkın RMS ve MSE değerleri ve hesaplama süreleri gürültülü durum için Tablo 5.2'de verilmiştir. Gürültü olarak 20 dB'lik beyaz gürültü kullanılmıştır.

Tablo 5.2. Gürültülü Durumda 1. Sistem ve 2. Sistem için YSA Eğitim Algoritmalarının Modellemedeki Performansları

Sistemler	**YSA Algoritmaları**	**Gürültülü Durum**				
		RMS		**MSE**		**Eğitim Süresi(s)**
		Eğitim	**Test**	**Eğitim**	**Test**	
1. Sistem	Trainbfg	0.0303	0.2469	$6.0853\text{x}10^{-4}$	0.0609	183.2970
	Trainb	0.3225	0.3324	0.0843	0.0923	97.2330
	Trainbr	0.2081	0.2761	$4.3316\text{x}10^{-4}$	0.0516	164.8040
	Traingd	0.3999	0.3185	0.0788	0.0953	32.0043
	Traincgf	0.0466	0.1320	0.0022	0.0184	38.6970
	Traincgb	0.0451	0.1517	0.0020	0.0182	39.3190
	Trainlm	0.0256	0.1655	$6.5554\text{x}10^{-4}$	0.0274	172.5360
2. Sistem	Trainbfg	0.0690	2.1797	0.0048	4.7511	163.8920
	Trainb	8.0507	9.5014	64.6113	121.5030	112.2470
	Trainbr	$2.4480\text{x}10^{-2}$	0.8760	$4.6081\text{x}10^{-4}$	0.4852	271.8750
	Traingd	16.3527	20.7518	267.4101	430.6371	118.8270
	Traincgf	5.3278	9.3582	28.3858	87.57500	124.3858
	Traincgb	0.2655	1.3828	0.0830	2.8163	121.6710
	Trainlm	$2.9224\text{x}10^{-4}$	3.0314	$8.5405\text{x}10^{-8}$	0.3552	150.7460

Tablo 5.1 ve 5.2'de verilen 1. Sistem ve 2. Sistem farklı öğrenme algoritmalarıyla eğitilmiştir. Bu sonuçlara göre elde edilen sonuçlar gürültüsüz durum için kabul edilebilir olsa da YSA'lar gürültülü sistemler için de iyi sonuçlar vermiştir. EK-1' YSA algoritmalarının içerisinde sistem modellemeye, Tablo 5.1 ve 5.2'den de görüleceği gibi, en iyi sonucu LM öğrenme algoritması vermiştir.

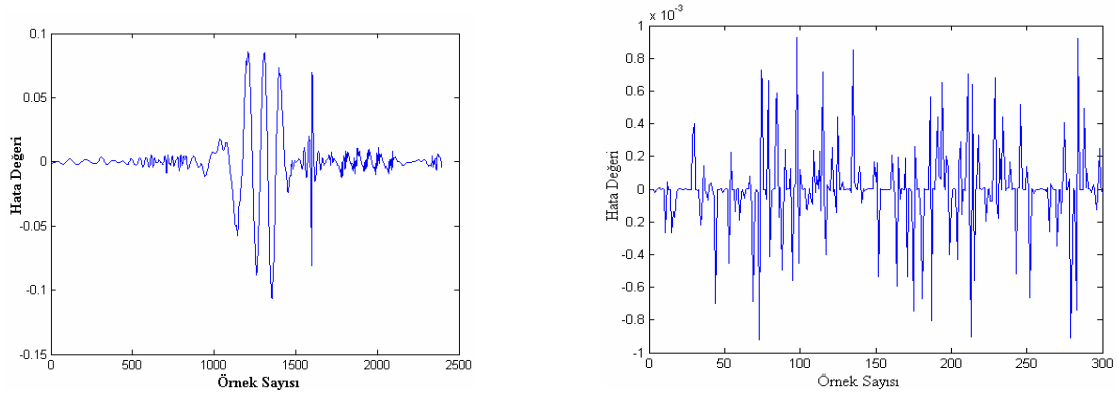


(a) (b)

Şekil 5.1.a) 1. Sistem b) 2. Sistem için LM eğitim algoritması ile eğitim esnasında oluşan hata değerleri.

1. Sistem ve 2. Sistem için LM eğitim algoritması ile eğitim esnasında oluşan hata değerleri Şekil 5.1'de verilmiştir.

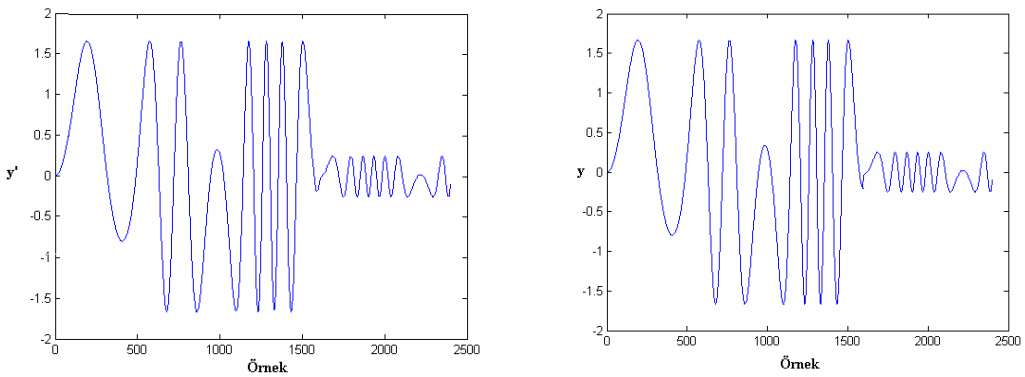
Şekil 5.2.'de 1. Sistem ve 2. Sistem için sistem çıkışı ile model çıkışı arasındaki hata değeri gösterilmiştir



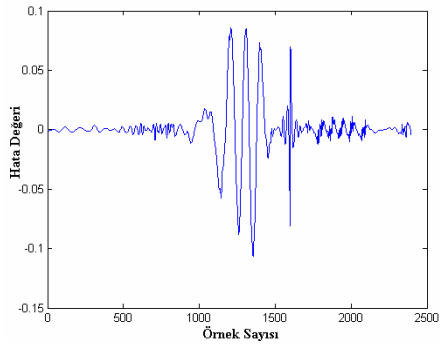
(a) (b)

Şekil 5.2. a) 1. Sistem b) 2. Sistem için Sistem çıkışı ile model çıkışı arasındaki hata değeri.

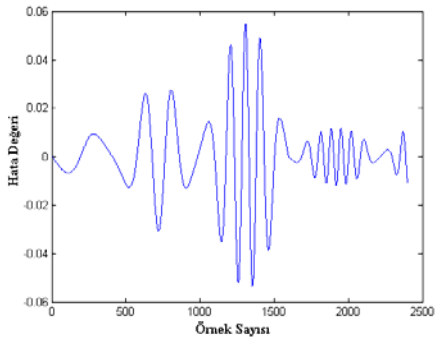
Şekil 5.3'de ise 1. Sistem'in sistem çıkışı, model çıkışı, sistem çıkışı ile model çıkışı arasındaki fark ve sistem çıkışı ile test çıkışı arasındaki fark gösterilmiştir.



(a) (b)



(c)



(d)

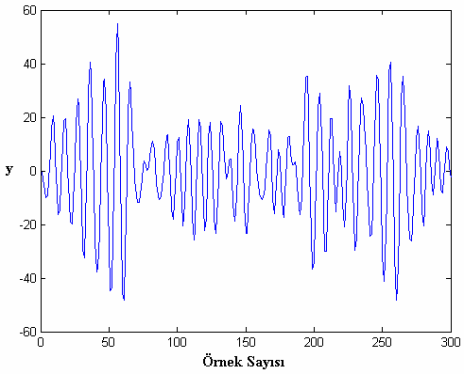
Şekil 5.3. 1. Sistem için;

a) Sistem çıkışı,

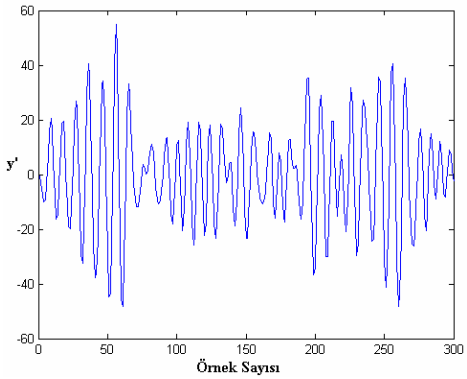
b) Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki fark,

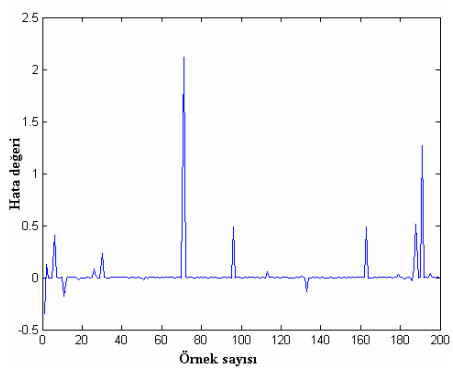
d) Testteki hata değeri.



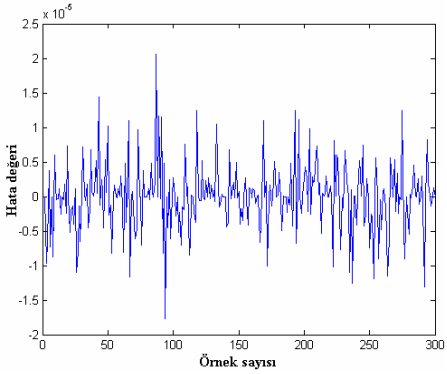
(a)



(b)



(c)



(b)

Şekil 5.4. 2. Sistem için;

a) Sistem çıkışı,

b) Model çıkışı,

c) Testteki hata miktarı,

d) Sistem çıkışı ile model çıkışı arasındaki fark.

5.1.2 YSA İle Doğrusal Olmayan Sistem Modelleme

Bu bölümde, bölüm 2.7.2'de bahsedilen bilineer sistemlerin YSA ile modellenmesi ele alınmıştır. Ayrıca bu sistem de hem gürültülü hem de gürültüsüz durum için test edilmiştir. En iyi öğrenme algoritmasının LM öğrenme algoritması olduğu deneme yanılma metodu ile bulunmuştur.

1. Sistem

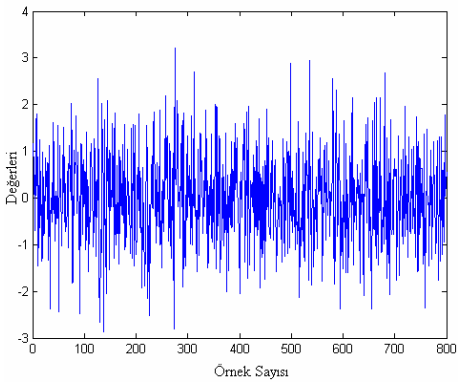
$$y(n)= 0.25*y(n-1) - 0.5*y(n-1)*x(n) + 0.05*y(n-1)*x(n-1) -0.5*x(n) + 0.5*x(n-1) \quad (5.3)$$

Bu sisteme gauss dağılımında, sıfır ortalamalı ve standart sapma değeri 1 olan n=800 adet elemana sahip giriş dizisi uygulanmış ve istenen çıkış dizisi elde edilmiştir. Giriş değişkenlerinin başlangıç değerleri sıfır olarak alınmıştır. Gürültü olarak gauss dağılımında, sıfır ortalamalı standart sapma değeri 1 olan dizi kullanılmıştır.

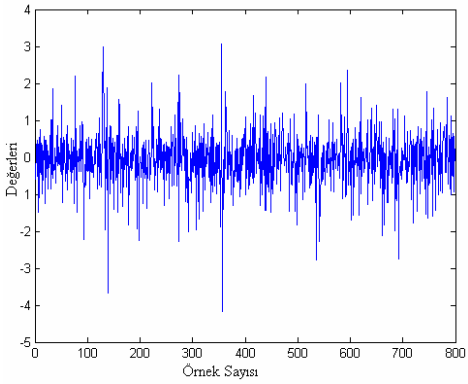
YSA yapısı giriş, iki adet gizli katman ve 1 adet nöronun yer aldığı çıkış katmanından ibarettir. YSA yapıları için ara katmanlarda hiperbolik tanjant sigmoid, çıkış katmanında ise doğrusal transfer fonksiyonu seçilmiştir. Kullanılan ağ yapıları deneme yanılma yoluyla en iyi sonucu verecek şekilde elde edilmiştir. YSA modeli, öğrenmede kullanılmayan başka bir gauss dağılımında, sıfır ortalamalı, standart sapma değeri 1 olan 200 elemanlı dizi ile teste tabii tutulmuştur. Deneme yanılma yoluyla bu sistem en iyi LM öğrenme algoritması ile eğitilmiştir. Farklı öğrenme algoritması ile eğitilmiş model çıkışı ile sistem çıkışı arasındaki farkın MSE değerleri ve korelasyon değerleri, gürültülü ve gürültüsüz durumlar için Tablo 5.3'de verilmiştir.

Tablo 5.3. Bilineer Sistem için Farklı Öğrenme Algoritmalarının Hata değerleri

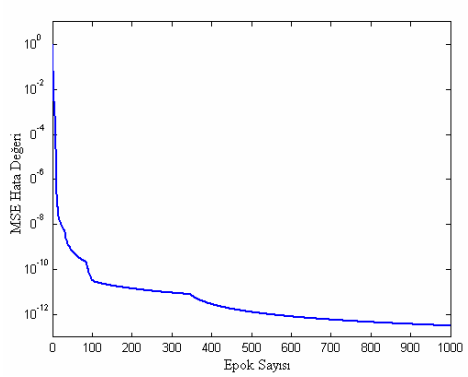
Sistemler	**YSA Algoritmaları**	**Gürültüsüz**		**Gürültülü**	
		MSE		**MSE**	
		Eğitim	**Korelasyon**	**Eğitim**	**Korelasyon**
Sistem1#	Trainbfg	$2.25x10^{-4}$	99.85	0.0424	95.26
	Trainb	0.0397	96.13	0.0532	94.32
	Trainbr	0.0055	98.89	0.0126	98.12
	Trainlm	$1.29x10^{-9}$	100	0.0113	98.29

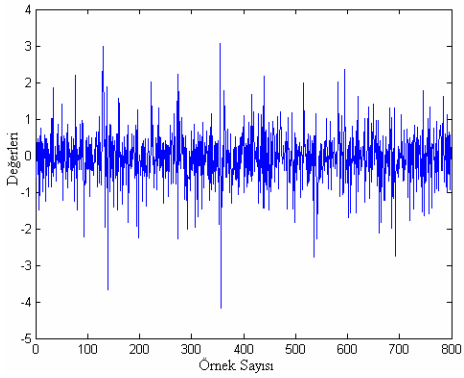

Değerleri
Örnek Sayısı

(a)


Değerleri
Örnek Sayısı

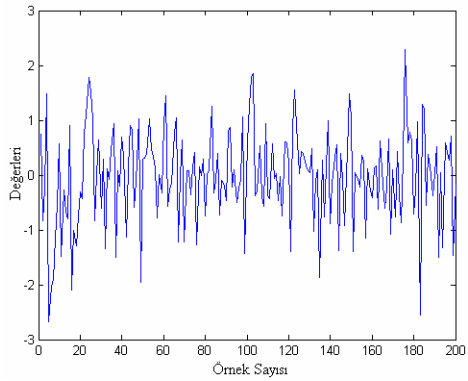
(b)

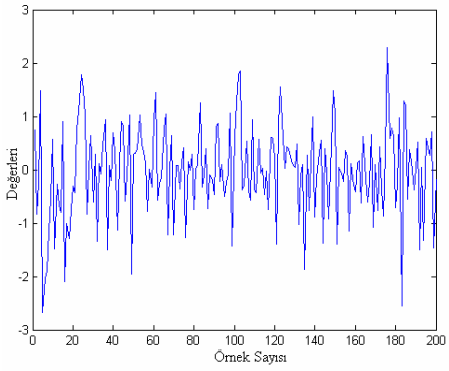

MSE Hata Değeri
Epok Sayısı


Değerleri
Örnek Sayısı

(c)

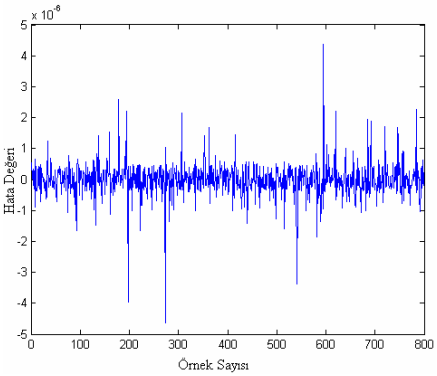
(d)


Değerleri
Örnek Sayısı

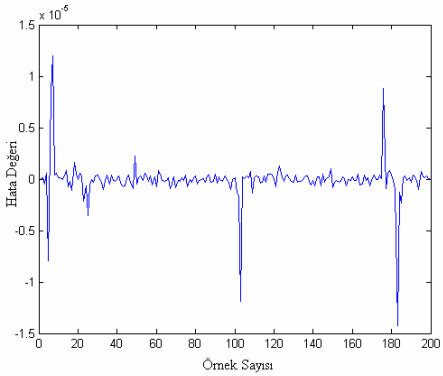

Değerleri
Örnek Sayısı

(e)

(f)



(g)



(h)

Şekil 5.5. 1. Sistem için;

a) Sistem girişi,

b) Sistem çıkışı,

c) LM eğitim algoritması ile eğitim esnasında oluşan hata değerleri,

d) Model çıkışı,

e) Sistemin test çıkışı,

f) Modelin test çıkışı,

g) Sistem çıkışı ile model çıkışı arasındaki fark,

h) Testeki hata miktarı.

Yapılan simülasyon sonuçlarına göre farklı öğrenme algoritmalarıyla eğitilen YSA yapıları ele alındığında gürültüsüz durumda elde edilen sonuçlar gürültülü duruma göre çok iyi olmaktadır fakat sistemin gürültülü olması durumunda elde edilen sonuçlar da kabul edilebilir şekildedir. YSA algoritmaları içerisinde LM öğrenme algoritması bilineer sistem modelleme açısından oldukça iyi sonuçlar veren algoritma olmuştur. Gürültülü ve gürültüsüz durumlar için YSA kullanılarak düşük modelleme hatası ile modelleme yapılabileceği ve sistem davranışının YSA tarafından iyi bir şekilde öğrenildiği gözlenmiştir.

Tablo 5.4.Değişik Gürültü Değerlerine Göre Sistemlerin MSE ve RMS Hataları

Sistemler	**Gürültü Değerleri**	**RMS**		**MSE**		**Eğitim Korelasyon**
		Eğitim	**Test**	**Eğitim**	**Test**	
AR Sistem	20 dB	0.0256	0.1655	6.5553×10^{-4}	0.0274	0.9998
	40 dB	0.0138	0.2067	1.9045×10^{-4}	0.0427	0.9999
	60 dB	0.0043	0.3668	1.8782×10^{-5}	0.1346	1
	80 dB	0.0040	0.6532	1.5094×10^{-5}	0.4267	1
ARMA	20 dB	2.9224×10^{-4}	0.0314	8.5405×10^{-8}	0.3552	1

Sistem	40 dB	$1.0950x10^{-4}$	0.4475	$1.9119x10^{-8}$	0.2002	1
	60 dB	$2.3316x10^{-6}$	0.0007	$5.4365x10^{-12}$	$6.66x10^{-7}$	1
	80 dB	$1.672x10^{-10}$	0.0001	$1.1389x10^{-14}$	$1.1x10^{-7}$	1
Bilineer Sistem	20 dB	$4.5436x10^{-5}$	0.0324	$2.0645x10^{-9}$	0.0010	1
	40 dB	$3.4411x10^{-5}$	0.0014	$1.1635x10^{-9}$	$1.91x10^{-6}$	1
	60 dB	$1.743x10^{-5}$	0.0006	$3.038x10^{-10}$	$4.49x10^{-5}$	1
	80 dB	$1.634x\ 10^{-6}$	0.0003	$2.9944x10^{-12}$	$1.03x10^{-7}$	1

Tablo 5.4.'de değişik miktarlarda uygulanan gürültünün sistemleri nasıl etkilediği gösterilmiştir. Gürültü miktarı düştükçe sistemin MSE ve RMS hata değerleri düşmektedir.

5.2. Bulanık Sinir Ağı ile Sistem Modelleme

Bu bölümde, Bulanık sinir ağları ile doğrusal ve doğrusal olmayan sistemler modellenmiştir. Bölüm 4'te anlatılan basamaklar ile sistemler modellenmiştir. Programlama için ANFIS yapısından yararlanılmıştır. ANFIS, Sugeno tip FIS tek çıkışlı sistemin üyelik fonksiyon parametrelerini belirlemek için hibrid (hybrid) öğrenme algorimasını kullanır. LS ve BGD (Backpropogation gradient descent-gradyan azaltılmış geri yayılım) metodunun birleşimini, giriş-çıkış bilgileri verilen FIS üyelik fonksiyon parametrelerinin eğitimi için kullanır.

İlk olarak doğrusal sistemlerden olan, AR, MA ve ARMA sistemler bulanık sinir ağı ile modellenmiştir. Daha sonra da Doğrusal olmayan sistemlerin modellenmesi gerçekleştirilmiştir.

5.2.1. Bulanık Sinir Ağı İle Doğrusal Sistem Modelleme

Burada üç tip doğrusal sistem incelenmiştir. Birinci sistem AR, ikinci sistem MA ve en son sistem de ARMA'dır.

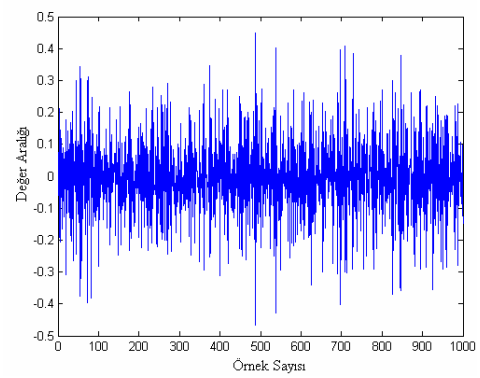
1. Sistem

Transfer fonksiyonu;

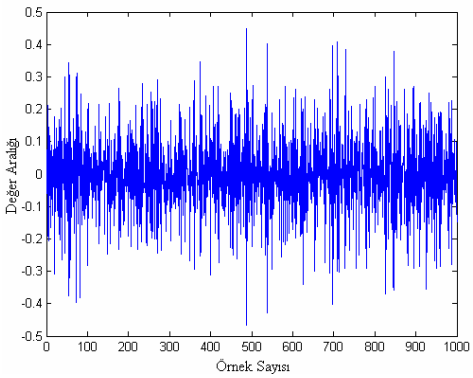
$$H(z) = \frac{1}{1 + 0.8 * z^{-1} + 0.15 * z^{-2}} \tag{5.4}$$

olan bir AR model alınmıştır. Bu sisteme gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 1000 adet giriş dizisi uygulanmış ve istenen çıkış dizisi elde edilmiştir. Ayrıca bu modelin testini yapmak içinde sisteme gauss dağılımı sıfır ortalamalı,

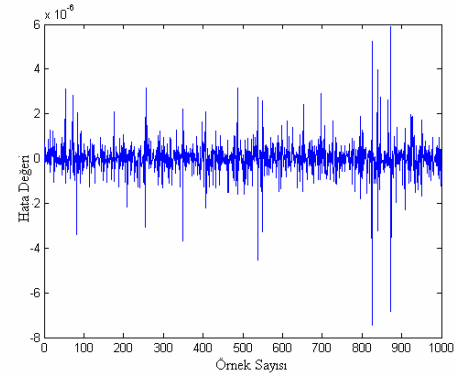
standart sapma değeri 1 olan 500 adet test giriş dizisi uygulanmıştır. Böylece 500x3'lük bir giriş dizisi oluşturulmuştur. ANFIS yapısı kullanıldığı için de bir tane çıkışı olacaktır. Yine ANFIS yapısından dolayı Sugeno tip bulanık çıkarım metodu kullanılmıştır.



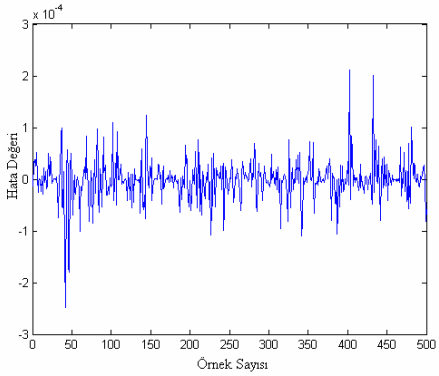
(a)



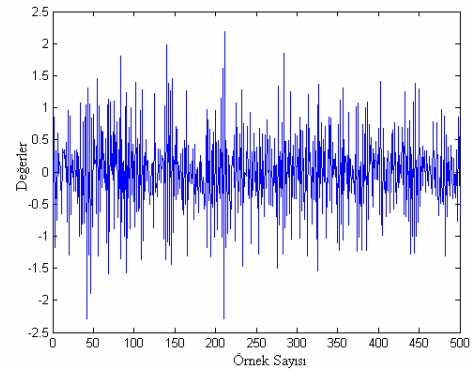
(b)



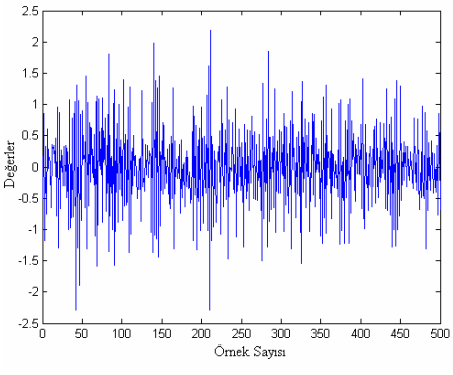
(c)



(d)



(e)



(f)

Şekil 5.6. 1. Sistem için;

a) Sistem çıkışı,

b) Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki fark,

d) Test hatası,

e) Sistemin test çıkışı,

f) Modelin test çıkışı.

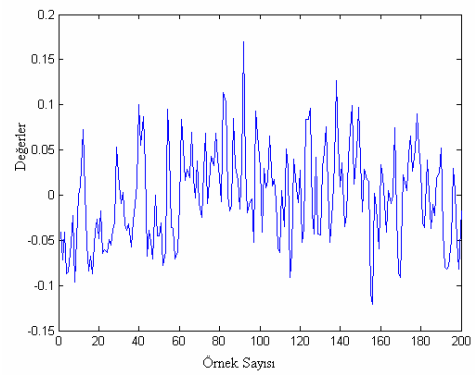
Bu yapıda 32 tane doğrusal parametre, 18 tane doğrusal olmayan toplam 50 parametre kullanılmıştır. Ayrıca 8 tane bulanık kural oluşturulmuştur. Eğitim için 1000 adet veri, test içinde 500 adet veri kullanılarak AR sistem modellenmiş ve Şekil 5.6'daki sonuçlar elde edilmiştir. Bu sonuçlara göre sistem ile model çıkışı birbirine çok benzemektedir aralarındaki hata ihmal edilebilir düzeydedir.

2. Sistem

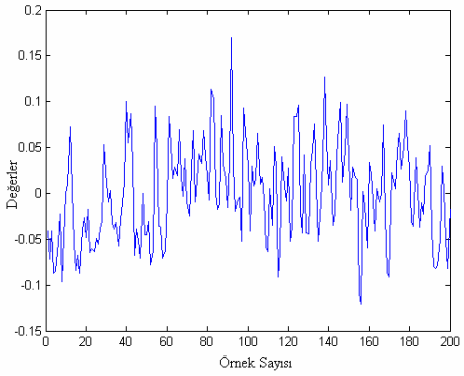
Bu çalışmada;

$$H(z) = 0.3 + 0.5 * z^{-1} \tag{5.5}$$

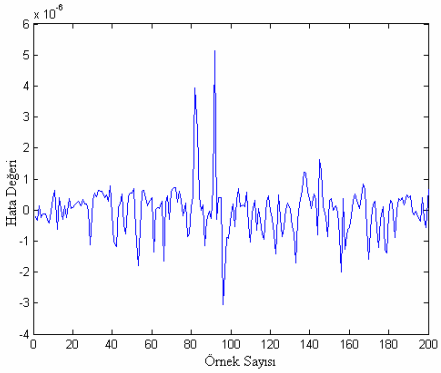
olan MA yapıda bir sistem modellenecektir. 2 girişli ve tek çıkışlı ANFIS yapısı kullanılmıştır. Giriş olarak gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 200 adet giriş dizisi uygulanmış ve istenen çıkış dizisi elde edilmiştir. Ayrıca bu modelin testini yapmak içinde sisteme gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 100 adet test giriş dizisi uygulanmıştır. Simülasyon sonuçlarından elde edilen şekiller, Şekil 5.7.'de verilmiştir.



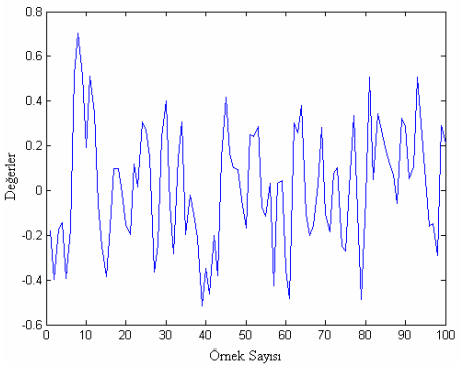
(a)



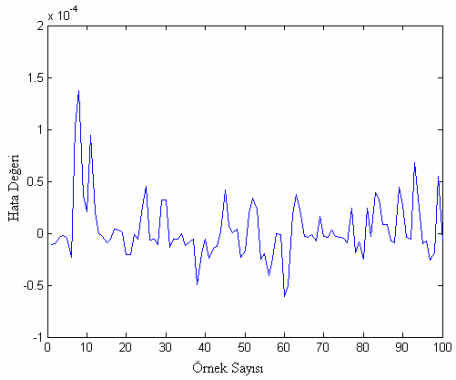
(b)



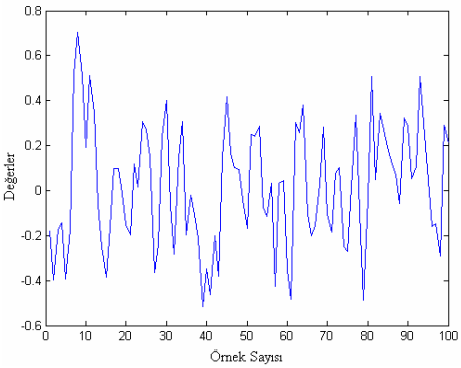
(c)



(d)



(e)



(f)

Şekil 5.7. 2. Sistem için;

a) Sistem çıkışı,

b) Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki hata değeri,

d) Sistemin test çıkışı,

e) Test hatası,

f) Modelin test çıkışı.

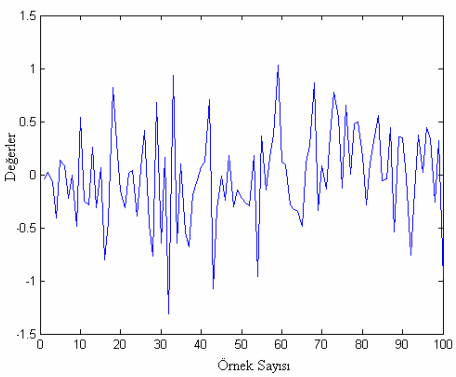
Bu yapıda 12 tane doğrusal parametre, 12 tane doğrusal olmayan toplam 24 parametre kullanılmıştır. Ayrıca 4 tane bulanık kural oluşturulmuştur. Eğitim için 200 adet veri, test içinde 100 adet veri kullanılarak MA sistem modellenmiş ve Şekil 5.7'deki sonuçlar elde edilmiştir. Bu sonuçlara göre sistem ile model çıkışı birbirine çok benzemektedir aralarındaki hata ihmal edilebilir düzeydedir.

3. Sistem

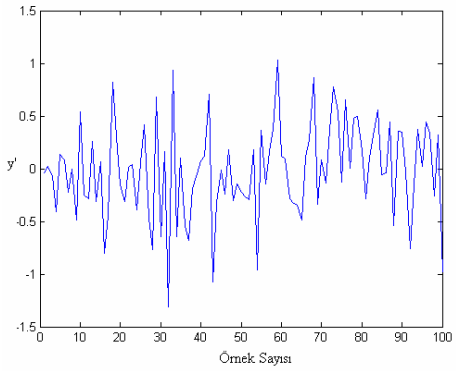
Bu çalışmada da hem AR hem de MA sistemin birleşiminden oluşan ARMA sistemin simülasyonu yapılmıştır. Transfer fonksiyonu (5.6) eşitliğinde verilen sistem incelenmiştir.

$$H(z) = \frac{0.3 + 0.5 * z^{-1}}{1 + 0.8 * z^{-1} + 0.15 * z^{-2}} \tag{5.6}$$

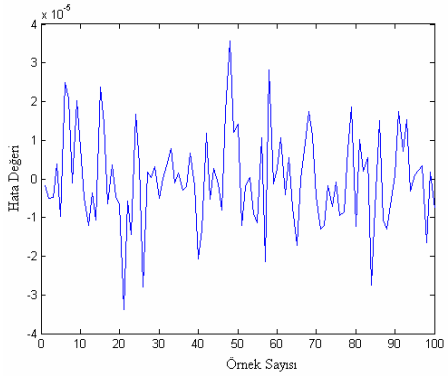
Bu çalışmada 4 girişli ve tek çıkışlı ANFIS yapısı kullanılmıştır. Giriş olarak gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 100 adet giriş dizisi uygulanmış ve istenen çıkış dizisi elde edilmiştir. Ayrıca bu modelin testini yapmak içinde sisteme gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 50 adet test giriş dizisi uygulanmıştır. Simülasyon sonuçlarından elde edilen şekiller, Şekil 5.8'de verilmiştir. Bu sisteme ait program Ek-1'de verilmiştir.



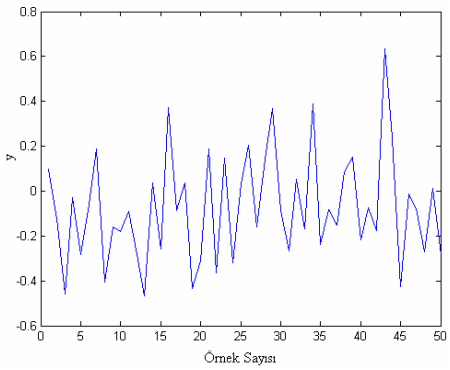
(a)



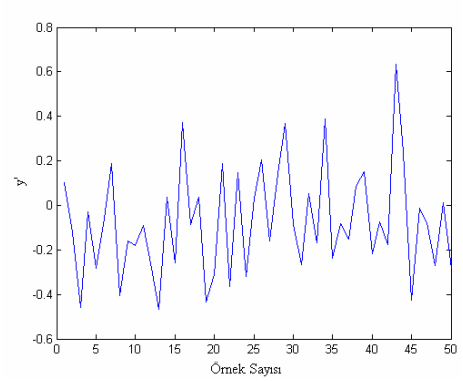
(b)



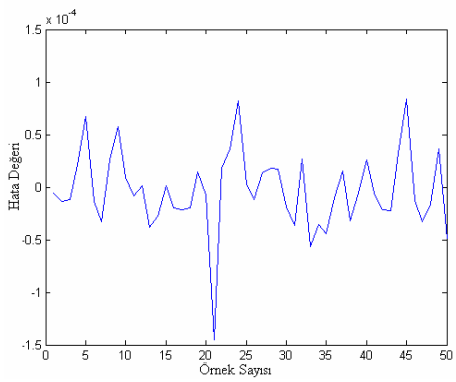
(c)



(d)



(e)



(f)

Şekil 5.8. 3. Sistem için;

a) Sistem çıkışı,

b) Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki fark,

d) Sistemin test çıkışı,

e) Modelin test çıkışı,

f) Test hatası.

Bu yapıda 80 tane doğrusal parametre, 24 tane doğrusal olmayan toplam 104 parametre kullanılmıştır. Eğitim için 100 adet veri, test içinde 50 adet veri kullanılarak ARMA sistem modellenmiş ve Şekil 5.8'deki sonuçlar elde edilmiştir. Bu sonuçlara göre sistem ile model çıkışı birbirine çok benzemektedir aralarındaki hata ihmal edilebilir düzeydedir. Bulanık kural sayısı 16'dır. Testteki hata miktarı eğitimdekinden büyük de olsa göz ardı edilebilir seviyededir.

Tablo5.5.Doğrusal Sistemlerin Farklı Üyelik Fonksiyonlarına Verdiği MSE Hata Değeri

Sistem	**Trimf**	**Dsigmf**	**Gaussmf**	**Gbellmf**	**Psigmf**
AR (1. Sistem)	$1.093x10^{-15}$	$1.853x10^{-14}$	$2.191x10^{-12}$	$2.573x10^{-12}$	$7.274x10^{-15}$
MA (2. Sistem)	$7.322x10^{-10}$	$3.557x10^{-10}$	$1.611x10^{-9}$	$2.941x10^{-10}$	$9.919x10^{-10}$
ARMA(3.Sistem)	$2.283x10^{-13}$	$1.077x10^{-9}$	$1.094x10^{-8}$	$1.573x10^{-8}$	$6.208x10^{-10}$

Tablo 5.5'e göre AR sistemde trimf üyelik fonksiyonu, MA sistemde gbell üyelik fonksiyonu ve ARMA sistemde yine trimf üyelik fonksiyonu, diğer fonksiyonlara göre daha iyi sonuç vermiştir.

4. Sistem

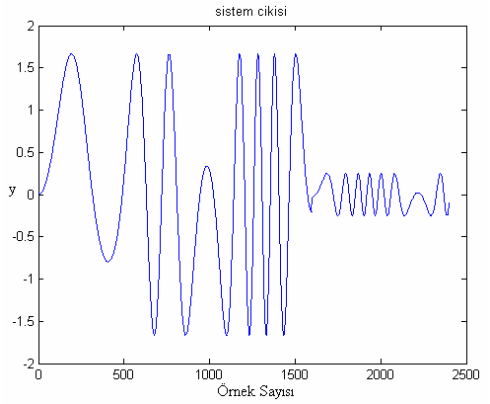
Burada YSA ile yapılan zamanla değişen AR sistem incelenmiştir.

$$H(z) = \frac{1}{1.1 - 0.5 * z^{-1}} \tag{5.7}$$

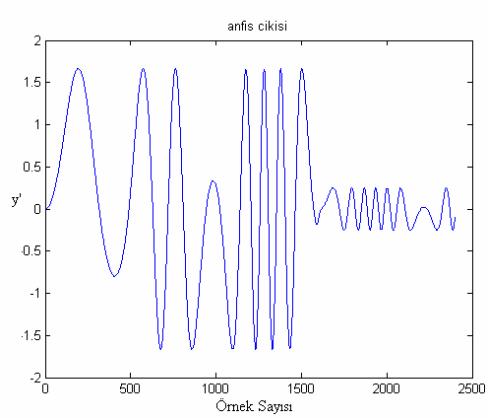
olarak verilmiştir. Bu fonksiyon 4 saniye sonra

$$H(z) = \frac{1}{5 - z^{-1}} \tag{5.8}$$

formuna dönüşmüştür.

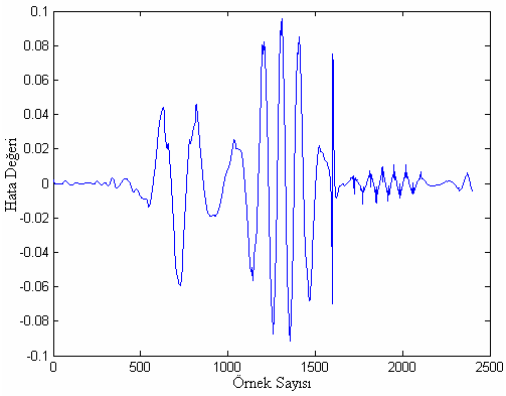


(a)



(b)

Şekil 5.9. 4. Sistemin
a) Sistem çıkışı,
b)Model çıkışı,
c)Sistem çıkışı ile model çıkışı arasındaki fark.



(c)

Bu sistemin hem YSA hem de bulanık sinir ağı ile modellenmesi sonucunda oluşan RMS, MSE ve korelasyon değerleri tablo 5.6'da verilmiştir.

Tablo 5.6. AR Sistemin YSA ve Bulanık Sinir Ağına Göre Hata değerleri

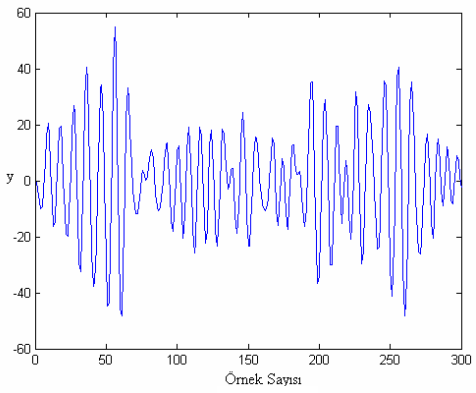
Yapay Zeka Tekniği	MSE	RMS	Korelasyon
Yapay Sinir Ağı	6.5554×10^{-4}	0.0244	0.09997
Bulanık Sinir Ağı	6.7596×10^{-4}	0.0260	0.09995

Bu yapıda 75 tane doğrusal parametre, 40 tane doğrusal olmayan toplam 115 parametre kullanılmıştır. Eğitim için 2400 adet veri kullanılarak AR sistem modellenmiş ve Şekil 5.9'daki sonuçlar elde edilmiştir. Bu sonuçlara göre sistem ile model çıkışı birbirine çok benzemektedir aralarındaki hata ihmal edilebilir düzeydedir. Bulanık kural sayısı 25'dır. Testteki hata miktarı eğitimdekinden büyük de olsa göz ardı edilebilir seviyededir. Bu sisteme deneme yanılma yöntemiyle trigonometrik (trimf) üyelik fonksiyonunun diğer üyelik fonksiyonlarına göre daha iyi sonuç verdiği görülmüştür.

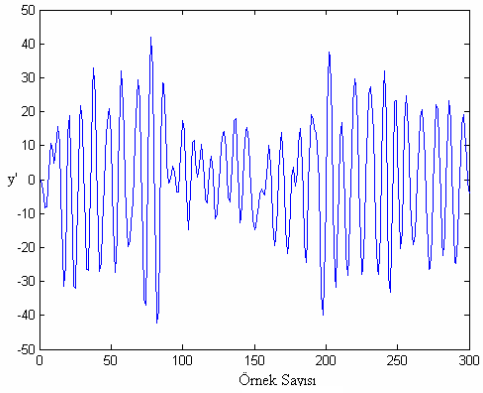
Tablo 5.6'daki değerlere göre YSA ve bulanık sinir ağı sistem modellemede başarılı olmuştur. Aralarındaki hata değerleri bibirine çok yakındır dolayısıyla performanslarıda birbirine eşittir.

5. Sistem

Bu sistemde yine YSA ile modellenmiş ARMA sistemin fark denklemi alınmış ve bu sistem bulanık sinir ağı ile eğitilmiştir.



(a)



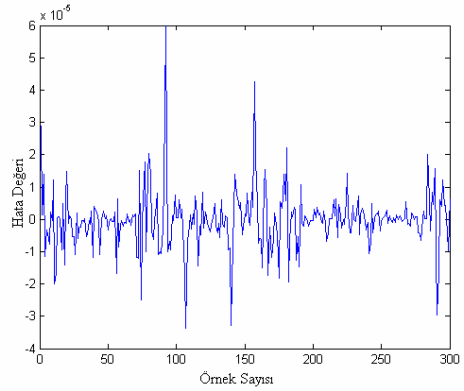
(b)

Şekil 5.10. 5.Sistemin

a) Sistem çıkışı

b)Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki fark.



(c)

Bu sistemin hem YSA hem de bulanık sinir ağı ile modellenmesi sonucunda oluşan RMS, MSE ve korelasyon değerleri tablo 5.4'de verilmiştir.

Tablo 5.7. ARMA Sistemin YSA ve Bulanık Sinir Ağına Göre Hata değerleri

Yapay Zeka Tekniği	**MSE**	**RMS**	**Korelasyon**
Yapay Sinir Ağı	9.6014×10^{-9}	9.7987×10^{-5}	1
Bulanık Sinir Ağı	8.0672×10^{-11}	8.9817×10^{-6}	1

Tablo 5.7'de verilen değerlere göre bulanık sinir ağı, YSA'ya göre daha iyi sonuç vermiştir.

Bu yapıda 1024 tane doğrusal parametre, 56 tane doğrusal olmayan toplam 1080 parametre kullanılmıştır. Eğitim için 300 adet veri kullanılarak ARMA sistem modellenmiş ve Şekil 5.10'daki sonuçlar elde edilmiştir. Bu sonuçlara göre sistem ile model çıkışı birbirine çok benzemektedir aralarındaki hata ihmal edilebilir düzeydedir. Bulanık kural sayısı 128'dir. Bu sisteme deneme yanılma yöntemiyle gaussmf üyelik fonksiyonunun diğer üyelik fonksiyonlarına göre daha iyi sonuç verdiği görülmüştür.

5.2.2. Bulanık Sinir Ağı İle Doğrusal Olmayan Sistem Modelleme

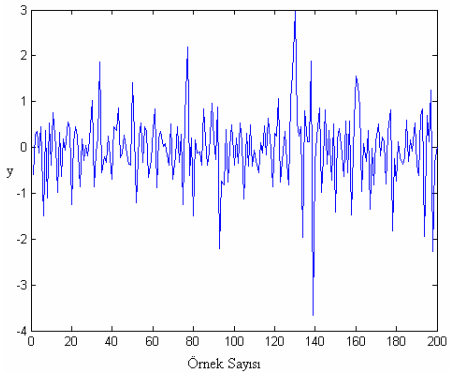
Burada doğrusal olmayan sistemlerden bilineer sistemin ve Mackey-Glass zaman gecikmeli sistemin simülasyonu yapılmıştır.

1. Sistem

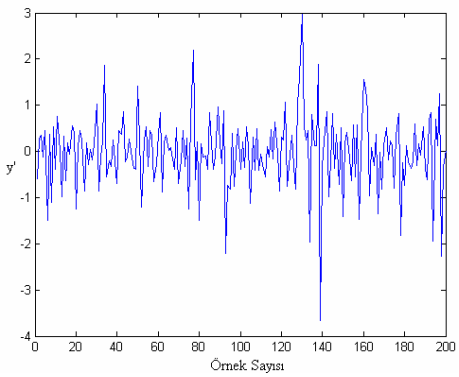
Bilineer sistem olarak aşağıda fark denklemi alınmıştır.

$$y(n)= 0.25*y(n-1) - 0.5*y(n-1)*x(n) + 0.05*y(n-1)*x(n-1) -0.5*x(n) + 0.5*x(n-1) \quad (5.9)$$

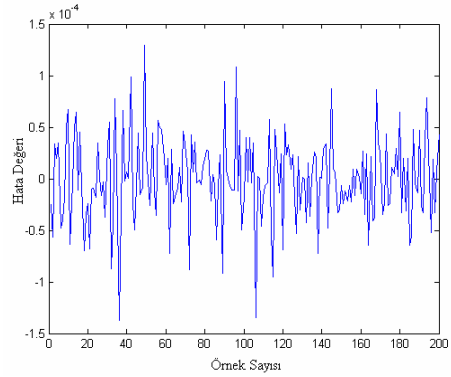
Bu sistemin başlangıç şartları sıfır olarak alınmıştır. Girişi olarak gauss dağılımı sıfır ortalamalı, standart sapma değeri 1 olan 200 adet giriş dizisi sisteme uygulanmış ve istenen çıkış dizisi elde edilmiştir.



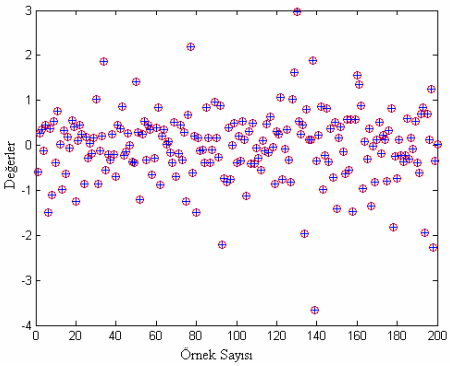
(a)



(b)



(c)



(d)

Şekil 5.11. 1. Sistem için;

a) Sistem çıkışı,

b) Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki fark,

d) Sistem ile modelin üst üste çizimi.

Şekil 5.11.c'deki verilere göre sistem çıkışı ile model çıkışı arasındaki fark 10^{-4}'ler mertebesinde ve bu hata ihmal edilebilir sevide olmaktadır. Oluşturulan FIS yapısında 192 tane doğrusal parametre, 20 tane doğrusal olmayan toplam 112 parametre mevcuttur. Ayrıca 32 tane bulanık kural oluşturulmuştur. Sistem beş girişe karşılık bir tane çıkıştan oluşmaktadır. Yine burada ANFIS kullanıldığı için Sugeno tipi bulanık çıkarım metodu kullanılmıştır. Bu sonuçlara göre bilineer doğrusal olmayan sistemlere bulanık sinir ağı başarılı bir şekilde uygulanmıştır.

Tablo 5.8. Bilineer Sistemin YSA ve Bulanık Sinir Ağına Göre Hata değerleri

Yapay Zeka Tekniği	**MSE**	**RMS**	**Korelasyon**
Yapay Sinir Ağı	$1.29\text{x}10^{-7}$	$3.59\text{x}10^{-4}$	1
Bulanık Sinir Ağı	$5.3354\text{x}10^{-8}$	$2.309\text{x}10^{-4}$	1

Tablo 5.8 bilineer sistemin hem yapay sinir ağları hemde bulanık sinir ağı ile modellenmesi sonucunda oluşan MSE ve RMS değerleri verilmiştir. Bu değerlere göre bulanık sinir ağı yapay sinir ağına göre daha iyi bir modelleme yapmıştır.

2. Sistem

Bu çalışmada doğrusal olmayan Mackey-Glass (MG) zaman gecikmeli fark eşitliği modellenmiştir. MG zaman gecikmeli fark eşitliği, eşitlik 5.8'de verildiği gibi alınmıştır.

$$\overset{*}{x}(t) = \frac{0.2 \times x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1 \times x(t) \qquad (5.10)$$

$x(0) = 1.2$, $\tau = 17$ ve $t\langle 0$ için $x(t) = 0$ olarak alınmıştır. ANFIS için giriş eğitim veri seti dört boyutlu bir vektör olarak seçilmiştir ve 6 birim zaman gecikmeli olarak verilmiştir.

$$w(t) = [x(t-18) \quad x(t-12) \quad x(t-6) \quad x(t)] \qquad (5.11)$$

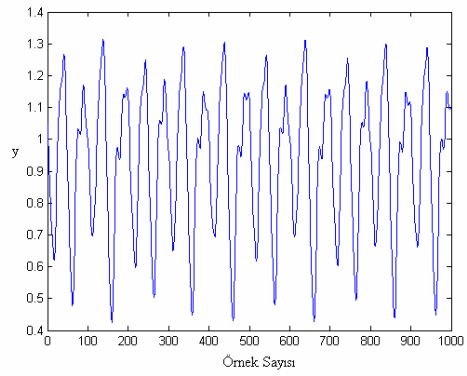
Buna karşılık sistem çıkışı da;

$$s(t) = x(t+6) \qquad (5.12)$$

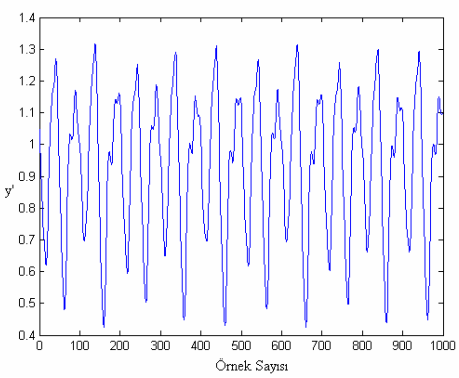
olarak verilmiştir.

Bu zaman serisi kaotiktir ve peiyodu açık bir şekilde tanımlanamaz. Bu seriler yakınsamaz veya ıraksamaz ve ayrıca eğrinin yörüngesi başlangıç şartlarına sıkı bir şekilde bağlıdır.

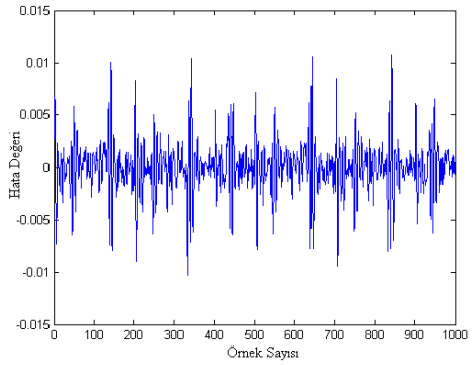
118 ve 1117 aralığında herbir t için, eğitim giriş çıkış verisi yapılandırılmıştır. Bu yapının ilk elemanı 4 boyutlu giriş matrisi w ve ikinci elemanı çıkış matrisi olan s 'dir. Burada 1000 tane giriş çıkış değeri mevcuttur. Bu 1000 verinin ilk 500'ü ANFIS eğitimi için, geri kalan 500'ü ise test verisi olarak kullanılmıştır. Sisteme ait elde edilen şekiller Şekil 5.10'da verilmiştir.



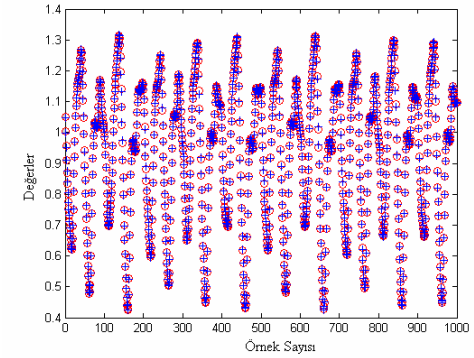
(a)



(b)



(c)



(d)

Şekil 5.12. 2. Sistem için;

a) Sistem çıkışı,

b) Model çıkışı,

c) Sistem çıkışı ile model çıkışı arasındaki fark,

d) Sistem ile modelin üst üste çizimi.

Şekil 5.12'daki şekillerde görüldüğü gibi sistem çıkışı ile model çıkışı arasındaki fark 10^{-3}'ler mertebesindedir. Model hem eğitimde hem de testde başarılı olmuştur. Oluşturulan FIS yapısında 80 tane doğrusal parametre, 24 tane doğrusal olmayan toplam 104 parametre mevcuttur. Ayrıca 16 tane bulanık kural yapı içerinde mevcuttur. Sistem 4 girişe karşılık bir tane çıkıştan oluşmaktadır. Yine burada ANFIS yapısından

dolayı Sugeno tipi bulanık çıkarım metodu kullanılmıştır. Bu sonuçlar neticesinde orjinal MG zaman serisi ile ANFIS çıkışı arasındaki farkın küçük olduğunu ve MG serisinin bulanık sinir ağı ile başarılı bir şekilde modellendiğini söyleyebilmekteyiz.

BÖLÜM 6

SONUÇ ve ÖNERİLER

Sistemlerin birçok alanda güvenli, verimli ve istenilen kalitede üretim yapabilmesi için kontrol düzeneğinin kurulması gerekir. Kontrol edilecek sistemin matematik modelinin bilinmesi zorunludur. Böylece sistemin modeli kullanılarak sistemin çıkışı arzu edilen şekilde düzenlenebilir.

Çalışmanın ilk bölümünde, AR ve ARMA yapısı şeklindeki doğrusal sistemlerin ve Bilineer sistem şeklindeki doğrusal olmayan sistemin YSA ile modellenmesi konusu incelenmiştir. YSA'ları ile yapılan testlerde gürültülü ve gürültüsüz durumlar farklı öğrenme algoritmaları ile test edilmiştir. Bu testlerde, LM öğrenme algoritması diğer öğrenme algoritmalarına göre üstün bir performans sağlamıştır. Farklı öğrenme algoritmaları ile eğitilen YSA, gürültülü ve gürültüsüz durumlarda başarılı olmuştur. Ancak MSE hata değeri bakımından gürültülü sistemler, gürültüsüz sistemlere nazaran daha yüksek hata değeri üretmiştir. Gürültülü sistemlerin eğitiminde geçen süre gürültüsüz sistemlere nazaran daha fazla olmuştur. YSA'da giriş sayısını artırdığımızda öğrenme artmakta ve hata değeri de düşmektedirSistemlere farklı büyüklükte gürültüler uygulanmış ve performansları karşılaştırılmıştır. Elde edilen sonuçlara göre sisteme uygulanan gürültü azaldıkça modelin hata miktarı düşerek modellemesini istediğimiz sisteme yaklaşmaktadır. Sonuçta YSA hem doğrusal sistemleri hem de doğrusal olmayan sistemleri, gürültülü ve gürültüsüz durumlarda başarılı bir şekilde modellemiştir.

Çalışmanın ikinci bölümünde, doğrusal ve doğrusal olmayan sistem örnekleri bulanık sinir ağı ile modellenmiştir. Bulanık mantığın belirsiz bilgileri işleme ve yapay sinir ağının öğrenme yeteneğinden yararlanılarak sistem modelleme gerçekleştirilmiştir. Bulanık mantık ve yapay sinir ağının birleştirilmesi, birbirlerinin dezavantajlarını da gidermektedir. Doğrusal sistem olarak AR, MA, ARMA sistemler ele alınmıştır.

Doğrusal olmayan sistem olarak bilineer sistem ve özel bir seri olan Mackey-Glass zaman gecikmeli fark serisi incelenmiştir. Burada sisteme en uygun yapı deneme yanılma yoluyla seçilmiştir. Bulanık sinir ağı hem sistem hakkındaki bilgileri, hem de sistem giriş çıkış verilerini kullanmaktadır. Sistem bilgileri, bulanık sistemle ilgili bilgileri içermektedir. Sistem giriş çıkış verileri ise ağın eğitilmesinde kullanılmaktadır. Bulanık sinir ağının parametreleri, ağın eğitilmesi ile elde edilmiştir. Bulanık sinir ağı ile yapılan eğitimler sonucunda, model çıkışı ile sistem çıkışı arasındaki hata incelenmiş ve kabul edilebilir seviyede olduğu görülmüştür. Ayrıca yapılan testlerde de hata miktarı oldukça küçük çıkmıştır. Bulanık sinir ağı ile yapılan modelleme işlemine gürültü eklenirse çıkışta tutarsız veya anlamsız bilgi alınmaktadır. Sonuç olarak Bulanık sinir ağı, hem doğrusal hemde doğrusal olmayan sistemlerin modellenmesinde başarılı sonuçlar vermiştir.

Simülasyonun ilk kısmında YSA ile modellenmiş AR, ARMA ve Bilineer sistemler simülasyonun ikinci kısmında bulanık sinir ağı ile modellenmiş ve sistem modellemede YSA ile bulanık sinir ağının performansları karşılaştırımıştır. Sonuçlara baktığımız da bulanık sinir ağı buradaki sistemleri daha iyi modellediği görülsede aralarındaki performans farkı çok büyük değildir.

Bulanık sinir ağı ile sistem modelleme, Voltera, PAR gibi doğrusal olmayan sistem modellemeye de uygulanabilir.

KAYNAKLAR

1. Perrot, M.H., and Cohen, R.J., An Efficiant Approach to ARMA Modelling of Biolagical Systems with Multiple Inputs and Delays. IEEE Transactions on Biomedical Engineering, 43, (1), 1-14, 1996.
2. Sjöberg, J., Benvestive, A., Nonlinear Black-Box Modelling in System Identification, Dept. Of Elec. Eng., Linkoping University, Linkoping, Sweeden, 1995.
3. Zadeh, L.A., From Circuit Theory to System Theory, Proc, IRE, 50, 856-865, 1962.
4. Johansen, T., Constructing NARMAX Model Using ARMAX Models Int., Control, 58, (5), 1125-1153, 1993.
5. Zhang, J., and Morrıs, A.J., Fuzzy Neural Network for Nonlinear System Modelling. IEEE Proceedings, Control Theory Applications, 42, (6), 551-561, Novenber, 1995.
6. Narendra, K. S., Parthasarathy, K., Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks, IEEE Trans. Neural Networks, 2, (2), 252-262, 1991.
7. Ljung, L., System Identification: Theory For The User, Englewood Cliffs, NJ., Prentice Hall, 1987.
8. Töderstorm S., System Identification, Prentice Hall, 1989.
9. Zorlu, H., Doğrusal Olmayan Sistemlerin Yapay Sinir Ağları Kullanılarak Modellenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi, Kayseri, 2004.
10. Widrow, B., Stearns, D., Adaptive Signal Processing, Prentice Hall, 1985.
11. Lee, J., Mathews, V.J., A Stability Condition For Certain Bilinear Systems, IEEE Trans. on Signal Processing, 42, 1871-1873, 1994.
12. Takagi, T., Fuzzy İdentification of Systems and It's Application to Modelling and Control, IEEE Transactions, SMC-15, (1), 116-132, 1985.
13. Kaplan, A., Nümerik Tabu Arama Algoritması, Doktora Tezi, Erciyes Üniversitesi, Kayseri, 2000.
14. Candy, J.V., Signal Processing: The model-based Approach, Mc Graw Hill, 1986.
15. Cioffi J. and Kailath T., Fast Recursive Least Squares Transversal Filters for Adaptive Processing, IEEE Trans. Acoust., Speech, Signal Processing, 34, 304-307 ,1984.
16. Honig, H.L., Messerschmitt, D.G., Adaptive Filters Structures, Algorithms and Apllications, Kluwer Academic Publishers, Berlin, 1995.
17. Faussett, L., Fundamentals of Neural Networks Architectures, Algorithms, and Applications, Prentice-Hall, 1994.
18. Zurada, J. M., Introduction to Artificial Neural Networks, West Publishing Company, 1992.
19. Demuth, H., Beale, M., Hagan, M. T., Neural Network Design, Wadsworth Publishing Company Pub., 1995.

20. Narendra, K. S., Parthasarathy, K., Identification and Control Of Dynamic Sistems Using Neural Networks, IEEE Trans. Neural Networks 1, 4-27, Mar.1990.
21. Giannakis G.B. and Mendel J.M., Identification Of Non-Minimum Phase Systems Using Higher-Order Statistics, IEEE Trans. Acoust., Speech, Signal Processıng, 37, 360-377, 1989.
22. Zhi, Q., A Quasi-ARMAX aproach to modelling of non-linear systems, İnternational Journal Of Control, 2001.
23. Yang, Z.J., Sagara, S., Tsuji, T., System Impulse Response Identification Using A Multiresolution Neural Network, Automatica, 33, 1345-1350, 1997.
24. Narendra, K.S., Mukhopadhyay, S., Adaptive Control Using Neural Networks and Approximate Models, IEEE Trans. on Neural Networks, 8, 475-485, 1997.
25. Chen, S., Billings, S. A., Neural Networks for Nonlinear Dynamic System Modeling and identification, intemational J. Control, 2, 319-346, 1992
26. Chen, S., Billings, S. A., Neural Networks for Nonlinear Dynamic System
27. Bossley, K.M, Neuro-Fuzzy Approaches in System Identification University of Southampton, Faculty of Engineering and Applied Sciense, Electronics and Computer Science Department, Doctor of Philosophy Thesis, England.1997.
28. Frye, R.C., Rıetman, E., and Wong, C.C., Back-Propogation Learning and Nonidealities in Analog Neural Network Hardware. IEEE Transactions on Neural Networks, 2, (1), 110-117, 1991.
29. Rauf, F., Nonlinear Adaptive Filtering: A Unified Approach, Ph.D. Thesis, Boston University, Boston, 1993.
30. Kosko, B., Neural Networks for Signal Processing" Prentice Hall, Englewood Cliffs, NJ 07632, 1992.
31. Abıyev, R., Neural Network Based Fuzzy Inference System For İdentification and Control Of Dinamic System, Near East University, IJCI Of Int XII. Turkish Symposium on Artificial Intelligence and Neural Networks, 1, (1), July 2003
32. Nauck, D., Lawonn. F., and Kruse, R., Fuzzy Sets, Fuzzy Controllers, and Neural Networks, Journal of the Humboldt-University of Berlin, Series Medicine 41, 1992.
33. Ross, T.J., Fuzzy Logic with Engineering Applications, Mc.Graw-Hill Publishing Co. New York, 1995.
34. Jınglu, H., Fuzzy Modelling and Identification, Katholieke Universiteit Leuven 1999.
35. Jario, J., A Fuzzy Identification Methot For Nonlinear System, Turk J Elec, 8, 2000.
36. Tanevo, A., Neuro-fuzzy Model: Used For İdentification and Predictive Controller Design in Matlab, Technical University of Sofia, 2000.
37. Lın, C.T., and Lu, Y.C., A Neural Fuzzy System with Linguistic Teaching Signals. IEEE Transactions on Fuzzy Systems, 3, (2), 169-189, 1995.

38. Lın, Y., and Cunnıngham, G., A New Approach to Fuzzy-Neural System Modelling, IEEE Transactions on Fuzzy Systemsa, 3, (2), 190-198, 1995.
39. Nauck, D., Beyond Neuro-Fuzzy Perspectives and Directions.Proc. Third European Congress on Intelligent Technigues and Soft Computing, 1159-1164, Aachen, 1995.
40. Nauck, D., and Klawonn.F., Neuro-Fuzzy Classification Initialized by Fuzzy Clusterig.Proc. Fourth European Congress on Intelligent Tecnigues and Soft Computing, Aachen 1996.
41. Nauck, D., and Kruse, R., Choosing Appropriate Neuro-Fuzzy Models. Proc. European Congress on Intelligent Technigues and Soft Computing, Aachen 1994.
42. Nauck, D., Klawonn, F.,and Kruse, R., Combining Neural Networks and Fuzzy Controllers Accepted for presentation at FLAI'93, Linz, Austria, 1993.
43. Park, Y.M., Choı, M.S., and Lee, K.Y., an Optimal Tracking Neuro-Controller for Nonlinear Dynamic Systems. IEEE Transactions on Neural Networks, 7, (5), 1099-1110, 1996.
44. Jogle, V., Artificial Neural Networks and Neuro-Fuzzy Systems For Modelling And Controlling Real Systems, A Comparative Study, Engineering Applications of Artificial Intelligence, 17, 265–273, 2004.
45. Cheng-Jion, C., Identification and Prediction Using Recurrent Compensatory Neuro-Fuzzy Systems İnternational Journal Of Fuzzy Sets And Systems, 21 July 2004.
46. Povilin, B., A Neurofuzzy Network Structure for Modelling and State Estimation of Nonlinear Systems, İnternational Journal Of Systems Science, 28, (4), 335-345, 1997.
47. Wilamowski, B. Neuro-Fuzzy Systems And Their Applications, University of Wyoming IECON, 1, IEEE, 1998.
48. Andreos, N., A Hierarchical Recurrent Neuro-Fuzzy Model For System İdentification, International Journal of Approximate Reasoning 32, 153–170, 2003.
49. Dang, Z. Extended Neuro-Fuzzy Models Of Multilayer Perceptrons, Fuzzy Sets and Systems 142, 221–242, 2004.
50. Jacceh, M., Neuro-Fuzzy System With Learning Tolerant To İmprecision, Fuzzy Sets and Systems 138, 427–439, 2003.
51. Khurram, M. U., Fast Learning Nonlinear Adaptive Filtering Structures, Ph.D. Thesis, University Of Boston, 1994.
52. Jyh, S., Neuro-Fuzzy Modelling and Control, IEEE Control and Simulation Group, March 1995.
53. Wilamowski, B., Neuro-Fuzzy Systems And Their Applications, University of Wyoming IECON 98 Vol.1 IEEE 1998.
54. Srinivasa, P., Fuzzy Identification And Control Of a Class Of Nonlinear Systems, İndian Institute Of Technology, Kanpur, 1999.
55. Marian, B., Neuro-Fuzzy Systems For Rule-Based Modelling Of Dynamic Process, ESIT 2000, 14-15 September, Germany, 2000.

56. Johansen, T., On Multı-Objective İdentification Of Takagı-Sugeno Fuzzy Model Parameters, Systems and Control Engineering Group, 2001.
57. Povilin, B., A Neurofuzzy Network Structure for Modelling and State Estimation of Nonlinear Systems, İnternational Journal Of Systems Science, 28, (4), 335-345, 1997.
58. Jaos, V., Dynamic System Control and İdentification For Fuzzy İnference Systems, IEEE ,1998.
59. Cheng-Jion, C., Identification and Prediction Using Recurrent Compensatory Neuro-fuzzy Sytems, İnternational Journal Of Fuzzy Sets and Systems, 21 July 2004.
60. Marsequera, M., Model Identification by Neuro-Fuzzy Techniques: Predicting The Water Level İn A Steam Generator OF A PWR, Progress in Nuclear Energy, 44, (3), 237-252, 2004.

EKLER

EK-1

```
clc;
clear all;
close all;

%giris dizisi
girisdizisi=randn(100,1);
xg=girisdizisi;

%arma katsayıları
akatsayilari=[0.3 0.5];
ak=akatsayilari;
bkatsayilari=[1 0.8 0.15];
bk=bkatsayilari;

%arma sistem cıkısı
cikis=filter(ak,bk,xg);

figure(1),plot(xg);title('xgirisleri')
figure(2),plot(cikis);title('arma sistem cikisi')

%nouralfuzzy girisler
xn=xg;
xn_1=[0;xn(1:99,1)];
yn=cikis;
yn_1=[0;yn(1:99,1)];
yn_2=[0; yn_1(1:99,1)];

%noural fuzzy egitim
giris=[xn xn_1 yn_1 yn_2];
egitim = [giris cikis];



fismat=genfis1(egitim);
xgt=0.5*randn(50,1);



cikist=filter(ak,bk,xgt);
```

```
xnt=xgt;
xn_1t=[0;xnt(1:49,1)];

ynt=cikist;
yn_1t=[0;ynt(1:49,1)];
yn_2t=[0; yn_1t(1:49,1)];

girist=[xnt xn_1t yn_1t yn_2t];

test = [girist cikist];
[fismat1,error1,ss,fismat2,error2] = ...
    anfis(egitim,fismat,[],[],test);
 figure(3),plot([error1; error2]);title('hata');
% anfis_output = evalfis([egitim; test], fismat2);

% cikis_fismat = anfis(egitim,fismat,epok);
anfis_cikis1 = evalfis(giris, fismat2);

anfis_cikis2 = evalfis(girist, fismat2);

t1=anfis_cikis1;
t2=anfis_cikis2;
a1=yn;
a2=ynt;
%ogrenme grafigi
figure(4);
plot(t1,'bo');hold on;plot(a1,'r+');
%test grafigi
figure(5);
plot(t2,'bo');hold on;plot(a2,'r+');




uyelikfonsayi  = [3];
uyelikfontipi  = str2mat('trimf');
epok= 20;
giris_fismat = genfis1(egitim,uyelikfonsayi,uyelikfontipi);
cikis_fismat = anfis(egitim,giris_fismat,epok);
cikisf=evalfis(giris,cikis_fismat);
```

```
figure(3),plot(cikisf);title('egitilmis sistemin cikisi')
figure(4),plot([cikis cikisf]);title('arma sistem cikisi ve egitilmis sistemin cikisi')
```

EK-2

```
load baslangıcdata;
x=randn(1,500);
giris=x;
b=1;
for n=1:300
   uk5(b)=giris(n);
   netin0(1,b) = uk5(b);
   netin0(2,b) = uk4(b);
   netin0(3,b) = uk3(b);
   netin0(4,b) = yk4(b);
   netin0(5,b) = yk3(b);
   netin0(6,b) = yk2(b);
   netin0(7,b) = yk1(b);
   yk5(b)=uk5(b)+0.433*uk4(b)+0.490*uk3(b)+2.595*yk4(b)-3.339*yk3(b)+2.2*yk2(b)-0.731*yk1(b);
   yk1(b+1)=yk2(b);
   yk2(b+1)=yk3(b);
   yk3(b+1)=yk4(b);
   yk4(b+1)=yk5(b);
   uk3(b+1)=uk4(b);
```

```
    uk4(b+1)=uk5(b);
    b=b+1;
end

y=yk5;
% w=ones(7,1);
 figure(1),plot(uk5);
 figure(2),plot(yk5);

%test verisi oluşturuluyor
a=1;
for m=301:500
    uk5t(a)=giris(m);

    netin1(1,a) = uk5t(a);
    netin1(2,a) = uk4t(a);
    netin1(3,a) = uk3t(a);
    netin1(4,a) = yk4t(a);
    netin1(5,a) = yk3t(a);
    netin1(6,a) = yk2t(a);
    netin1(7,a) = yk1t(a);
    yk5t(a)=uk5t(a)+0.433*uk4t(a)+0.490*uk3t(a)+2.595*yk4t(a)-3.339*yk3t(a)+2.2*yk2t(a)-
0.731*yk1t(a);
    yk1t(a+1)=yk2t(a);
    yk2t(a+1)=yk3t(a);
    yk3t(a+1)=yk4t(a);
    yk4t(a+1)=yk5t(a);
    uk3t(a+1)=uk4t(a);
    uk4t(a+1)=uk5t(a);
    a=a+1;
end
 figure(3),plot(uk5t);
 figure(4),plot(yk5t);

 randn('seed',33265);
net=newff(minmax(netin0),[10,20,1],{'tansig','tansig','purelin'},'trainlm');
net.trainparam.show=10;
net.trainparam.epochs=1000;
net.trainparam.goal=0;
net.trainparam.lr=0.00000001;
```

```
to=clock;
net=train(net,netin0,yk5);
time_=etime(clock,to);
fprintf('toplam egitim süresi %5.4f saniyedir.\n',time_);
out1=sim(net,netin0);
out2=sim(net,netin1);
cikis=out1;
cikis=[out1];
cikist=[out2];
mse_egitim_hata=mse(yk5-cikis)
mse_test_hata=mse(yk5t-cikist)
rms_egitim_hata=sqrt(mse_egitim_hata)
rms_test_hata=sqrt(mse_test_hata)
cor_egitim=corr2(yk5,cikis)
cor_test=corr2(yk5t,cikist)

%ogrenme grafigi
figure(5);
plot(yk5,'bo');hold on;plot(cikis,'r+');
%test grafigi
figure(6);
plot(yk5t,'bo');hold on;plot(cikist,'r+');
```

ÖZGEÇMİŞ

Ali DURMUŞ, 1980 yılında Kayseri'de doğdu. İlk, orta ve lise öğrenimini Kayseri'de çeşitli okullarda tamamladı. 1998 yılında girdiği Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü' nden 2003 yılında mezun olarak mühendis ünvanını kazandı. Aynı yıl Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Anabilim Dalı'nda Yüksek Lisans öğrenimine başladı.

Adres: Erciyes evler mah. 30 Ağustos cad.

Yenipınar Apt. No:13 Kocasinan/KAYSERİ

Tel: 2341524