

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ADAPTİF AĞ YAPISINA DAYALI BULANIK ÇIKARIM SİSTEMİNİN (ANFIS)
SAYISAL İŞARET İŞLEMCİ İLE GERÇEKLEŞTİRİLMESİ VE UYGULAMASI

Neşet BAYSAL

YÜKSEK LİSANS TEZİ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİMDALI

DANIŞMAN
Yrd. Doç. Dr. Ömer Faruk ÖZGÜVEN

MALATYA
HAZİRAN 2009

Tezin Başlığı : Adaptif Ağ Yapısına Dayalı Bulanık Çıkarım Sisteminin (ANFIS)
Sayısal İşaret İşlemci İle Gerçekleştirilmesi ve Uygulaması

Tezi Hazırlayan: Neşet BAYSAL

Sınav Tarihi : 18 Haziran 2009

Yukarıda adı geçen tez, jürimizce değerlendirilerek Elektrik Elektronik Mühendisliği.
Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Jürisi Üyeleri

Doç. Dr. Nusret TAN.....

Yrd. Doç. Dr. Ö. Faruk ÖZGÜVEN.....

Yrd. Doç. Dr. Mehmet Emin TAĞLUK ...

.....

.....

.....

İnönü Üniversitesi Fen Bilimleri Enstitüsü Onayı

Prof. Dr. İsmail ÖZDEMİR
Enstitü Müdürü

ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduğum “Adaptif Ağ Yapısına Dayalı Bulanık Çıkarım Sisteminin (ANFIS) Sayısal İşaret İşlemci ile Gerçekleştirilmesi ve Uygulaması” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Neşet BAYSAL

ÖZET

Yüksek Lisans Tezi

ADAPTİF AĞ YAPISINA DAYALI BULANIK ÇIKARIM SİSTEMİNİN (ANFIS) SAYISAL İŞARET İŞLEMCI İLE GERÇEKLEŞTİRİLMESİ VE UYGULAMASI

Neşet BAYSAL

İnönü Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

85 + xi sayfa

2009

Danışman : Yrd.Doç.Dr. Ömer Faruk ÖZGÜVEN

Günümüzde sayısal işaret işlemcilerin gelişmesi ile birlikte kontrol sistemlerinde klasik yöntemlerin yerine farklı algoritmalar kullanılmaya başlanmıştır. Bu tezde yaygın olarak kullanılmakta olan algoritmalarından biri olan ANFIS algoritması kullanılarak bir DC Motorun hız kontrol gerçekleştirildi. ANFIS algoritmasının gerçekleştirilmesi için sayısal işaret işlemci olarak Microchip firmasının ürettiği dsPic30f2010 kullanıldı.

DC Motor kontrol uygulamasında, motor dönüş hızı gerilime dönüştürülerek sayısal işaret işlemciye aktarıldı. Bu gerilim değeri ile referans giriş gerilim değeri arasındaki hata ve hatanın değişimi hesaplanarak, ANFIS girişleri olarak kullanıldı. Hesaplanan ANFIS çıkış kontrol değeri PWM sinyaline dönüştürüldü ve bu sinyal ile MOSFET devre elemanı sürülerek motor akımı kontrol edildi.

ANFIS yapısının eğitimi için kullanılacak veriler PID ile kontrol edilen bir DC Motor sistemi üzerinden elde edildi. Kontrol edilen sistemde, referans değer ile motorun hız değeri arasındaki hata ve hatanın değişimi ANFIS'in giriş eğitim verileri, PID denetleyicisi çıkışı ise ANFIS çıkışında olması istenen referans değeri olarak kullanıldı. Bu değerler bilgisayardaki veri toplama kartı kullanılarak Matlab programı aracılığı ile elde edildi.

Bilgisayarda Matlab Simulink'te gerçekleştirilen PID ve ANFIS uygulamalarına ait sonuçlar ile sayısal işaret işlemcide gerçekleştirilen ANFIS uygulaması sonuçları karşılaştırıldı ve bu sistemlerin ayrı ayrı performansları incelendi.

ANAHTAR KELİMELEER: ANFIS, DC Motor, PID, eğitim verisi, Bulanık Mantık, Yapay Sinir Ağları, Sayısal İşaret İşlemci, dsPic30f2010

ABSTRACT

Graduate Thesis

THE APPLICATION AND REALIZATION OF NEURAL - FUZZY INFERENCE SYSTEM CONTROLLER FOR DC-MOTOR WITH MICRCONTROLLER

Neşet BAYSAL

Inonu University

Graduate School of Natural and Applied Science

Department of Electrical and Electronics Engineering

85 + xi pages

2009

Supervisor : Ass. Yrd.Doç.Dr. Ömer Faruk ÖZGÜVEN

With the development of digital signal processing integrated circuits (DSPIC) instead of conventional methods variety of algorithms have been developed and used in control systems. In this thesis, the speed of a DC motor (30 V; 2A; 3100 rpm) was controlled using an adaptive neuro-Fuzzy inference system (ANFIS). The system was primarily simulated in Matlab[®] environment, and afterward was realized by embedding the soft computer program into a DSPIC as the controlling system.

In the training phase of the ANFIS, the output error and the error evolution were used as the input parameters while the output of the PID controller used as the desired reference output for the designed ANFIS system. The trained ANFIS system was first tested for controlling the speed of the DC motor in the Matlab environment and afterward the developed ANFIS algorithm was embedded into the DSPIC which is widely used in the automation media and the speed of the DC motor was controlled.

Through the designed embedded system, the speed of the motor was controlled by driving the MOSFET current driver in the hardware circuit by the pulse width modulated (PWM) signal produced by ANFIS. The performance of the systems were all analyzed and compared together.

KEYWORDS: ANFIS, DC Motor, PID, training data, Fuzzy Logic, Neural Network,
Digital Signal Processor, dsPic30f2010

TEŐEKKÜR

Çalıőmamın başarıya ulaşmasında en büyük paya sahip olan değerli hocam Yrd.Doç.Dr.Ömer Faruk ÖZGÜVEN'e;

Sayısal işaret işlemci programlamasında yardımcı olan Erdoğan HAN'a;

Tez çalışmamda bana destek olan ve her türlü kolaylığı sağlayan tüm mesai arkadaşlarıma;

Her zaman güven ve sevgilerini yanımda hissettiğim aileme

teşekkür ederim.

İÇİNDEKİLER

ÖZET	iii
ABSTRACT.....	iv
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	x
SİMGELER VE KISALTMALAR DİZİNİ	xi
SİMGELER VE KISALTMALAR DİZİNİ	xi
1. GİRİŞ:.....	1
2. KURAMSAL TEMELLER:	3
2.1. Bulanık Kontrol:.....	3
2.1.1. Bulanık işlemler (Kuralların gerçekleştirilmesi, Bağınrtı):.....	5
2.1.1.1. Mamdani modeli:.....	7
2.1.1.2. Larsen modeli:	8
2.1.1.3. Takagi-Sugeno-Kang (TSK) modeli:.....	9
2.1.1.4. Tsukamoto modeli:.....	10
2.1.2. Netleştirme (Defuzzification) :	11
2.1.2.1. Ağırlık merkezi metodu:	11
2.1.2.2. Yüksek değerlik ağırlık merkezi metodu:.....	12
2.1.2.3. Maksimum Ölçüt metodu:	12
2.1.2.4. İlk Maksimum Değer Metodu :	12
2.1.2.5. Maksimum Nokta Ortalaması Metodu:	13
2.2. Yapay Sinir Ağları (Neural Networks):.....	14
2.2.1. Öğrenme:.....	16
2.2.2. Delta Kural ve Geri Yayılım (Backpropagation):.....	16
2.3. Adaptif Ağ Yapısına Dayalı Bulanık Mantık Sistemleri (Adaptive Network Fuzzy Inference System (ANFIS)) :	24
2.3.1. Takagi-Sugeno-Kang (TSK) modeli:.....	24
2.3.2. ANFIS Yapısı:	25
2.3.3. Çok Girişli ve Çok Çıkışlı ANFIS Yapısının Geri Yayılım (Backpropagation) Metodu ile Katsayılarının Hesaplanması:	29
2.3.3.1. Sistem Katsayılarının Rastgele Seçilmesi:	29
2.3.3.2. Mevcut Katsayılar ile Çıkışların Hesaplanması	32
2.3.3.2.1. Her bir eğitim çift için üyelik fonksiyonlarını hesaplanması:	32
2.3.3.2.2. Her bir eğitim çift için kural çıkışlarının hesaplanması:	32
2.3.3.2.3. Her bir eğitim çift için Kuralların Normalleştirilmesi:.....	35
2.3.3.2.4. Her bir eğitim çift için Çıkış Fonksiyonlarının hesaplanması:.....	35
2.3.3.2.5. Her bir eğitim çift için Çıkışların Hesaplanması:	36
2.3.3.3. Hatanın Hesaplanması ve ANFIS Katsayılarının Eğitilmesi:.....	37
2.3.3.3.1. Hatanın Hesaplanması:.....	37
2.3.3.3.2. Çıkış Fonksiyon Katsayılarının Hesaplanması:	37
2.3.3.3.3. Üyelik Fonksiyonları Katsayılarının Hesaplanması:.....	40
2.3.4. Çok Girişli ve Çok Çıkışlı ANFIS Yapısının Hybrid (Melez) Metodu ile Katsayılarının Hesaplanması:	46
2.3.4.1. Hybrid Modeli Çıkış Fonksiyon Katsayılarının Hesaplanması:.....	46
2.3.5. "Matlab GUIDE (GUI Builder)" Yardımı ile Oluşturulan ANFIS Eğitim Programı:.....	49
3. MATERYAL VE YÖNTEM:.....	52
3.1. Matlab Simulink'te ANFIS ile DC Motor Kontrol Uygulama Simülasyonu:...	52
3.2. ANFIS ile DC Motor Kontrol Uygulamaları:.....	57
3.2.1. Uygulamada Kullanılan Motorun Teknik Özellikleri:.....	59
3.2.2. PID Denetleyiciden ANFIS Eğitim Verilerinin Alınması:	59
3.2.3. Bilgisayar Ortamında ANFIS Uygulamasının Gerçekleştirilmesi:.....	65

3.2.4.	DC Motor Devir Kontrolünün dsPic30f2010 ile Yapılması:.....	67
3.2.4.1.	dsPic30f2010'de Yazılan ANFIS Algoritması:.....	67
3.2.4.1.1.	ANFIS Algoritması Hesaplama Alt Yordamı:	68
3.2.4.1.2.	Üyelik Fonksiyon Çıkışlarının Hesaplanması Alt Fonksiyonu:	70
3.2.4.2.	dsPic30f2010 Bacak Bağlantıları:	70
4.	ARAŞTIRMA BULGULARI:.....	72
4.1.	PID ve ANFIS Bilgisayar Uygulama Sonuçları ile dsPic30f2010 Uygulama Sonuçlarının Karşılaştırılması:	72
4.1.1.	Birim Basamak Cevapları:.....	72
4.1.2.	Yük Durumunda Sistem Cevapları:	74
5.	SONUÇ VE ÖNERİLER:	80
6.	KAYNAKLAR	83
	ÖZGEÇMİŞ	85

ŞEKİLLER DİZİNİ

Şekil 2.1.	Giriş Üyelik fonksiyonları	3
Şekil 2.2.	Çıkış Üyelik Fonksiyonları	4
Şekil 2.3.	Yaygın Kullanılan Üyelik Fonksiyon Örnekleri	5
Şekil 2.4.	Giriş Üyelik Fonksiyonları Grafiği.....	6
Şekil 2.5.	Çıkış Üyelik Fonksiyonları Grafiği.....	6
Şekil 2.6.	Mamdani Modeli Bağintı Sonucu ($x=1.25$ için).....	8
Şekil 2.7.	Larsen Modeli Kompozisyon Sonucu ($x=1.25$ için)	9
Şekil 2.8.	Takagi-Sugeno-Kang Modeli Üyelik Fonksiyonları.....	10
Şekil 2.9.	Takagi-Sugeno-Kang Modeli Bağintısı	10
Şekil 2.10.	Tsukamoto Modeli Kompozisyonu	11
Şekil 2.11.	Ağırlık Merkezi Metodu ile Netleştirme	12
Şekil 2.12.	Sinir Hücresi Yapısı.....	14
Şekil 2.13.	“r” Adet Girişli Bir Yapay Sinir Hücresi Modeli.....	15
Şekil 2.14.	Sigmoid Fonksiyonu	15
Şekil 2.15.	Yapay Sinir Ağı Katsayılarının Ayarlanması Şematik Gösterimi.....	16
Şekil 2.16.	Örnek Hata Katsayı Değişim Grafiği	17
Şekil 2.17.	3 Katmanlı Yapay Sinir Ağı.....	19
Şekil 2.18.	Takagi-Sugeno-Kang Modeline Ait Örnek Üyelik Fonksiyonları	24
Şekil 2.19.	TSK Modeli Çıkış Fonksiyonu.....	25
Şekil 2.20.	Üçgen Fonksiyon.....	25
Şekil 2.21.	2 Girişli 3'er Üyelik Fonksiyonlu ANFIS Yapısı.....	28
Şekil 2.22.	Geri Yayılım Algoritması.....	29
Şekil 2.23.	Çok Girişli ve Çok Çıkışlı ANFIS Modeli	30
Şekil 2.24.	Başlangıç Üyelik Fonksiyonları.....	31
Şekil 2.25.	Mevcut Katsayılar ile Çıkış Değerlerinin Hesaplanması Algoritması	32
Şekil 2.26.	1.Katman Bütün Eğitim Verileri için Üyelik Fonksiyon Hesap Algoritması.....	33
Şekil 2.27.	2.Katman Bütün Eğitim Verileri için Kural Çıkışlarının Hesap Algoritması.....	34
Şekil 2.28.	3.Katman Normalleştirme İşlemi Algoritması	35
Şekil 2.29.	4.Katman Bütün Eğitim Verileri için Çıkış Fonksiyonlarının Hesap Algoritması	36
Şekil 2.30.	5.Katman Bütün Eğitim Verileri için Çıkış Hesap Algoritması.....	37
Şekil 2.31.	Çıkış Fonksiyon Katsayılarının Hesaplanması Algoritması	39
Şekil 2.32.	Üçgen Fonksiyonu.....	42
Şekil 2.33.	Üyelik Fonksiyonları Katsayılarının Hesaplanması Algoritması.....	44
Şekil 2.34.	Üyelik Fonksiyon Katsayısı Alt Algoritması (Türev Hesapla).....	45
Şekil 2.35.	Hyrbrid Modeli Algoritması	47
Şekil 2.36.	Hybrid Metodunun Hesaplanması Algoritması.....	48
Şekil 2.37.	Matlab GUI Builder ile Yapılan Eğitim Programı	49
Şekil 2.38.	Matlab GUI Builder ile Yapılan Eğitim Program Eğitim Sonuç Kontrolü ..	51
Şekil 3.1.	ANFIS ile DC Motor Kontrolü Simulink Modellemesi.....	52
Şekil 3.2.	Eğitim Verilerinin Alındığı Simulink Modeli.....	52
Şekil 3.3.	Giriş Sinyali	53
Şekil 3.4.	Çıkış Sinyali	53
Şekil 3.5.	Hata Sinyali	53
Şekil 3.6.	Hatanın Değişimi Sinyali.....	54
Şekil 3.7.	PID Kontrol Devri.....	54
Şekil 3.8.	PID Denetleyici Modeli	55
Şekil 3.9.	Eğitim Programı Kontrol Yüzeyi.....	55
Şekil 3.10.	Anfisedit Programı Kontrol Yüzeyi	55
Şekil 3.11.	Anfisedit Programı Kontrol Yüzeyi (Hybrid Yöntemi ile Eğitim)	56
Şekil 3.12.	DC Motor Hız Kontrolü Simülasyon Sonuçları	56
Şekil 3.13.	Uygulama Devresi Blok Diyagramı	57
Şekil 3.14.	ANFIS Uygulaması için Kullanılan Devre Şeması.....	58

Şekil 3.15	Uygulamada Kullanılan Motora Ait Fotoğraf	59
Şekil 3.16.	Eğitim için Uygulanan Referans İşareti	60
Şekil 3.17.	Eğitim Verisi Hata İşareti	60
Şekil 3.18.	ANFIS Eğitim Verilerinin Alındığı PID Denetleyicisi Modeli	61
Şekil 3.19.	Eğitim Verisi Hatanın Değişimi Eğrisi.....	62
Şekil 3.20.	Eğitim Verisi PID Kontrol Değeri Değişimi İşareti.....	62
Şekil 3.21.	ANFIS Eğitim Programı Kontrol Yüzeyi (Geri Yayılım).....	63
Şekil 3.22.	Anfisedit Programı Kontrol Yüzeyi (Geri Yayılım)	63
Şekil 3.23.	Anfisedit Programı Kontrol Yüzeyi (Hybrid).....	64
Şekil 3.24.	PID Uygulaması Modeli	65
Şekil 3.25.	Bilgisayar Ortamında ANFIS Uygulanması Modeli.....	66
Şekil 3.26.	dsPic30f2010 ANFIS Programı.....	68
Şekil 3.27.	ANFIS Algoritması Hesaplama Alt Yordamı.....	69
Şekil 3.28.	Üyelik Fonksiyon Çıktılarının Hesaplanması Alt Fonksiyonu.....	70
Şekil 3.29.	dsPic30f2010 Bacak İsimleri.....	70
Şekil 3.30.	dsPic30f20102'nun Devre Bağlantıları.....	71
Şekil 4.1.	1000 Devir için ANFIS ve PID Birim Basamak Cevapları	72
Şekil 4.2.	1500 Devir için ANFIS ve PID Birim Basamak Cevapları	72
Şekil 4.3.	2000 Devir için ANFIS ve PID Birim Basamak Cevapları	73
Şekil 4.4.	2500 Devir için ANFIS ve PID Birim Basamak Cevapları	73
Şekil 4.5.	3000 Devir için ANFIS ve PID Birim Basamak Cevapları	74
Şekil 4.6.	1000 Devirde 1 Ω 'luk Direnç ile Yükleme Durumu Cevabı	74
Şekil 4.7.	1000 Devirde 5 Ω 'luk Direnç ile Yükleme Durumu Cevabı	75
Şekil 4.8.	1500 Devirde 2 Ω 'luk Direnç ile Yükleme Durumu Cevabı	75
Şekil 4.9.	1500 Devirde 5 Ω 'luk Direnç ile Yükleme Durumu Cevabı	76
Şekil 4.10.	1500 Devirde 10 Ω 'luk Direnç ile Yükleme Durumu Cevabı	76
Şekil 4.11.	2000 Devirde 5 Ω 'luk Direnç ile Yükleme Durumu Cevabı	77
Şekil 4.12.	2000 Devirde 10 Ω 'luk Direnç ile Yükleme Durumu Cevabı	77
Şekil 4.13.	2500 Devirde 15 Ω 'luk Direnç ile Yükleme Durumu Cevabı	78

ÇİZELGELER DİZİNİ

Çizelge 2.1. 3 Girişli İkişer Üyelik Fonksiyonlu ANFIS'in Kural Çizelgesi	41
Çizelge 2.2. Eğitim Sonucu Katsayıların Komuta Penceresi Çıktısı	51
Çizelge 3.1. Eğitim Programında Kullanılan Eğitim Katsayıları	64

SİMGELER VE KISALTMALAR DİZİNİ

ANFIS	: Adaptive Neuro-Fuzzy Inference Systems (Adaptif Ağ Yapısına Dayalı Bulanık Çıkarım Sistemi)
η	: Eğitim katsayısı,
E	: Hata,
F :	: ANFIS çıkış fonksiyonlarına ait değerler değişkeni,
UYE	: ANFIS Üyelik fonksiyonları katsayıları,
K	: ANFIS Üyelik fonksiyonları arasında oluşturulan kurallarının değerlerini tanımlar,
KT	: ANFIS kurallarının toplamı,
N	: ANFIS kurallarının normal değerleri,
Z	: ANFIS çıkış fonksiyon katsayıları,
w	: Yapay sinir ağıları sistem katsayısı,
R	: Bulanık Mantık Kompozisyonu,
O	: Yapay sinir ağı sistemlerinde hesaplanan çıkış değeri,

1. GİRİŞ:

Günümüzde sayısal işaret işlemcilerin gelişmesi ile birlikte kontrol sistemlerinde klasik yöntemlerin yerine daha farklı, yeni kontrol algoritmaları tercih edilmeye başlanmıştır. Bu çalışmamızdaki asıl hedef, yaygın olarak kullanılmakta olan algoritmalarından biri olan Adaptif Ağ Yapısına Dayalı Bulanık Denetleyici (ANFIS, Adaptive Neural-Fuzzy Inference System) algoritması kullanarak bir uygulama yapmaktır. Uygulama olarak basit yapıları ve birçok alanda kullanımı olan DC Motorlar için hız kontrol uygulaması yapılacaktır. Sayısal işaret işlemci olarak Microchip firmasının ürettiği dsPic30f2010 kullanılacaktır.

ANFIS ile bir uygulama gerçekleştirilmesi için ANFIS yapısının çok iyi özümsemiş olması gerekmektedir. 1993 yılında Jang tarafından ortaya koyulan ANFIS, hem bulanık mantık hem de yapay sinir ağları sistemlerinin beraber kullanıldığı bir yapıya sahiptir [1]. Bu nedenle ANFIS'in daha iyi anlaşılabilmesi için öncelikle bulanık kontrol sistemleri anlatılacak, daha sonra da yapay sinir ağ yapısı, delta kuralı ve geri yayılım yöntemi kullanılarak yapay sinir ağlarında eğitimin nasıl gerçekleştiği anlatılacaktır [2].

Bir sonraki adımda ANFIS'in girişleri ve çıkışları arasındaki hesaplamaların nasıl yapıldığı anlatılacaktır. Yapay sinir ağlarında kullanılan geri yayılım hesaplamalarının mantığından yola çıkarak, ANFIS katsayılarının eğitim algoritmaları kullanılarak nasıl elde edildikleri, denklemlerin nasıl çıkartıldığı ve bunların bilgisayar algoritmalarına dönüştürülmesi anlatılacaktır. Elde edilen bilgisayar algoritmalarına çok çıkışlı ANFIS yapısı hesaplamalarını yapabilecektir [7].

ANFIS yapısı ve eğitimi özümsemişikten sonra uygulamaya ışık tutması için uygulamanın Matlab Simulink'te simülasyonu gerçekleştirilecektir.

Simülasyondan sonra uygulama için gerekli devre hazırlanacak, analog/dijital çevirici (ADC) kullanılarak uygulamanın Matlab Simulink ile bağlantısı sağlanacaktır. Bağlantı sağlandıktan sonra simulink'te PID denetleyici hazırlanarak DC Motor hız kontrolü gerçekleştirilip, yapılan sistem üzerinden eğitim verileri elde edilecektir. Daha sonra aynı bağlantı yapısını kullanarak Simulink "Fuzzy Logic Controller" ünitesi kullanılarak simulink ile ANFIS modeli yapılarak DC Motor kontrolü gerçekleştirilecektir.

En son olarak da sayısal işaret işlemciye ANFIS programı ve ANFIS katsayıları aktarılarak DC Motor hız kontrolü sayısal işaret işlemci ile gerçekleştirilecektir. Elde edilen sonuçlar, eğitim verilerinin elde edildiği PID denetleyicisi ve simulink ile yapılan ANFIS uygulaması sonuçları ile karşılaştırılacaktır.

Kuramsal temellerin anlatımında, "A First Course in Fuzzy and Neural Control" [2] kitabından faydalanılmış ve özellikle bulanık kontrol bölümünün anlatımı için örneklerinden faydalanılmıştır.

Tez için aynı konuda yapılan çalışmaların araştırılması sonucunda aşağıdaki çalışmalar özetlenebilir.

Şekil 2.23'te verilen çok çıkışlı ANFIS sistemi yapısının oluşturulmasında, Avdhesh Sharma et al. tarafından hazırlanan "Intelligent Dual Input Power System Stabilizers For Multi-Machine System" [7] çalışmada kullanılan çift çıkışlı ANFIS yapısından faydalanılmıştır.

AVCI et al. tarafından yapılan "Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi ile Doğru Akım Motorlarının Hız Denetimi" [13] çalışmasında, denetlenecek sistemin ANFIS ters modeli çıkarılarak ANFIS'in eğitim safhası gerçekleştirilmiştir. Yapılan uygulamasında da ANFIS girişleri olarak hata ve hatanın değişimi kullanılmıştır.

Niasar et al. tarafından yapılan "Adaptive Neuro-Fuzzy Control with Fuzzy Supervisory Learning Algorithm for Speed Regulation of 4-Switch Inverter Brushless DC Machines" [17] çalışmasında, ANFIS'in eğitimi için bulanık denetleyici kullanılmış ve sonuçlar PI denetleyici sonuçları ile karşılaştırılmıştır.

Ponce et al. tarafından yapılan "A Novel DC Drive Based On Fuzzy Logic Inverse Plant Model Optimised by ANFIS" [18] çalışmasında, ANFIS yapısı oluşturulurken, sistemin ANFIS ters modeli çıkarılması yöntemi kullanılmıştır.

GÜN tarafından hazırlanan "The Position Control Of The DC Machine by PID Algorithm and Training with Adaptive Neuro Fuzzy Inference System" çalışmasında bir DC makinesinin konum kontrolü ANFIS kullanılarak gerçekleştirilmiştir. Bu makalede ANFIS'in eğitimi için PID algoritması kullanılmıştır.

ANFIS yapısının eğitim için değişik uygulamalar kullanılmaktadır. Biz bu çalışmamızda, ANFIS yapısının eğitimi için gerekli olan eğitim verilerini, PID ile kontrol edilen bir DC Motor sistemi üzerinden elde ettik. ANFIS girişleri için, motor dönüş hızındaki hata ve hatanın değişimini, ANFIS sisteminin olması istenen çıkış değerleri için de PID kontrol sinyalini kullandık.

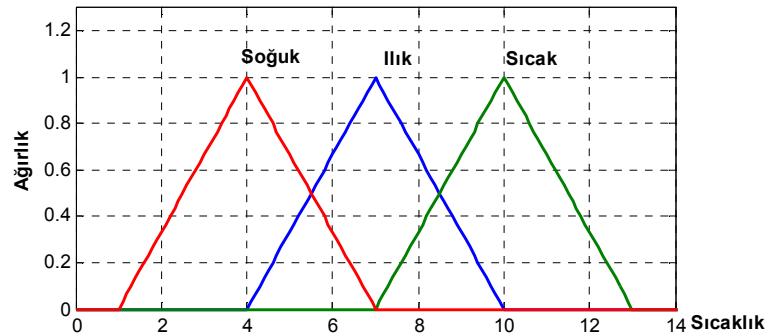
2. KURAMSAL TEMELLER:

2.1. Bulanık Kontrol:

İnsanlar arabanın hızını ayarlarken, eğer hız arzu edilen değerden yavaş ise hızı artırır, eğer fazla ise azalt, eğer normal ise değişiklik yapma gibi kurallar ile bir sistemi kontrol eder. Hızın artırılıp ya da azaltılacağına karar verdikten sonra miktarını belirleyerek (önceden öğrenilmiş bilgiler ışığında ya da yeni bir işlem ise hata oranını azaltacak şekilde düzenlemeler yaparak) arabanın hızı istenen değere ayarlanabilir.

İnsanlar bir nesnenin sıcaklık derecesini bilmez, sıcak, ılık ya da soğuk şeklinde yorum yaparlar. Bir nesnenin sıcaklığını belirlerken, hissettiğimiz sıcaklığı önceden edindiğimiz tecrübeler ışığında sıcak, soğuk ya da ılık değerlendirme gruplarından hangisine girdiğine karar veririz. Günlük hayatımızda yaptığımız bütün faaliyetlerde bu gibi değerlendirmeler yaparak işlerimizi gerçekleştiririz.

Bu mantık doğrultusunda düzenlemeler yapılarak sistemlerin kontrol edilmesine bulanık kontrol denilir. Bir nesnenin sıcaklığını kontrol eden bir sistem yaptığımızda sıcaklık değerleri için Şekil 2.1'deki gibi bir küme yapısını oluşturmamız gerekir. Şekilde de görüldüğü gibi soğuk, ılık ve sıcak iç içe girmişlerdir. Aynen insan düşünce yapısında olduğu gibi burada da kesin ayrımlar yapılamamakta ve bu sayede bulanıklık dediğimiz olguyu oluşturabilmekteyiz.



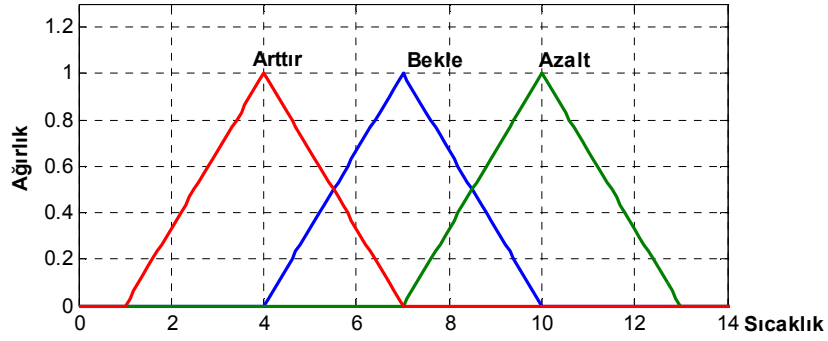
Şekil 2.1. Giriş Üyelik fonksiyonları

Sıcaklığın hangi ağırlıklara karşılık geleceği Şekil 2.1'de belirledikten sonra sıcaklık kümeleri için hangi işlemin yapılması gerektiği ve kurallarını belirlememiz gerekir. Kurallar;

1. **Eğer** Sıcaklık Soğuk **ise**, Isıyı Arttır
2. **Eğer** Sıcaklık Ilık **ise**, Bekle
3. **Eğer** Sıcaklık Sıcak **ise**, Azalt

olarak belirlenmiştir.

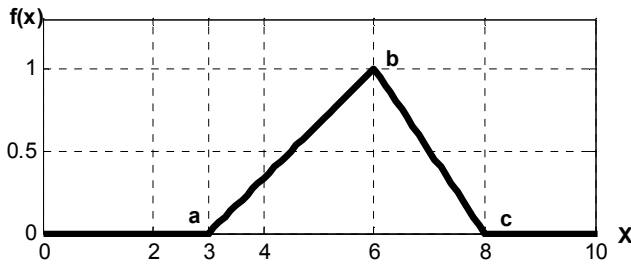
Isıda yapılacak deęişiklik durumları için yukarıda yaptığımız gibi ısı deęişimini gösteren bir küme yapısı Şekil 2.2’de oluşturuldu.



Şekil 2.2. Çıkış Üyelik Fonksiyonları

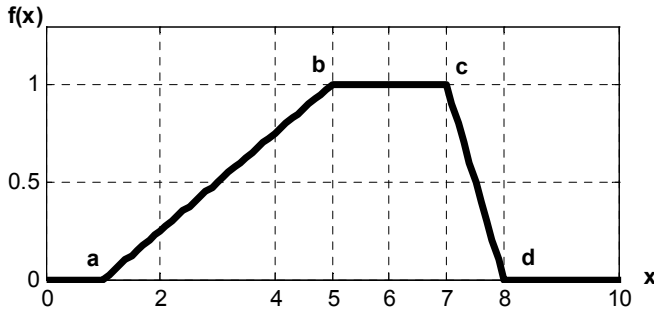
Giriş ve çıkış işlemleri için oluşturulan bu kümelere üyelik fonksiyonları adı verilir [2].

Kullanılan üyelik fonksiyonlarına ait örnekler Şekil 2.3’te gösterildiği gibidir [2].



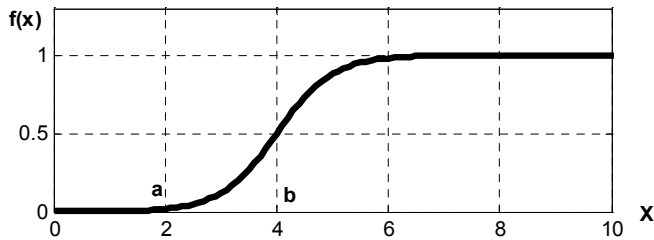
$$f(x, a, b, c) = \begin{cases} 0 & , \quad x \leq a \\ \frac{x-a}{b-a} & , \quad a \leq x \leq b \\ \frac{c-x}{c-b} & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases}$$

a. Üçgen



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

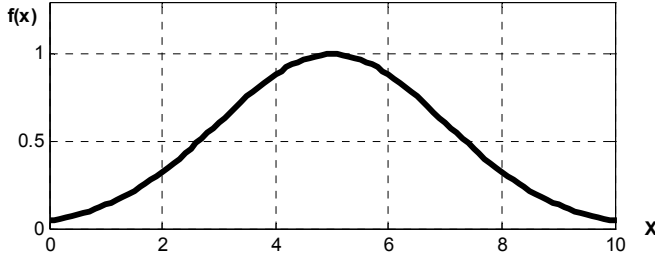
b. Trapezoidal



$$f(x, a, b) = \frac{1}{1 + e^{-a(x-b)}}$$

a = 2; b = 4

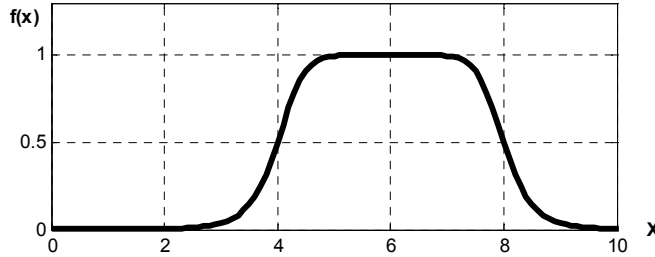
c. Sigmoidal



$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

$$\sigma = 2; c = 5$$

d. Gaussian



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

$$a = 2; b = 4; c = 6$$

e. Generalized Bell

Şekil 2.3. Yaygın Kullanılan Üyelik Fonksiyon Örnekleri

Üyelik fonksiyonlarını belirledikten sonra, giriş ve çıkış üyelik fonksiyonları arasında gerçekleştirilecek işlemler belirlenerek sistemin sıcaklık kontrolü yapması sağlanır.

2.1.1. Bulanık işlemler (Kuralların gerçekleştirilmesi, Bağıntı):

Oluşturulan üyelik fonksiyonları ve kuralların gerçekleştirilmesi için tercih edilen işlemlerin gerçekleştirilmesine bağıntı denilir [2]. Yaygın olarak kullanılan çeşitli işlem modelleri mevcuttur. Kullanılan bu işlem modelleri tek giriş ve tek çıkışlı bir sistem üzerinden incelenecektir.

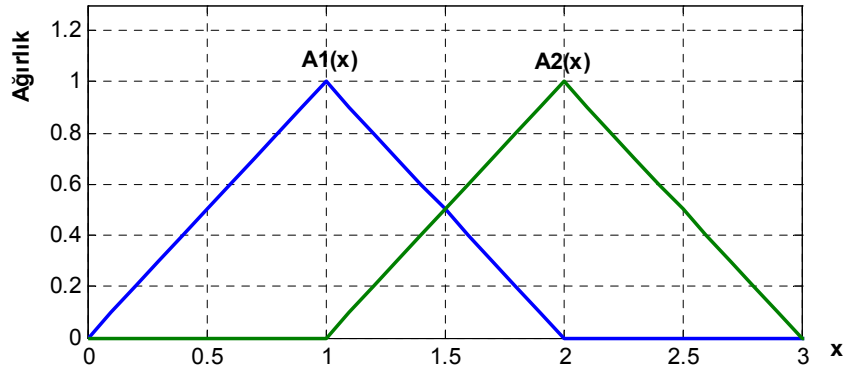
x'e bağlı girişe ait üyelik fonksiyonları $A_1(x)$ ve $A_2(x)$, y'e bağlı çıkışa ait üyelik fonksiyonları ise $B_1(y)$ ve $B_2(y)$ olsun. Giriş ve çıkış arasındaki bağıntı;

$$R(x,y)=A(x)\otimes B(y)$$

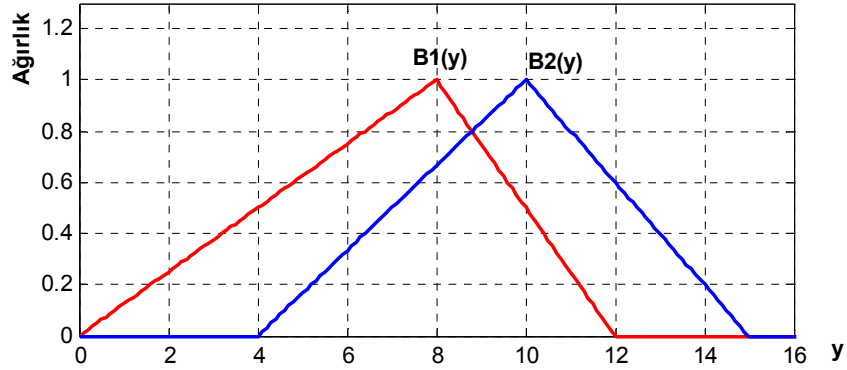
şeklinde gösterilir [2]. \otimes işlemi tasarımcının belirleyeceği bir işlem olacaktır. Yaygın olarak kullanılan işlemler "VEYA" (\vee), "VE" (\wedge), "ÇARPMA" (\cdot) işlemleridir.

$$A_1(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ 2-x, & 1 \leq x \leq 2 \\ 0, & \text{diger} \end{cases} ; \quad A_2(x) = \begin{cases} x-1, & 1 \leq x \leq 2 \\ 3-x, & 2 \leq x \leq 3 \\ 0, & \text{diger} \end{cases} \quad (2.1)$$

$$B_1(y) = \begin{cases} \frac{1}{8}y, & 0 \leq y \leq 8 \\ -\frac{1}{4}y+3, & 8 \leq y \leq 12 \\ 0, & \text{diger} \end{cases} ; \quad B_2(y) = \begin{cases} \frac{1}{6}y-\frac{2}{3}, & 4 \leq y \leq 10 \\ -\frac{1}{5}y+3, & 10 \leq y \leq 15 \\ 0, & \text{diger} \end{cases} \quad (2.2)$$



Şekil 2.4. Giriş Üyelik Fonksiyonları Grafiği



Şekil 2.5. Çıkış Üyelik Fonksiyonları Grafiği

Tek girişli ve tek çıkışlı bir sistem için üyelik fonksiyonları Şekil 2.4 ve Şekil 2.5'deki gibi seçilmiş olup, kurallar aşağıda olduğu gibidir.

“Eğer x girişi $A_i(x)$ üyelik fonksiyonuna ait ise y çıkışı $B_i(y)$ üyelik fonksiyonuna aittir.”

Bundan sonraki kural tanımlamaları için “Eğer $x = A_i(x)$ ise $y = B_i(y)$ ” şekilde kısa gösterilim kullanılacaktır. Burada kullanılan “=” simgesi x'in değerinin $A_i(x)$ üyelik fonksiyonunda bir değere sahip olduğu anlamına gelir.

2.1.1.1. Mamdani modeli:

Bu bağıntı için uygulanacak işlem “ve” (\wedge) işlemidir ve bağıntı formülü denklem 2.3’de verildiği gibidir.

$$R(x, y) = \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \quad (2.3)$$

Her bir $x = x_1, x_2, \dots, x_k$ değeri için R_x bağıntısı denklem 2.4’de olduğu gibidir.

$$R_{x=x_k}(y) = \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \quad (2.4)$$

k adet girişe sahip bir sistem için oluşturulacak n adet kural denklem 2.5’de olduğu gibi tanımlanabilir.

R_i : Eğer $x_1=A_{i1}$ ve $x_2=A_{i2}$ ve ve $x_k=A_{ik}$ ise $y=B_i$ ’dir $i=1,2,\dots,n$

$$R_{x=x_k}(y) = R(x_1, x_2, \dots, x_k, y) = \bigvee_{i=1}^n [(A_{i1}(x_1) \wedge A_{i2}(x_2) \wedge \dots \wedge A_{ik}(x_k)) \wedge B_i(y)] \quad (2.5)$$

Örneğimiz için aşağıdaki gibi tanımlanan iki adet kural belirlenmiştir.

Eğer $x = A_1(x)$ ise $y = B_1(y)$ ’dir.

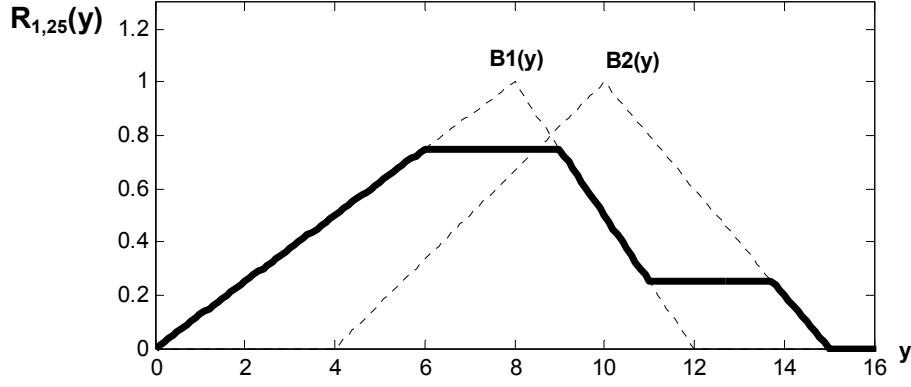
Eğer $x = A_2(x)$ ise $y = B_2(y)$ ’dir.

Bu kurallar ile oluşturulan bağıntı denklem 2.6’dadır.

$$R(x, y) = (A_1(x) \wedge B_1(y)) \vee (A_2(x) \wedge B_2(y)) \quad (2.6)$$

Bu bağıntı $x = 1.25$ için denklem 2.7’de olduğu gibi hesaplanarak ve Şekil 2.6’daki sonuç elde edilir.

$$R_{1,25}(y) = \left\{ \begin{array}{ll} \left(\frac{3}{4} \wedge \frac{1}{8} y\right) & , 0 \leq y \leq 4 \\ \left(\frac{3}{4} \wedge \frac{1}{8} y\right) \vee \left(\frac{1}{4} \wedge \left(\frac{1}{6} y - \frac{2}{3}\right)\right) & , 4 \leq y \leq 8 \\ \left(\frac{3}{4} \wedge \left(-\frac{1}{4} y + 3\right)\right) \vee \left(\frac{1}{4} \wedge \left(\frac{1}{6} y - \frac{2}{3}\right)\right) & , 8 \leq y \leq 10 \\ \left(\frac{3}{4} \wedge \left(-\frac{1}{4} y + 3\right)\right) \vee \left(\frac{1}{4} \wedge \left(-\frac{1}{5} y + 3\right)\right) & , 10 \leq y \leq 12 \\ 0 & , \text{diger} \end{array} \right. \quad (2.7)$$



$$A_1(1,25) \wedge B_1(y) \vee [A_2(1,25) \wedge B_2(y)]$$

Şekil 2.6. Mamdani Modeli Bağını Sonucu (x=1.25 için)

2.1.1.2. Larsen modeli:

Bu bağını için uygulanacak işlem çarpma (.) işlemidir ve bağını formülü denklem 2.8'de olduğu gibi tanımlanır.

$$R(x, y) = \bigvee_{i=1}^n (A_i(x) \cdot B_i(y)) \quad (2.8)$$

Her bir $x = x_1, x_2, \dots, x_k$ değeri için R_x bağını denklem 2.9'da olduğu gibidir.

$$R_{x=x_k}(y) = \bigvee_{i=1}^n (A_i(x) \cdot B_i(y)) \quad (2.9)$$

k adet girişe sahip bir sistem için oluşturulacak n adet kural denklem 2.10'da olduğu gibi tanımlanabilir.

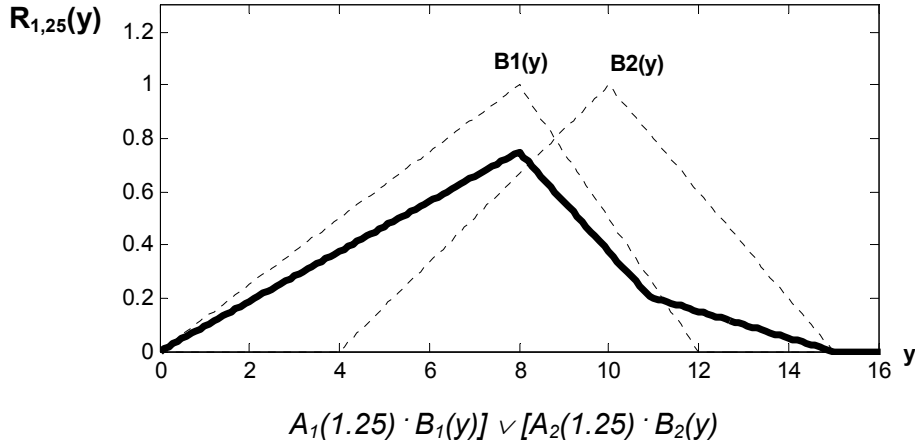
R_i : Eğer $x_1=A_{i1}$ ve $x_2=A_{i2}$ ve ve $x_k=A_{ik}$ ise $y=B_i$ 'dir $i=1,2,\dots,n$

$$R_{x=x_k}(y) = R(x_1, x_2, \dots, x_k, y) = \bigvee_{i=1}^n [(A_{i1}(x_1) \wedge A_{i2}(x_2) \wedge \dots \wedge A_{ik}(x_k)) \cdot B_i(y)] \quad (2.10)$$

Mamdani modelinde oluşturulan iki kural için oluşturulacak Larsen Modeli bağını denklem 2.11'de olduğu gibidir .

$$R(x, y) = (A_1(x) \cdot B_1(y)) \vee (A_2(x) \cdot B_2(y)) \quad (2.11)$$

Bu bağını $x = 1.25$ için hesaplandığında Şekil 2.6'daki sonuç elde edilir.



Şekil 2.7. Larsen Modeli Kompozisyon Sonucu (x=1.25 için)

2.1.1.3. Takagi-Sugeno-Kang (TSK) modeli:

Bu modelli ANFIS yapısı anlatılırken daha ayrıntılı olarak ele alınacaktır. Bu modelin özel hali ANFIS yapısını meydana getirir. Fonksiyon katsayıları ve girişlerin üyelik fonksiyonları yapay sinir ağlarında kullanılan öğrenme sisteminin kullanılması ile elde edilir.

Bu modelde bağıntı için kurallar aşağıdaki şekilde tanımlanır [2];

“Eğer x girişi $A_i(x)$ üyelik fonksiyonuna ait ise çıkış $f_i(x)$ fonksiyonudur.”

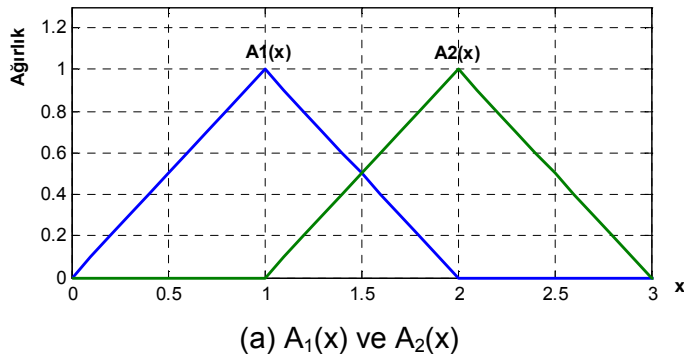
k adet girişe sahip bir sistem için n adet kural aşağıdaki gibi tanımlanabilir.

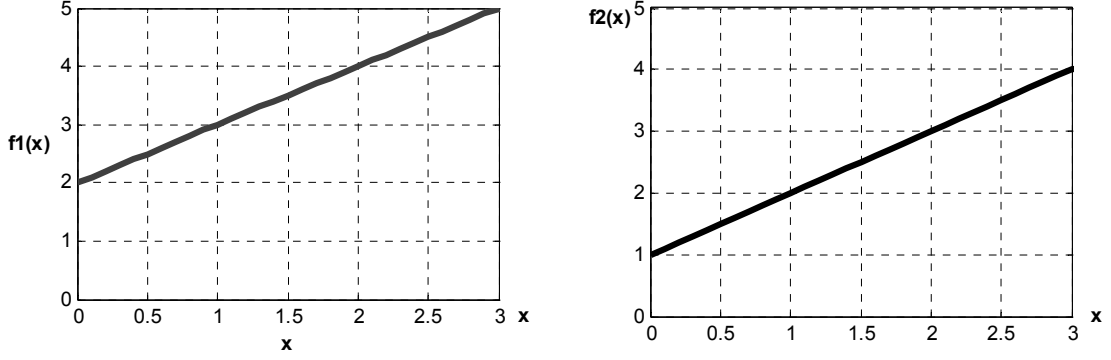
R_i :Eğer $x_1=A_{i1}(x_1)$ ve $x_2=A_{i2}(x_2)$ veve $x_k=A_{ik}(x_k)$ ise $f_i(x_1,x_2,\dots,x_k)$, $i=1,2,\dots,n$

Tek girişli bir sistem için bağıntı denklem 2.12’de olduğu gibi tanımlanır [2].

$$R(x) = \frac{A_1(x)f_1(x) + A_2(x)f_2(x) + \dots + A_k(x)f_k(x)}{A_1(x) + A_2(x) + \dots + A_k(x)} \quad (2.12)$$

Çıkış fonksiyonları $f_1(x) = 2+x$, $f_2(x) = 1+x$ olan ve üyelik fonksiyonları Şekil 2.8’deki gibi olan bir sistem için; bağıntı Şekil 2.9’daki gibi olacaktır [2].

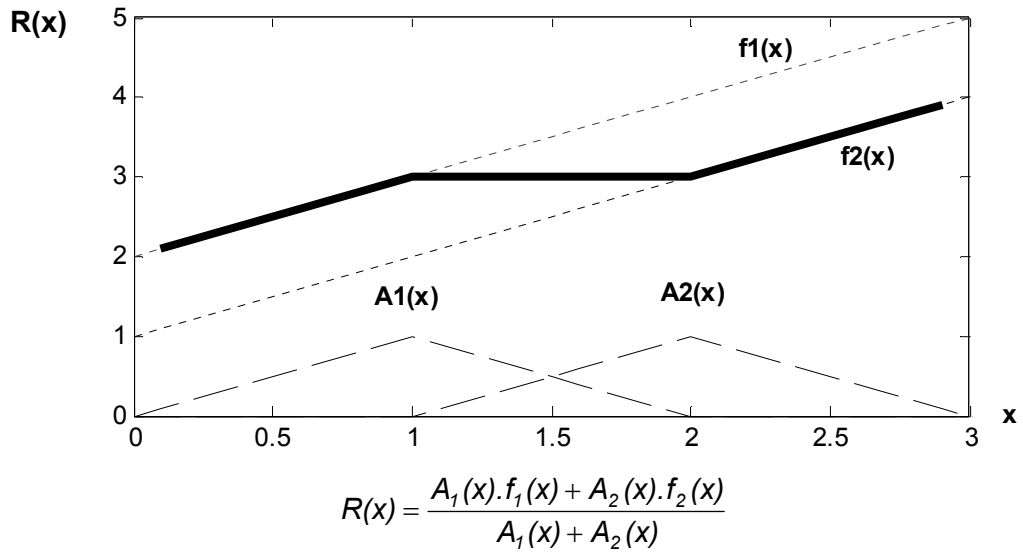




(b) $f_1(x) = 2 + x$

(c) $f_2(x) = 1 + x$

Şekil 2.8. Takagi-Sugeno-Kang Modeli Üyelik Fonksiyonları



Şekil 2.9. Takagi-Sugeno-Kang Modeli Bağıntısı

2.1.1.4. Tsukamoto modeli:

Bu modelde kullanılan bağlantı aşağıdaki gibi olacaktır.

R_i : Eğer $x = A_i(x)$ ise $y_i = C_i(A_i(x))$, $i=1, 2, \dots, n$

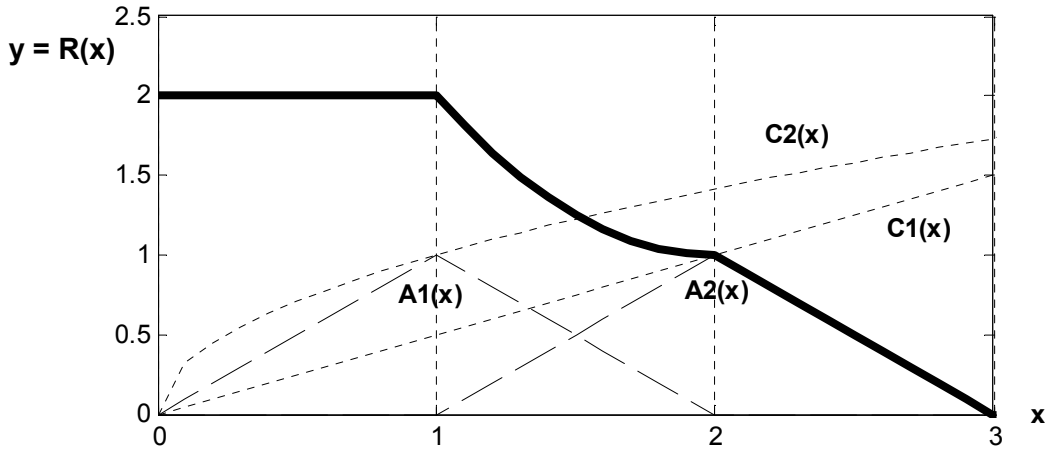
$$y = \frac{\sum_{i=1}^n C_i^{-1}(A_i(x))}{\sum_{i=1}^n A_i(x)}$$

$A_1(x)$ ve $A_2(x)$ üyelik fonksiyonları yukarıdaki diğer modellerdeki üyelik fonksiyonları gibi olan ve çıkış fonksiyonları denklem 2.13 ve 2.14'de olduğu gibi olan bir sistem için bağlantı sonucu denklem 2.15'de olduğu gibi hesaplanarak, Şekil 2.10'daki sonuç elde edilir.

$$C_1(y) = y/2 \quad C_1^{-1}(z) = 2z \quad (2.13)$$

$$C_2(y) = \sqrt{y} \quad C_2^{-1}(z) = z^2 \quad (2.14)$$

$$y = R(x) = \frac{\sum_{i=1}^n C_i^{-1}(A_i(x))}{\sum_{i=1}^n A_i(x)} = \begin{cases} 2 & , 0 \leq x \leq 1 \\ 5 - 4x - x^2 & , 1 \leq x \leq 2 \\ 3 - x & , 2 \leq x \leq 3 \end{cases} \quad (2.15)$$



$$R(x) = \frac{\sum_{i=1}^n C_i^{-1}(A_i(x))}{\sum_{i=1}^n A_i(x)}$$

Şekil 2.10. Tsukamoto Modeli Kompozisyonu

2.1.2. Netleştirme (Defuzzification) :

Önceki bölümlerde bulanıklaştırma ve bağıntıların nasıl elde edildiği anlatıldı. Netleştirme, bağıntı sonuçlarından sistemin çıkış değerinin elde edilmesi şeklinde kısaca açıklanabilir. Netleştirme için kullanılan 2 teknik mevcuttur. Bir tanesi composite moments (bütün hareketler), diğeri composite maximum (bütün maksimumlar) tekniğidir [2]. Composite moments tekniğinde belirli bir alan kullanılarak, composite maximum tekniğinde bağıntının maksimum değerine ulaştığı noktalar için belirli teknikler kullanılarak netleştirme yapılır.

Ağırlık Merkezi ve Yüksek Değer Ağırlık Merkezi teknikleri birinci metoda, Maksimum Kriteri, İlk Maksimum tekniği ve Ortalama Maksimum tekniği ikinci metoda ait örneklerdir.

2.1.2.1. Ağırlık merkezi metodu:

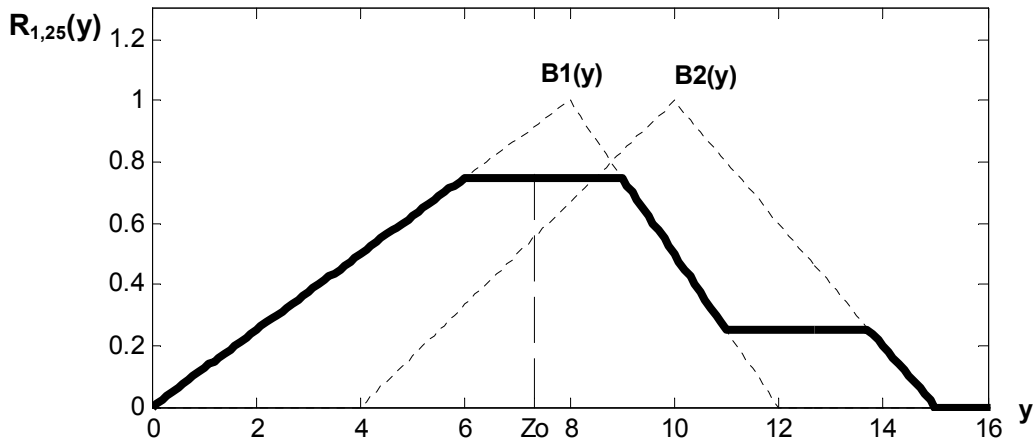
Bağıntı eğrisinin altında kalan alanın ağırlık merkezinin bulunması metodudur. Bağıntı eğrisinin fonksiyonu $C(z)$ ise ve integral alınabilir bir fonksiyon ise çıkış denklem 2.16'da olduğu gibi hesaplanır [2];

$$z_0 = \frac{\int_a^b z \cdot C(z) \cdot dz}{\int_a^b C(z) \cdot dz} \quad (2.16)$$

Eğer sınırlı bir fonksiyon ise ya da integral almak mümkün değilse örnekleme metodu ile de hesaplama denklem 2.17'de olduğu gibi hesaplanır.

$$z_0 = \frac{\sum_{j=1}^n z_j \cdot C(z_j)}{\sum_{j=1}^n C(z_j)} \quad (2.17)$$

Şekil 2.6'da hesaplanan bağıntı için ağırlık merkezi hesaplanarak netleştirme işlemi yapıldığında Şekil 2.11'deki sonuç elde edilir.



Şekil 2.11. Ağırlık Merkezi Metodu ile Netleştirme

2.1.2.2. Yüksek değerlik ağırlık merkezi metodu:

Elde edilen bağıntı eğrisinde belirli bir değer üzerinde bulunan kısmına ait alanın ağırlık merkezinin bulunması yöntemidir. Belirlenen değer altında kalan değerlerin sıfır kabul edilmesi sonucu kalan eğrinin yukarıdaki formüller ile ağırlık merkezi hesaplanarak sonuç bulunur [2].

Mamdani modeli ile elde ettiğimiz bağıntı 0,5 değerinin üzerindeki değerleri için ağırlık merkezi hesaplanır ise sonuç $z_0=7.0606$ olur.

2.1.2.3. Maksimum Ölçüt metodu:

Bu metotta en yüksek bağıntı sonucuna sahip değerler arasından rasgele bir değer seçilir. Mamdani Modeline ait bağıntıda maksimum değerler [6 9] aralığındadır. Sonuç bu aralıkta rasgele seçilir.

2.1.2.4. İlk Maksimum Değer Metodu :

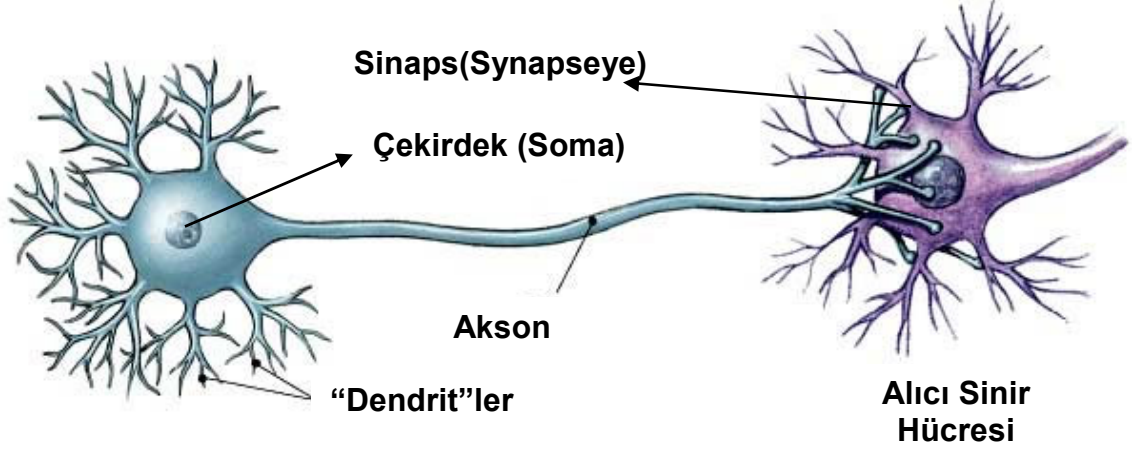
Bu metotta bağıntı fonksiyonun ilk maksimum değerine ulaştığı nokta sonuç olarak alınır.

2.1.2.5. Maksimum Nokta Ortalaması Metodu:

Bu metotta maksimum noktaları veren deęerlerin orta noktası alınır. Baęıntılı fonksiyonu birden fazla tepe noktasına sahip olduęu zaman bu metod biraz zayıf kalabilir [2].

2.2. Yapay Sinir Ağları (Neural Networks):

Yapay Sinir Ağları (YSA) sistemleri sinir hücrelerinin çalışma yapısına yönelik bir sistemin oluşturulması amacı ile ortaya çıkarılan bir yapıdır. Bir sinir hücresinin yapısı Şekil 2.12’de gösterildiği gibidir.

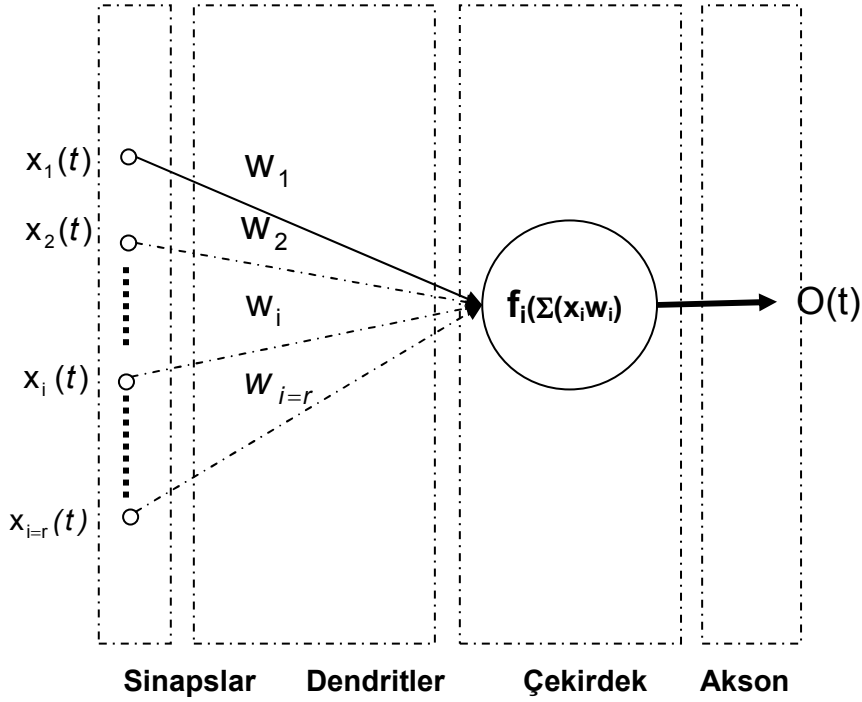


Şekil 2.12. Sinir Hücresi Yapısı

Şekil 2.12’de gösterildiği gibi temel bir biyolojik sinir hücresi sinapsler, çekirdek (soma), axon ve dendritlerden oluşur. Sinaps iki sinir hücresinin birbiri ile bağlantı noktalarıdır. Bu bağlantı fiziksel bir bağlantı olmayıp, bir hücreden diğerine elektrik sinyallerinin geçmesini sağlayan boşluklardır. Bu sinyaller dendritlerden hücre çekirdeğine ulaşır. Hücre çekirdeğine gelen uyarılar belirli bir uyarı seviyesine geldiğinde hücre bir tepki üretir. Bu tepki akson vasıtası ile sinapslara oradan da diğer sinir hücrelerine bağlantılar sağlanmış olur. Sonuç olarak dendritler vasıtası ile giriş verileri hücre çekirdeğine ulaştırılıp, burada işleminden geçirildikten sonra akson vasıtası ile sinapslara çıkış olarak ulaşır.

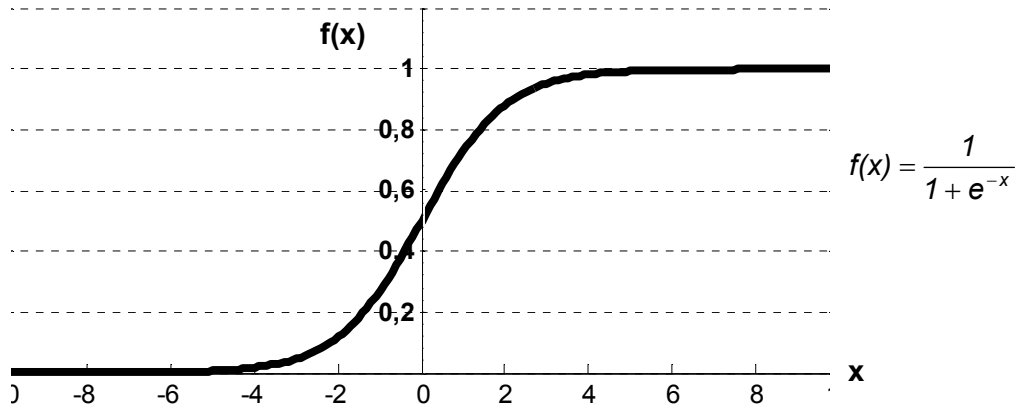
Biyolojik sinir yapısının yapısından kısaca bahsettik sonra bu yapının ışığında tasarlanmış yapay sinir hücresi yapısı Şekil 2.13’de gösterilen şekildedir. r adet girişe sahip yapay sinir hücresinin matematiksel modeli denklem 2.18’de olduğu gibidir.

$$O(t) = f_i \left(\sum_{i=1}^r x_i(t) \cdot w_i \right) \quad (2.18)$$



Şekil 2.13. “r” Adet Girişli Bir Yapay Sinir Hücresi Modeli

Yapay sinir hücresinin yapacağı işlevi, dendrit katsayıları (w_i) ve çekirdek fonksiyonu belirler. Çekirdek fonksiyonu olarak değişik fonksiyonlar kullanılabilir; ancak amacımız yapay sinir ağlarının kısaca anlatmak olduğu için fonksiyon çeşitlerinden bahsetmeden yaygın olarak kullanılan sigmoid fonksiyonunu üzerinden yapay sinir ağları anlatılacaktır. Step fonksiyonu da kullanılabilir ancak bulanık sonuçlar elde edebilmek amacı ile sigmoid fonksiyonu tercih edilmiştir.

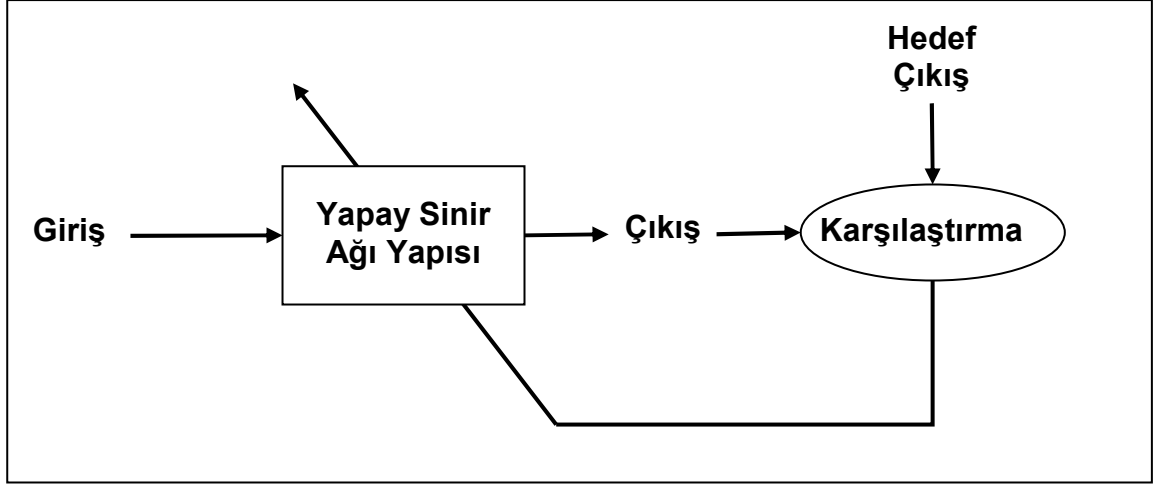


Şekil 2.14. Sigmoid Fonksiyonu

Çekirdek fonksiyonu belirlendikten sonra kullanılacak olan dendrit katsayılarının (buradan sonra sadece katsayı denilecektir) belirlenmesi gerekir. Bu aşamada katsayılar rasgele seçilmekte, öğrenme adı verilen bir takım işlemlerden geçirilerek katsayılar hesaplanabilmektedir.

2.2.1. Öğrenme:

Öğrenme, yapay sinir ağı sistemindeki katsayıların çeşitli yöntemler kullanılarak hesaplanma işlemine denir.



Şekil 2.15. Yapay Sinir Ağı Katsayılarının Ayarlanması Şematik Gösterimi

Öğrenmenin gerçekleştirilebilmesi için yapay sinir ağının amacının belirlenmesi gerekir. Bu da giriş ve çıkışlarının neler olması gerektiğinin belirlenmesi demektir. Bir sistemin bütün giriş ve çıkışlarının elde edilmesi mümkün olmayacağı için bütün giriş çıkış uzayının gerçekleştirilmesini sağlayacak bir örnekleme uzayı oluşturularak bu problem ortadan kaldırılabilir. Bu örnekleme uzayına eğitim verileri, eğitim çiftleri vs. denilebilmektedir.

Eğitim çiftleri oluşturulduktan sonra Şekil 2.15'te gösterilen şematik yapıda olduğu gibi bütün eğitim girişleri için sistem çıkışı hesaplanır. Daha sonra hesaplanan çıkışlar ile eğitim veri çıkışları karşılaştırılarak sistemin hatası hesaplanır. En son olarak da eğitim algoritmaları kullanılarak sistem eğitilmesi sağlanır. En yaygın kullanılan öğrenme metodu delta kuralı ve geri yayılım metodudur.

2.2.2. Delta Kural ve Geri Yayılım (Backpropagation):

Bir yapay sinir hücresi çıkışı Denklem 2.18'de olduğu şekilde hesaplanır. Bu hesaplanan çıkış değerlerinin istenen çıkış değerlerini eşit olabilmesi için bir eğitim yöntemi kullanılarak, sistem katsayılar değiştirilip sistemin eğitilmesi gerekir.

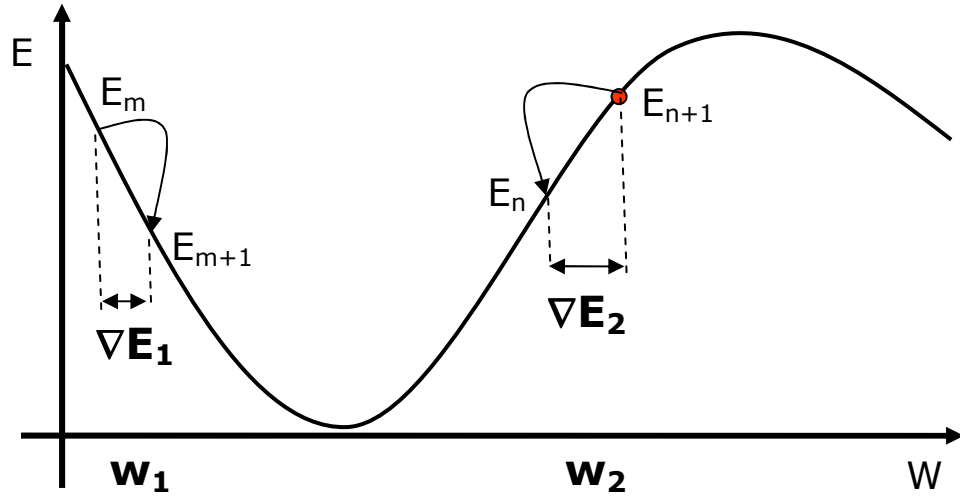
Delta Kuralı, istenen çıkış değerleri ile hesaplanan çıkış değerleri arasındaki hatanın, yeniden değeri hesaplanacak olan "w" katsayısına göre türevinin, negatif değerinin belirli bir oranda katsayıya tekrar eklenmesi ile gerçekleşir. Ekleme oranına eğitim katsayısı denilmekte olup "η" ile gösterilir.

Sistemin hata fonksiyonu denklem 2.19 ve 2.20'de olduğu gibi hesaplanır.

$$E = \sum_{q=1}^N E(q) \quad (2.19)$$

$$E(q) = \frac{1}{2} \sum_{j=1}^p (O_j^4(q) - Y_j(q))^2 \quad (2.20)$$

Hata fonksiyonu E'nin w_i noktasındaki gradyeni ∇E , $\frac{\partial E}{\partial w_i}$ şeklindedir. Hata fonksiyonunun değişik " w_i " katsayılarına ilişkin grafiği Şekil 2.16'de verildiği gibidir.



Şekil 2.16. Örnek Hata Katsayı Değişim Grafiği

Amaç hatanın minimum olduğu noktadaki w katsayısını bulmaktır. ∇E_1 negatif sonuçlu, ∇E_2 de pozitif sonuçludur. Görüldüğü gibi hatanın gradyeni her zaman üst tepe noktasının doğrultusunu gösterecektir. Bu nedenle hatanın minimum noktasına ulaşmak için, negatif gradyen değeri güncelleme için kullanılır.

$$w_i \rightarrow w_i + \Delta w_i \quad (2.21)$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \quad (2.22)$$

N adet eğitim verisi olan bir sistem için, q deneme sayısını belirtmek üzere, hatanın " w_i " katsayısına göre türevi denklem 2.23'de verildiği gibidir.

$$\frac{\partial E}{\partial w_i} = \sum_{q=1}^N \frac{\partial E^q}{\partial w_i} \quad (2.23)$$

Tek bir yapay sinir hücresi üzerinde delta kuralı anlattıktan sonra, çok katmanlı yapay sinir ağları sisteminde, son katmandan ilk katmana kadar olan katsayıların eğitilmesi anlatılacaktır. Bu yöntem geri yayılım metodu denilir [2].

Şekil 2.17'de verilen 3 katmanlı yapay sinir ağı yapısı üzerinden geri yayılım metodu anlatılacaktır.

Her bir katmana ait nöronların çıkışlarının, her bir deneme için hesaplanan çıkış değerleri aşağıdaki gibi tanımlanır.

$$1. \text{ Katman k. nöron için çıkış : } O_k^{2q}$$

$$2. \text{ Katman i. nöron için çıkış : } O_i^{3q}$$

$$3. \text{ Katman j. nöron için çıkış : } O_j^{4q}$$

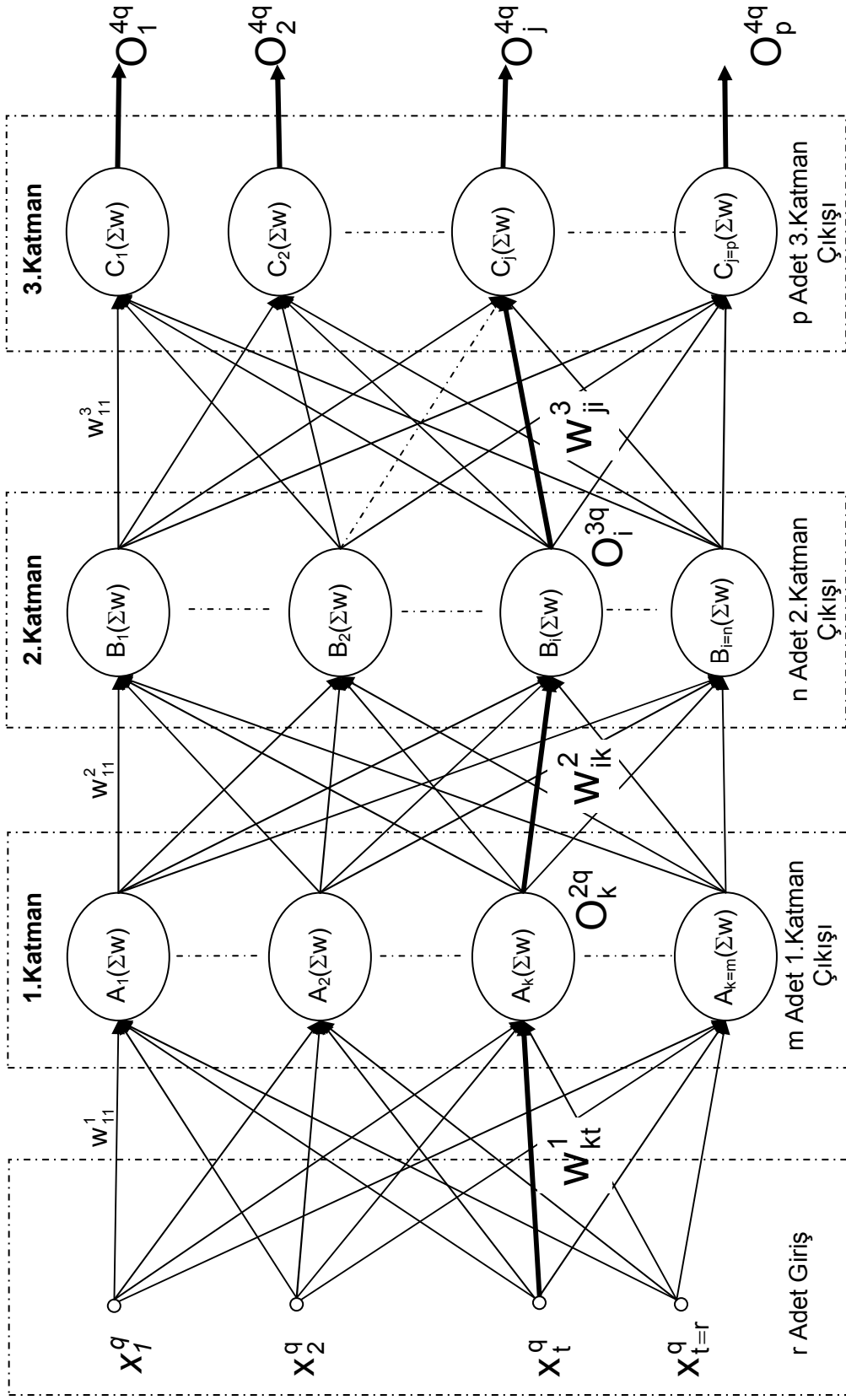
3. katman çıkışları sistem çıkışlarını ifade eder. Girişler için x ifadesi yerine O_k^{1q} ifadeleri kullanılacaktır. Bunun sebebi de, her bir katman çıkışı bir sonraki katmanın girişini ifade etmekte olduğu için, ilk girişler için de benzerliğin sağlanmasıdır. Yapay sinir ağları içerisindeki Σw ifadesi, yapay sinir hücresi ile bağlantısı olan girişlerin w katsayıları ile çarpımlarının toplamını gösterir.

$$O_k^{2q} = f_k \left(\sum_{t=1}^r w_{kt}^1 \cdot O_t^{1q} \right) \quad (2.24)$$

$$O_i^{3q} = f_i \left(\sum_{k=1}^m w_{ik}^2 \cdot O_k^{2q} \right) \quad (2.25)$$

$$O_j^{4q} = f_j \left(\sum_{i=1}^n w_{ji}^1 \cdot O_i^{3q} \right) \quad (2.26)$$

Katman çıkışları Denklem 2.24, 2.25 ve 2.26'da olduğu gibi hesaplanır.



Şekil 2.17. 3 Katmanlı Yapay Sinir Ağı

3. katman çıkışı olan O_j^{4q} , aynı zamanda sistem çıkışı olup, eğitim verilerinin çıkışı Y_j^q olmak üzere, sistemin hata denklemleri denklem 2.27 ve 2.28'de olduğu gibi hesaplanır. N, eğitim deneme adedini tanımlamakta olup, denklem 2.28'deki denklem kullanılarak, her bir eğitim denemesi için hata hesaplanarak toplam hata bulunur.

$$E = \sum_{q=1}^N E^q \quad (2.27)$$

$$E^q = \frac{1}{2} \sum_{j=1}^p (O_j^{4q} - Y_j^q)^2 \quad (2.28)$$

Geri yayılım metodunda en son katmandan geriye doğru katsayılar güncellenir.

$$w_{ji}^{3'} = w_{ji}^3 + \Delta w_{ji}^3 \quad (2.29)$$

$$\Delta w_{ji}^3 = \sum_{q=1}^N \Delta^q w_{ji}^3 \quad (2.30)$$

$$\Delta^q w_{ji}^3 = -\eta \frac{\partial E^q}{\partial w_{ji}^3} \quad (2.31)$$

$$\frac{\partial E^q}{\partial w_{ji}^3} = \frac{\partial E^q}{\partial O_j^{4q}} \cdot \frac{\partial O_j^{4q}}{\partial w_{ji}^3} \quad (2.32)$$

$$\frac{\partial E^q}{\partial O_j^{4q}} = O_j^{4q} - Y_j^q \quad (2.33)$$

$$\frac{\partial O_j^{4q}}{\partial w_{ji}^3} = \frac{\partial}{\partial w_{ji}^3} f_j(w_{j1}^3 \cdot O_1^{3q} + w_{j2}^3 \cdot O_2^{3q} + \dots + w_{ji}^3 \cdot O_i^{3q} + \dots + w_{jn}^3 \cdot O_n^{3q}) \quad (2.34)$$

$$\frac{\partial O_j^{4q}}{\partial w_{ji}^3} = O_i^{3q} \cdot f_j' \left(\sum_{i=1}^n w_{ji}^3 \cdot O_i^{3q} \right) \quad (2.35)$$

$$\Delta^q w_{ji}^3 = -\eta (O_j^{4q} - Y_j^q) \cdot f_j' \left(\sum_{i=1}^n w_{ji}^3 \cdot O_i^{3q} \right) \cdot O_i^{3q} \quad (2.36)$$

$$\delta_j^{3q} = (O_j^{4q} - Y_j^q) \cdot f_j' \left(\sum_{i=1}^n w_{ji}^3 \cdot O_i^{3q} \right) \quad (2.37)$$

Denklem 2.37'deki gibi tanımlama yapılarak [2] kısaltma yapılırsa eğitim denemesi için katsayıya eklenecek değer denklem 2.38'de olduğu gibi yazılabilir. Bu kısaltma her bir katman için yapılacaktır.

$$\Delta^q w_{ji}^3 = -\eta \cdot O_i^{3q} \cdot \delta_j^{3q} \quad (2.38)$$

3. Katman hesaplandıktan sonra 2. Katman katsayılarının hesaplanmasında da aynı işlem basamakları uygulanacaktır.

$$w_{ik}^{2'} = w_{ik}^2 + \Delta w_{ik}^2 \quad (2.39)$$

$$\Delta w_{ik}^2 = \sum_{q=1}^N \Delta^q w_{ik}^2 \quad (2.40)$$

$$\Delta^q w_{ik}^2 = -\eta \frac{\partial E^q}{\partial w_{ik}^2} \quad (2.41)$$

$$\frac{\partial E^q}{\partial w_{ik}^2} = \frac{\partial E^q}{\partial O_i^{3q}} \cdot \frac{\partial O_i^{3q}}{\partial w_{ik}^2} \quad (2.42)$$

$$\frac{\partial O_i^{3q}}{\partial w_{ik}^2} = \frac{\partial}{\partial w_{ik}^2} f_i(w_{i1}^2 \cdot O_1^{2q} + w_{i2}^2 \cdot O_2^{2q} + \dots + w_{ik}^2 \cdot O_k^{2q} + \dots + w_{im}^2 \cdot O_m^{2q}) \quad (2.43)$$

$$\frac{\partial O_i^{3q}}{\partial w_{ik}^2} = O_k^{2q} \cdot f_i' \left(\sum_{k'=1}^m w_{ik'}^2 \cdot O_{k'}^{2q} \right) \quad (2.44)$$

Hatanın 2. katmanın bir çıkışı için türevi alınırken kullanılacak denklem 2.45'de olduğu gibidir. Burada 2. katman çıkışının 3. katman çıkışlarının her birinde bir etkisi mevcut olmasından dolayı 3. katmandaki bütün çıkışlar için türev hesaplanması gerekir.

$$\frac{\partial E^q}{\partial O_i^{3q}} = \sum_{j=1}^p \left(\frac{\partial E^q}{\partial O_j^{4q}} \frac{\partial O_j^{4q}}{\partial O_i^{3q}} \right) \quad (2.45)$$

$$\frac{\partial E^q}{\partial O_j^{4q}} = O_j^{4q} - Y_j^q \quad (2.46)$$

$$\frac{\partial O_j^{4q}}{\partial O_i^{3q}} = \frac{\partial}{\partial O_i^{3q}} f_j(w_{j1}^3 \cdot O_1^{3q} + w_{j2}^3 \cdot O_2^{3q} + \dots + w_{ji}^3 \cdot O_i^{3q} + \dots + w_{jn}^3 \cdot O_n^{3q}) \quad (2.47)$$

$$\frac{\partial O_j^{4q}}{\partial O_i^{3q}} = w_{ji}^3 \cdot f_j' \left(\sum_{i=1}^n w_{ji}^3 \cdot O_i^{3q} \right) \quad (2.48)$$

$$\frac{\partial E^q}{\partial O_i^{3q}} = \sum_{j=1}^p \left((O_j^{4q} - Y_j^q) \cdot w_{ji}^3 \cdot f_j' \left(\sum_{i'=1}^n w_{ji}^3 \cdot O_{i'}^{3q} \right) \right) \quad (2.49)$$

Denklem 2.37'de kullanılan kısaltma terimi denklem 2.49'da yerine konulursa denklem 2.50 elde edilir.

$$\frac{\partial E^q}{\partial O_i^{3q}} = \sum_{j=1}^p (w_{ji}^3 \cdot \delta_j^{3q}) \quad (2.50)$$

Denklem 2.44 ve 2.50 birleştirildiğinde hatanın katsayıya göre türevi denklem 2.51 elde edilir.

$$\frac{\partial E^q}{\partial W_{ik}^2} = O_k^{2q} \cdot f_i' \left(\sum_{k'=1}^m w_{ik'}^2 \cdot O_{k'}^{2q} \right) \cdot \sum_{j=1}^p (w_{ji}^3 \cdot \delta_j^{3q}) \quad (2.51)$$

$$\delta_i^{2q} = f_i' \left(\sum_{k'=1}^m w_{ik'}^2 \cdot O_{k'}^{2q} \right) \cdot \sum_{j=1}^p (w_{ji}^3 \cdot \delta_j^{3q}) \quad (2.52)$$

3. Katmanın hesaplanmasında olduğu gibi denklem 2.52, denklem 2.51'de yerine konulursa denklem 2.53 elde edilerek kısaltma yapılabilir.

$$\Delta^q W_{ik}^2 = -\eta \cdot O_k^{2q} \cdot \delta_i^{2q} \quad (2.53)$$

1. Katman katsayılarının hesaplanması:

$$w_{kt}^{1'} = w_{kt}^1 + \Delta w_{kt}^1 \quad (2.54)$$

$$\Delta w_{kt}^1 = \sum_{q=1}^N \Delta^q w_{kt}^1 \quad (2.55)$$

$$\Delta^q w_{kt}^1 = -\eta \frac{\partial E^q}{\partial w_{kt}^1} \quad (2.56)$$

$$\frac{\partial E^q}{\partial w_{kt}^1} = \frac{\partial E^q}{\partial O_k^{2q}} \cdot \frac{\partial O_k^{2q}}{\partial w_{kt}^1} \quad (2.57)$$

$$\frac{\partial O_k^{2q}}{\partial w_{kt}^1} = \frac{\partial}{\partial w_{kt}^1} f_k (w_{k1}^1 \cdot O_1^{1q} + w_{k2}^1 \cdot O_2^{1q} + \dots + w_{kt}^1 \cdot O_t^{1q} + \dots + w_{kr}^1 \cdot O_r^{1q}) \quad (2.58)$$

$$\frac{\partial O_k^{2q}}{\partial w_{kt}^1} = O_t^{1q} \cdot f_k' \left(\sum_{t'=1}^r w_{kt'}^1 \cdot O_{t'}^{1q} \right) \quad (2.59)$$

$$\frac{\partial E^q}{\partial O_k^{2q}} = \sum_{i=1}^n \left(\frac{\partial E^q}{\partial O_i^{3q}} \frac{\partial O_i^{3q}}{\partial O_k^{2q}} \right) \quad (2.60)$$

$$\frac{\partial E^q}{\partial O_i^{3q}} = \text{Denklem 2.50'de hesaplandı.} \quad (2.61)$$

$$\frac{\partial O_i^{3q}}{\partial O_k^{2q}} = \frac{\partial}{\partial O_k^{2q}} f_i(w_{i1}^2 \cdot O_1^{2q} + w_{i2}^2 \cdot O_2^{2q} + \dots + w_{ik}^2 \cdot O_k^{2q} + \dots + w_{im}^2 \cdot O_m^{2q}) \quad (2.62)$$

$$\frac{\partial O_i^{3q}}{\partial O_k^{2q}} = w_{ik}^2 \cdot f_i' \left(\sum_{k'=1}^m w_{ik'}^2 \cdot O_{k'}^{2q} \right) \quad (2.63)$$

Denklem 2.50 ve denklem 2.63 birleştirildiğinde denklem 2.64 elde edilir.

$$\frac{\partial E^q}{\partial O_k^{2q}} = \sum_{i=1}^n \left(w_{ik}^2 \cdot f_i' \left(\sum_{k'=1}^m w_{ik'}^2 \cdot O_{k'}^{2q} \right) \cdot \sum_{j=1}^p (w_{ji}^3 \cdot \delta_j^{3q}) \right) \quad (2.64)$$

$$\delta_i^{2q} = f_i' \left(\sum_{k'=1}^m w_{ik'}^2 \cdot O_{k'}^{2q} \right) \cdot \sum_{j=1}^p (w_{ji}^3 \cdot \delta_j^{3q}) \quad (2.65)$$

3 ve 2. Katmanın hesaplanmasında olduğu gibi denklem 2.65, denklem 2.64'de yerine konulursa denklem 2.66 elde edilerek kısaltma yapılabilir.

$$\frac{\partial E^q}{\partial W_{kt}^1} = O_t^{1q} \cdot f_k' \left(\sum_{t'=1}^r w_{kt'}^1 \cdot O_{t'}^{1q} \right) \cdot \sum_{i=1}^n (w_{ik}^2 \cdot \delta_i^{2q}) \quad (2.66)$$

$$\delta_k^{1q} = f_k' \left(\sum_{t'=1}^r w_{kt'}^1 \cdot O_{t'}^{1q} \right) \cdot \sum_{i=1}^n (w_{ik}^2 \cdot \delta_i^{2q}) \quad (2.67)$$

$$\Delta^q W_{kt}^1 = -\eta \cdot O_t^{1q} \cdot \delta_k^{1q} \quad (2.68)$$

2.3. Adaptif Ağ Yapısına Dayalı Bulanık Mantık Sistemleri (Adaptive Network Fuzzy Inference System (ANFIS)) :

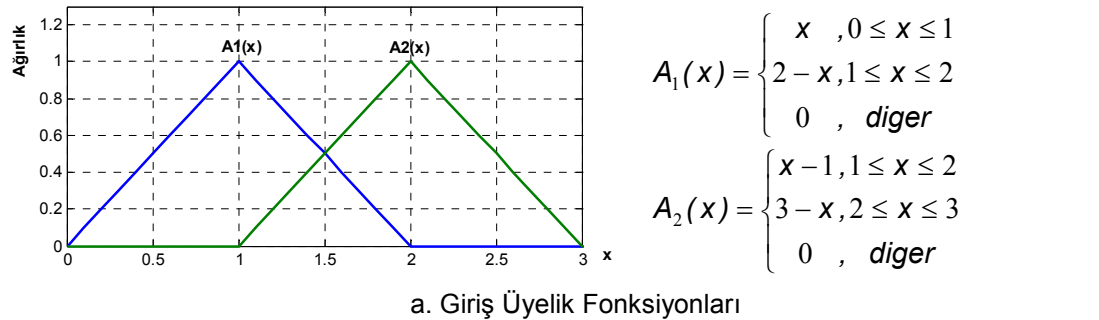
ANFIS yapısı Takagi-Sugeno-Kang (TSK) modelinin, yapay sinir ağı yapısında kullanılan öğrenme algoritmalarının kullanılarak elde edilmesi çerçevesinde meydana çıkarılmış özel bir bulanık mantık modelidir.

Bu model, bulanık mantık sistemlerine, yapay sinir ağlarındaki öğrenme kabiliyetinin kazandırılmak amacı ile geliştirilmiştir. Yapay sinir ağları yapısında nöronlara ait katsayılar değiştirilmekte olup nöronların sahip olduğu fonksiyonlar tasarımcı tarafından belirlenmektedir. Bulanık mantık sistemlerinde ise üyelik fonksiyonları katsayıları ve aralarındaki kurallar tasarımcının tecrübelerine dayalı bir yapıya sahiptir.

ANFIS yapısı sayesinde sistem katsayıları ile birlikte üyelik fonksiyonları da değiştirilebilir.

2.3.1. Takagi-Sugeno-Kang (TSK) modeli:

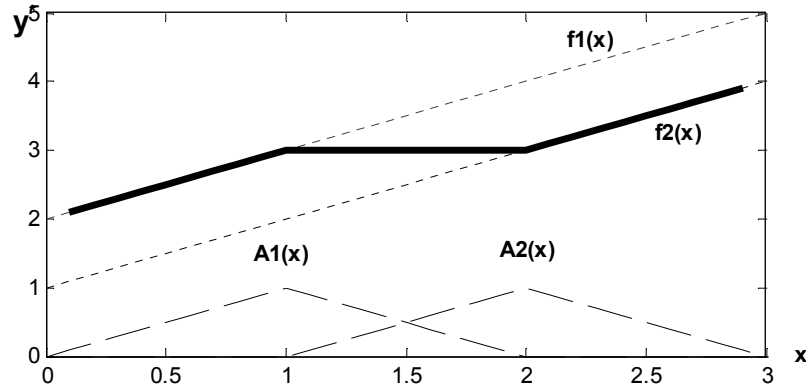
x girişine ait üyelik fonksiyonları $A_1(x)$ ve $A_2(x)$ olan, y çıkışı ile ilişkili $f_1(x)$ ve $f_2(x)$ fonksiyon bilgileri aşağıdaki gibi olan bir sistemde;



Şekil 2.18. Takagi-Sugeno-Kang Modeline Ait Örnek Üyelik Fonksiyonları

TSK modeline ait i. kural; "Eğer $x = A_i(x)$ ise $y = f_i(x)$ şeklindedir ve sisteme ait çıkış denklem 2.69'da olduğu gibi hesaplanır ve sonucun grafik olarak gösterimi Şekil 2.19'da olduğu gibidir.

$$y^* = \frac{A_1(x)f_1(x) + A_2(x)f_2(x)}{A_1(x) + A_2(x)} \quad (2.69)$$

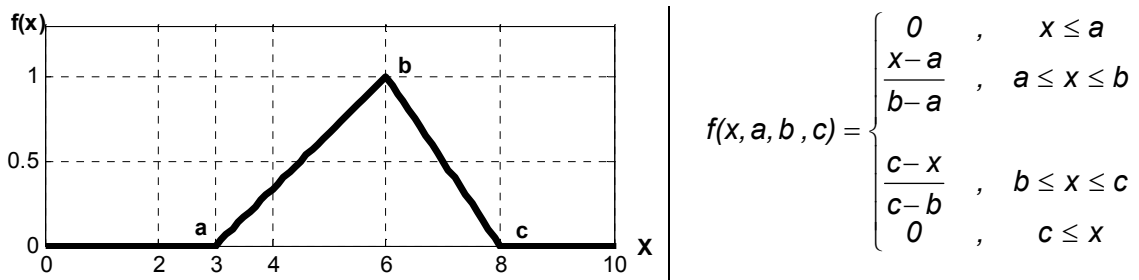


Şekil 2.19. TSK Modeli Çıkış Fonksiyonu

TSK modelinde giriş ve çıkışlar arasındaki bağıntı Şekil 2.19'de olduğu gibidir. Bu model ANFIS yapısına uyarlanıp, yapay sinir ağlarında kullanılan öğrenme algoritmaları ile hesaplamalar yapıldıktan sonra, girişlere ait üyelik fonksiyonlarının katsayıları ve çıkış fonksiyonlarına ait katsayıları elde edilir.

2.3.2. ANFIS Yapısı:

Şekil 2.21'de 2 Girişli ve 3'er üyelik fonksiyonlu ANFIS yapısı modellenmiştir. Bu model üzerinde giriş ve çıkışlar arasındaki işlemlerin nasıl gerçekleştiğini özümseydikten sonra; farklı sayılarda üyelik fonksiyonlarına sahip çok girişli ve çok çıkışlı bir sistemlerde, geri yayılım (backpropagation) ve melez (hybrid) öğrenme algoritmaları kullanılarak sistem katsayılarının nasıl elde edildiği ve öğrenme algoritmalarının nasıl çalıştığını incelenecektir. Kullanacağımız üyelik fonksiyonu üçgen fonksiyonu olacaktır.



Şekil 2.20. Üçgen Fonksiyon

2 girişli ve 3'er üyelik fonksiyonlu ANFIS için yapısına ait kurallar;

Eğer $x_1=A1_1(x_1)$ ve $x_2=A2_1(x_2)$ İSE $y_1=f_1(z,x_1,x_2)$

Eğer $x_1=A1_1(x_1)$ ve $x_2=A2_2(x_2)$ İSE $y_1=f_2(z,x_1,x_2)$

Eğer $x_1=A1_1(x_1)$ ve $x_2=A2_3(x_2)$ İSE $y_1=f_3(z,x_1,x_2)$

Eğer $x_1=A1_2(x_1)$ ve $x_2=A2_1(x_2)$ İSE $y_1=f_4(z,x_1,x_2)$

Eğer $x_1=A1_2(x_1)$ ve $x_2=A2_2(x_2)$ İSE $y_1=f_5(z,x_1,x_2)$

Eğer $x_1=A1_2(x_1)$ ve $x_2=A2_3(x_2)$ İSE $y_1=f_6(z,x_1,x_2)$

Eğer $x_1=A1_3(x_1)$ ve $x_2=A2_1(x_2)$ İSE $y_1=f_7(z,x_1,x_2)$

Eğer $x_1=A1_3(x_1)$ ve $x_2=A2_2(x_2)$ İSE $y_1=f_8(z,x_1,x_2)$

Eğer $x_1=A1_3(x_1)$ ve $x_2=A2_3(x_2)$ İSE $y_1=f_9(z,x_1,x_2)$

Bu kuralla doğrultusunda sistem çıkışı 5 aşamada hesaplanmaktadır.

1. aşamada üyelik fonksiyonlarının çıkışları hesaplanır.

1. giriş x_1 , üyelik fonksiyonları $A1_1(x_1)$, $A1_2(x_1)$ ve $A1_3(x_1)$, 2. giriş x_2 ve üyelik fonksiyonları $A2_1(x_2)$, $A2_2(x_2)$ ve $A2_3(x_2)$ 'dir. x_1 ve x_2 değerleri için bütün üyelik fonksiyonlarının çıkışları hesaplanır.

2. aşamada üyelik fonksiyon çıkışları arasındaki ilişkiler oluşturularak, bu ilişkilerin çıkış değerleri hesaplanır. Üyelik fonksiyonları arasındaki ilişkiler, 1. girişteki her bir üyelik fonksiyonu ile 2. girişteki her bir üyelik fonksiyonun eşlenmesi ile elde edilir. 1. giriş için üç ve 2. giriş için üç üyelik fonksiyonu olduğuna göre toplam 9 adet ilişki elde edilecektir.

1. ilişki : $K_1(x_1,x_2) = A1_1(x_1).A2_1(x_2)$

2. ilişki : $K_2(x_1,x_2) = A1_1(x_1).A2_2(x_2)$

3. ilişki : $K_3(x_1,x_2) = A1_1(x_1).A2_3(x_2)$

4. ilişki : $K_4(x_1,x_2) = A1_2(x_1).A2_1(x_2)$

5. ilişki : $K_5(x_1,x_2) = A1_2(x_1).A2_2(x_2)$

6. ilişki : $K_6(x_1,x_2) = A1_2(x_1).A2_3(x_2)$

7. ilişki : $K_7(x_1,x_2) = A1_3(x_1).A2_1(x_2)$

8. ilişki : $K_8(x_1,x_2) = A1_3(x_1).A2_2(x_2)$

9. ilişki : $K_9(x_1,x_2) = A1_3(x_1).A2_3(x_2)$

3. aşamada üyelik fonksiyonları arasındaki ilişkiler normalleştirme denilen bir işlemde geçirilir. İlişkiye ait normalleştirilmiş değer, ilişki çıkış değerinin bütün ilişkilerin çıkış değerlerinin toplamına bölünmesi ile elde edilir. Bütün ilişkilerin toplamı (K_7) aşağıdaki gibi hesaplanır.

$$K_T = A1_1(x_1).A2_1(x_2)+A1_1(x_1).A2_2(x_2)+A1_1(x_1).A2_3(x_2)+A1_2(x_1).A2_1(x_2)+$$

$$A1_2(x_1).A2_2(x_2)+A1_2(x_1).A2_3(x_2)+A1_3(x_1).A2_1(x_2)+A1_3(x_1).A2_2(x_2)+ A1_3(x_1).A2_3(x_2)$$

İlişkilere ait normal değerleri aşağıda verildiği gibi hesaplanır.

$$\text{Normal 1: } N_1(x_1, x_2) = \frac{A1_1(x_1).A2_1(x_2)}{K_T} = \frac{K_1}{K_T}$$

.

.

$$\text{Normal 9: } N_9(x_1, x_2) = \frac{A1_3(x_1).A2_3(x_2)}{K_T} = \frac{K_9}{K_T}$$

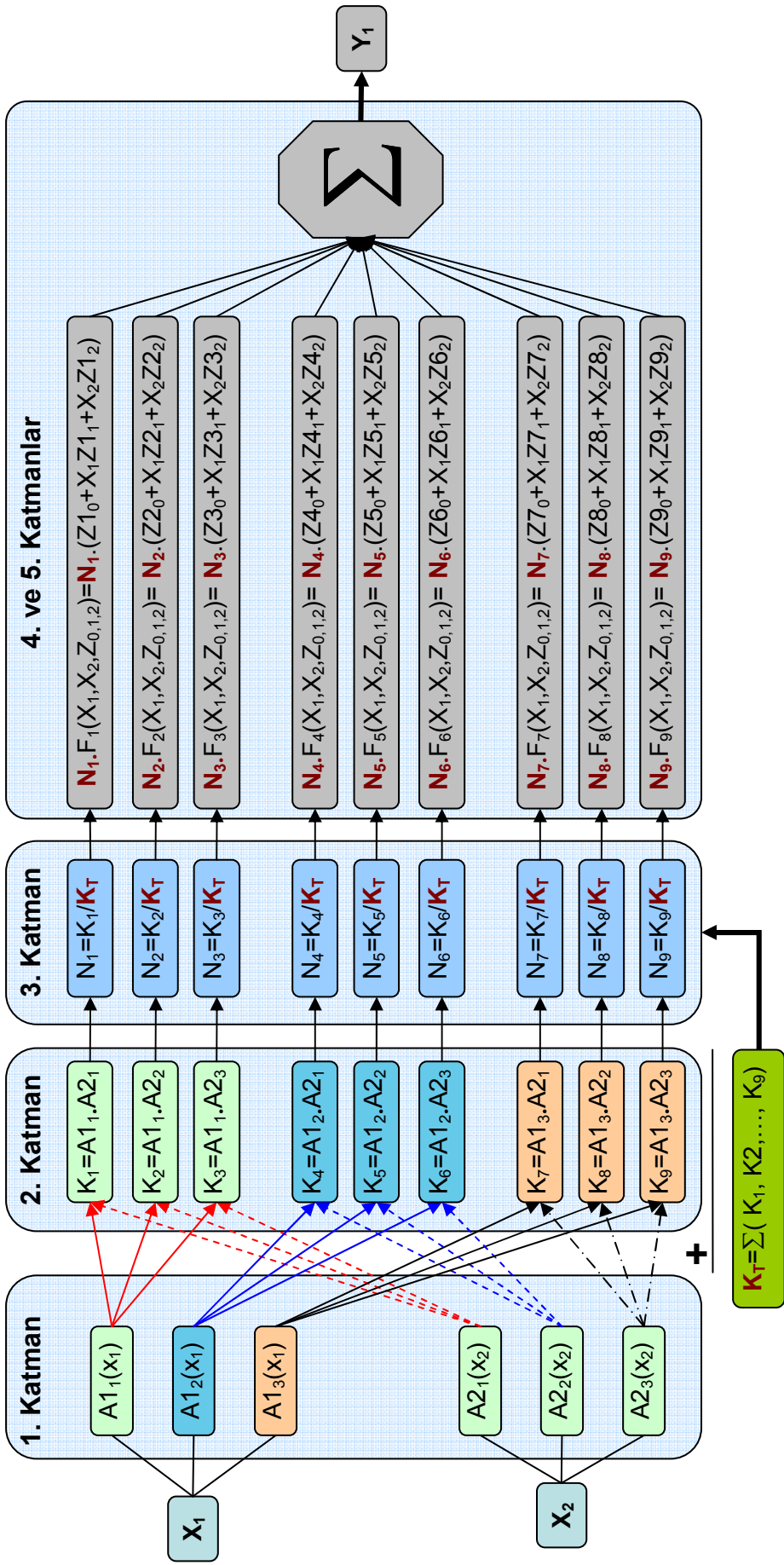
4. aşamada sistemin çıkış fonksiyonları hesaplanır.

$$F_i(x_1; x_2; Z_{0,1,2}) = Z_{i0} + x_1.Z_{i1} + x_2.Z_{i2} \quad (i=1...9)$$

5. aşamada normalleştirme çıkışları ile sistem çıkış fonksiyonları çarpılmakta ve elde edilen çarpım sonuçları toplanarak, sistem çıkış değeri denklem 2.70 ve 2.71'de olduğu gibi elde edilir.

$$y(x_1, x_2) = \frac{K_1.F_1 + K_2.F_2 + \dots + K_8.F_8 + K_9.F_9}{K_1 + K_2 + \dots + K_8 + K_9} \quad (2.70)$$

$$y(x_1, x_2) = N_1.F_1 + N_2.F_2 + \dots + N_8.F_8 + N_9.F_9 \quad (2.71)$$



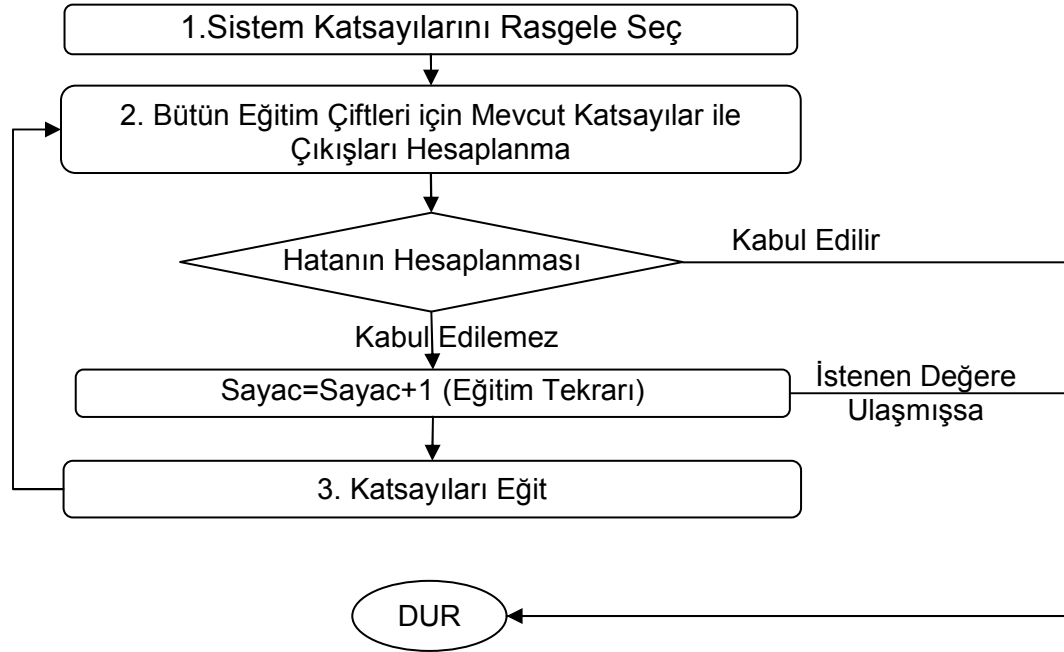
Şekil 2.21. 2 Girişli 3'er Üyelik Fonksiyonu ANFIS Yapısı

2.3.3. Çok Girişli ve Çok Çıkışlı ANFIS Yapısının Geri Yayılım (Backpropagation) Metodu ile Katsayılarının Hesaplanması:

Bu bölümde yapay sinir ağları bölümünde anlatılan delta kuralı ve geri yayılım metodu kullanarak çok girişli ve çok çıkışlı bir ANFIS'in katsayılarının hesaplamaları yapılacaktır.

Geri yayılım metodu, mevcut katsayılar ile hesaplanan çıkış değerlerinin hataları hesaplanarak, hatanın katsayıya göre türevinin eğitim katsayısı adı verilen ($\eta \in 0-1$) bir katsayı ile çağrılarak katsayıya eklenmesi ile gerçekleştirilir.

Giriş-Çıkış eğitim verileri Giriş: x (q : eğitim veri no, k : giriş no) ve Çıkış: y (q : eğitim veri no, s : çıkış no) olarak tanımlanır.

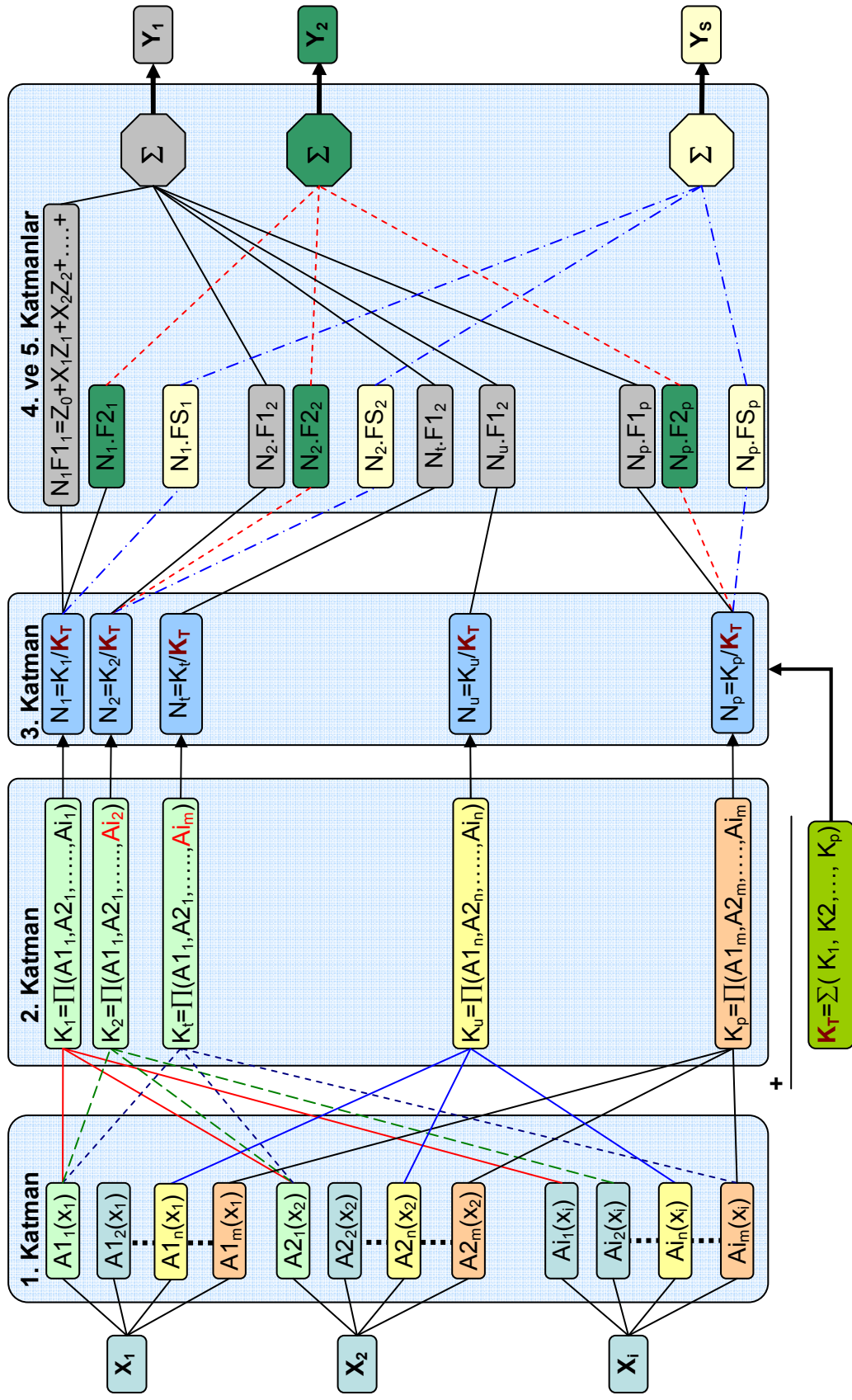


Şekil 2.22. Geri Yayılım Algoritması

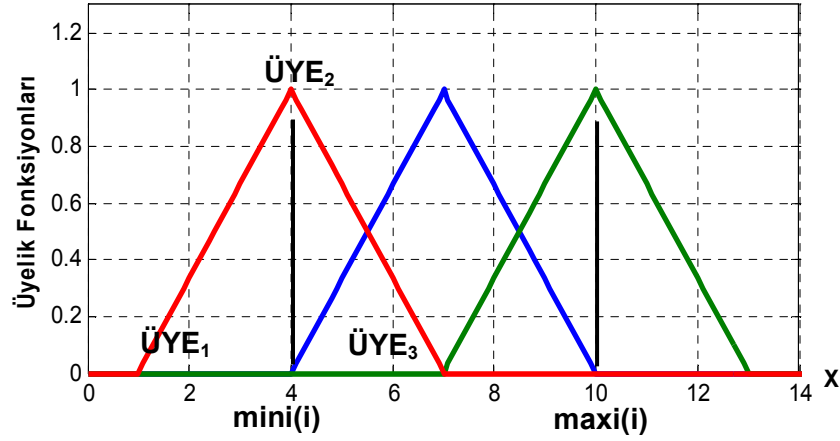
2.3.3.1. Sistem Katsayılarının Rastgele Seçilmesi:

Üyelik fonksiyon katsayılarını belirlemek için girişe ait eğitim verilerinin maksimum ve minimum noktalarını belirlenerek, üyelik fonksiyonları bu aralıkta eşit olarak dağıtılır. Şekil 2.24'de örnek üyelik fonksiyonları gösterilmiştir.

Çıkış fonksiyon katsayıları sıfır olarak seçilmiştir.



Şekil 2.23. Çok Girişli ve Çok Çıkışlı ANFIS Modeli



Şekil 2.24. Başlangıç Üyelik Fonksiyonları

Şekil 2.20'de gösterildiği gibi verilen üçgen üyelik fonksiyonunun tanımlanması için 3 adet değişken kullanılır. ANFIS eğitim programı algoritmaları anlatılırken değişkenlerin daha anlaşılır olması için, Şekil 2.24'de de gösterildiği üzere, a değişkeni için ÜYE₁, b için ÜYE₂ ve c için ÜYE₃ değişkeni tanımlamaları yapılmıştır.

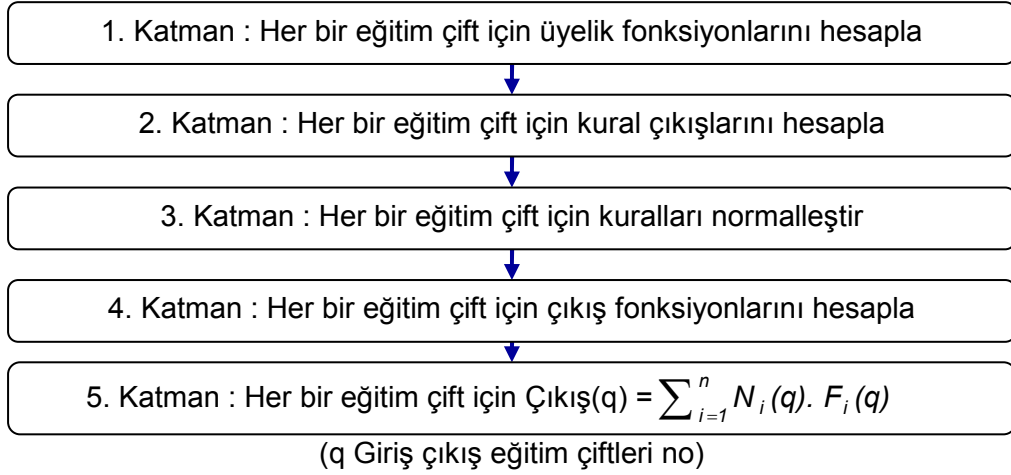
Kullanılan değişkenlerin program akışı içerisinde dizi gösteriminin tanımlanması aşağıda gösterildiği gibidir.

“Üyelik Fonksiyon Tanımlaması: UYE (k: giriş no, i: Üyelik No, Katsayı No)”

$$f(x, a : UYE(k, i, 1), b : UYE(k, i, 2), c : UYE(k, i, 3)) = \begin{cases} 0 & , \quad x \leq a \\ \frac{x-a}{b-a} & , \quad a \leq x \leq b \\ \frac{c-x}{c-b} & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases} \quad (2.72)$$

Çıkış fonksiyonları katsayılarının tanımlanması için “Z” değişkeni kullanılmış olup, değişkenin dizi gösterimi “Z (Çıkış No, İlişki No, Giriş No)” şeklinde tanımlanmıştır. Giriş numarası için, giriş sayısı artı bir olarak kullanılan değer ile fonksiyondaki sabit değer tanımlanmıştır.

2.3.3.2. Mevcut Katsayılar ile Çıkışların Hesaplanması:



Şekil 2.25. Mevcut Katsayılar ile Çıkış Değerlerinin Hesaplanması Algoritması

Burada rastgele belirlenen katsayılar ile bütün eğitim verileri için sistem çıkışı hesaplanır. Hesaplamalar esnasında her bir katmanda hesaplanan değerler programın ilerleyen bölümlerinde kullanılmak amacıyla belirli değişkenlerde saklanır.

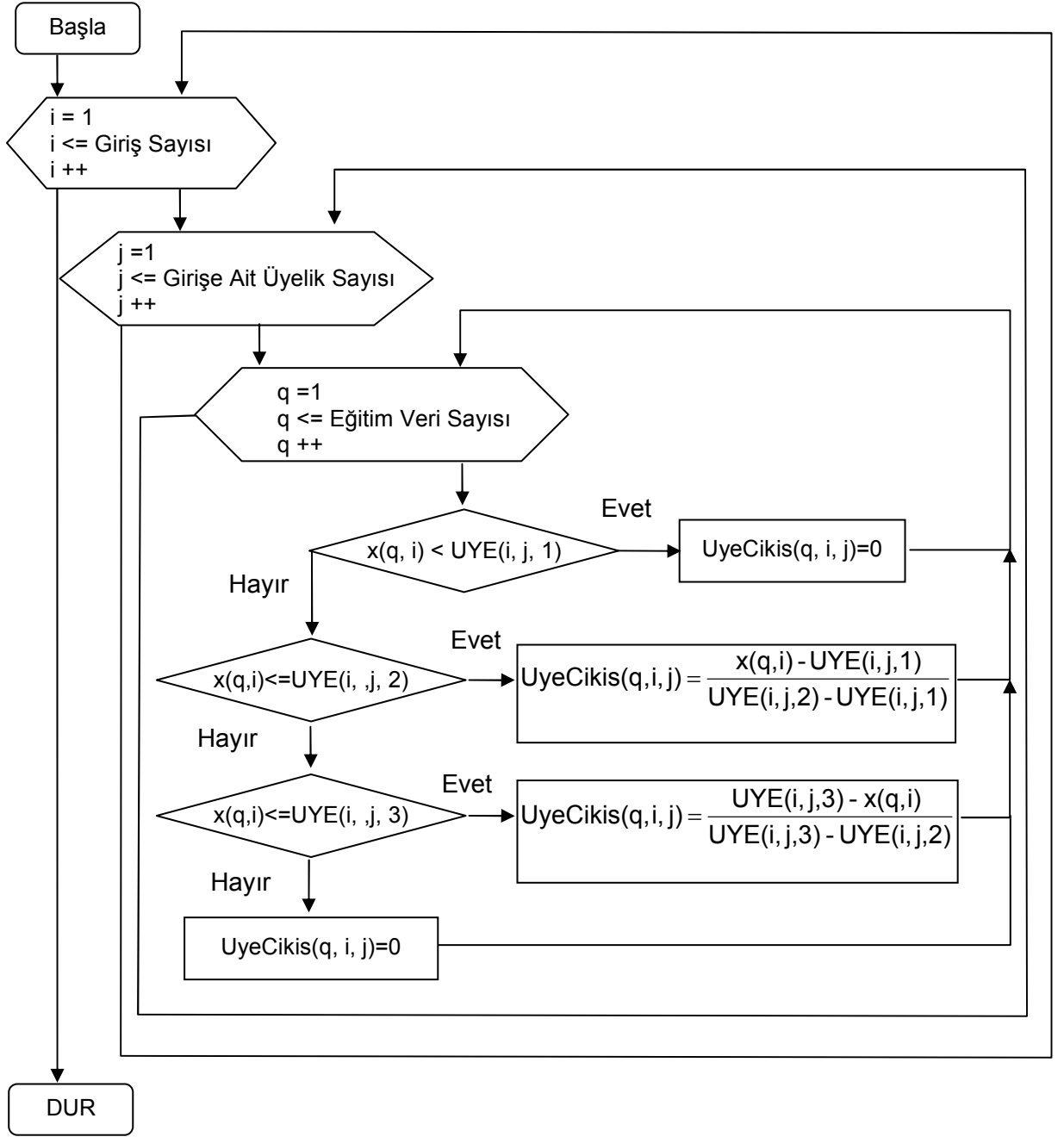
2.3.3.2.1. Her bir eğitim çift için üyelik fonksiyonlarını hesaplanması:

Her bir eğitim veri çiftine ait üyelik fonksiyonlarının çıkışları hesaplanarak üç boyutlu bir değişkende saklanır. Saklanan değişken UyeCikis (q: Eğitim veri no, i: Giriş No, j: Üyelik Fonksiyon No) şeklinde tanımlanmıştır. Üyelik fonksiyonlarının hesaplanma algoritması Şekil 2.26'dadır.

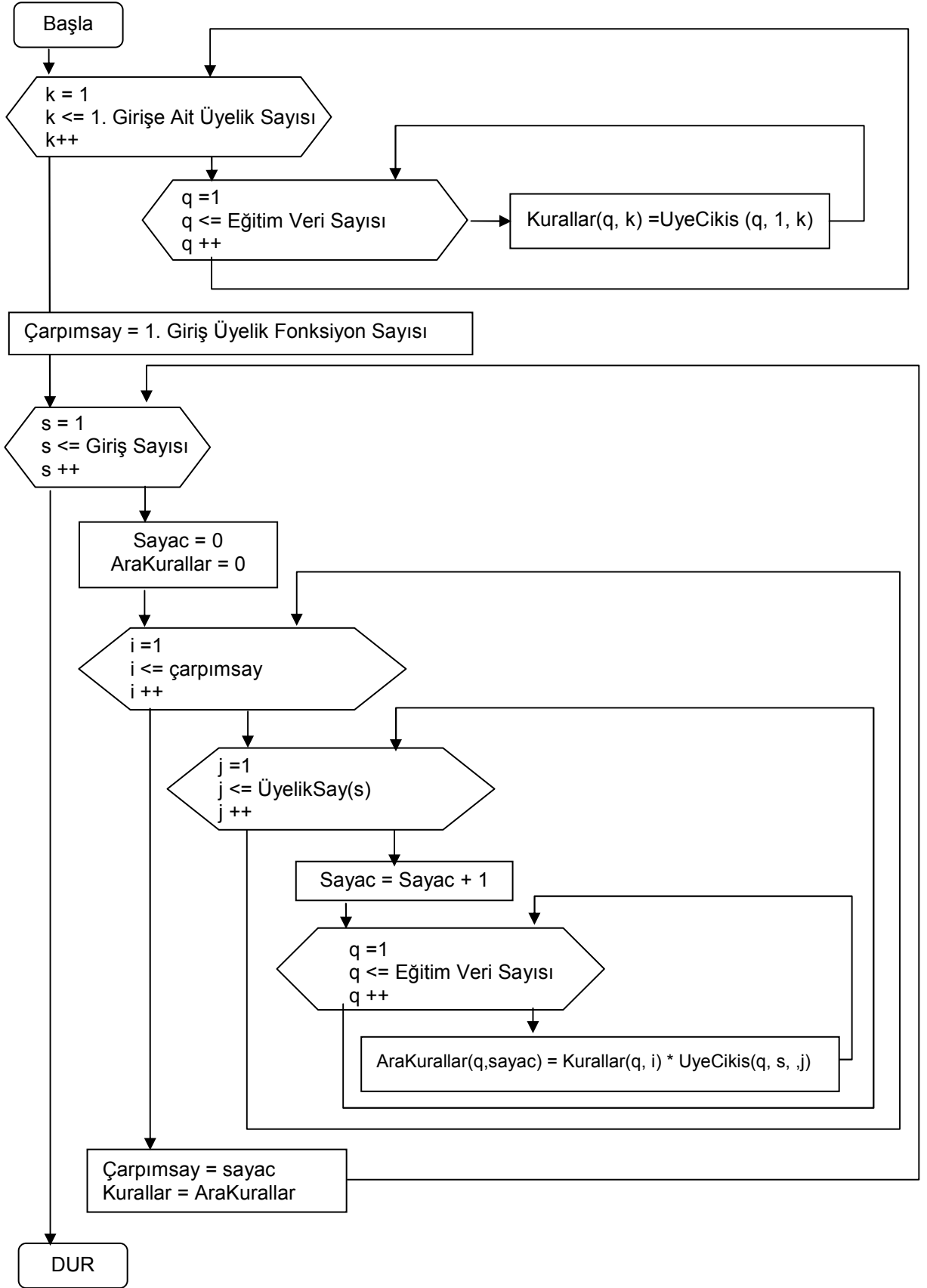
2.3.3.2.2. Her bir eğitim çift için kural çıkışlarının hesaplanması:

Her bir eğitim veri çifti için üyelik fonksiyonları arasında oluşturulan ilişkilere ait çıkış değerleri hesaplanır. Hesaplanan değerler iki boyutlu bir değişkende saklanmış olup değişken "Kurallar (q: Eğitim veri no, i: ilişki no)" şeklinde tanımlanmıştır.

İlk olarak ilk girişe ait üyelik fonksiyon çıkışları kurallar olarak tanımlanır. Daha sonra bu kuralların her bir elamanı bir sonraki girişin üyelik fonksiyonu ile çarpılarak, yeni kurallar oluşturulur. Bu işlem bütün girişler için devam ettirilir. Bu işlemlerin yapılması için kullanılan algoritma Şekil 2.27' da verilmiştir.



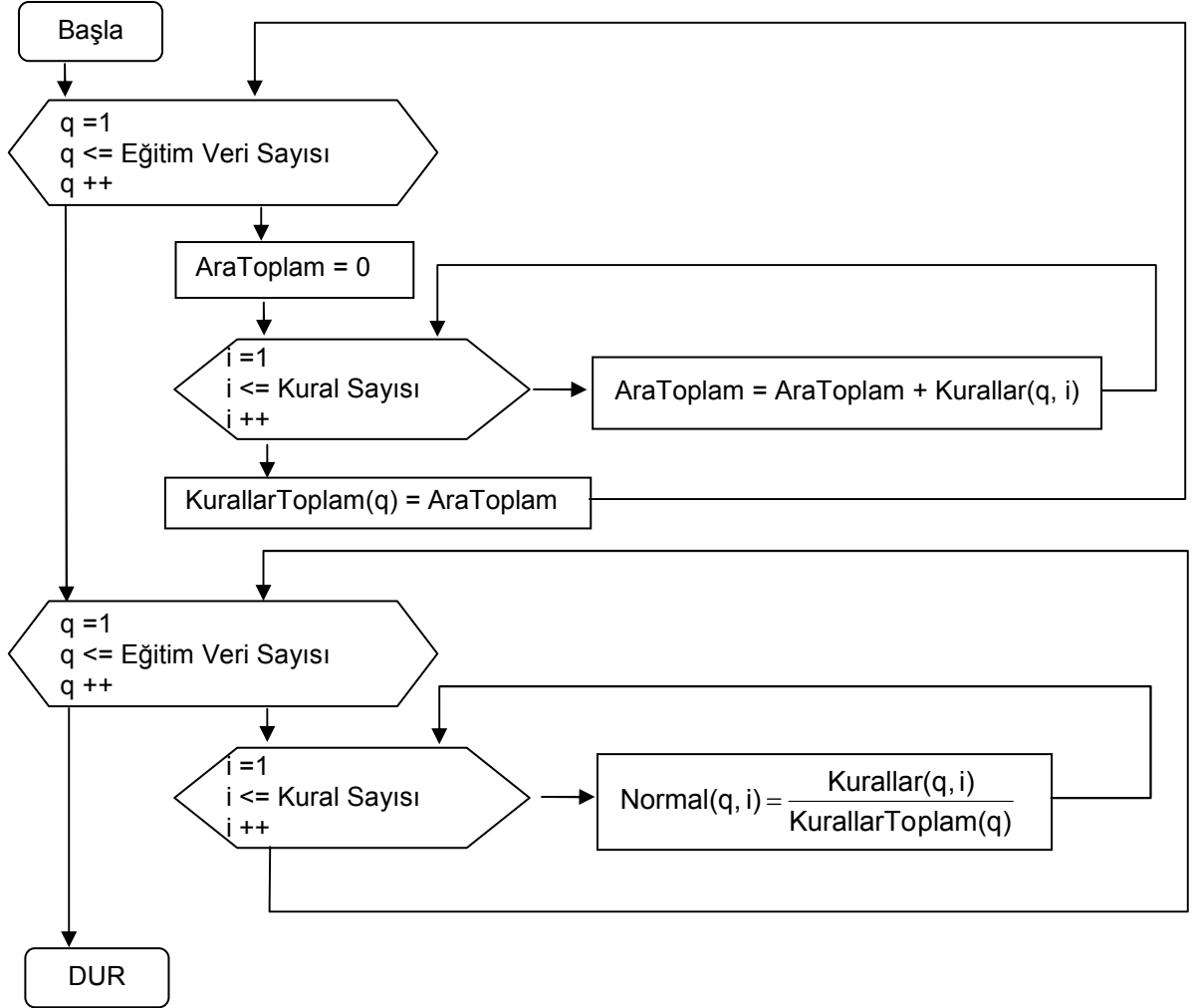
Şekil 2.26. 1.Katman Bütün Eğitim Verileri için Üyelik Fonksiyon Hesap Algoritması



Şekil 2.27. 2.Katman Bütün Eğitim Verileri için Kural Çıktılarının Hesap Algoritması

2.3.3.2.3. Her bir eğitim çift için Kuralların Normalleştirilmesi:

Her bir eğitim verisi çift için o çifte ait kuralların toplamları bulunur. Kural çıkışları ait oldukları eğitim verisi çifti grubundaki kural toplamlarına bölünerek normalleştirme işlemi gerçekleştirilir. Değerler iki boyutlu "Norma(q: Eğitim veri no, i : İlişki no)" değişkeninde saklanır. Bu işlemlerin yapıldığı algoritma Şekil 2.28'da verilmiştir.

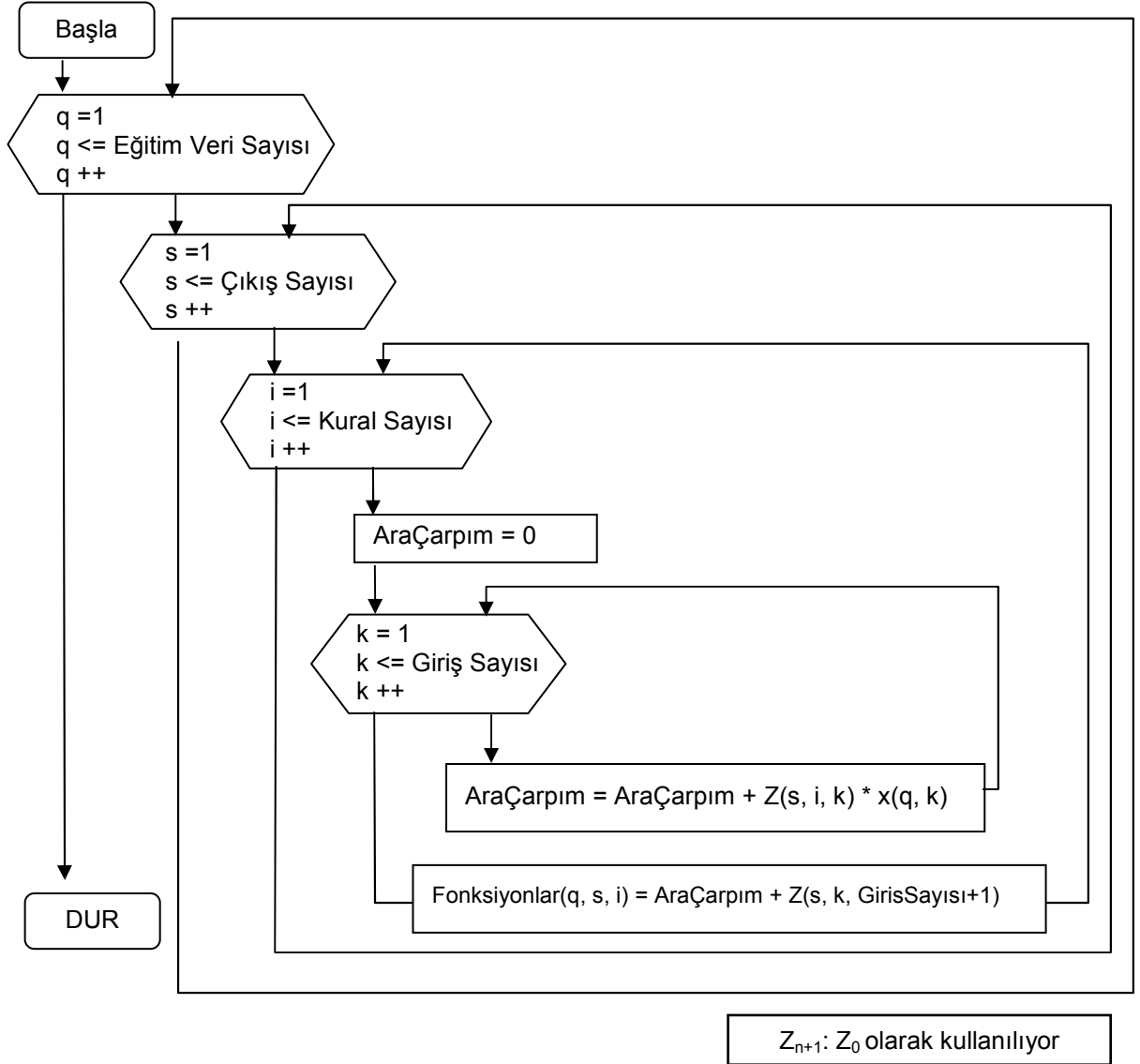


Şekil 2.28. 3.Katman Normalleştirme İşlemi Algoritması

2.3.3.2.4. Her bir eğitim çift için Çıkış Fonksiyonlarının hesaplanması:

Her bir eğitim verisi çift için çıkış fonksiyonları hesaplanarak değerler üç boyutlu "Fonksiyonlar (q: Eğitim Çift No, s : Çıkış No, i: İlişki No)" değişkeninde saklanmıştır. Bu işlemlerin yapıldığı algoritma Şekil 2.29'dadır. m tane kurala sahip, n adet girişe ve s adet çıkışsa sahip ANFIS çıkış fonksiyonu Denklem 2.73'dedir.

$$fs_j(z,x) = \begin{bmatrix} fs_1(z,x) \\ fs_2(z,x) \\ \vdots \\ fs_m(z,x) \end{bmatrix} = \begin{bmatrix} Zs1_0 & Zs1_1 & Zs1_2 & \dots & Zs1_n \\ Zs2_0 & Zs2_1 & Zs2_2 & \dots & Zs2_n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Zsm_0 & Zsm_1 & Zsm_2 & \dots & Zsm_n \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.73)$$



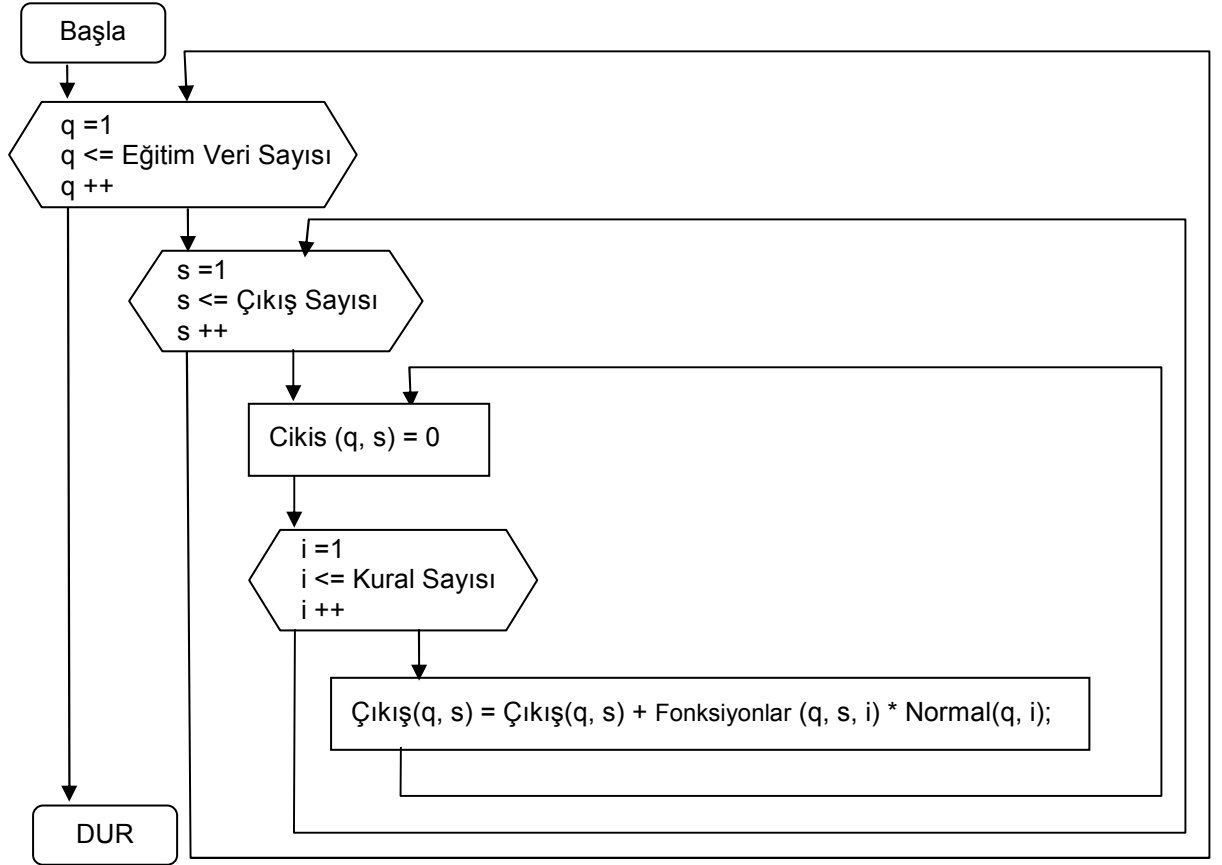
Şekil 2.29. 4.Katman Bütün Eğitim Verileri için Çıkış Fonksiyonlarının Hesap Algoritması

2.3.3.2.5. Her bir eğitim çift için Çıkışların Hesaplanması:

En son adım olarak bütün eğitim veri çiftleri için değer hesaplanarak iki boyutlu

Cikis : Eğitim Veri No, s : Çikişi No) = $\sum_{i=1}^n Normal_i(q)$. Fonksiyon $_i(q)$

değişkeninde saklanmıştır. Çıkışların hesaplanması algoritması Şekil 2.30'dadır.



Şekil 2.30. 5.Katman Bütün Eğitim Verileri için Çıkış Hesap Algoritması

2.3.3.3. Hatanın Hesaplanması ve ANFIS Katsayılarının Eğitilmesi:

2.3.3.3.1. Hatanın Hesaplanması:

Eğitim verisi çıkışları $\text{Çıkış}_o(q)$: Eğitim veri no, s: Çıkış no) : $Y_{so}(q)$

Hesaplanan çıkışlar $\text{Çıkış}_n(q)$: Eğitim veri no, s: Çıkış no) : $Y_s^*(q)$ olmak üzere hata;

$$\text{Hata}_s : E_s = \sum_{q=1}^N E_s^q \quad E_s^q = \frac{1}{2} (Y_s^*(q) - Y_{so}(q))^2 \quad (s: \text{Çıkış No}) \quad (2.74)$$

olarak hesaplanır. Her bir çıkış için ayrı ayrı hata hesaplanmakta ve geri yayılım yöntemi ile katsayıların hesaplanması her bir çıkış için ayrı ayrı yapılır. Her bir çıkış için çıkış fonksiyonları ayrı, üyelik fonksiyonları ise ortaktır.

$$Y_s^*(q) = N(q,1).F(q,s,1) + N(q,2).F(q,s,2) + \dots + N(q,n).F(q,s,n) \quad (2.75)$$

2.3.3.3.2. Çıkış Fonksiyon Katsayılarının Hesaplanması:

Yapay sinir ağları yapısında kullanılan delta kuralı burada da kullanılır.

Çıkış fonksiyonları katsayısı Z (s:Çıkış No, i: Kural No, k:Giriş No) değişkenini denklemlerde Z_{sik} olarak kullanılmıştır.

$$Zs_{i_k}(t+1) = Zs_{i_k}(t) + \Delta Zs_{i_k}(t) \quad (2.76)$$

$$\Delta Zs_{i_k}(t) = \sum_{q=1}^N \Delta^q Zs_{i_k}(t) \quad (2.77)$$

Eğitim katsayısı η olarak tanımlanmıştır.

$$\Delta^q Zs_{i_k}(t) = -\eta \cdot \frac{\partial E_s^q}{\partial Zs_{i_k}(t)} \quad (2.78)$$

$$\Delta^q Zs_{i_k}(t) = -\eta \cdot \frac{\partial E_s^q}{\partial Zs_{i_k}(t)} = -\eta \cdot \frac{\partial E_s^q}{\partial Y_s^{*q}} \cdot \frac{\partial Y_s^{*q}}{\partial Zs_{i_k}(t)} \quad (2.79)$$

Denklem 2.74 hata denkleminin $Y_s^*(q)$ 'ye göre türevi alınırsa;

$$\frac{\partial E_s^q}{\partial Y_s^{*q}} = Y_s^*(q) - Y_{so}(q) \quad (2.80)$$

n kural sayısı olmak üzere hesaplanan çıkış değerlerinin, çıkış fonksiyon katsayılarına göre türevi denklem 2.81'de olduğu hesaplanır.

$$\frac{\partial Y_s^{*q}}{\partial Zs_{i_k}(t)} = \frac{\sum_{i=1}^n N_i^q \cdot Fs_i^q}{\partial Zs_{i_k}(t)} = \frac{N_1^q \cdot Fs_1^q + N_2^q \cdot Fs_2^q + \dots + N_i^q \cdot Fs_i^q + \dots + N_n^q \cdot Fs_n^q}{\partial Zs_{i_k}(t)} \quad (2.81)$$

Denklem 2.81'de $(N_i^q \cdot Fs_i^q)$ terimlerinden sadece Zs_{i_k} katsayısını içeren terimin türevi vardır, diğer terimlerin türevleri sıfır olacaktır.

$$\frac{\partial Y_s^{*q}}{\partial Zs_{i_k}(t)} = \frac{N_i^q \cdot Fs_i^q}{\partial Zs_{i_k}(t)} \quad (2.82)$$

$$\frac{N_i^q \cdot Fs_i^q}{\partial Zs_{i_k}(t)} = \frac{N_i^q (Zs_{i0} + Zs_{i1} \cdot x_1 + Zs_{i2} \cdot x_2 + \dots + Zs_{ik} \cdot x_k + \dots + Zs_{in} \cdot x_n)}{\partial Zs_{i_k}(t)} \quad (2.83)$$

$k = 1, 2, \dots, n$ için türev;

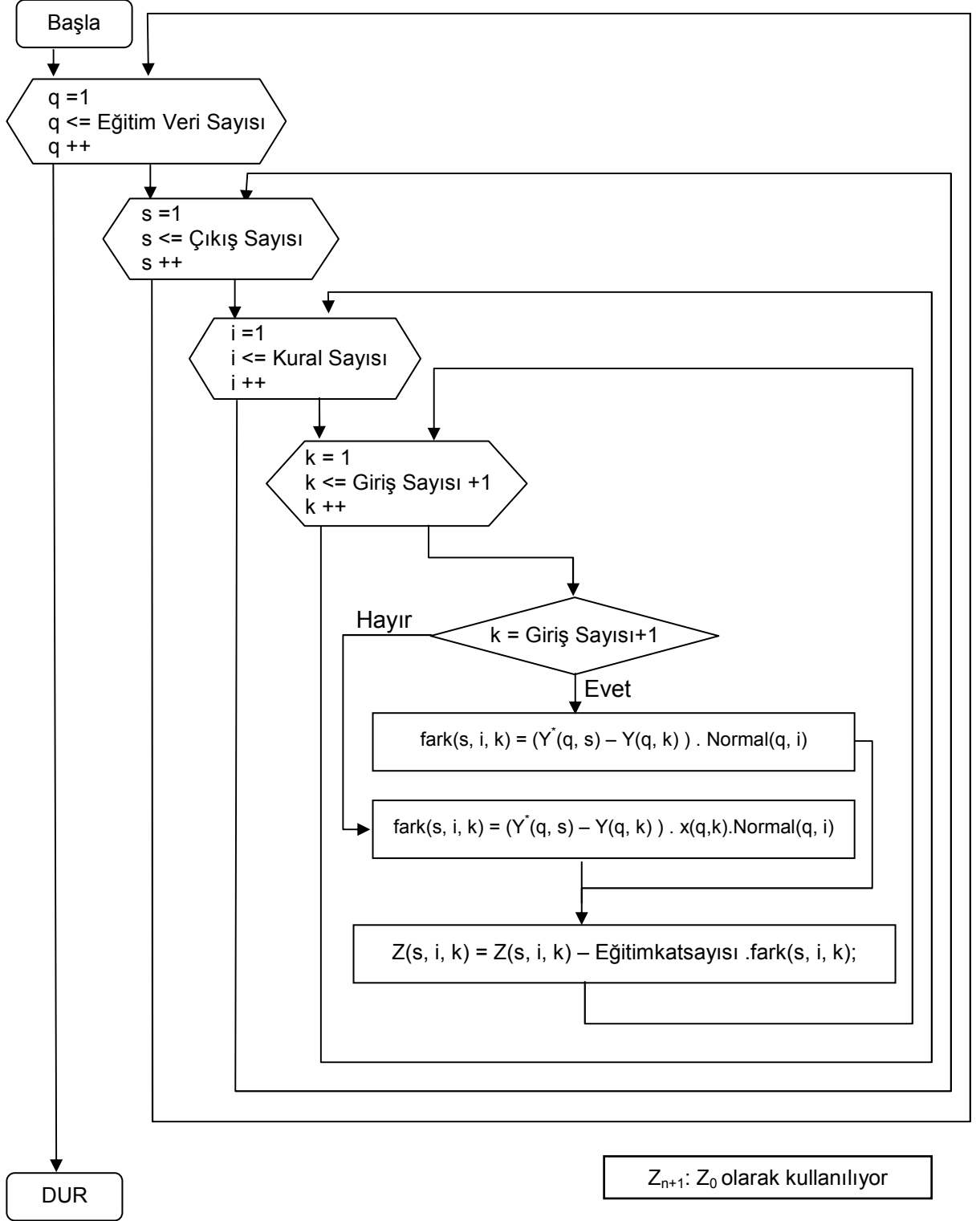
$$\frac{\partial Y_s^{*q}}{\partial Zs_{i_k}(t)} = N_i^q \cdot x_k \quad (2.84)$$

$$\Delta^q Zs_{i_k}(t) = -\eta \cdot (Y_s^*(q) - Y_{so}(q)) \cdot (N_i^q \cdot x_k) \quad (2.85)$$

$k = 0$ için türev;

$$\frac{\partial Y_s^{*q}}{\partial Zs_{i_k}(t)} = N_i^q \quad (2.86)$$

$$\Delta^q Zs_{i_k}(t) = -\eta \cdot (Y_s^*(q) - Y_{so}(q)) \cdot (N_i^q) \quad (2.87)$$



Şekil 2.31. Çıkış Fonksiyon Katsayılarının Hesaplanması Algoritması

2.3.3.3.3. Üyelik Fonksiyonları Katsayılarının Hesaplanması:

Aynı şekilde delta kuralı üyelik fonksiyonları katsayılarının hesaplanmasında da kullanılır.

Şekil 2.24'deki gibi tanımlanan üyelik fonksiyonları katsayıları olarak program akışı içerisinde kullanılan UYE₁, UYE₂ ve UYE₃ değişkenleri için denklemlerde yazım kolaylığı açısından a, b ve c tanımlamaları kullanılacaktır.

a_{kj}: gösteriminde k: Giriş numarasını, j: Üyelik fonksiyon numarasını gösterir.

$$UYE1_{kj}(t+1) = UYE1_{kj}(t) + \Delta UYE1_{kj}(t) \quad (2.88)$$

$$a_{kj}(t+1) = a_{kj}(t) + \Delta a_{kj}(t) \quad (2.89)$$

N deneme sayısı olmak üzere;

$$\Delta a_{kj}(t) = \sum_{q=1}^N \Delta^q a_{kj}(t) \quad (2.90)$$

P çıkış sayısını belirtmek üzere;

$$\Delta^q a_{kj}(t) = -\eta \cdot \sum_{s=1}^P \frac{\partial E_s^q}{\partial a_{kj}(t)} \quad (2.91)$$

$$\Delta^q a_{kj}(t) = -\eta \cdot \sum_{s=1}^P \left(\frac{\partial E_s^q}{\partial Y_s^{*q}} \cdot \frac{\partial Y_s^{*q}}{\partial a_{kj}} \right) \quad (2.92)$$

$$\sum_{s=1}^P \frac{\partial E_s^q}{\partial a_{kj}} = \left(Y_1^{*q} - Y_{1o}^q \right) \cdot \frac{\partial Y_1^{*q}}{\partial a_{kj}} + \left(Y_2^{*q} - Y_{2o}^q \right) \cdot \frac{\partial Y_2^{*q}}{\partial a_{kj}} + \dots + \left(Y_P^{*q} - Y_{Po}^q \right) \cdot \frac{\partial Y_P^{*q}}{\partial a_{kj}} \quad (2.93)$$

Çıkış denklemini T kural sayısı olmak üzere;

$$Y_s^{*q} = \frac{K_1^q \cdot F_{s1}^q + K_2^q \cdot F_{s2}^q + \dots + K_T^q \cdot F_{sT}^q}{KurallarToplam^q = K_1^q + K_2^q + \dots + K_T^q} \quad (2.94)$$

$$\frac{\partial Y_s^{*q}}{\partial a_{kj}} = \sum_{n=1}^T \left(\frac{(K_n^q)' \cdot F_{sn}^q}{KurallarToplam^q} - \frac{K_n^q \cdot F_{sn}^q \cdot (KurallarToplam^q)'}{[KurallarToplam^q]^2} \right) \quad (2.95)$$

$$\sum_{n=1}^T \left(\frac{(K_n^q)' \cdot F_{sn}^q}{KurallarToplam^q} \right) \quad (2.96)$$

Denklem 2.96 hesaplamasında a_{kj} teriminin ait olduğu üyelik fonksiyonlarını barındıran kuralların türevleri olacak, diğer kuralların türevleri sıfır olacaktır.

$$\sum_{n=1}^T \left(-\frac{K_n^q \cdot F_{sn}^q \cdot (KurallarToplam^q)'}{[KurallarToplam^q]^2} \right) \quad (2.97)$$

Denklem 2.97 hesaplamasında kuralların toplamının türevi için sadece a_{kj} teriminin ait olduğu üyelik fonksiyonlarını barındıran kurallar için türev mevcut olup diğer terimlerin türevleri sıfır olacaktır.

Bir üyelik fonksiyonunun ait olduğu kuralların bulunması için öncelikle kuralların nasıl oluşturulduğu gözden geçirilmelidir. Bunun için 3 giriş ve ilk iki girişi ikişer üyelik fonksiyonlu 3. girişi 3 üyelik fonksiyonlu ANFIS yapısı göz önüne alınsın.

Üyelik Fonksiyonları AK_j (K: Giriş No, j: Üyelik Fonksiyon No)

Kural1	$A_{1_1} \cdot A_{2_1} \cdot A_{3_1}$
Kural2	$A_{1_1} \cdot A_{2_1} \cdot A_{3_2}$
Kural3	$A_{1_1} \cdot A_{2_1} \cdot A_{3_3}$
Kural4	$A_{1_1} \cdot A_{2_2} \cdot A_{3_1}$
Kural5	$A_{1_1} \cdot A_{2_2} \cdot A_{3_2}$
Kural6	$A_{1_1} \cdot A_{2_2} \cdot A_{3_3}$
Kural7	$A_{1_2} \cdot A_{2_1} \cdot A_{3_1}$
Kural8	$A_{1_2} \cdot A_{2_1} \cdot A_{3_2}$
Kural9	$A_{1_2} \cdot A_{2_1} \cdot A_{3_3}$
Kural10	$A_{1_2} \cdot A_{2_2} \cdot A_{3_1}$
Kural11	$A_{1_2} \cdot A_{2_2} \cdot A_{3_2}$
Kural12	$A_{1_2} \cdot A_{2_2} \cdot A_{3_3}$

Çizelge 2.1. 3 Girişli İkişer Üyelik Fonksiyonlu ANFIS'in Kural Çizelgesi

Çizelge 2.1'de görüldüğü gibi ilk giriş ait üyelik fonksiyonları birer adet 6'lı grup şeklinde toplanmıştır. İkinci girişe ait üyelik fonksiyonları ikişer adet 3'lü grup şeklinde toplanmış ve en son girişe ait üyelik fonksiyonları dörder adet tekli gruplar şeklinde dağılmıştır.

Bir üyelik fonksiyonunun hangi kurallar ile ilişkisinin olduğunu bulmak için

1. Üyelik fonksiyonunu içeren ne kadar kural olduğunun bilinmesi gerekir. Bu değer toplam kural sayısının o girişe ait üyelik fonksiyon sayısına bölünmesi ile elde edilir.

2. Üyelik fonksiyonun dağılım gruplarındaki miktarının bilinmesi gerekir. Bu değer toplam kural sayısının o girişe ait ve önceki girişlere ait üyelik fonksiyonları sayılarına bölünmesi ile bulunur.

3. Üyelik fonksiyonu içeren kaç adet grup olduğunun bilinmesi gerekir.

Bu değer P, 1. değere M ve 2. değere N denirse $P = M / N$ şeklinde bulunur.

k. girişe ait i. üyelik fonksiyonu için kaç grup ve gruplarda kaçar eleman olduğu hesaplandıktan sonra kural numarasını bulmak için yapılacak işlem grupların belirlenmesi için bir döngü, bu döngü içerisinde de grup elemanlarını bulmak için ikinci bir döngü kullanılması gerekir.

Grubu sırasını tanımlayan değişken “p”, ve grup içerisindeki yeri tanımlayan değişken “g” ve k. girişe ait üyelik fonksiyon sayısı Y (ÜyelikSayısı (k:GirişNo) dizi değişkeninden elde edilir.) olmak üzere kural numarası denklem 2.98’de olduğu gibi hesaplanır.

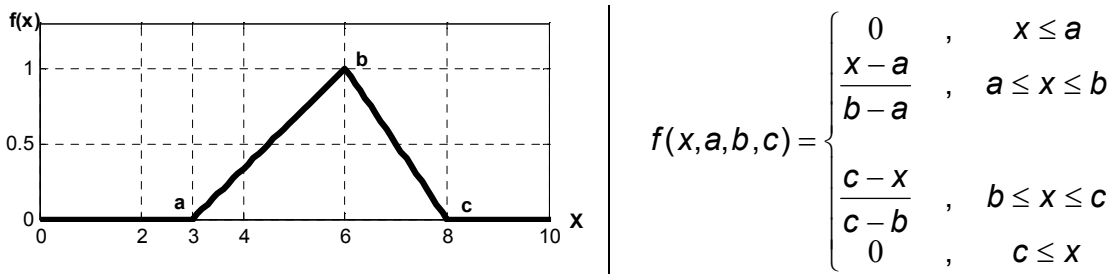
$$\text{Kural No} = (p - 1).N.Y + g \quad (2.98)$$

Kural numarasının elde edilmesi için kullanılan döngü içerisinde denklem 2.96 ve denklem 2.97 için gerekli türev hesaplamaları yapılır. Hesaplama döngüsü içerisinde Denklem 2.96’daki n için a_{ki} terimini içeren kural numarası kullanılır. K_n^q türevi hesaplanırken, a_{ki} teriminin ait olduğu üyelik fonksiyonu H olarak tanımlanır ve H üyelik fonksiyonu ile birlikte K_n^q kuralını oluşturan diğer üyelik fonksiyonlarını $D_n = K_n^q / H$ şeklinde bulunur. Buradan da K_n^q ’nin türevi denklem 2.99’da olduğu şekilde yazılabilir.

$$(K_n^q)' = \frac{\partial K_n^q}{\partial a_{kj}} = D_n \cdot \frac{\partial H}{\partial a_{kj}} \quad (2.99)$$

Buradan da a_{ki} terimini içeren kural numaraları n^* olarak tanımlarsak denklem 2.96, denklem 2.100’deki gibi yazılabilir.

$$\sum_{n=1}^T \left(\frac{(K_n^q)' \cdot F_{sn}^q}{\text{KurallarToplam}^q} \right) : \frac{\partial H}{\partial a_{kj}} \cdot \sum_{n=1}^{T^*} \left(\frac{D_n \cdot F_{sn}^q}{\text{KurallarToplam}^q} \right) \quad (2.100)$$



Şekil 2.32 Üçgen Fonksiyonu

Üçgen fonksiyonunun a, b ve c’ye göre türevleri denklem 2.101, 2.102 ve 2.103’de verildiği gibidir.

$$\frac{\partial f(x, a, b, c)}{\partial a} = \begin{cases} 0 & , \quad x \leq a \\ \frac{x-b}{(b-a)^2} & , \quad a \leq x \leq b \\ 0 & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases} \quad (2.101)$$

$$\frac{\partial f(x, a, b, c)}{\partial b} = \begin{cases} 0 & , \quad x \leq a \\ \frac{a-x}{(b-a)^2} & , \quad a \leq x \leq b \\ \frac{c-x}{(c-b)^2} & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases} \quad (2.102)$$

$$\frac{\partial f(x, a, b, c)}{\partial c} = \begin{cases} 0 & , \quad x \leq a \\ 0 & , \quad a \leq x \leq b \\ \frac{x-b}{(c-b)^2} & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases} \quad (2.103)$$

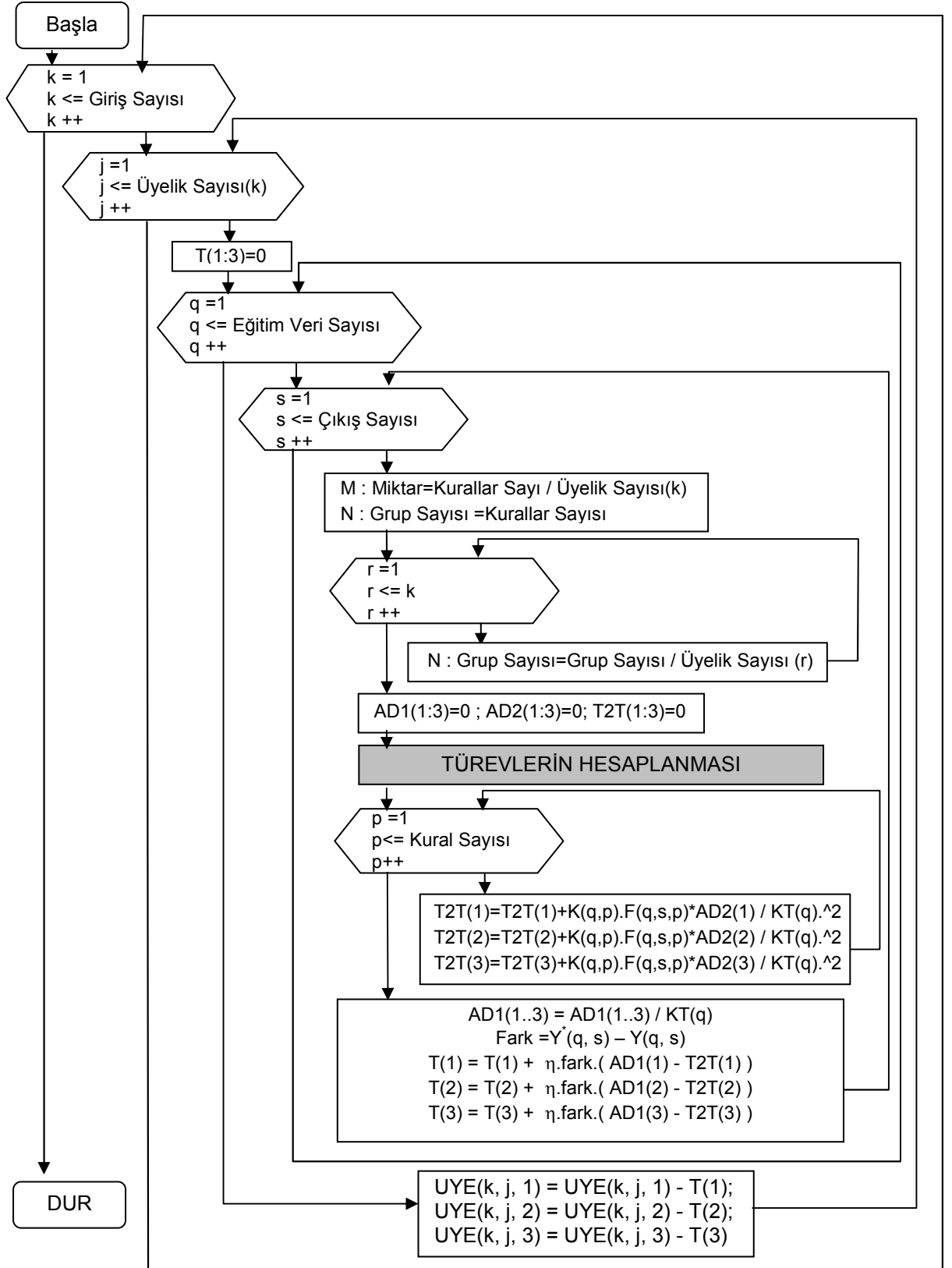
Denklem 2.97 olarak verilen terimdeki KuralToplam^q değeri denklem 2.104'teki şekilde hesaplanır.

$$(KurallarToplam^q)' = \frac{\partial}{\partial a_{kj}} \sum_{n=1}^T (K_n^q) \quad (2.104)$$

Yukarıda yapılan tanımlama da olduğu gibi a_{ki} teriminin ait olduğu üyelik fonksiyonu H olarak tanımlanırsa, denklem 2.104, denklem 2.105 şeklinde yazılabilir. Denklem 2.105'de n*, H terimini içeren kural numaralarını gösterir. H terimini içermeyen kuralların türevleri sıfır olacaktır.

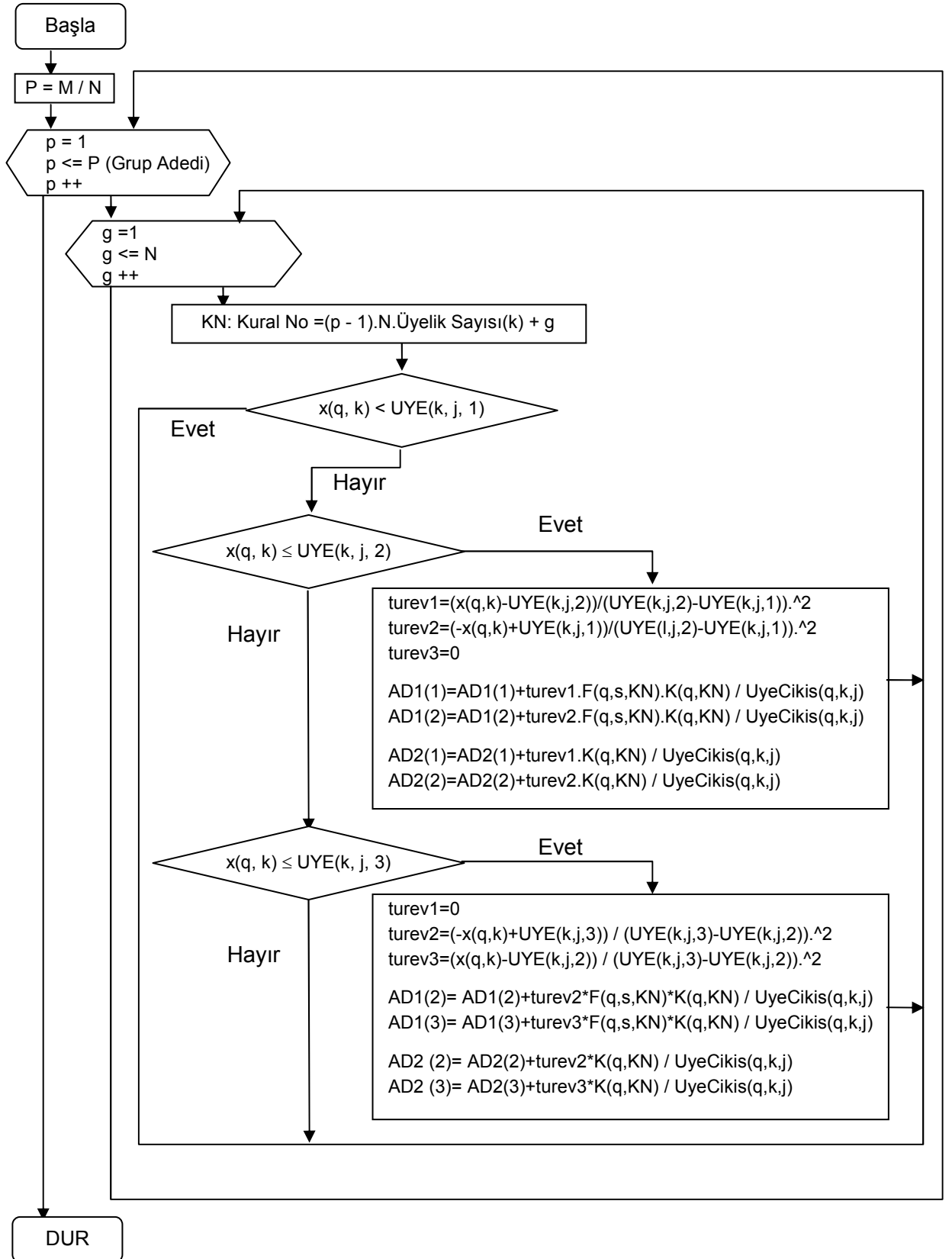
$$\frac{\partial}{\partial a_{kj}} \sum_{n=1}^T (K_n^q) = \sum_{n=1}^T \frac{\partial K_n^q}{\partial a_{kj}} : \sum_{n^*=1}^T \left(D_n \cdot \frac{\partial H}{\partial a_{kj}} \right) = \frac{\partial H}{\partial a_{kj}} \cdot \sum_{n=1}^N D_n \quad (2.105)$$

Şekil 2.33'de üyelik fonksiyon katsayılarının hesaplanması algoritmasında; miktar, üyelik fonksiyonunu içeren kural sayısı için kullanılan değişken olup fonksiyonlar için "F", Kurallar için "K" ve Kurallar Toplamı için "KT" ifadeleri kullanılmıştır. Arada yapılan işlemler için "AD1" ve "AD2" ifadeleri kullanılmıştır. "AD1" denklem 2.96 olarak verilen terimin sonucunu, "AD2" denklem 2.97 olarak verilen terimdeki türev sonucunu saklayan değişkenlerdir.



Şekil 2.33. Üyelik Fonksiyonları Katsayılarının Hesaplanması Algoritması

Türev Hesapla



Şekil 2.34. Üyelik Fonksiyon Katsayısı Alt Algoritması (Türev Hesapla)

2.3.4. Çok Girişli ve Çok Çıkışlı ANFIS Yapısının Hybrid (Melez) Metodu ile Katsayılarının Hesaplanması:

Hybrid algoritması çıkış fonksiyonlarının katsayılarının eğitilmesi hariç geri yayılım metodu ile aynıdır. ANFIS yapısında iki farklı katsayı grubu eğitilir. Birisi giriş üyelik fonksiyonlarının katsayıları grubu, birisi de çıkış fonksiyonları katsayıları grubudur. Hybrid metodunda bu iki grup iki ayrı yöntem ile eğitilmekte oldukları için bu ad ile anılır.

Bu model için; sadece çıkış fonksiyonlarının nasıl hesaplandığı anlatılacaktır.

2.3.4.1. Hybrid Modeli Çıkış Fonksiyon Katsayılarının Hesaplanması:

$$\begin{aligned}
 Y_1^*(1) &= N_1(1).F1_1(1) + N_2(1).F1_2(1) + \dots + N_n(1).F1_n(1) \\
 Y_1^*(2) &= N_1(2).F1_1(2) + N_2(2).F1_2(2) + \dots + N_n(2).F1_n(2) \\
 &\vdots \\
 Y_1^*(q) &= N_1(q).F1_1(q) + N_2(q).F1_2(q) + \dots + N_n(q).F1_n(q) \\
 &\vdots \\
 Y_1^*(Q) &= N_1(Q).F1_1(Q) + N_2(Q).F1_2(Q) + \dots + N_n(Q).F1_n(Q)
 \end{aligned} \tag{2.106}$$

Denklem 2.106'da gösterilen eğitim verileri ile hesaplanan çıkış fonksiyonu

“Eğitim çifti çıkışı=(Bilinen Sistem değerleri Matrisi) X (Çıkış Fonksiyon Katsayıları)”

Şeklindeki matrissel gösterimi haline dönüştürülecektir.

$$Y_o = \begin{bmatrix} Y_o^*(1) \\ Y_o^*(2) \\ \vdots \\ Y_o^*(q) \\ \vdots \\ Y_o^*(Q) \end{bmatrix}$$

$$A = \begin{bmatrix} \boxed{N_1(1) \quad N_1(1).x_1(1) \quad \dots \quad N_1(1).x_k(1)} & \dots & \dots & \boxed{N_i \quad N_i(1).x_1(1) \quad \dots \quad N_i(1).x_k(1)} \\ N_1(2) \quad N_1(2).x_1(2) \quad \dots \quad N_1(2).x_k(2) & \dots & \dots & N_i \quad N_i(2).x_1(2) \quad \dots \quad N_i(2).x_k(2) \\ \vdots & \vdots & \vdots & \vdots \\ N_1(q) \quad N_1(q).x_1(q) \quad \dots \quad N_1(q).x_k(q) & \dots & \dots & N_i \quad N_i(q).x_1(q) \quad \dots \quad N_i(q).x_k(q) \\ \vdots & \vdots & \vdots & \vdots \\ N_1(Q) \quad N_1(Q).x_1(Q) \quad \dots \quad N_1(Q).x_k(Q) & \dots & \dots & N_i \quad N_i(Q).x_1(Q) \quad \dots \quad N_i(Q).x_k(Q) \end{bmatrix}$$

1. Kural'a ait değerler
i. Kural'a ait değerler

$$Z = [Z1_0 \quad Z1_1 \quad Z1_2 \quad \dots \quad Z1_k \quad \dots \quad Z2_0 \quad Z2_1 \quad Z2_2 \quad \dots \quad Z2_k \quad \dots \quad Zi_0 \quad Zi_1 \quad Zi_2 \quad \dots \quad Zi_k]^T$$

1. Kural Çıkış Katsayıları
2. Kural Çıkış Katsayıları
i. Kural Çıkış Katsayıları

Eđitim verilerine ait ıkıř deęerleri Y_o matrisi, kurallara ait deęerleri oluřturan matris A matrisi ve ıkıř fonksiyonlarını oluřturan katsayılara ait matris Z matrisi olmak üzere

$$Y_o = A.Z \quad (2.107)$$

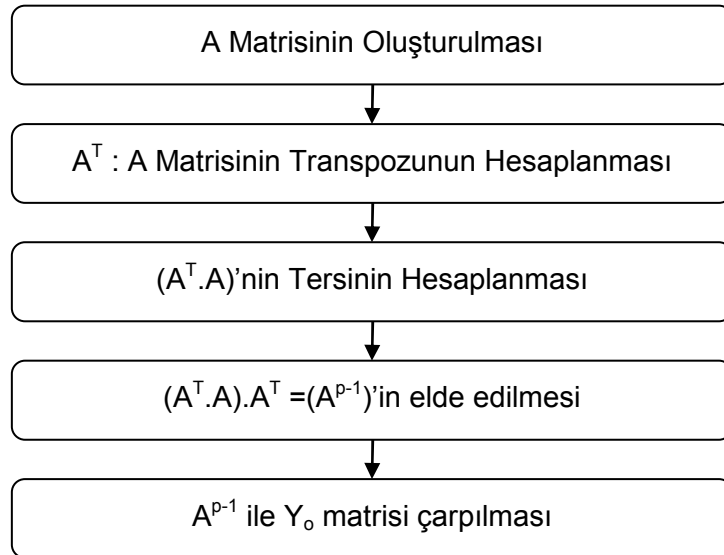
Denklem 2.94'de verilen ANFIS yapısının matrissel gsterimi denklem 2.107'de olduęu gibi elde edilir.

Denklem 2.107'de, eđitim verilesi olarak verilen, sistemin olması istenen ıkıř deęerleri Y_o matrisi ve mevcut üyelik fonksiyon katsayıları ile hesaplanabilen "Normalleřtirilmiř" deęerleri ile giriř deęerlerinin arpımından oluřan A matrisi bilinmektedir. Bilinmeyen ve hesaplanmak istenen Z matrisi, Denklem 2.107'nin zlmesi ile elde edilecektir.

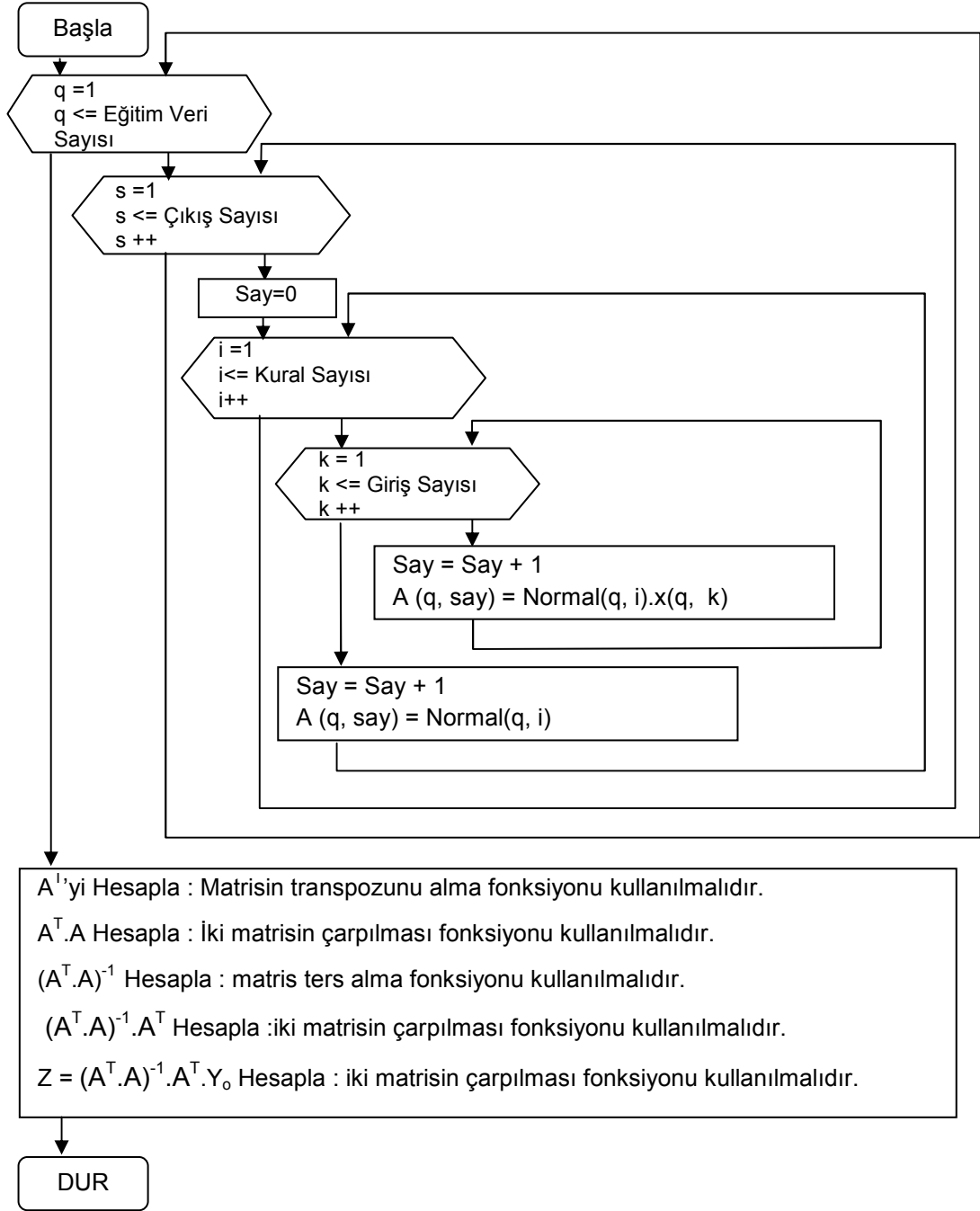
Denklem 2.107 lineer denkleminin zm, denklemin her iki tarafının A matrisinin tersi ile arpılması sonucunda bulunur. Ancak A matrisinin kare matris olmadıęı durumlar iin A matrisinin pseudo tersi ile arpılması gerekir.

$$Z^* = (A^T.A)^{-1}.A^T.Y_o = A^{p-1}.Y_o \quad (2.108)$$

$A^T.A$ matrisinin tekil matris olduęu durumlarda determinantı "Sıfır" olacaęı iin tersi alınamadıęından bu denklem zm mmkn olmayacaktır. Bu durumlar iin farklı yntemler kullanılabilir. Burada sadece denklem 2.108 iin hybrid modeli algoritması ıkarılacaktır.



řekil 2.35. Hyrbrid Modeli Algoritması



Şekil 2.36. Hybrid Metodunun Hesaplanması Algoritması

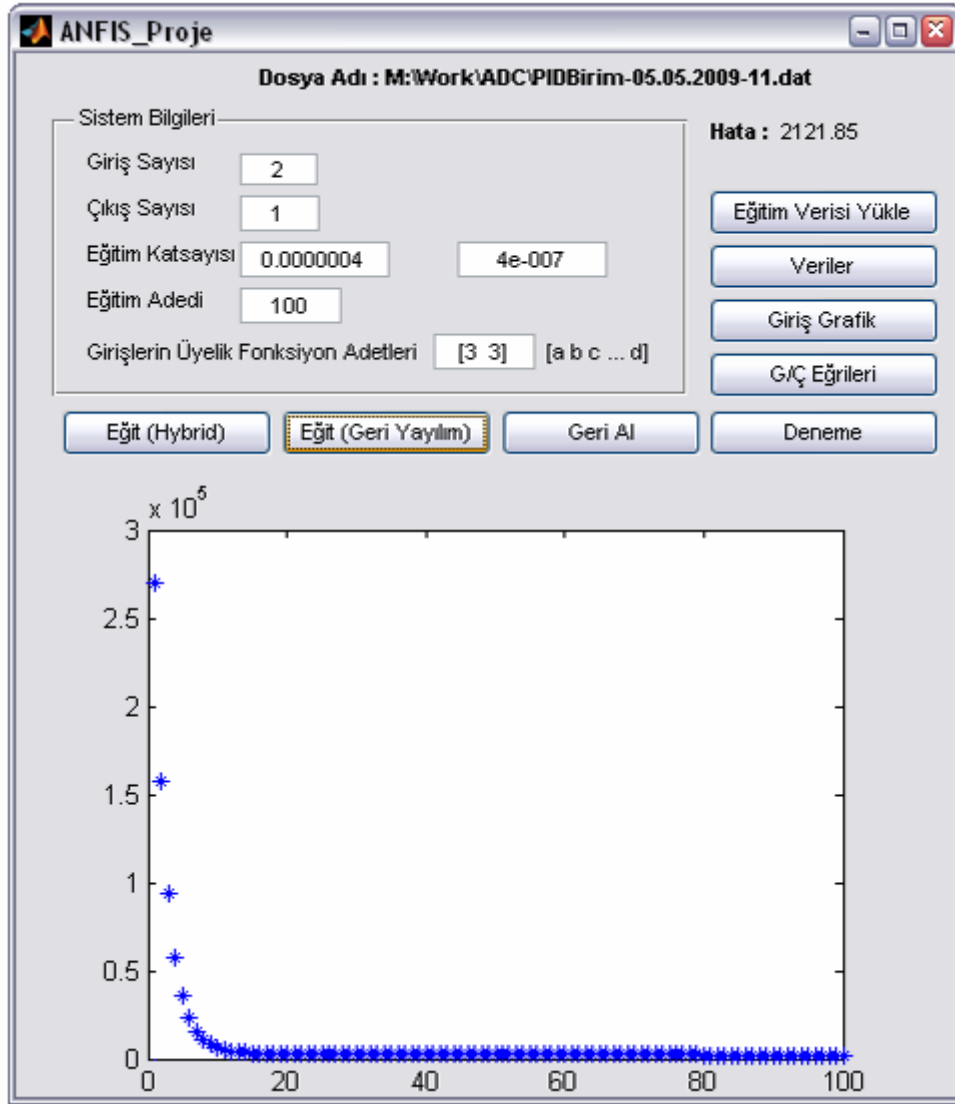
Şekil 2.36'de gösterilen hybrid metodu algoritmasının hesaplanması için bir matrisin transpozunun alınması fonksiyonunun, iki matrisin çarpılması fonksiyonunun ve matrisin tersinin alınması fonksiyonunun ayrı olarak yapılması gerekir. ANFIS programının yapılması için Matlab kullanıldığı için matris işlemleri için Matlab fonksiyonları kullanılmıştır.

Hybrid modeli, çıkış fonksiyonlarının hesaplanması için lineer denklem çözümü kullanıldığı için, geri yayılıma göre daha hızlı çözüme ulaşılır. Bu modeldeki problem

$A^T.A$ matrisinin tekil matris olduğu durumlarda determinantı “Sıfır” olduğundan tersi alınamayacak ve bu model ile çözüm elde edilemeyecektir.

2.3.5. “Matlab GUIDE (GUI Builder)” Yardımı ile Oluşturulan ANFIS Eğitim Programı:

Yukarıda anlatılan algoritmaların birleştirilmesi ile ANFIS sisteminin katsayılarının eğitilmesi için gerekli program elde edilmiştir. Veri giriş işlemlerinin kolaylıkla yapılabilmesi için “Matlab GUIDE (GUI Builder)” programından faydalanılmıştır. GUI Builder ile yapılmış olan eğitim programı Şekil 2.37’dedir.



Şekil 2.37. Matlab GUI Builder ile Yapılan Eğitim Programı

“Eğitim Verisi Yükle” Düğmesi ile eğitim verilerinin bulunduğu dosyadan veriler alınarak giriş çıkış değişkenleri olarak atanabilmekte ve sistemin çıkış adedi girilerek veriler giriş ve çıkış verileri olarak yüklenebilir. Girişlere ait üyelik fonksiyonları için farklı sayıda üyelik fonksiyonu belirlenebilmesini sağlamak amacı ile veri girişi olanağı sağlanmıştır.

“Veriler”, “Giriş Grafikleri” ve “G/Ç Eğrileri” düğmeleri mevcut verilerin kontrol edilmesi için kullanılır.

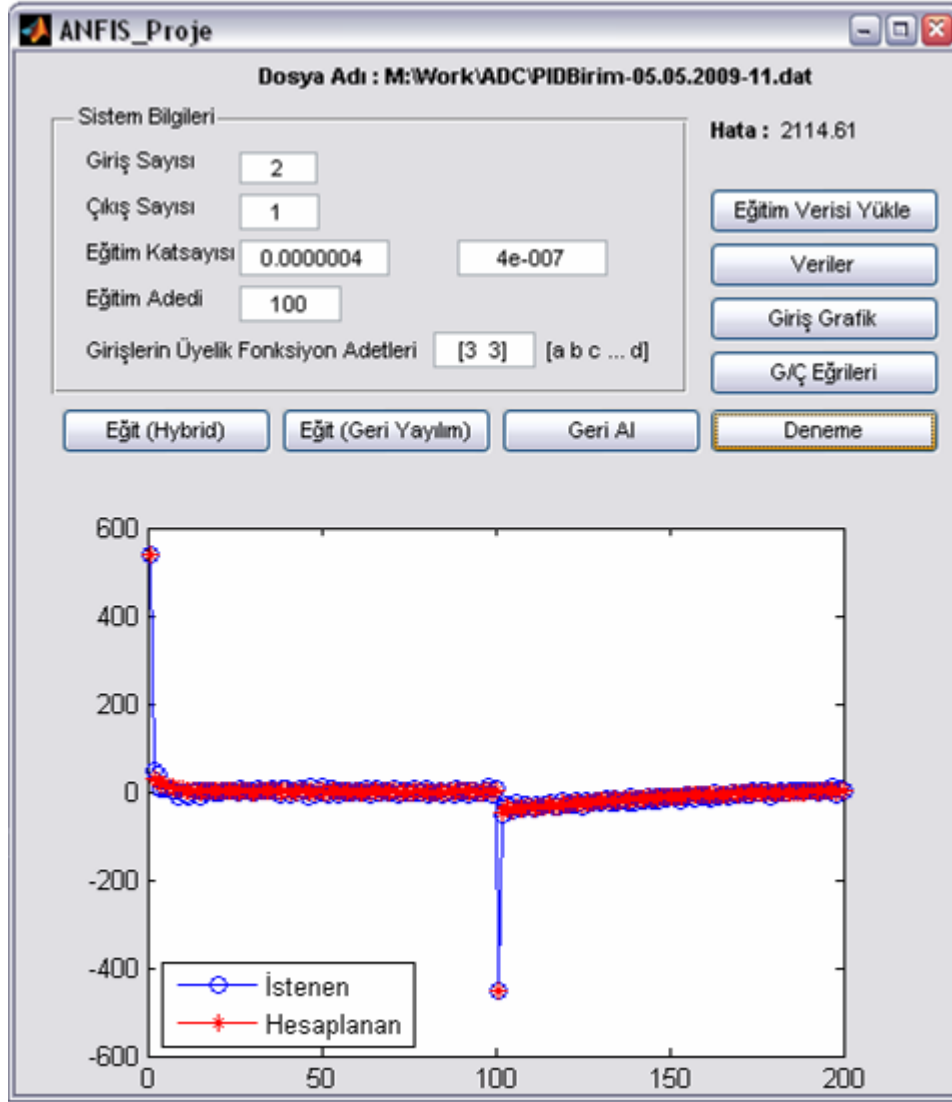
ANFIS için gerekli veri girişleri yapıldıktan sonra Hyrbid ya da Geri Yayılım ile eğitim yöntemleri kullanılabilir. Mevcut eğitim katsayısı ile çözüme ulaşamayabilir. Bu durumda yapılan eğitim “Geri Al” işlem ile iptal edilip, eğitim katsayısı değiştirilerek tekrar eğitim işlemi yapılabilir. Bu işlem uygun eğitim katsayısı bulunana kadar yapılabilir. Eğitim katsayısı değişimi için her seferinde eğitim katsayısı 10^{-1} ile çarpılarak değiştirilmesi kısa sürede uygun eğitim katsayısının elde edilmesini sağlayacaktır.

Hyrbid Modeli lineer denklem çözümü yaptığı için kısa sürede çözüme ulaşacaktır. Bunun için eğitim adedi fazla olmasına gerek yoktur. 1 ile 5 arasında bir sayı seçilebilir. 3 uygun bir sayı olacaktır.

Geri Yayılım yöntemi için eğitim adedi büyük sayılar seçilebilir. Hatanın değişim oranına ve tahmini yeterli çözüme ulaşma zamanına göre eğitim adedi belirlenebilir. Eğitim adedinin 50 ya da 100 olarak belirlenmesi eğitim sürecinin nasıl ilerleyebileceği konusunda yardımcı olabilir. Geri Yayılım kullanılarak yapılan eğitimler arasında daha kısa sürede eğitimi tamamlayabilmek için eğitim katsayısı değiştirebilir.

Eğitim sürecinde eğer hata bir önceki hatadan büyük ise eğitim süreci durdurulur. Bu durumda en son yapılan eğitim “Geri Al” düğmesi yardımı geri alınıp, yeni belirlenen eğitim katsayısı ile eğitim tekrarlanabilir. Bu işlem sayesinde geri yayılım yöntemi ile yapılan eğitimler arasında eğitim katsayısı küçük küçük artırılıp, en büyük eğitim katsayısı ile eğitim yapılabilir. Hatanın artması nedeni ile program süreci durmadığı sürece eğitim katsayısı artırılabilir. Eğitim sürece durması durumunda eğitim katsayısının küçültülmesi eğitimin devam etmesini sağlayacaktır.

Yeterli eğitim yapıldıktan sonra “Deneme” düğmesi ile sonuçlar kontrol edilebilir. Şekil 2.38’de programa ait sonuçların kontrol edildiği bölüm gösterilmiştir. Ayrıca Matlab komut penceresinde (Command Window) ANFIS katsayılarına ait veriler Çizelge 2.2’de görüldüğü gibi listelenir. Komut penceresinde değişkenler Matlab “Anfisedit” programına ait “*.fis” dosyasının içeriğinde katsayıların yazıldığı formata listelenir. Bu sayede “Anfisedit” programı ile kaydedilen “*.fis” dosyasındaki katsayılar program ile elde edilebilen katsayılar ile kolaylıkla değiştirilerek sonuçlar kontrol edilebilir.



Şekil 2.38. Matlab GUI Builder ile Yapılan Eğitim Program Eğitim Sonuç Kontrolü

```

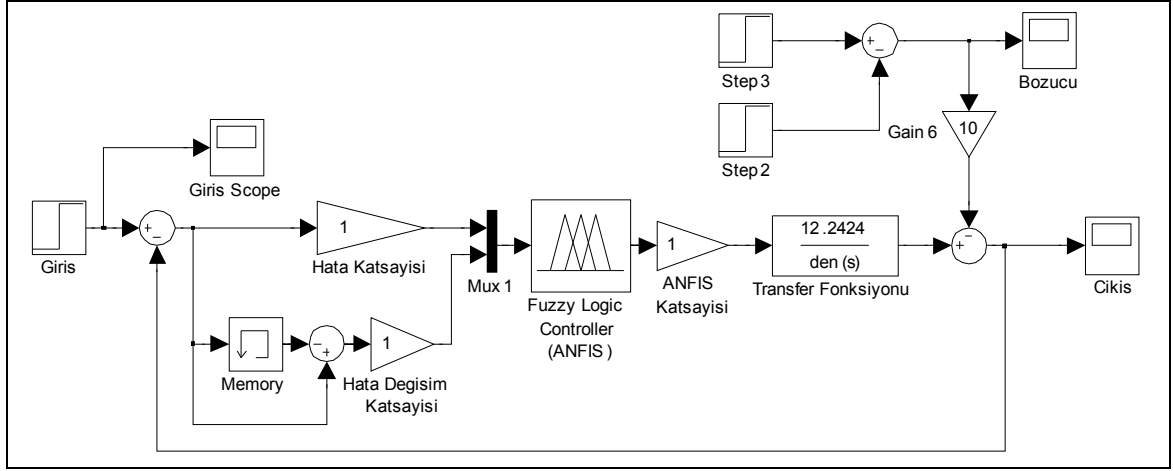
MF1='out1mf1': 'linear',[0.472586 0.459875 -0.001057]
MF2='out1mf2': 'linear',[0.049091 0.024862 -0.001045]
MF3='out1mf3': 'linear',[0.000000 0.000000 0.000000]
MF4='out1mf4': 'linear',[-0.031887 0.029802 -0.000424]
MF5='out1mf5': 'linear',[0.021692 0.144438 -0.000848]
MF6='out1mf6': 'linear',[0.000000 0.000000 0.000000]
MF7='out1mf7': 'linear',[-0.061246 0.014121 -0.000232]
MF8='out1mf8': 'linear',[0.057923 0.074049 -0.000613]
MF9='out1mf9': 'linear',[0.449999 0.449999 0.000753]
***** 1. Giriş Ait Üyelik Fonksiyonları Katsayıları *****
MF1='in1mf1': 'trimf',[-1060.147000 -508.805970 46.959033]
MF2='in1mf2': 'trimf',[-508.253256 46.405970 596.641000]
MF3='in1mf3': 'trimf',[45.853000 597.194000 1152.959000]
***** 2. Giriş Ait Üyelik Fonksiyonları Katsayıları *****
MF1='in2mf1': 'trimf',[-1055.349100 -506.101777 47.552721]
MF2='in2mf2': 'trimf',[-505.551190 47.001777 595.147300]
MF3='in2mf3': 'trimf',[46.450900 595.698200 1149.352700]

```

Çizelge 2.2. Eğitim Sonucu Katsayıların Komuta Penceresi Çıktısı

3. MATERYAL VE YÖNTEM:

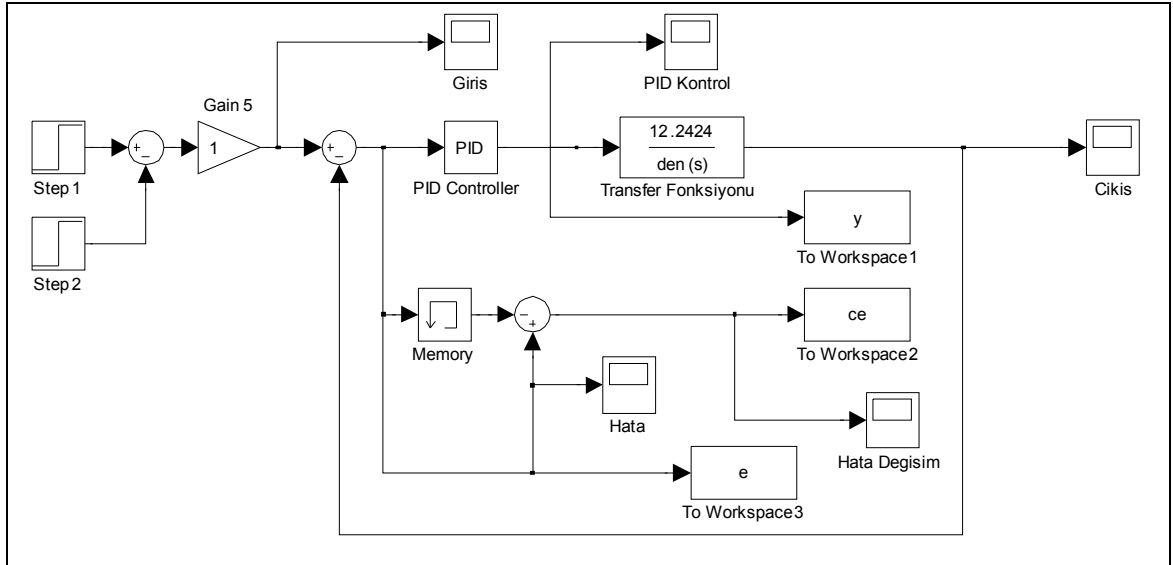
3.1. Matlab Simulink'te ANFIS ile DC Motor Kontrol Uygulama Simülasyonu:



Şekil 3.1. ANFIS ile DC Motor Kontrolü Simulink Modellemesi

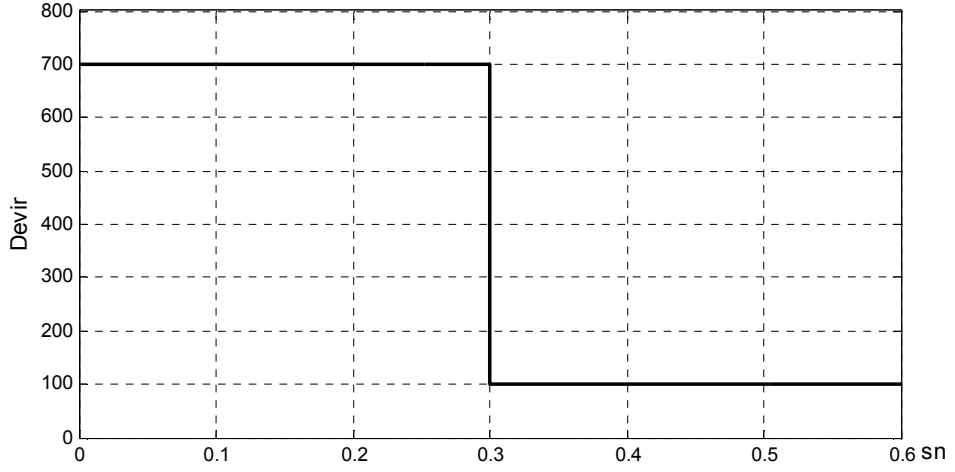
Şekil 3.1'de bir DC motorun ANFIS ile kontrolüne ait simulink modellemesi gösterilmiştir. ANFIS yapısı için simulinkte "Fuzzy Logic Controller" kullanılır. "Anfisedit" programı ile yaratılmış "*.fis" dosyası "workspace"e bir ad verilerek gönderilir. Takiben "Fuzzy Logic Controller" bu dosya ile ilişkilendirilerek, ANFIS yapısı simulink modellemesine aktarılmış olur.

DC motor kontrolünde kullanılan ANFIS için eğitim verileri Şekil 3.2'deki modelleme üzerinden alınmıştır. Bu modellemede motor PID ile kontrol edilmekte olup, hata ve hatanın değişimine karşılık, PID çıkışı eğitim verileri olarak alınmıştır.

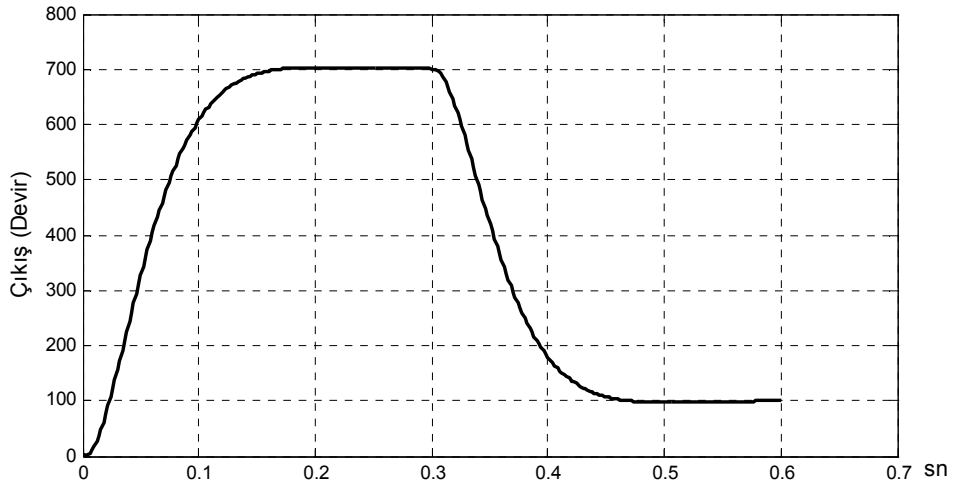


Şekil 3.2. Eğitim Verilerinin Alındığı Simulink Modeli

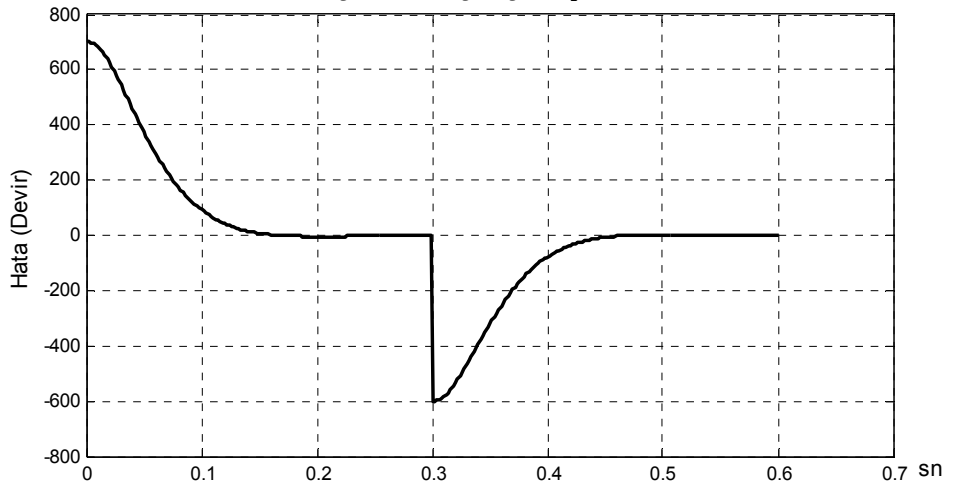
Şekil 3.2'de Matlab Workspace'e veri gönderilirken kullanılan örnekleme zamanı $2 \cdot 10^{-3}$ sn'dir. Girişe uygulanan sinyal Şekil 3.3'te, çıkıştan alınan sinyal Şekil 3.4'te, hata sinyali Şekil 3.5'te ve hatanın değişim sinyali Şekil 3.6'da verildiği gibidir.



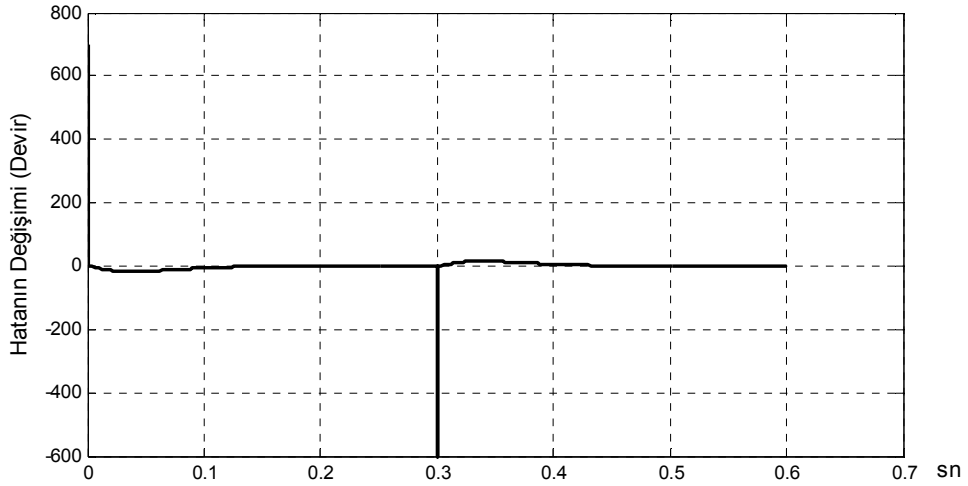
Şekil 3.3. Giriş Sinyali



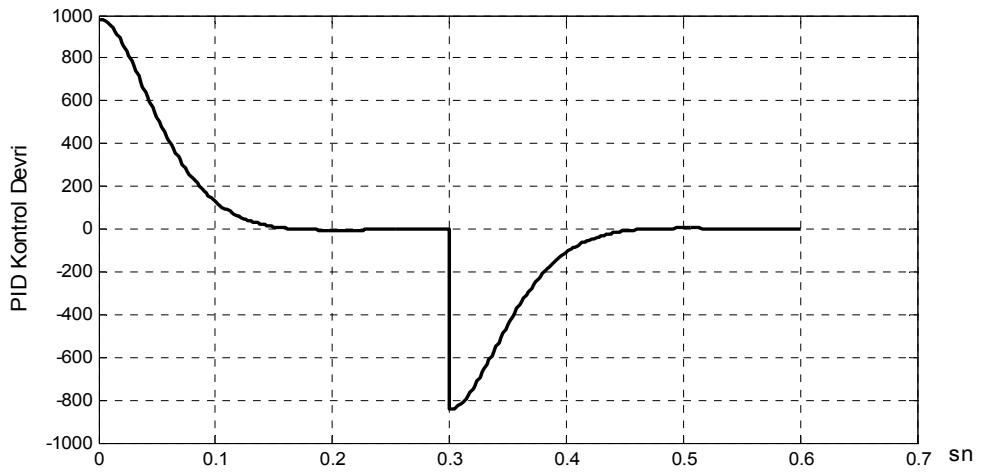
Şekil 3.4. Çıkış Sinyali



Şekil 3.5. Hata Sinyali



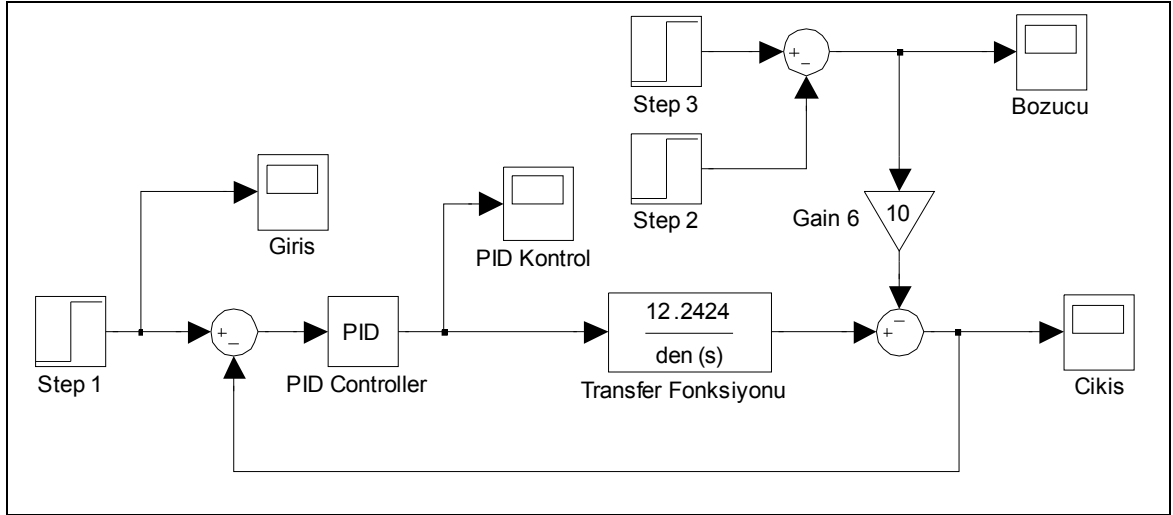
Şekil 3.6. Hatanın Değişimi Sinyali



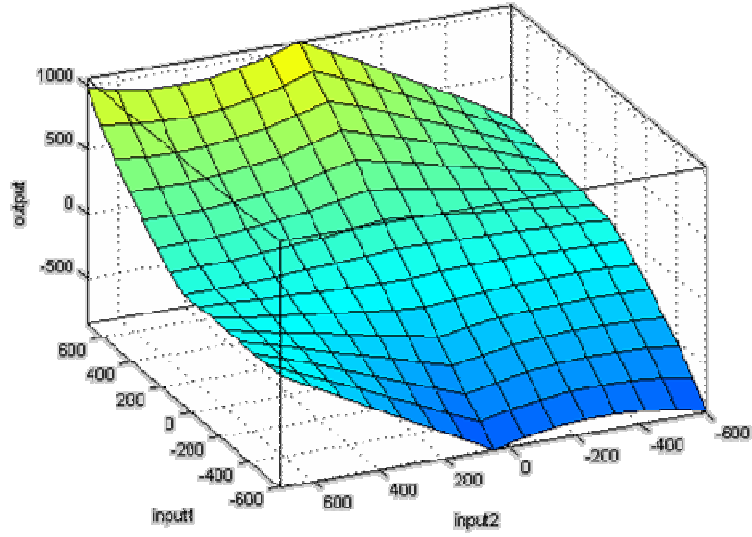
Şekil 3.7 PID Kontrol Devri

Alınan eğitim verileri kullanılarak eğitim programında yapılan eğitim sonucunda elde edilen kontrol yüzeyi Şekil 3.9'da, anfisedit programı kullanılarak yapılan eğitim sonucunda elde edilen kontrol yüzeyi Şekil 3.10'dadır. Şekil 3.9'de eğitim programı kullanılarak elde edilen kontrol yüzeyi için hata oranı 2,7047 olup, 300 adet eğitim tekrarı yapılmıştır. (1. 100'lük eğitim adedi için eğitim katsayısı 1.10^{-7} ; 2. 100'lük için 2.10^{-7} ; 3. 100'lük için 4.10^{-7} kullanılmıştır). Şekil 3.10'de anfisedit kullanılarak elde edilen kontrol yüzeyi için eğitim adedi 300 olup hata oranı 2,6241'dir.

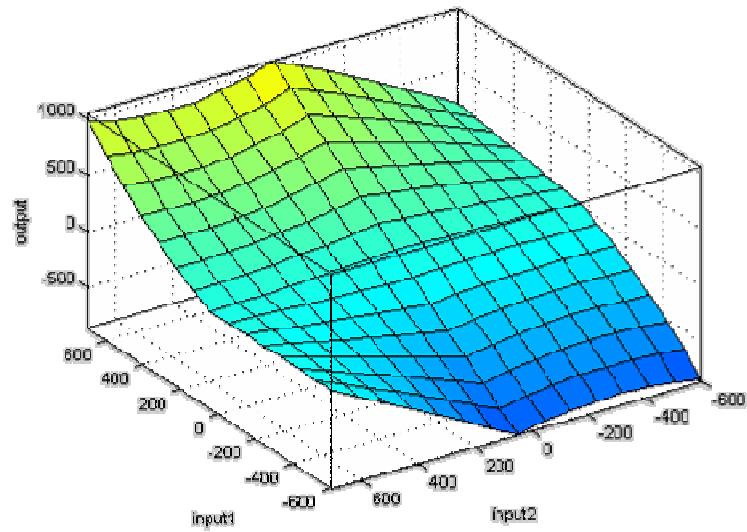
Eğitim sonucunda elde edilen ANFIS sonuçları matlab workspace'e gönderilerek Şekil 3.1'deki model üzerinden DC Motor ANFIS denetleyici simülasyon sonuçları elde edilecektir. Eğitim programı ve "anfisedit" programı kullanılarak elde edilen sonuçların, eğitim verilerinin elde edildiği PID denetleyicisine ait sonuçlar ile karşılaştırılma sonuçları Şekil 3.12'dedir. Elde edilen sonuçların birbirlerine çok yakın oldukları gözlemlenmiştir.



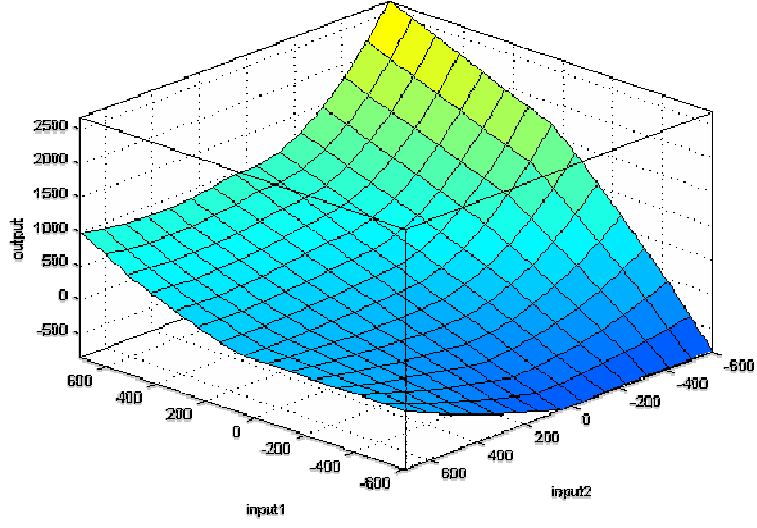
Şekil 3.8. PID Denetleyici Modeli



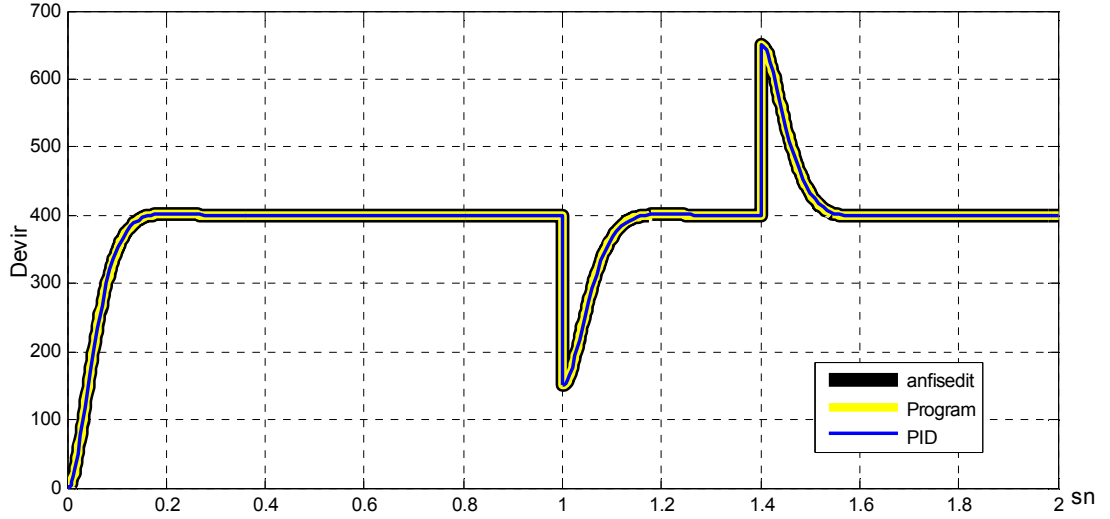
Şekil 3.9. Eğitim Programı Kontrol Yüzeyi



Şekil 3.10. Anfisedit Programı Kontrol Yüzeyi

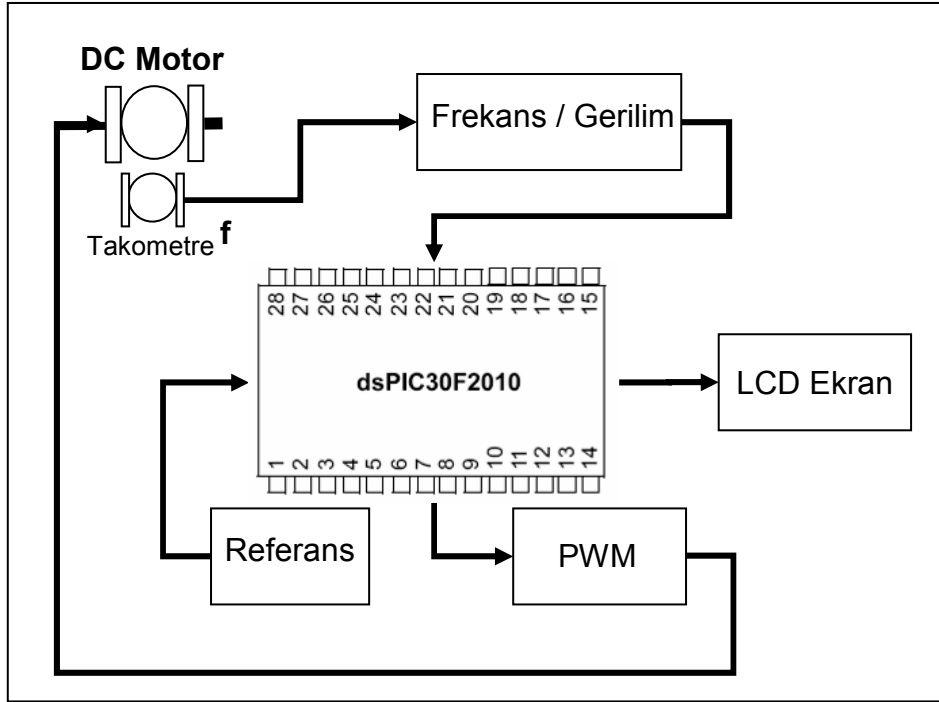


Şekil 3.11. Anfisedit Programı Kontrol Yüzeyi (Hybrid Yöntemi ile Eğitim)



Şekil 3.12. DC Motor Hız Kontrolü Simülasyon Sonuçları

3.2. ANFIS ile DC Motor Kontrol Uygulamaları:

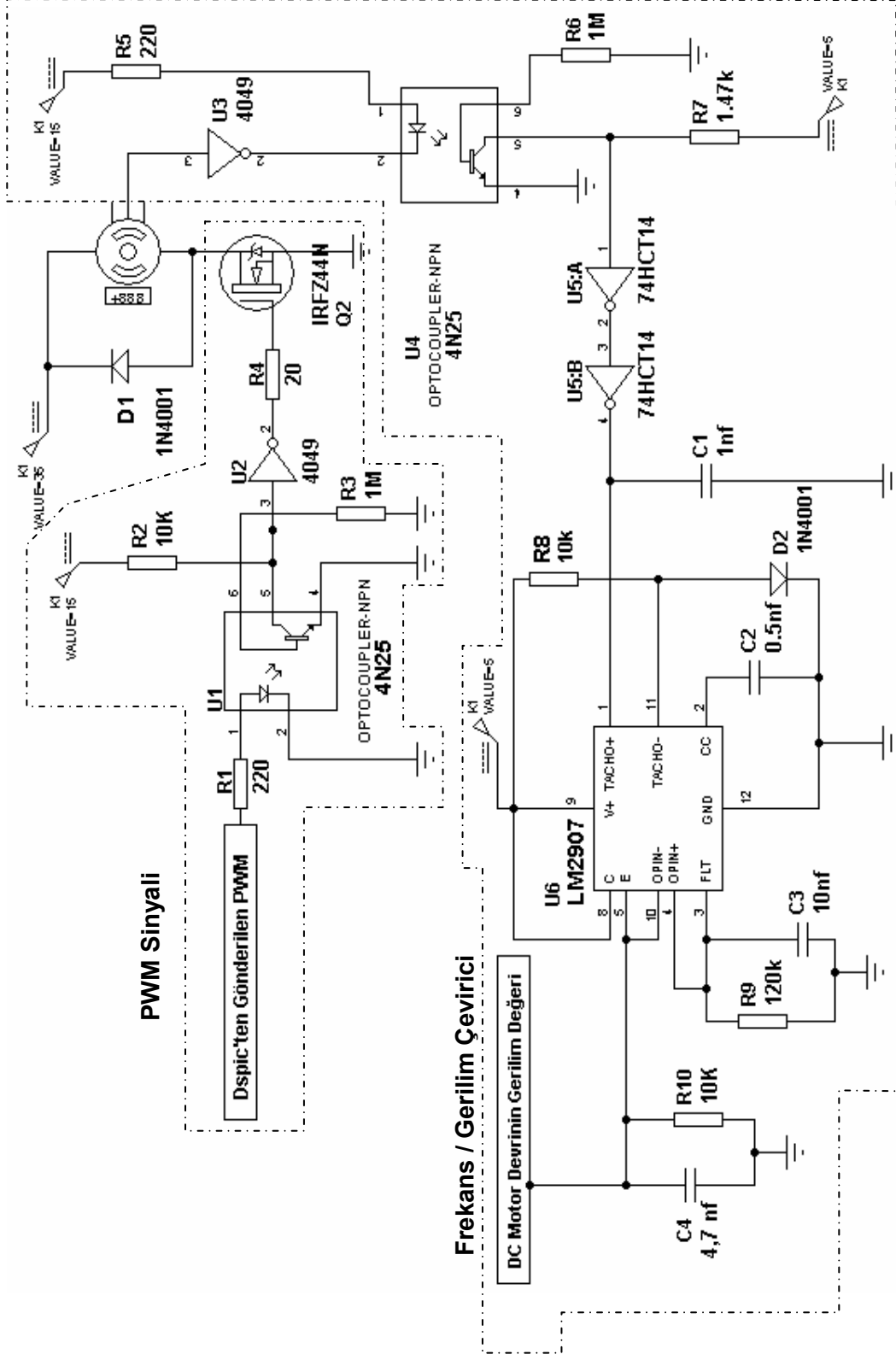


Şekil 3.13 Uygulama Devresi Blok Diyagramı

DC motor hız kontrolü için yapılacak olan uygulamaya ait devrenin blok diyagramı Şekil 3.13'de verilmiştir. Bir takometre aracılığı ile DC motor devri frekans olarak elde edilir (Takometre bir devir için 200 kare dalga üretmektedir). Motor devri frekans olarak elde edildikten sonra frekans/gerilim çevirici kullanılarak gerilime çevrilir. dsPic30f2010'nun kendi içerisinde bulunan analog/dijital çevirici (ADC) kullanılarak, motor devri sayısal işaret işlemciye aktarılır.

DC motor'un hızı için bir referans gerilimi uygulanmaktadır. Yine bu referans gerilimi, dsPic30f2010'nun bünyesindeki ADC kullanılarak, sayısal işaret işlemciye aktarılır. Referans gerilimi ve motor hızı gerilim olarak dsPic30f2010'a aktarıldıktan sonra, hata ve hatanın değişimi bulunmakta ve ANFIS algoritması çalıştırılarak çıkış değeri hesaplanmaktadır. Dspic30f2010' için kullanılan ANFIS algoritması için akış diyagramı Şekil 3.26'da verilmiştir. Hesaplanan çıkış değerine karşılık, dsPic30f2010 bünyesinde bulunan PWM üretici vasıtası ile kontrol PWM değeri üretilerek motor hız kontrolü yapılır.

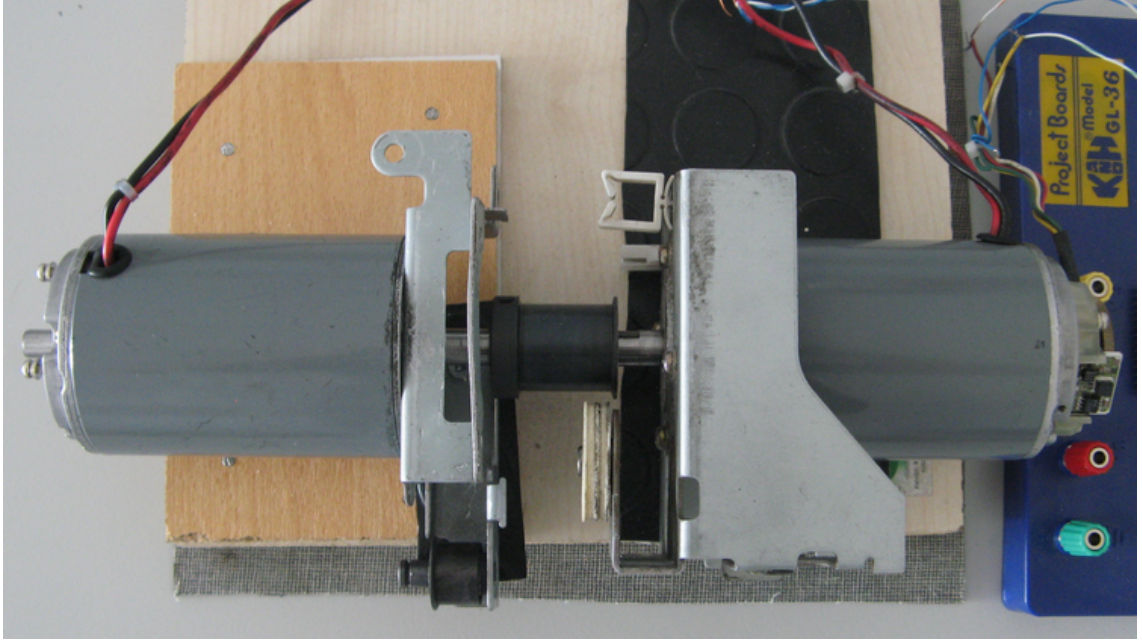
Şekil 3.14'de uygulamaya ait dsPic30f2010'da üretilen PWM sinyalini DC motora ulaştıran ve motor devrinin frekans değerini gerilime çeviren devre şeması verilmiştir.



Şekil 3.14. ANFIS Uygulaması için Kullanılan Devre Şeması

3.2.1. Uygulamada Kullanılan Motorun Teknik Özellikleri:

Şekil 3.15'de uygulama için kullanılan motora ait fotoğraf verilmiştir. Burada Kontrol edilen motora (sağdaki motor) jeneratör görevi gören ikinci bir motor bağlanmıştır. Kontrol edilen motorun yüklenmesi için jeneratör görevi gören motordan elde edilen gerilim uçlarına 1Ω, 2Ω, 5Ω, 10Ω ve 15Ω'luk dirençler bağlanarak sistemin yüklenmesi sağlanmıştır.



Şekil 3.15 Uygulamada Kullanılan Motora Ait Fotoğraf

Motora ait Teknik bilgiler;

Adı : HITACHI DC Motor

Modeli : D06D03E200

Volt : 30 V

Amper : 2.0 A

Devir : 3100 RPM

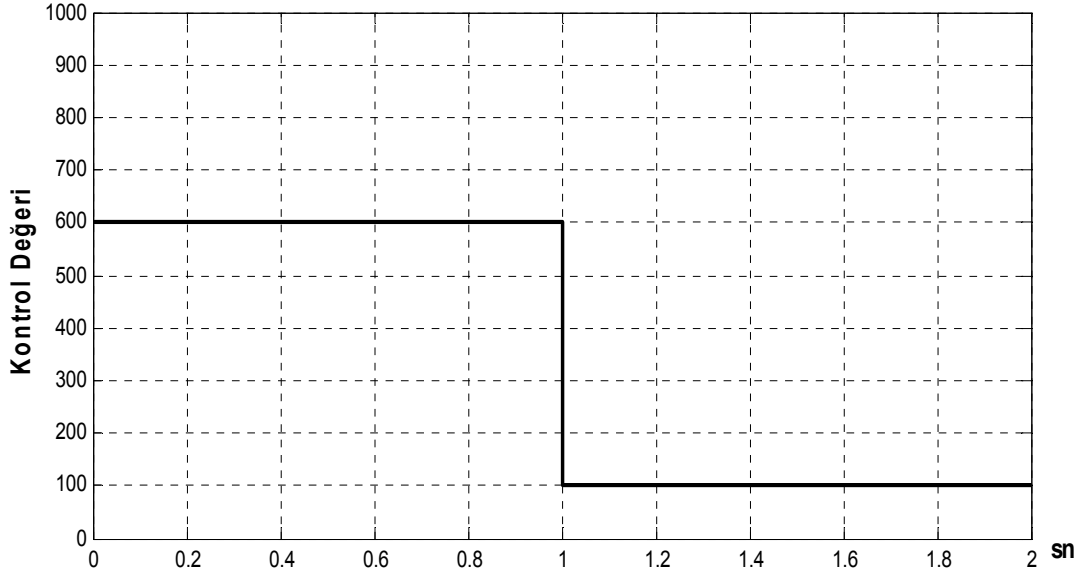
Encoder : 200 P/R

MFG NO : M87.3

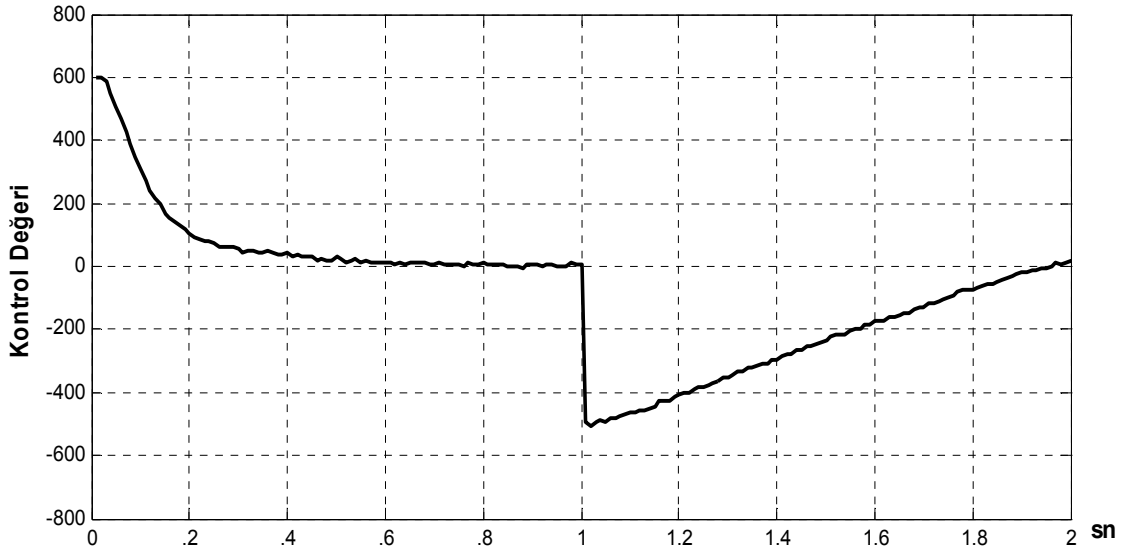
3.2.2. PID Denetleyicinden ANFIS Eğitim Verilerinin Alınması:

Şekil 3.14'de elde edilen DC Motor hızının gerilim değeri ADC kullanılarak Şekil 3.18'de verilen Matlab Simulink modeline aktarılarak, eğitim için gerekli veriler elde edilmiştir. ADC'de yaşanan problem nedeni ile gerilim değeri +1V arttırılarak Matlab'e aktarıldığı için, gelen değerden 1 çıkarılarak hata düzeltilmesi yapılmıştır.

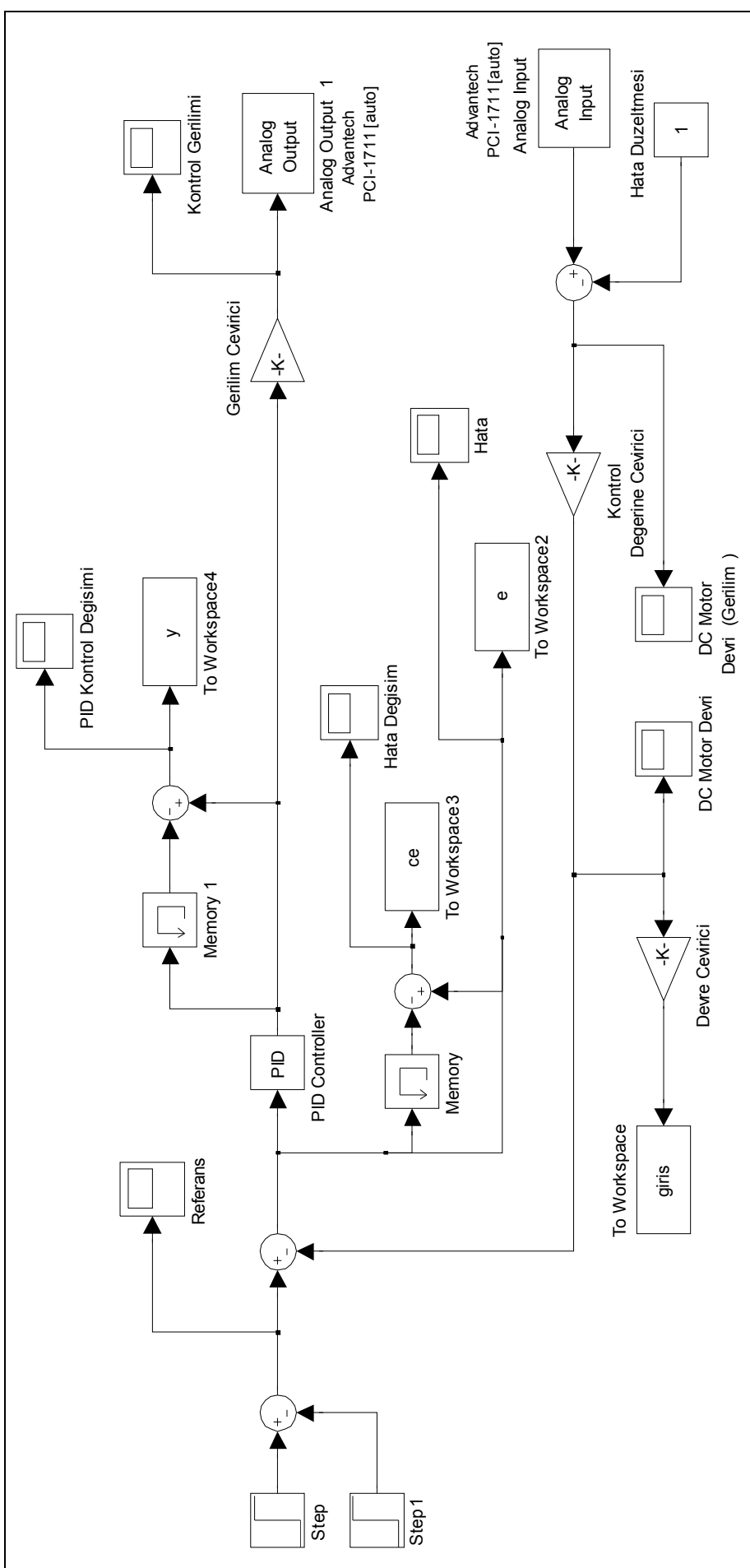
DC Motor gerilimi doğru olarak Simulink'e aktarıldıktan sonra dsPic30f2010'da kullanılan gerilim sayısal karşılığı için kullanılan değer aralığına çevrilir. Bu işlemlerden sonra Şekil 3.16'de verilen referans uygulanır ve Şekil 3.17'deki hata, Şekil 3.19'daki hatanın değişimi ve Şekil 3.20'deki PID Kontrol değeri değişimi işaretleri elde edilir.



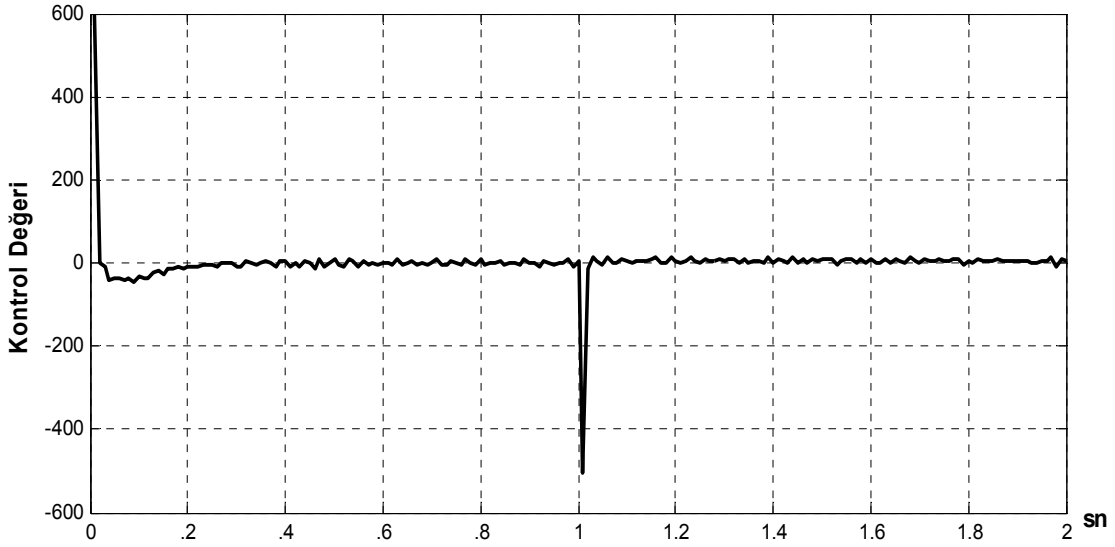
Şekil 3.16. Eğitim için Uygulanan Referans İşareti



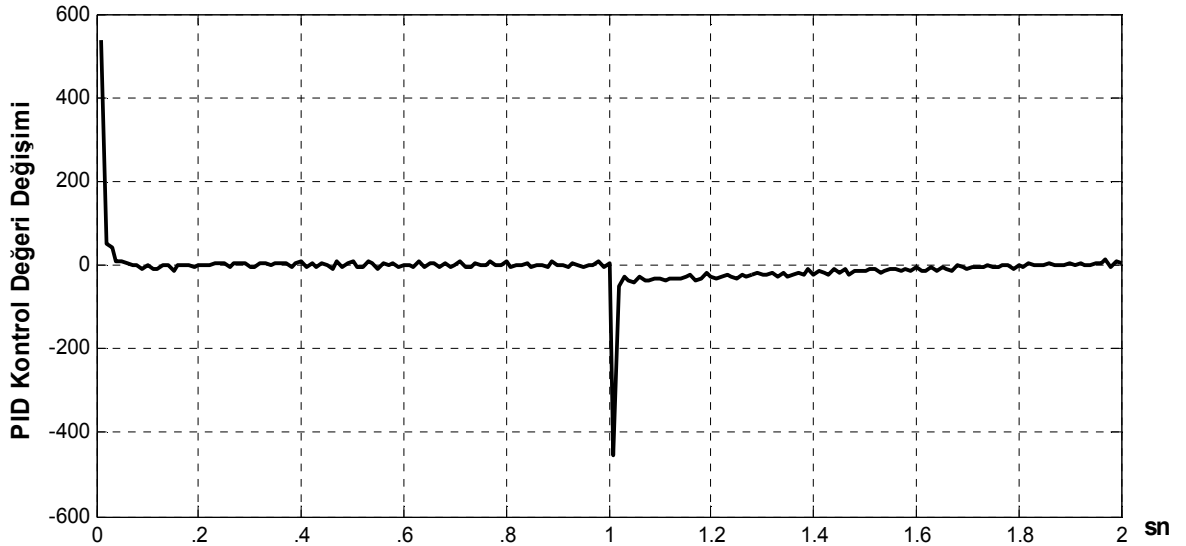
Şekil 3.17. Eğitim Verisi Hata İşareti



Şekil 3.18. ANFIS Eğitim Verilerinin Alındığı PID Denetleyicisi Modeli



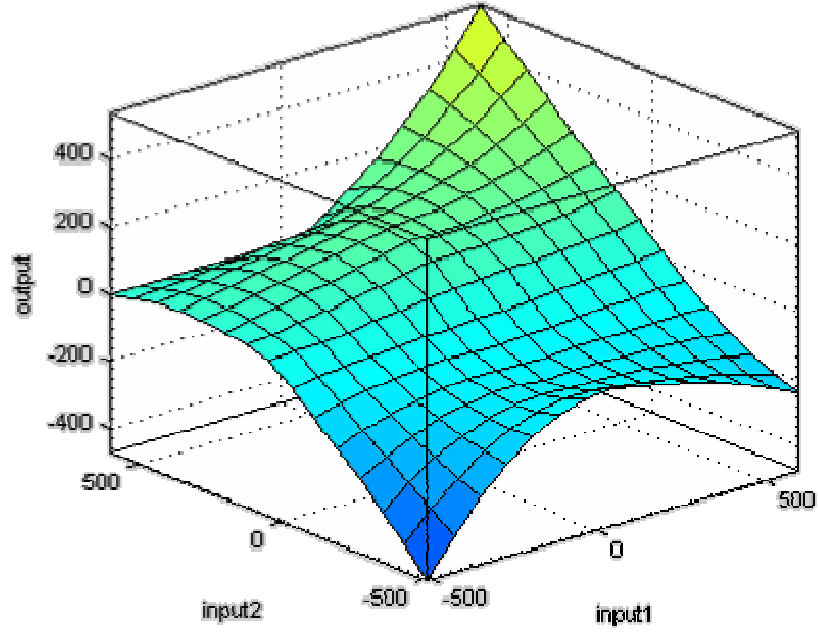
Şekil 3.19. Eğitim Verisi Hatanın Değişimi Eğrisi



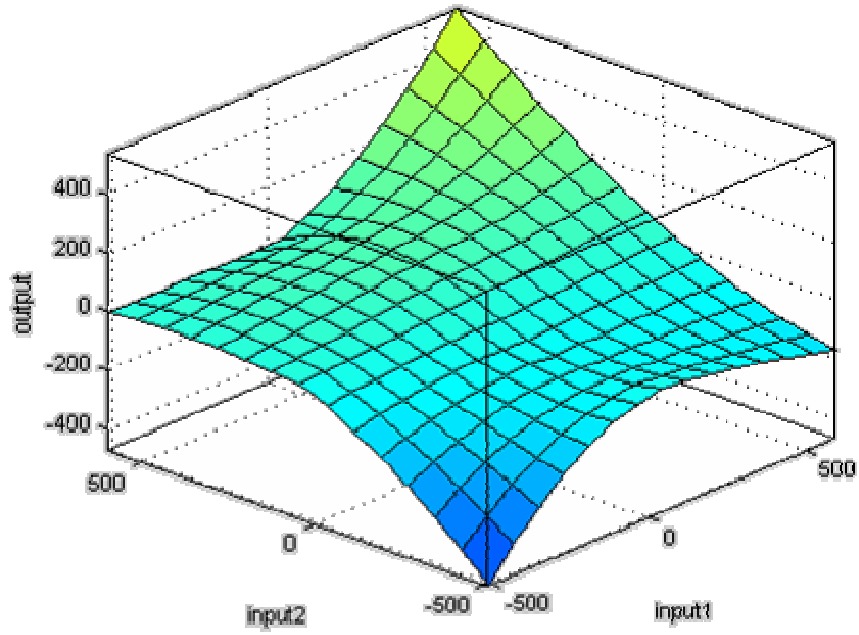
Şekil 3.20. Eğitim Verisi PID Kontrol Değeri Değişimi İşareti

PID kontrol işareti elde edildikten sonra, ADC yardımı ile dsPic30f2010'a gönderilir ve gelen gerilim değerine karşılık PWM sinyali üretilir. Gerilimin PWM sinyaline dönüştürülmesi için dsPic30f2010'da ayrı bir program oluşturulmuştur.

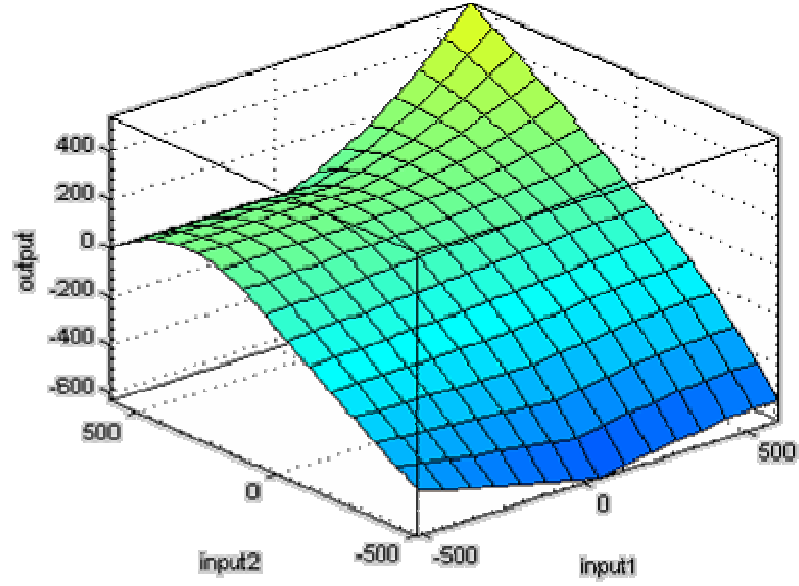
Eğitim verileri oluşturulduktan sonra ANFIS eğitim programında, geri yayılım yöntemi ile 700 adet eğitim yapılarak elde edilen kontrol yüzeyi Şekil 3.21'de, anfisedit programında geri yayılım yöntemi ile yapılan eğitim sonucunda elde edilen kontrol yüzeyi Şekil 3.22'de ve anfisedit programında hybrid yöntemi ile elde edilen kontrol yüzeyi Şekil 3.23'dedir.



Şekil 3.21. ANFIS Eğitim Programı Kontrol Yüzeyi (Geri Yayılım)



Şekil 3.22. Anfisedit Programı Kontrol Yüzeyi (Geri Yayılım)



Şekil 3.23. Anfisedit Programı Kontrol Yüzeyi (Hybrid)

Şekil 3.21'deki kontrol yüzeyini elde etmek için kullanılan eğitim katsayıları Çizelge 3.1 verilmekte olup, eğitim sonucundaki hata oranı 2.1553'tür. Şekil 3.22'deki kontrol yüzeyi Anfisedit programı ile 100'er adet 7 eğitim sonucunda elde edilmiş olup hata oranı 1.1111'dir. Şekil 3.23'deki kontrol yüzey için 3 adet hybrid eğitimi uygulanmış olup hata oranı 0.19072'dir.

SN	Eğitim Katsayısı	Eğitim Adedi
1	1.10^{-7}	100
2	1.10^{-7}	100
3	1.10^{-7}	100
4	1.10^{-7}	100
5	1.10^{-7}	100
6	1.10^{-7}	100
7	1.10^{-7}	100

Çizelge 3.1. Eğitim Programında Kullanılan Eğitim Katsayıları

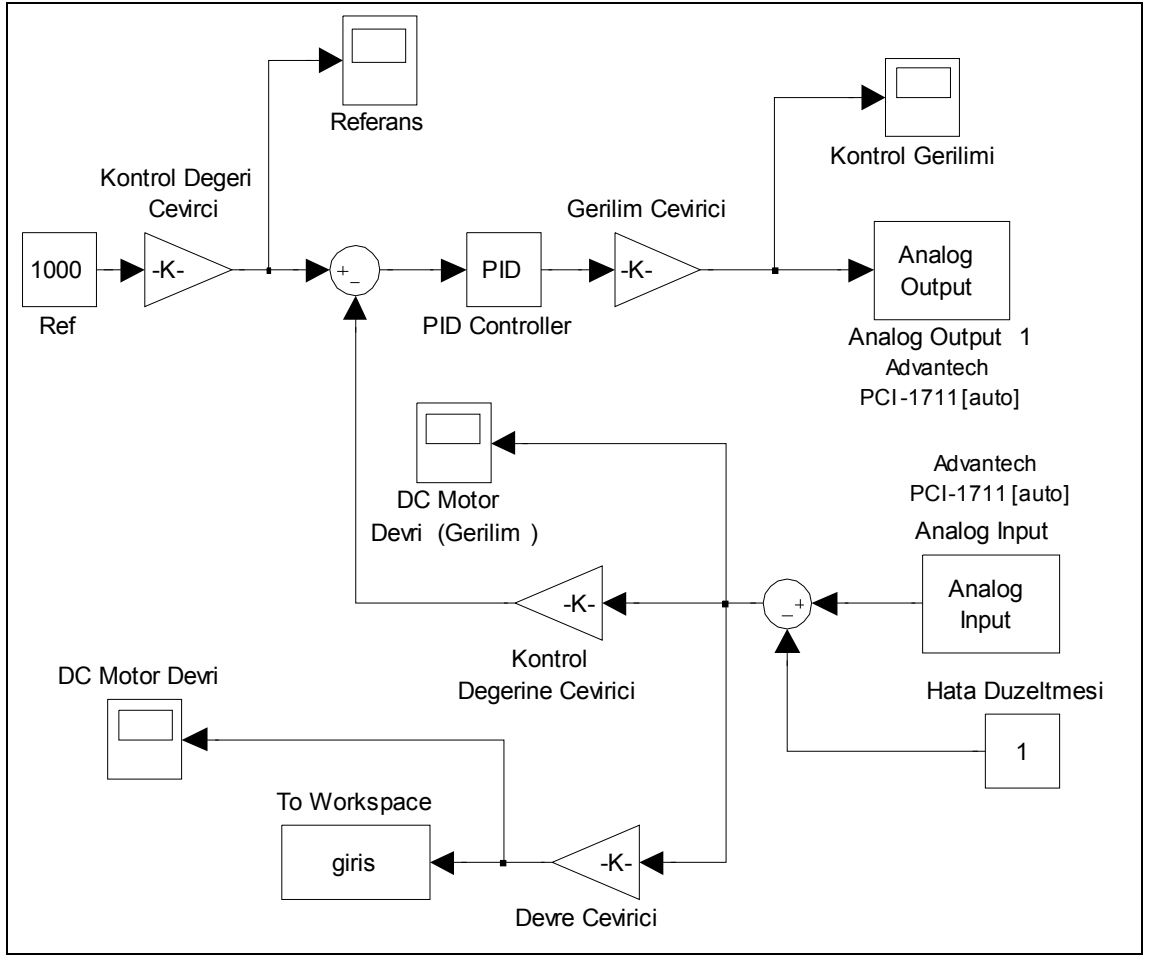
Eğitim verilerinin alındığı modelde PID denetleyicisinde, kazanç için 0,9, integral için 8 ve türev için 0 değerleri kullanılmıştır.

3.2.3. Bilgisayar Ortamında ANFIS Uygulamasının Gerçekleştirilmesi:

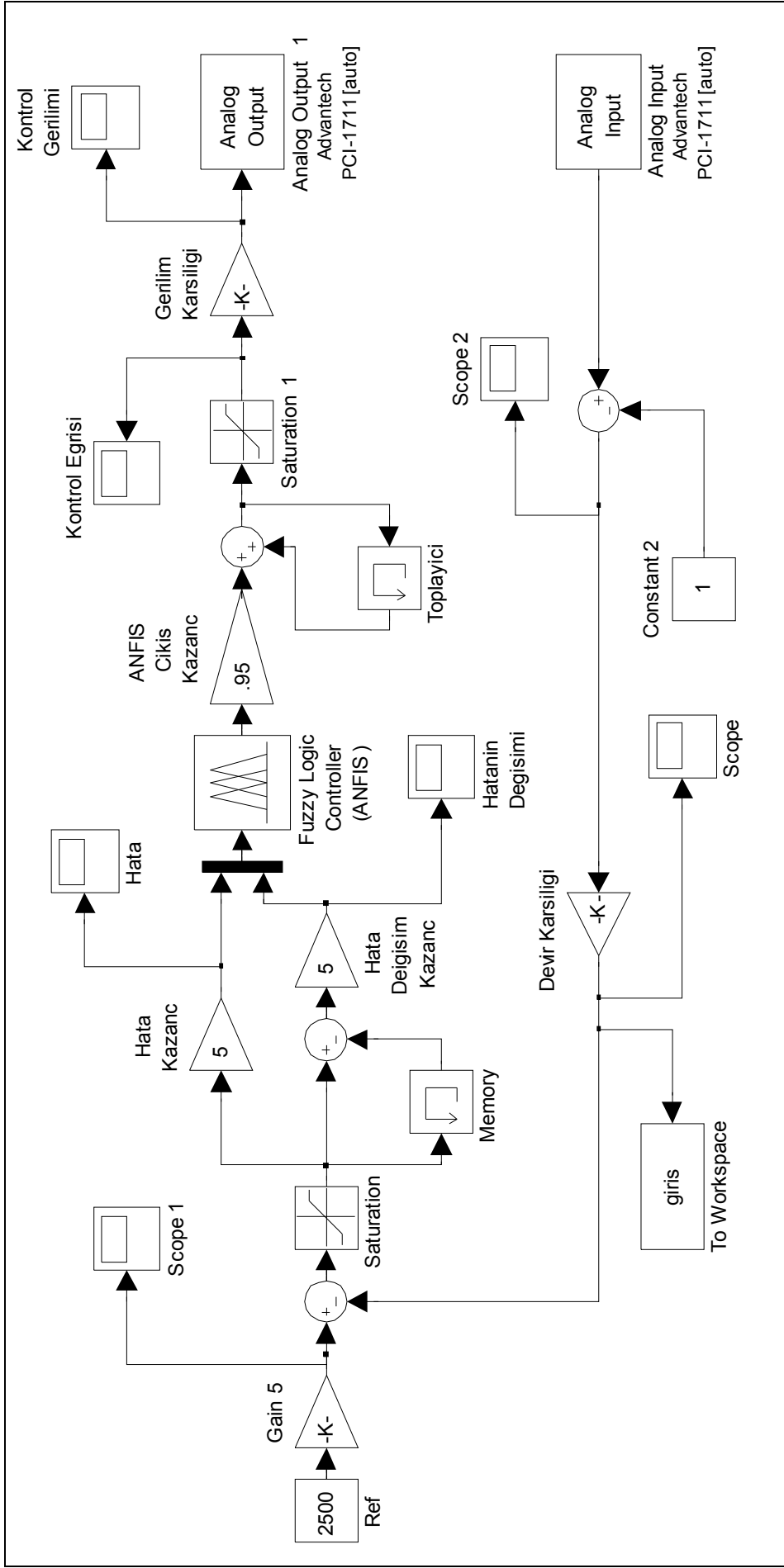
Eğitim verilerinin elde edilmesi işlemi tamamlandıktan sonra, eğitim verileri kullanılarak matlab anfisedit programında ANFIS yapısı oluşturulmuş ve Şekil 3.25'deki modele aktararak bilgisayar uygulaması gerçekleştirilmiştir.

Ayrıca Şekil 3.24'de verilen model kullanılarak, eğitim verilerinin alındığı PID ile bir DC motor kontrolü gerçekleştirilmiştir.

Gerçekleştirilen iki uygulamadan, farklı devirler için birim basamak cevapları ve bu devir değerlerinde yük uygulanması durumundaki sistem cevaplarına ait sonuçlar elde edilmiştir. Elde edilen sonuçlar Araştırma Bulguları bölümünde dsPic30f2010 uygulama sonuçları ile karşılaştırmalı olarak verilecektir.



Şekil 3.24. PID Uygulaması Modeli



Şekil 3.25. Bilgisayar Ortamında ANFIS Uygulanması Modeli

3.2.4. DC Motor Devir Kontrolünün dsPic30f2010 ile Yapılması:

Eđitim verileri elde edildikten ve bilgisayar uygulamaları yapıldıktan sonra, ANFIS algoritmasının dsPic30f2010'a aktararak sayısal iřaret iřlemci uygulaması gerekleřtirilecektir.

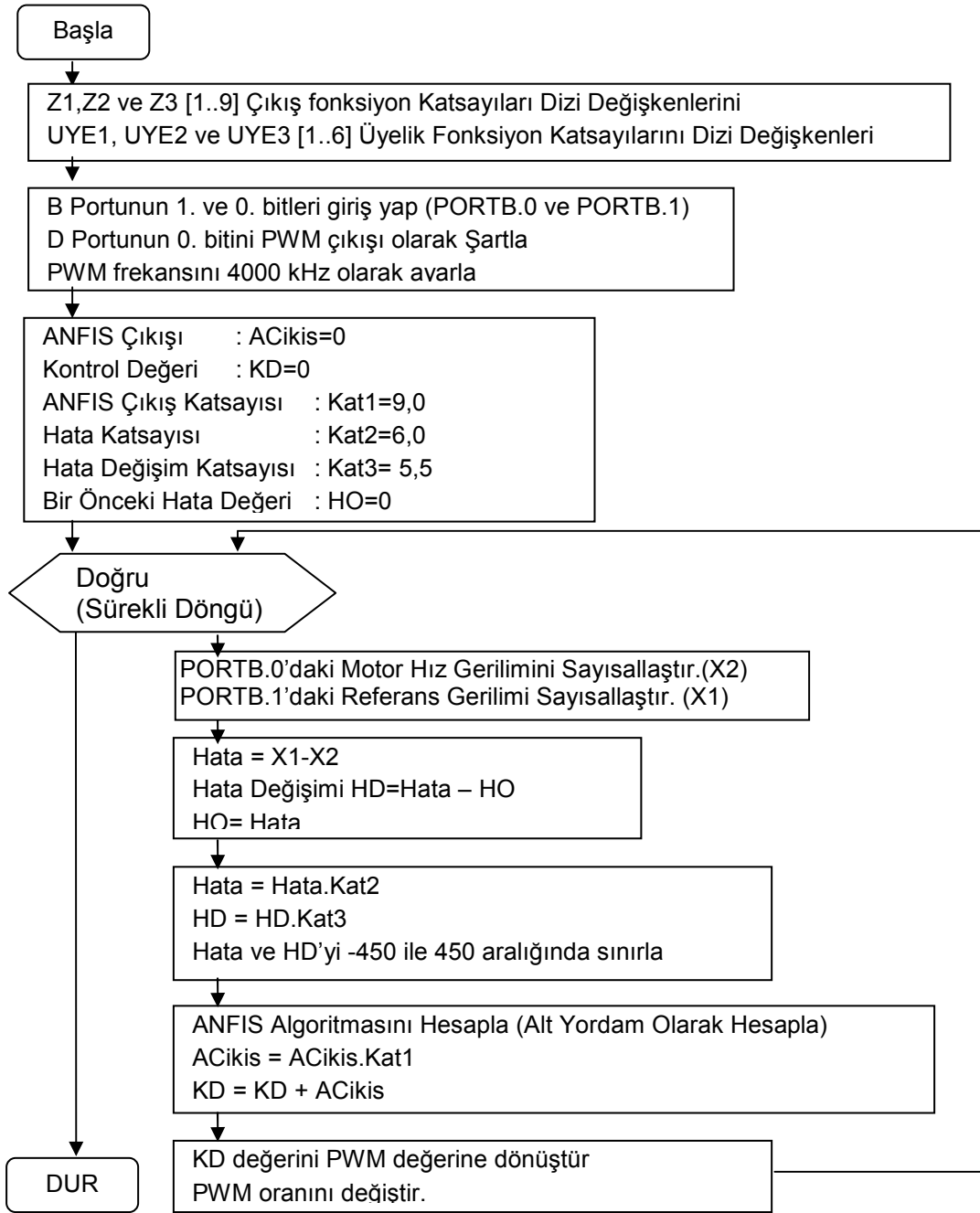
3.2.4.1. dsPic30f2010'de Yazılan ANFIS Algoritması:

Uygulamada 2 giriřli ve tek ıkıřlı bir ANFIS yapısı kullanılmıřtır. Giriřler 3'er üyelik fonksiyonuna sahip olup, üçgen fonksiyon kullanılmıřtır.

ıkıř fonksiyonları $f_i=z1_i.x1+z2_i.x2+z3_i$ řeklinde tanımlanmıř olup, program ierisinde katsayılar tek boyutlu dizi deđiřken olarak tanımlanmakta olup, Z1(9), Z2(9) ve Z3(9) olarak tanımlanmıřlardır.

Üyelik fonksiyonu katsayıları da řekil 2.24'te olduđu řekliyle kullanılarak UYE1(6), UYE2(6) ve UYE3(6) tek boyulu dizi deđiřkenleri olarak tanımlanmıřtır. Dizi deđiřkenlerdeki ilk 3 deđer birinci giriře ve son 3 deđer de ikinci giriře ait üyelik fonksiyon katsayılarını tanımlar.

ANFIS'in hata giriři iin 6, hatanın deđiřimi iin 5,5 ve ANFIS ıkıřı iin 9 katsayıları kullanılmıřtır. Bilgisayar ile yapılan uygulama da ise hata ve hatanın deđiřimi iin 5, ANFIS ıkıřı iin 0,95 katsayıları kullanılmıřtır.

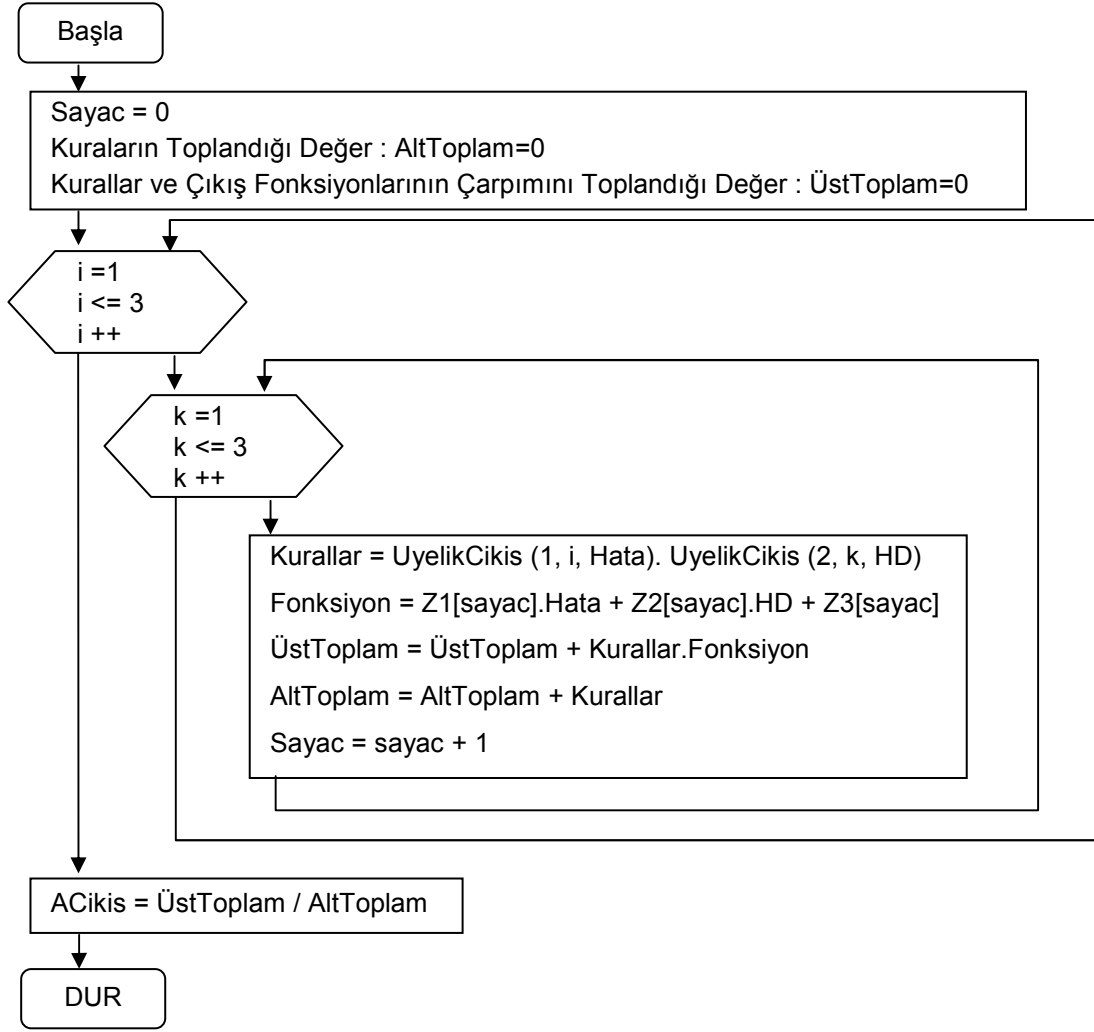


Şekil 3.26. dsPic30f2010 ANFIS Programı

3.2.4.1.1. ANFIS Algoritması Hesaplama Alt Yordamı:

$$y(x_1, x_2) = N_1 \cdot F_1 + N_2 \cdot F_2 + \dots + N_8 \cdot F_8 + N_9 \cdot F_9 \quad (3.1)$$

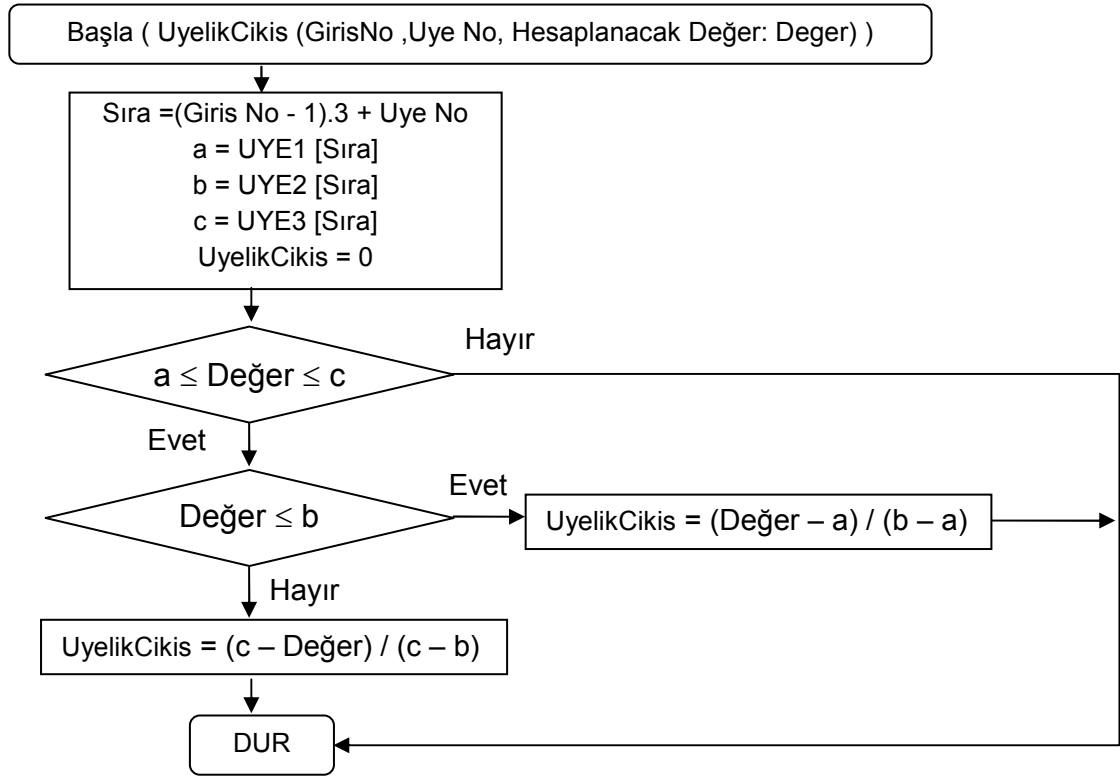
Denklem 3.1 hesaplanarak “ACikis” değişkenine atanır. Üyeli fonksiyonlarının hesaplanması için “UyelikCikis” alt fonksiyonu yapılmıştır. Giriş numarası, üyelik numarası ve hangi değere karşılık üyelik fonksiyon çıkışının hesaplanacağı fonksiyon değişkenleridir ve “UyelikCikis (Giriş No, Üyelik Fonksiyon No, Değer)” şeklinde tanımlanır.



Şekil 3.27. ANFIS Algoritması Hesaplama Alt Yordamı

3.2.4.1.2. Üyelik Fonksiyon Çıktılarının Hesaplanması Alt Fonksiyonu:

ANFIS algoritmasının çıkışının hesaplandığı alt yordam içerisinde bulunan “UyelikCikis” alt fonksiyonu Şekil 3.28’dedir. Üyelik fonksiyon katsayısının, UYE_i olarak tanımlanan dizi değişkenlerde, hangi sıradaki değer olduğu bulunarak, katsayı a, b ve c değişkenlerine atanmakta ve hesaplanması istenen giriş değerine göre bu katsayılar ile karşılaştırarak üyelik fonksiyonunun değeri hesaplanmaktadır.



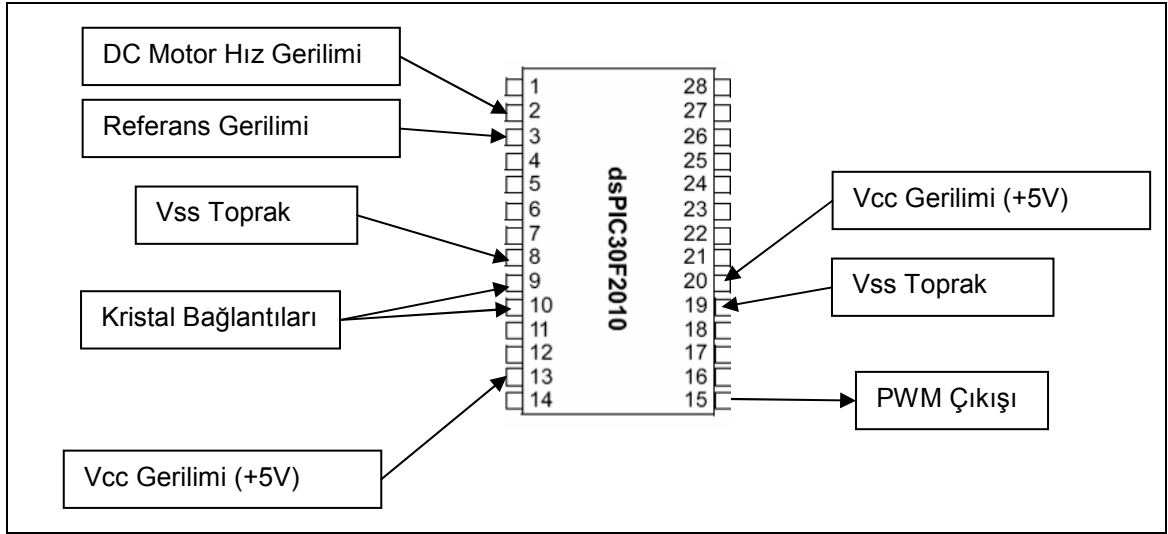
Şekil 3.28. Üyelik Fonksiyon Çıktılarının Hesaplanması Alt Fonksiyonu

3.2.4.2. dsPic30f2010 Bacak Bağlantıları:

MCLR	1	28	AVDD
EMUD3/AN0/VREF+/CN2/RB0	2	27	AVSS
EMUC3/AN1/VREF-/CN3/RB1	3	26	PWM1L/RE0
AN2/SS1/CN4/RB2	4	25	PWM1H/RE1
AN3/INDX/CN5/RB3	5	24	PWM2L/RE2
AN4/QEA/IC7/CN6/RB4	6	23	PWM2H/RE3
AN5/QEB/IC8/CN7/RB5	7	22	PWM3L/RE4
VSS	8	21	PWM3H/RE5
OSC1/CLKI	9	20	VDD
OSC2/CLKO/RC15	10	19	VSS
EMUD1/SOSCI/T2CK/U1ATX/CN1//RC13	11	18	PGC/EMUC/U1RX/SDI1/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	12	17	PGD/EMUD/U1TX/SDO1/SCL/RF3
VDD	13	16	FLTA/INT0/SCK1/OCFA/RE8
EMUD2/OC2/IC2/INT2/RD1	14	15	EMUC2/OC1/IC1/INT1/RD0

Şekil 3.29. dsPic30f2010 Bacak İsimleri

Şekil 3.14'deki devre şeması ile dsPic30f2010'nun birbirleri ile bağlantıları ve dsPic30f20'nun çalışmasının sağlanması için gerekli bağlantılar şekilde verilmiştir.



Şekil 3.30. dsPic30f20102'nun Devre Bağlantıları

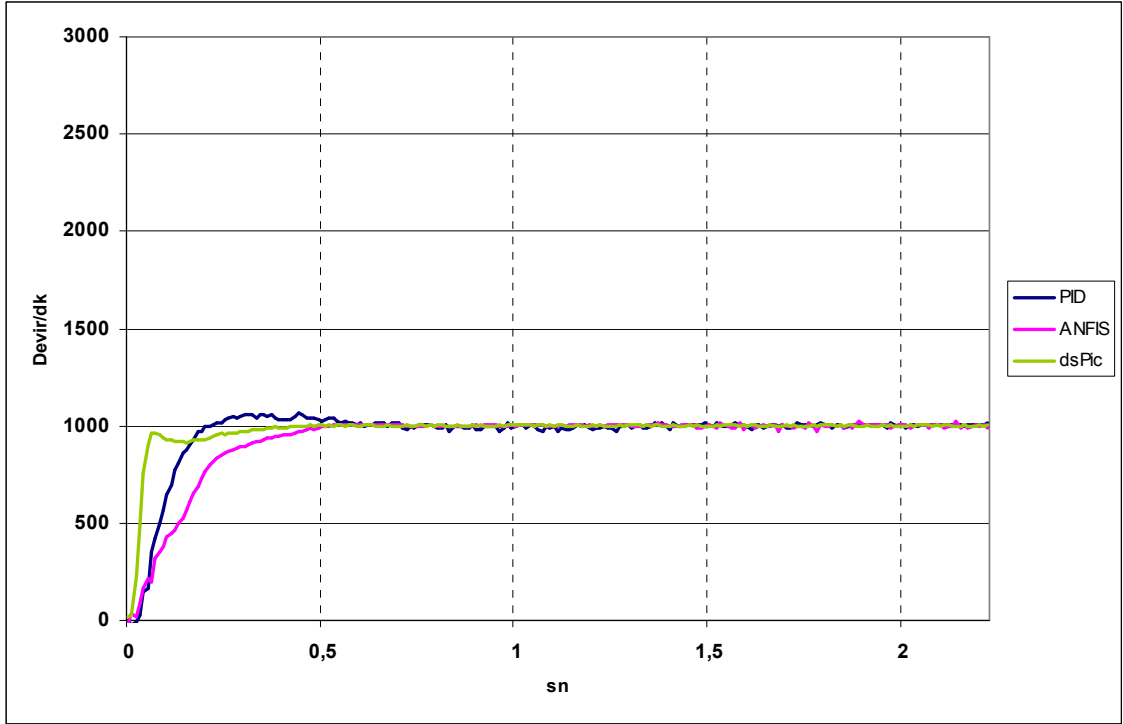
2 ve 3 numaralı bacaklara gelen gerilimler dsPic30f2010'nun kendi bünyesindeki ADC'nin kullanımı ile sayısallaştırılır. 2 ve 3 numaralı bacakların ADC'ye uygun olarak koşullanmaları gerekir.

ANFIS hesaplamaları yapıldıktan sonra elde edilen kontrol değerinin PWM dönüştürülür. İstenen PWM oranı hesaplandıktan sonra 15 numaralı bacaktan PWM sinyali alınır. 15 numaralı bacaktan PWM sinyalinin alınması için gerekli koşullamaların yapılması gerekir. Uygulamaya ait sonuçlar, Araştırma Bulguları bölümünde PID ve ANFIS bilgisayar uygulama sonuçları ile karşılaştırmalı olarak verilecektir.

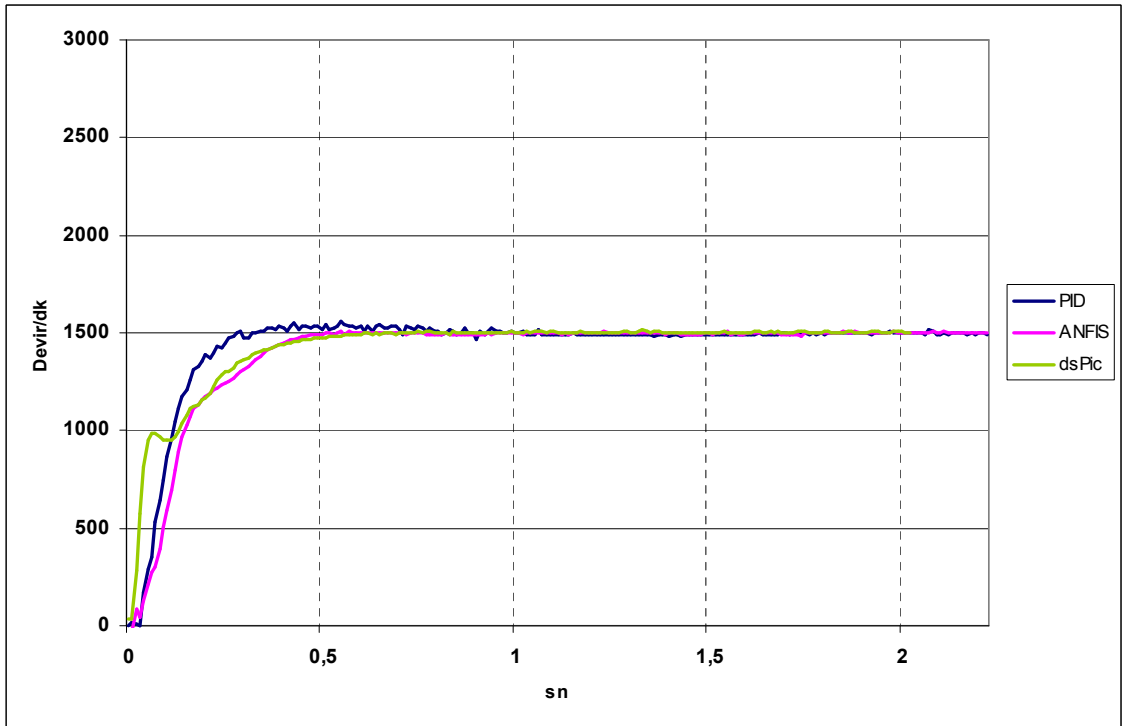
4. ARAŞTIRMA BULGULARI:

4.1. PID ve ANFIS Bilgisayar Uygulama Sonuçları ile dsPic30f2010 Uygulama Sonuçlarının Karşılaştırılması:

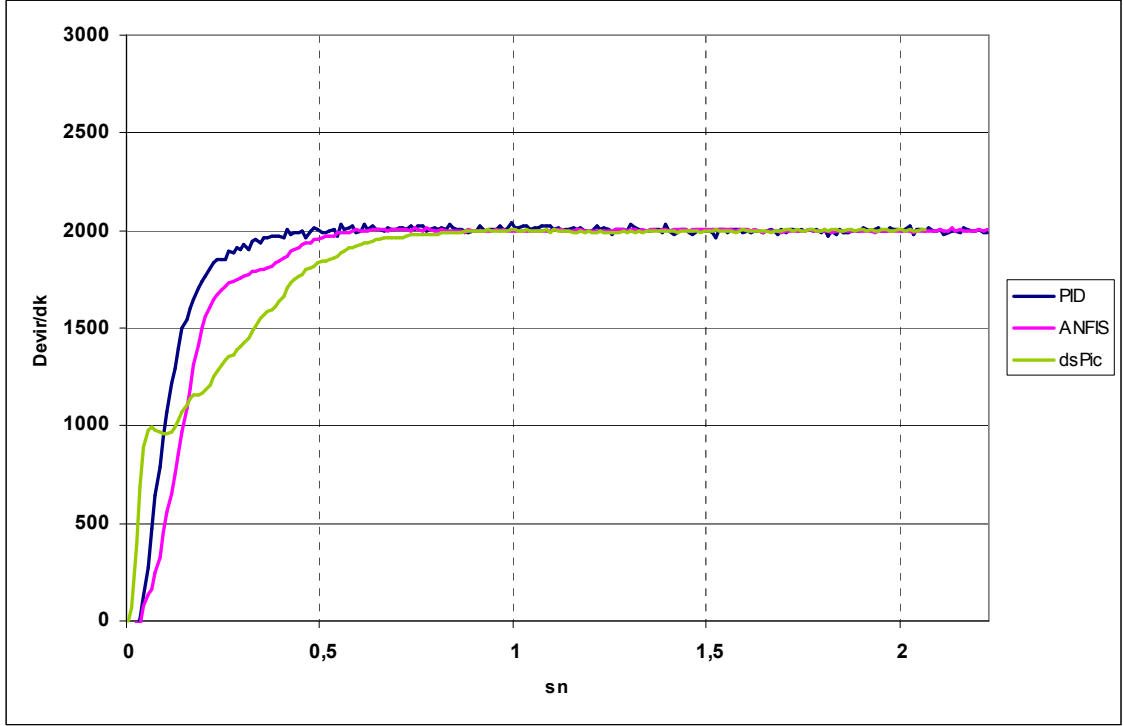
4.1.1. Birim Basamak Cevapları:



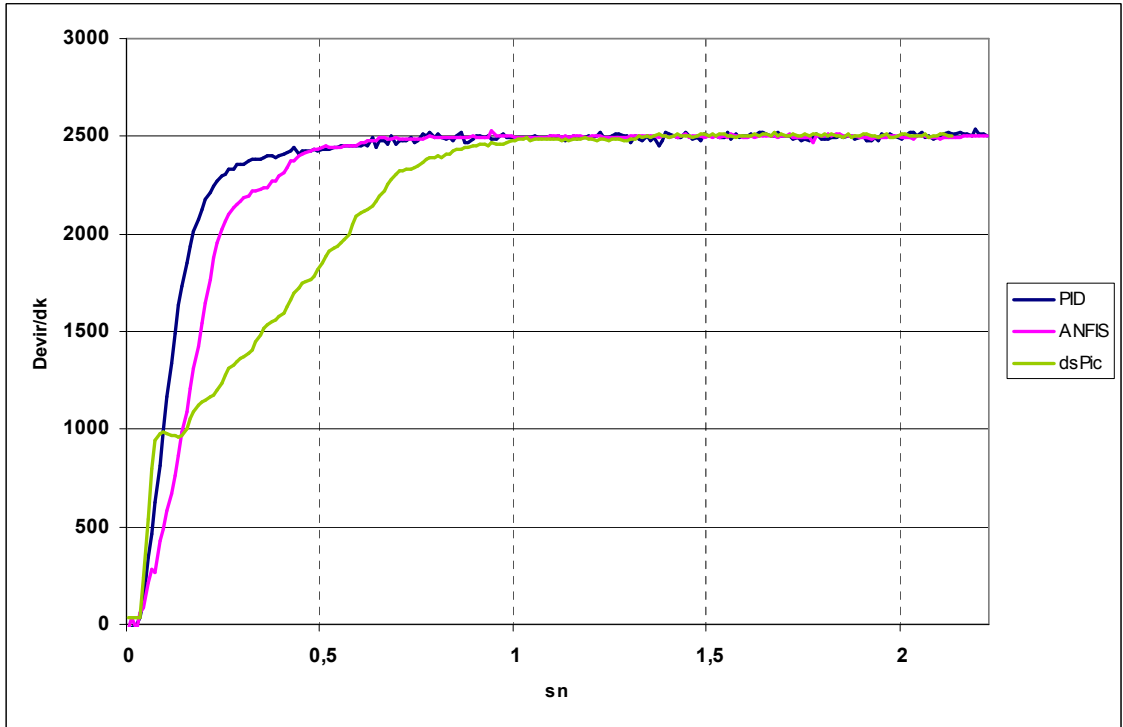
Şekil 4.1. 1000 Devir için ANFIS ve PID Birim Basamak Cevapları



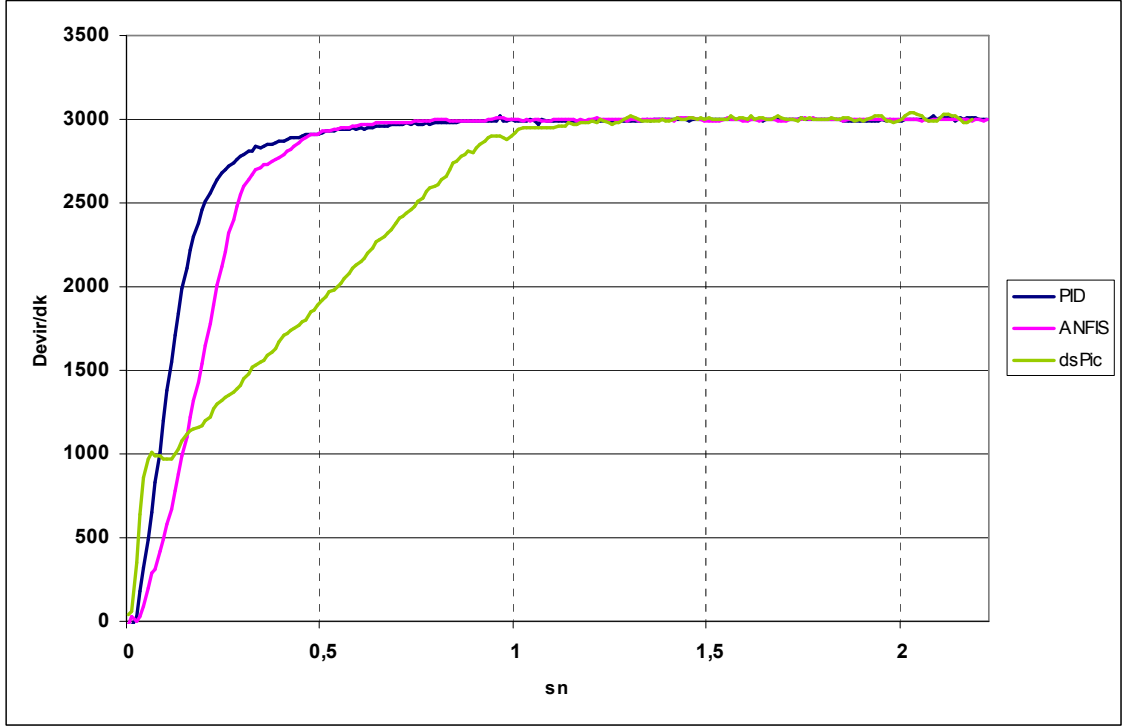
Şekil 4.2. 1500 Devir için ANFIS ve PID Birim Basamak Cevapları



Şekil 4.3. 2000 Devir için ANFIS ve PID Birim Basamak Cevapları

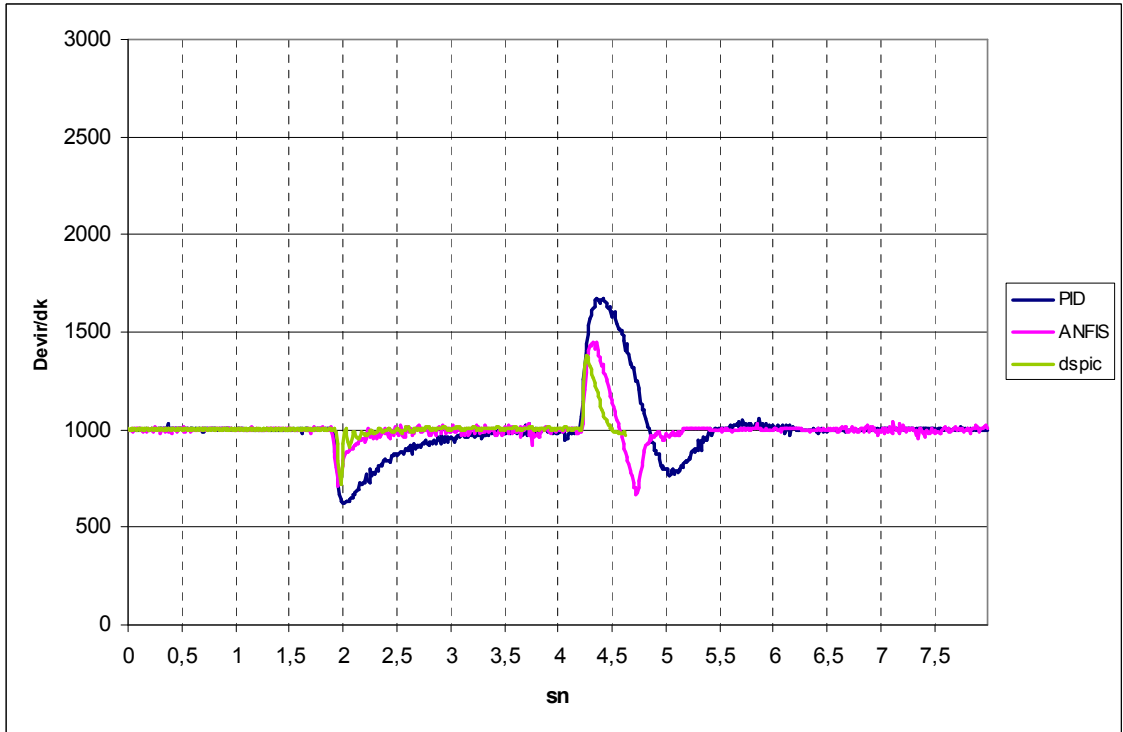


Şekil 4.4. 2500 Devir için ANFIS ve PID Birim Basamak Cevapları

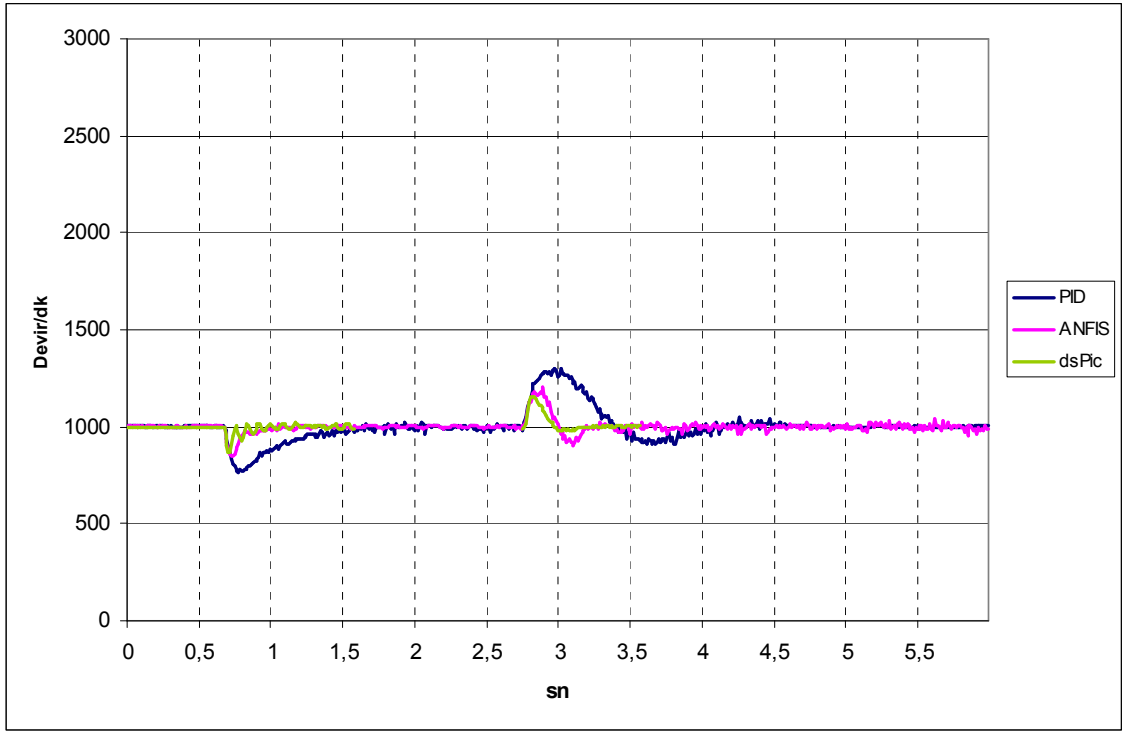


Şekil 4.5. 3000 Devir için ANFIS ve PID Birim Basamak Cevapları

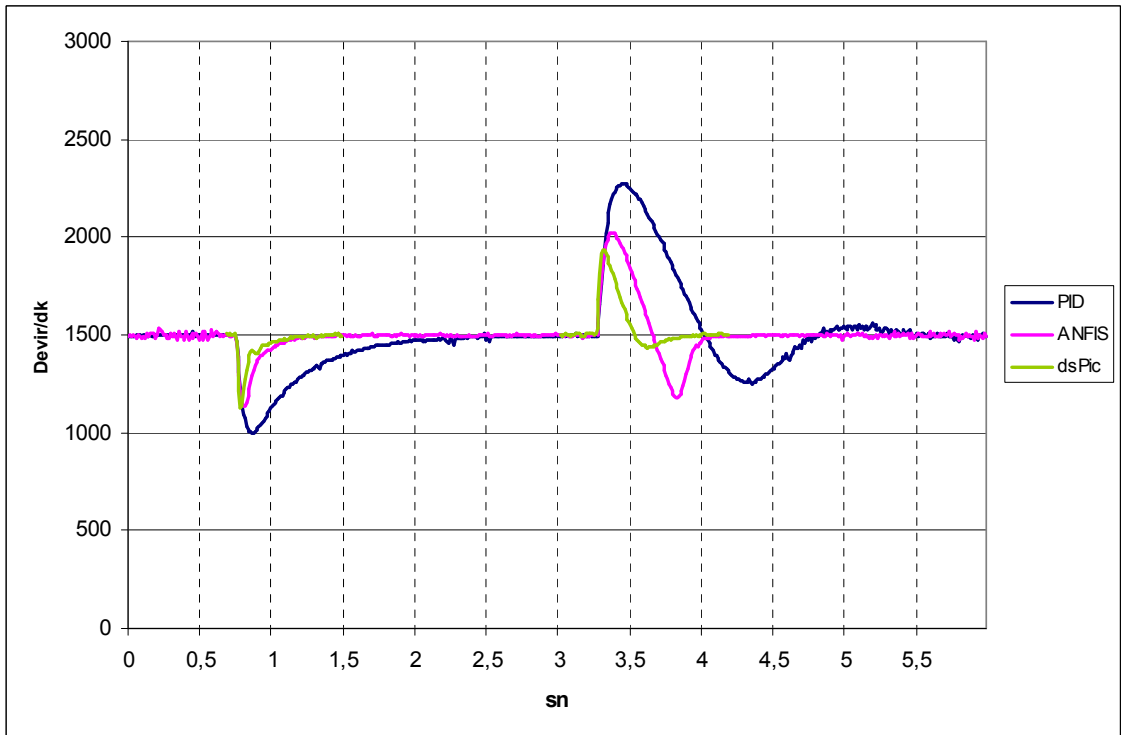
4.1.2. Yük Durumunda Sistem Cevapları:



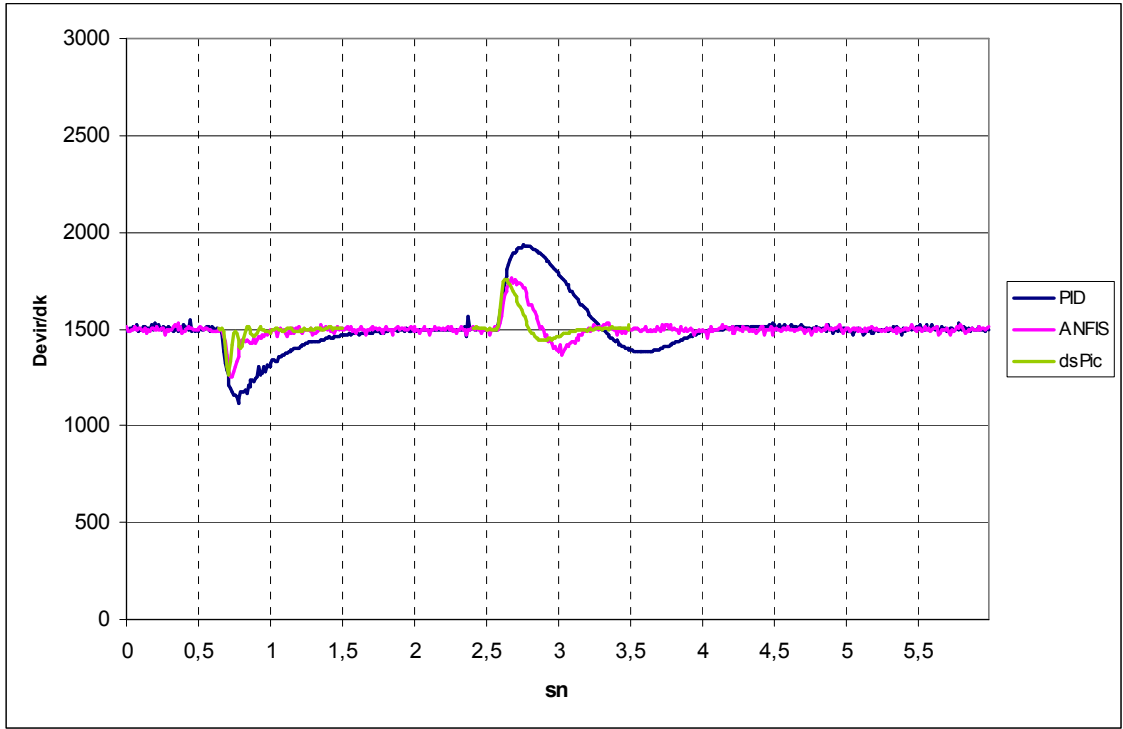
Şekil 4.6. 1000 Devirde 1Ω'luk Direnç ile Yükleme Durumu Cevabı



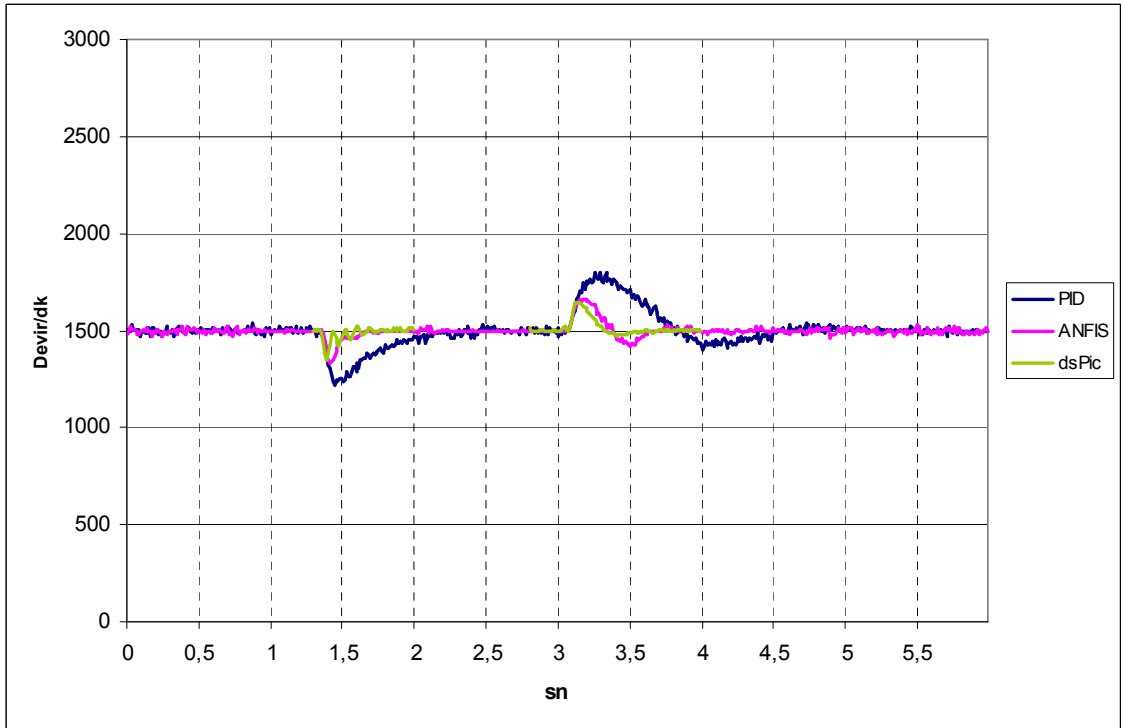
Şekil 4.7. 1000 Devirde 5Ω'luk Direnç ile Yükleme Durumu Cevabı



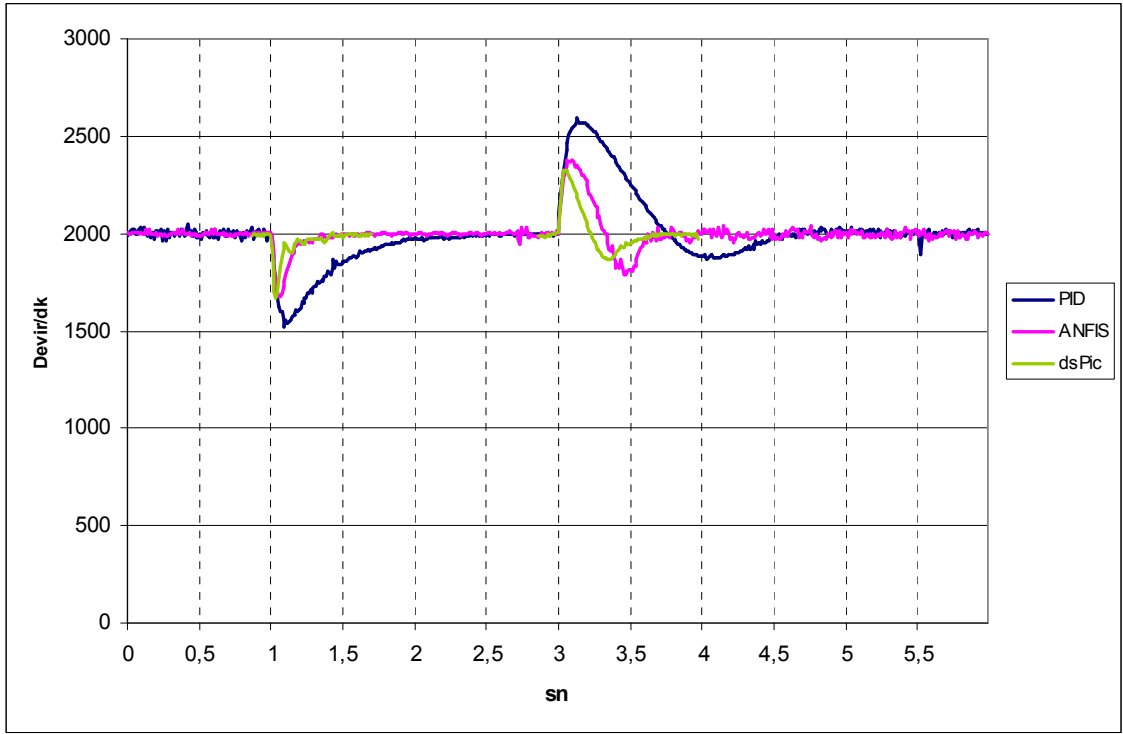
Şekil 4.8. 1500 Devirde 2Ω'luk Direnç ile Yükleme Durumu Cevabı



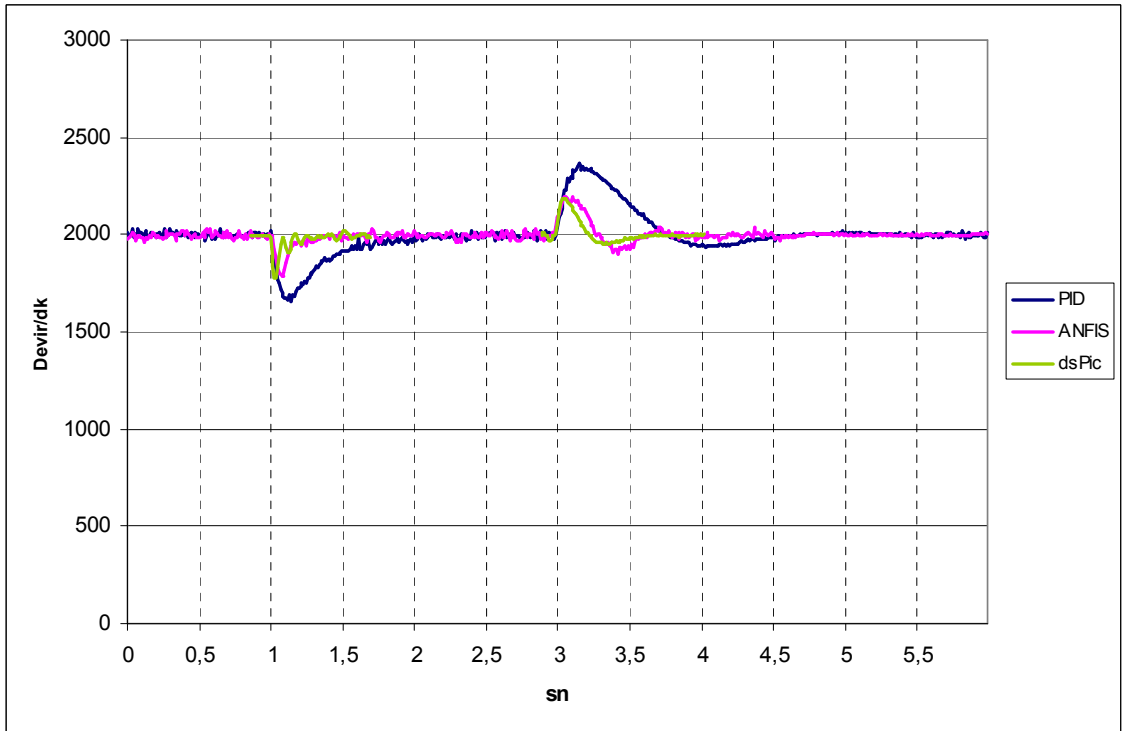
Şekil 4.9. 1500 Devirde 5Ω'luk Direnç ile Yükleme Durumu Cevabı



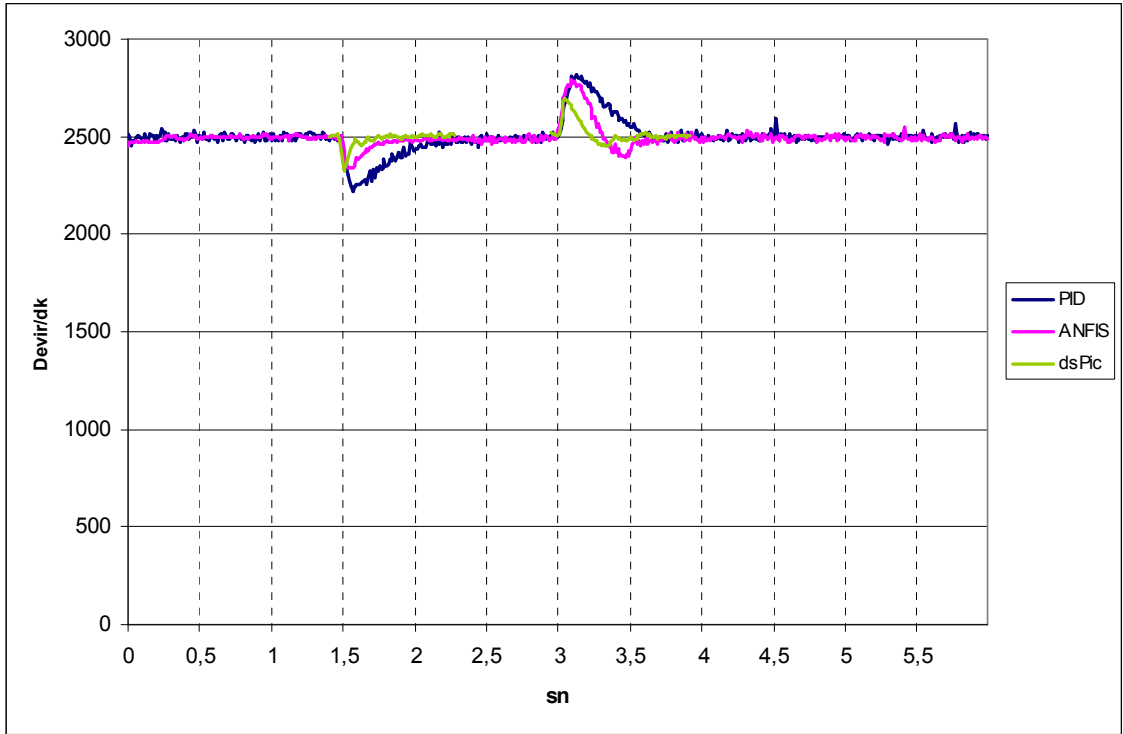
Şekil 4.10. 1500 Devirde 10Ω'luk Direnç ile Yükleme Durumu Cevabı



Şekil 4.11. 2000 Devirde 5Ω'luk Direnç ile Yükleme Durumu Cevabı



Şekil 4.12. 2000 Devirde 10Ω'luk Direnç ile Yükleme Durumu Cevabı



Şekil 4.13. 2500 Devirde 15Ω'luk Direnç ile Yükleme Durumu Cevabı

Yapılan üç uygulamada sistemlere ait katsayıların belirlenmesinde, birim basamak cevapları için sistemi yüzde aşımına götürmeyecek ve yük durumları için de en kısa zamanda cevap verecek şekilde düzenleme yapıldı.

dsPic30f2010 ile yapılan uygulama için sistem katsayıları ayarlanırken, birim basamak cevabından çok, yükleme durumlarında sistemin daha iyi cevap vermesi üzerinde duruldu. dsPic30f2010 için birim basamak cevabında 1000 devirden sonra lineer bir artışla sistem istenen değere ayarlandığı gözlenmektedir. ANFIS'e ait giriş ve çıkış katsayılarının değiştirilmesi için bilgisayar uygulama sonuçları ile benzer sonuçlara elde etmek mümkün olmuştur. Ancak bu durumlar için yük durumunda elde edilen sonuçlar, mevcut verilerdeki kadar iyi olmamış, bilgisayar uygulaması ile benzer sonuçlar elde edilmiştir.

PID ve bilgisayarda gerçekleştirilen ANFIS uygulaması birim basamak cevapları birbirlerine çok yakın olarak elde edilmiştir. ANFIS'te düşük devirlerde gözlenen yüzde aşımı gerçekleşmemiştir.

Sistemin yüklendiği durumlarda, her iki ANFIS uygulamasının da PID'den daha kısa bir sürede cevap verdiği gözlenmiştir. Ancak yük durumunun ortadan kaldığı durumlarda için bilgisayar ortamında gerçekleştirilen ANFIS uygulamasında çok daha fazla çökme gerçekleşmekte ancak buna rağmen PID'ye göre daha kısa bir sürede dengeye ulaşabilmektedir.

dsPic30f2010 ile gerekleřtirilen uygulama yk durumunda verdiđi cevaplar deđerlendirildiđinde diđer iki sisteme gre daha kısa srede tepki vererek, sistemi en kısa srede dengeye ulařtırdıđı gzlenmiřtir. Ancak dengeye ulařma noktasında sistemde kk salınımlar gzlenmiřtir. Sistemin yk durumu iin kısa srede verdiđi tepki gz nne alındıđında bu salınımların nemsiz olduđu deđerlendirilmiřtir.

dsPic30f2010 ile yapılan ve bilgisayar ortamında gerekleřtirilen ANFIS uygulamaları iin farklı katsayılar kullanılmıřtır. Bilgisayarda gerekleřtirilen uygulamadaki iin ANFIS katsayıları, dsPic30f2010 uygulamasında olduđu kadar arttırılamamıřtır. Katsayılarının arttırılması durumunda sistemde ok byk salınımlar gzlemlenmiřtir.

dsPic30f210 uygulamasında ANFIS'in hata giriři iin 6, hatanın deđerimi iin 5,5 ve ANFIS ıkıřı iin 9 katsayıları kullanılmıř, bilgisayar ile yapılan uygulama da ise hata ve hatanın deđerimi iin 5, ANFIS ıkıřı iin 0,95 katsayıları kullanılmıřtır. Grldđ gibi zellikle ıkıř katsayıları arasında ok byk bir fark vardır. Bu nedenle dsPic30f2010 uygulaması, bilgisayar uygulamasına gre yk durumunda daha iyi cevap vermiřtir.

5. SONUÇ VE ÖNERİLER:

Sonuç olarak öncelikli hedefimiz olan ANFIS ile DC Motor hız kontrolünü sayısal işaret işlemci kullanarak gerçekleştirdik.

Tez çalışması boyunca, ANFIS çalışma sistemi anlatıldı, ANFIS katsayılarının eğitimine yönelik algoritmalar çıkartıldı, Matlab Anfisedit programına alternatif olarak eğitim programı hazırlandı.

ANFIS yapısı ve eğitim sistemleri özümzendikten sonra uygulamaya ışık tutması bakımından Simulink programında simülasyon uygulaması gerçekleştirildi. Simülasyonda eğitim verilerinin elde edilmesi yöntemleri değerlendirilerek, PID denetleyici kullanılarak eğitim verilerinin elde edilmesine karar verildi. Simulink simülasyon sonuçları karşılaştırıldığında PID ve ANFIS arasında bir fark gözlemlenemedi. Bunun nedeni olarak eğitim verilerinin PID yapısı kullanılarak alınması olmasının yanında kontrolün çok kısa bir zamanda gerçekleşmesi sebebiyle farkın gözlemlenememiş olması değerlendirildi.

Simülasyon çalışmaları tamamlandıktan sonra uygulama için gerekli devre hazırlandı, eğitim verilerinin alınması için ADC kullanılarak veriler Matlab programı yardımı ile sistemden alındı. Sistemin Matlab programına bağlantısının sağlanmış olmasından yararlanılarak, Matlab ANFIS denetleyicisi modeli tasarlanarak DC motor hız kontrolü önce bilgisayarda gerçekleştirildi.

Bilgisayar ile yapılan uygulama sonuçların değerlendirilmesi amacı ile eğitim verilerinin alındığı PID ile de uygulamalar yapıldı. Sonuçlar karşılaştırıldığında birim basamak cevapları, PID için daha hızlı olmuş ancak yüksek devirlerde yüzde aşımı ANFIS'e göre biraz fazla olmuştur. Yük durumlarında ise ANFIS'in daha hızlı ve kısa zamanda sistemi kontrol ettiği gözlemlendi.

Bilgisayar uygulamasından sonra asıl hedef olan; sayısal işaret işlemci (dspic30f2010) ile DC Motor hız kontrolü gerçekleştirildi. Bilgisayar uygulaması ile karşılaştırdığında, öncelikle girişler ve çıkış için kullanılan katsayıların belirlenmesinde farklılık yaşandı. Bilgisayar uygulamasında katsayıların artırılması çok daha fazla salınımına sebep olurken, sayısal işaret işlemcide daha az salınım olmakta ve katsayılar artırılarak sistem performans artırılması sağlanmıştır. Bunun sebebi olarak bilgisayar ve sayısal işaret işlemcinin farklı frekanslarda çalışması ve ADC sistemlerinin farklı olması değerlendirildi.

Çalışmamızın başında microcontroller olarak pic16f877A kullanılarak DC motor hız kontrolü gerçekleştirildi. 16f877A maksimum çalışma frekansı olan 20 MHz'de

çalıştırılmasına rağmen istenen performans elde edilmedi. 16f877A ANFIS algoritmasını 9.45ms hesaplarken, dsPic30f2010 1.17ms'de hesaplayabilmektedir. dsPic30f2010 80 MHz'de çalıştırılmış olup, maksimum 120MHz'de çalışabilmektedir. ANFIS ile yapılacak uygulamalar için ANFIS algoritmasını hesaplayabilecek kapasitede microcontroler seçilmesi gerekir.

Sayısal işaret işlemci uygulaması sonucunda sisteme de az miktarda salınım olmaktadır. ANFIS ile sistemin ne kadar hızlı cevap verebileceğinin değerlendirilmesi amacıyla az miktarda salınıma göz yumarak katsayılar artırılmış ve sistem gözlemlenmiştir. Sistem, tatmin edecek kadar kısa bir sürede yükleme durumlarına tepki vererek kontrolü gerçekleştirmektedir. Mevcut salınımlar ve sistem performansının artırılması için farklı yöntemler kullanılabilir. Bunlar;

1. Eğitim verilerinin alınması için kullanılan referans giriş işareti değiştirilebilir.
2. Sistem kontrolü girişi olarak kullanılan hata ve hatanın değişimi yerine ya da ek olarak farklı girişlerde kullanılabilir. Örnek olarak hatanın büyüklük derecesi eklenebilir.
3. ANFIS yapısında kullanılmakta olan üçgen üyelik fonksiyonu yerine gauss fonksiyonu gibi değişik fonksiyonlar kullanılabilir.
4. Çıkış fonksiyonu $F(x_1, x_2, Z_1, Z_2, Z_3)=Z_1.X_1 + Z_2.X_2 + Z_3$ yerine daha farklı fonksiyonlar kullanılabilir. Örnek olarak X 'ler yerine X^2 'ler ya da $\sin(x)$ kullanılarak sistem performansı değerlendirilebilir.
5. Katsayıların eğitimi 700. denemeden daha fazla eğitilerek, daha az hata oranı elde edilebilir.
6. Girişler için 3'er üyelik fonksiyonu yerine daha farklı sayılardaki üyelik fonksiyonu adetleri için sistem kontrolü değerlendirilebilir.

Uygulamanın daha iyi olması için alternatif yöntemler geliştirmek mümkün olabilir. ANFIS eğitim programının algoritmalarının çıkarılması ve programının hazırlanmasındaki amaç, ANFIS yapısını tüm ayrıntıları ile iyi analiz edilmesi ve alternatif yöntemler geliştirilmesine ışık tutması olmuştur. Bunun yanın eğitim verilerinin elde edilmesi ve eğitimin yapılması için bilgisayar kullanılmıştır. ANFIS eğitim algoritmasının bilinmesi, eğitimin sayısal işaret işlemcide gerçekleştirilebilmesi yolunda bize fayda sağlayacaktır.

Uygulamanın yapılmasında öncelikli olan konunun ANFIS olması nedeni ile kontrolü basit olan DC motor uygulaması tercih edilmiştir. ANFIS yapısı eğitilebilir olması nedeni ile yapısında ufak değişiklikler yapılarak farklı sistemlerin kontrollerinde

kullanılabilirler. Genellikle kontrol edilmesi zor olan non-lineer sistemlerin kontrollerinde tasarım kolaylığı sağlayabilirler. Ancak ANFIS'in uygulanabilirliğinde karşılaşılan en zor bölüm eğitim verilerinin ne olacakları problemiştir.

6. KAYNAKLAR

- [1] Jang, J.-S.R. ANFIS, Adaptive Network-based Fuzzy Inference System, (1993)
- [2] Hung T.Nguyen, N.R.Prasad, C.L.Walker E.A.Walker, A First Course in Fuzzy and Neural Control,A CRC Press Company (2002).
- [3] M.Önder Efe ve Okyay Kaynak, Yapay Sinir Ağları ve Uygulamaları, Boğaziçi Üniversitesi, (2000)
- [4] Ranadhir Ghosh, A Novel Hybrid Learning Algorithm for Artificial Neural Networks, School of Information Technology Faculty of Engineering and Information Tecnology Griffith University, (2002)
- [5] Piero P. Bonissone GE CRD,Schenectady, NY USA: Adaptive Neural Fuzzy Inference Systems (ANFIS): Analysis and Applications
- [6] Heikki Koivo, Adaptive Neuro-Fuzzy Inference System, (2006)
- [7] Avdhesh Sharma and M.L.Kothari, Intelligent Dual Input Power System Stabilizers For Multi-Machine System, (2002)
- [8] MATLAB 7.5.0 Guide Fuzzy Logic Toolbox, Anfis and the ANFIS Editor GUI
- [9] MATLAB 7.5.0 Help
- [10] dsPic30f2010 DataSheet
- [11] Dr. Ünal Dikmen, Matlab Programlama Diline Giriş, JFM224 Sayısal Analiz ve Programlama III
- [12] LM2907 DataSheet
- [13] Engin Avcı, Z.Hakan Akpolat, Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi ile Doğru Akım Motorlarının Hız Denetimi
- [14] E. E. El-kholy, A. M. Dabroom, and Adel El-kholy, Principle Of Adaptive Fuzzy Logic Controllers For DC Motor Drives
- [15] Mehmet Akar, İsmail Temiz, Motion Controller Design For The Speed Control Of DC Servo Motor (2007)
- [16] Ercan Öztemel, Yapay Sinir Ağları (2006)
- [17] A. Halvaei Niasar, H. Moghbelli and A. Vahedi, Adaptive Neuro-Fuzzy Control with Fuzzy Supervisory Learning Algorithm for Speed Regulation of 4-Switch Inverter Brushless DC Machines
- [18] Pedro Ponce, Rodrigo Blancas, Cristian Tena and Manuel Rana, A Novel DC Drive Based On Fuzzy Logic Inverse Plant Model Optimised by ANFIS.
- [19] S.Ushakumari, Adaptive Neuro-Fuzzy Controller for Improved Performance of a Permanent Magnet Brushless DC Motor (2001).

- [20] Ayhan Gün, The Position Control Of The DC Machine by PID Algoritm and Training with Adaptive Neuro Fuzzy Inference System (2007)
- [21] M.Ali Akcayol, Application of adaptive neuro-fuzzy controller for SRM (2004)
- [22] Ferhat Daldaban, Nurettin Ustkoyuncu ve Kerim Guney, Phase Inductance Estimation for Switched Reluctance Motor Using Adaptive Neuro-Fuzzy Inference System (2006)
- [23] Kosko, B., 1991, Neural Networks and Fuzzy Systems, A Dynamical Systems Approach, Englewood Ciffs., NJ: Prentice Hall.

ÖZGEÇMİŞ

Neşet BAYSAL, 1981 yılında İZMİR'de doğdu. İlk ve orta öğrenimini İZMİR'de tamamladı. Yüksek öğrenimini ise Hava Harp Okulu Elektronik Mühendisliği Bölümünde, 2003 yılında tamamladı. 2006 yılında İnönü Üniversitesi Elektrik Elektronik Mühendisliği Bölümünde yüksek lisans öğrenimine başladı. Halen Hava Kuvvetleri Komutanlığı personeli olarak görev yapmaktadır.