

**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**MİKROİŞLEMCİ KULLANILARAK NEURO-FUZZY PROSES KONTROL  
UYGULAMASI**

Elektrik Müh. İbrahim ALIŞKAN

FBE Elektrik Mühendisliği Anabilim Dalı Kontrol ve Otomasyon Programında Hazırlanan

**YÜKSEK LİSANS TEZİ**

**Tez Danışmanı : Yrd. Doç. Dr. Kayhan GÜLEZ**

**İSTANBUL, 2006**

# İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ .....	v
KISALTMA LİSTESİ .....	vi
ŞEKİL LİSTESİ .....	vii
ÇİZELGE LİSTESİ .....	ix
ÖNSÖZ.....	x
ÖZET .....	xi
ABSTRACT .....	xii
1. GİRİŞ.....	1
2. KONTROL SİSTEMLERİ.....	3
2.1 Kontrol Teorisi (Tarihi Geçmiş).....	3
2.2 Kontrol Sistemlerine Genel Bakış .....	4
2.2.1 Kontrol Sistemlerinin Sınıflandırılması.....	7
2.2.1.1 Geri Beslemeli Kontrol.....	8
2.2.2 Kontrol Sistemlerinin Karakteristiği.....	14
3. YAPAY SİNİR AĞLARI ile KONTROLÜN TEMEL ELEMANLARI.....	17
3.1 Yapay Sinir Hücresi.....	17
3.2 Yapay Sinir Ağı Mimarileri.....	19
3.3 Yapay Sinir Ağının Eğitilmesi .....	20
3.3.1 Öğrenme Stratejileri.....	21
3.3.2 Öğrenme Kuralları .....	21
3.4 Perceptron ve MLP (Multi Layer Perceptron).....	23
3.4.1 Perceptron (Basit Algılayıcı Modeli).....	23
3.4.2 MLP (çok Katmanlı Algılayıcı).....	25
3.5 Yazılımsal ve Donanımsal Yapay Sinir Ağı Tasarım Elemanları .....	32
3.5.1 Analog Yapay Sinir Ağı Donanımları .....	34
3.5.2 Dijital Yapay Sinir Ağı Donanımları.....	35
3.5.3 Hibrid Yapay Sinir Ağı Tasarımları .....	36
3.5.4 Yapay Sinir Ağı Donanımlarının Performanslarının Ölçülmesi ve Karşılaştırılması .....	37
3.6 Yapay Siniri Ağlarının Genel Kullanım Alanları.....	37
4. BULANIK MANTIK KONTROLÜN TEMELLERİ .....	42
4.1 Tarihi Geçmiş .....	43
4.2 Bulanık Mantık Nedir? .....	44

4.3	Bulanık Küme Teorisi.....	45
4.3.1	Üyelik Fonksiyonları .....	46
4.4	Dilsel Niceleyiciler .....	51
4.5	Bulanık Mantık Kontrol Sistemleri .....	52
4.5.1	Bulanıklaştırma.....	55
4.5.2	Bulanık Çıkarım.....	55
4.5.3	Netleştirme.....	55
4.5.4	Bulanık Kontrol Algoritması .....	57
4.6	Kontrol Uygulamalarında Bulanık Mantık.....	60
4.7	Klasik Kontrol ile Bulanık Mantık Tabanlı Kontrolün Karşılaştırılması.....	62
5.	SİNİRSEL BULANIK MANTIK SİSTEMLER .....	65
5.1	Giriş .....	65
5.2	Sinirsel-Bulanık Mantık Ağ Yapıları.....	67
5.3	Sinirsel-Bulanık Mantık Ağ Kuramı .....	69
5.4	Sinirsel-Bulanık Mantık Ağ Yapılarında Çıkarım Yöntemleri .....	70
5.5	Bulanık Kuralların Öğrenilmesi .....	71
5.6	Üyelik İşlevlerinin Öğrenilmesi .....	71
6.	pH NÖTRALİZASYON PROSESİ.....	73
6.1	Proses ve Proses Kontrol .....	73
6.2	pH Kavramı ve Nötralizasyon İşlemi .....	75
6.2.1	Asitler ve Bazlar .....	75
6.2.2	pH Ölçeği.....	77
6.2.3	Suyun Kendi Kendine İyonlaşması .....	78
6.2.4	Asitlerin ve Bazların Kuvveti .....	79
6.2.5	Zayıf Asitler ve Zayıf Bazlar.....	80
6.2.6	Asit-Baz Titrasyonları .....	81
6.2.6.1	Kuvvetli Asit-Kuvvetli Baz Titrasyonları .....	81
6.2.6.2	Zayıf Asit-Kuvvetli Baz Titrasyonları.....	82
6.2.7	Nötralizasyon İşlemi .....	84
6.2.7.1	Atık Sularda pH Nötralizasyon Prosesi .....	84
6.2.7.2	Prosesin Matematiksel Modelinin Çıkarımı .....	85
7.	BİLGİSAYAR ORTAMINDA SİSTEMİN SİMÜLASYONU .....	88
7.1	Sistemin Matematiksel Modelinin Doğruluğu.....	88
7.2	pH Nötralizasyon Prosesine PID Kontrolör Tasarımı ve Uygulanması .....	89
7.2.1	pH Nötralizasyon Prosesine PID Kontrolör Tasarımı .....	90
7.2.2	pH Nötralizasyon Prosesine PID Kontrolör Uygulanması .....	91
7.3	pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Tasarımı ve Uygulanması ...	92
7.3.1	YSA Kullanım Nedenleri ve YSA nın Genel Durumu.....	92
7.3.1.1	Yapay Sinir Ağlarının Kullanım Nedenleri.....	92
7.3.1.2	YSA ile Klasik Yazılımlar Arasındaki Farklar.....	93
7.3.1.3	MLP ve Fuzzy Kullanım Nedenleri?.....	94
7.3.2	pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Tasarımı .....	95
7.3.2.1	YSA Tabanlı Modelin Tasarımı .....	96
7.3.2.2	Bulanık Tabanlı Kontrolörün Tasarımı .....	97
7.3.3	pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Uygulanması .....	101
8.	YAPILAN UYGULAMA .....	103

8.1	Mikrokontrolör ile pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Tasarımı ve Uygulanması .....	103
8.1.1	Kullanılan Mikrokontrolörün Teknik Özellikleri .....	104
8.1.2	Mikrokontrolöre Nöro-Fuzzy Kontrolörün Uyarlanması .....	108
8.1.3	pH Nötralizasyon Prosesine Ait Matematiksel Modelin Mikrokontrolöre Uyarlanması .....	112
8.2	Uygulama Sonuçlarının Elde Edilmesi.....	113
9.	SONUÇLAR ve ÖNERİLER .....	114
	KAYNAKLAR.....	118
	ÖZGEÇMİŞ.....	120

## SİMGE LİSTESİ

$A_i$	YSA daki her bir girişe ait ağırlık değeri.
$\alpha$	1/6...1/20 aralığındaki herhangi bir sabit değer.
$\alpha_i$	BM de i. kuralın üyelik derecesi.
$\beta^a$	YSA da ara katman eşik değer ağırlıkları.
$\beta^c$	ÇKA da çıktı katmanındaki proses elemanına ait eşik değer ağırlığı.
$\beta_j$	YSA da ara katmanda bulunan j. elemana bağlanan eşik değer elemanının ağırlığıdır.
$C$	Referans değeri aşma miktarı.
$C_a$	Giriş asit derişimi (mol/l).
$C_b$	Giriş baz derişimi (mol/l).
$C_{max}$	Maksimum üst aşım değeri.
$e$	Hata sinyali (değeri).
$E(t)$	Hatanın zamana bağlı değışimi.
$E_m$	YSA da çıktı katmanında m. proses elemanı için oluşan hata.
$F_a$	Asit akış hızı (l/dk).
$F_b$	Baz akış hızı (l/dk).
$G_c$	İletim fonksiyonu.
$G_i$	YSA ya gelen girişler.
$j$	Hatanın karesinin veya hatanın karesinin zamanla çarpımının zamana bağlı integrali.
$K_a$	Asitin sudaki çözünürlük sabiti.
$K_c$	Oransal kontrolörde çarpan.
$K_p$	Ziegler-Nichols kurallarına uygun olarak atanan PID kontrolörde oransal sabit.
$K_{su}$	Suyun iyonlar çarpımı sabiti.
$P_b$	Orandal bant geçiren yapının değeri.
$pH$	Asitlik derecesi.
$P_i$	BM-YSA tabanlı yapıda çıkış ürünü.
$pOH$	Bazlık derecesi.
$pK_a$	$-\log K_a$ değeridir.
$TH$	YSA da çıktı katmanında oluşan toplam hata.
$T_{omax}$	Çıkışın $C_{max}$ a ulaşması için geçen süre.
$T_r$	Sistem çıkışının, referans değerin 10%-90% aralığındaki değışimine karşılık gelen süre.
$T_s$	Sistem çıkışının, referans değerin $\pm 2\%$ lik bandına oturmasına kadar geçen süre.
$\tau$	Zaman sabiti.
$u$	BM kontrolörün çıkış değeri.
$V$	Reaksiyon tankının hacmi.
$Z$	BM kontrolde netleştirme aşamasındaki ağırlık merkezi metodu ile hesaplanan merkez değeri.
$z_j$	BM kontrolde netleştirme aşamasındaki kuantalama seviyesinde kontrol çıkışının miktarı.
$x_a$	Reaksiyon tankı içindeki asit derişimi.
$x_b$	Reaksiyon tankı içindeki baz derişimi.
$w$	Açısal hız.
$\Delta A^i$	ÇKA da ara katman elemanlarının ağırlık değışim değeri.
$\delta_m$	ÇKA da m. çıktı ünitesinin ahat değeri.
$\lambda$	YSA da öğrenme katsayısı.
$q$	BM kontrolde netleştirme aşamasındaki kuantalama seviyesi.

## KISALTMA LİSTESİ

AO	Air to Open
ART	Adaptif Rezonans Teorisi
BM	Bulanık Mantık
ÇKA	Çok Katmanlı Algılayıcı
DSP	Dijital İşaret İşleyici
EEPROM	Elektriksel Olarak Yazılabilir ve Okunabilir Sabit Hafıza
FIS	Bulanık Mantık Çıkarım Sistemi
FLC	Bulanık Mantık Kontrolör
GA	Genetik Algoritma
GP32	Freescale Firmasına Ait MC68HC08GP32 8 Bitlik Mikrodenetleyici
ISE	Hatanın Karesinin Integrali
ITEA	Mutlak Hatanın Zaman Çarpımlı Integrali
ITSE	Hatanın Karesinin Zaman Çarpımlı Integrali
MLP	Çok Katmanlı Perseptron
P	Oransal Kontrolör
PB	Oransal Band
PBNN	Olasılık Tabanlı Yapay Sinir Ağı
PD	Oransal Türev Alan Kontrolör
PI	Oransal Integral Alan Kontrolör
PID	Oransal Integral ve Türev Alan Kontrolör
PLC	Programlanabilir Lojik Kontroller
PV	Proses Çıkış Değeri
RAM	Geçici Bellek Ünitesi
RBF	Radyal Tabanlı Fonksiyon
RBN	Radyal Tabanlı Yapay Sinir Ağı
SCI	Seri Haberleşme Arayüzü
SIMD	Tek Komut ve Çoklu Veri
SP	Referans Değeri
YSA	Yapay Sinir Ağı
XOR	Exlusive OR Kapısı

## ŞEKİL LİSTESİ

Sayfa

Şekil 2.1	James Watts'ın 1788'de buhar motorunun hız kontrolü için geliştirdiği "fly-ball governor" .....	4
Şekil 2.2	Kapalı çevrim kontrol sistemi blok diyagramı (Sarıoğlu, 1995) .....	8
Şekil 2.3	Geribeslemeli sıcaklık kontrolünün proses üzerindeki gösterimi (Akgün,2003) .....	9
Şekil 2.4	Geribeslemeli sıcaklık kontrolünün blok diyagramı (Akgün,2003) .....	9
Şekil 2.5	PID kontrolörün parametrelerinin atanmasında kullanılan, sistem birim basamak referans işaretin uygulanışı (a), sistemin bu işarete vermiş olduğu cevap (b).....	13
Şekil 2.6	Kapalı çevrim kontrol sistemi performans parametreleri .....	15
Şekil 3.1	Yapay sinir hücresinin yapısı .....	18
Şekil 3.2	Yapay sinir ağları sınıflandırıcıları .....	20
Şekil 3.3	Bir basit algılayıcı yapısı (Öztemel, 2003) .....	23
Şekil 3.4	Eşik değerinin aktivasyon fonksiyonunun konumuna etkisi (Öztemel, 2003) .....	24
Şekil 3.5	ÇKA modeli .....	26
Şekil 3.6	Temel analog yapay sinir ağı yongası (Öztemel, 2003) .....	35
Şekil 3.7	Dijital yapay sinir ağı yonga yapısı (Öztemel, 2003) .....	35
Şekil 4.1	Bulanık kümeler; a) Yaşın üç gruba ayrıldığı kümeler, b) Diğer bir kümelendirme yapısı (Elmas, 2003) .....	45
Şekil 4.2	Tablo 4.1 de tanımlana bulanık kümelerin eğrisi.....	46
Şekil 4.3	Çeşitli üyelik fonksiyonları.....	47
Şekil 4.4	Üçgen üyelik fonksiyonu .....	48
Şekil 4.5	Yamuk üyelik fonksiyonu.....	49
Şekil 4.6	Çan üyelik fonksiyonu .....	49
Şekil 4.7	Monolitik üyelik fonksiyonu.....	50
Şekil 4.8	Monotonik üyelik fonksiyonu.....	51
Şekil 4.9	Dilsel niceleyicilerin ifade edilmesi (Dalcı, 2000) .....	52
Şekil 4.10	Bir bulanık kontrolörün yapısı (Özbay, 1995).....	53
Şekil 4.11	Bir bulanık kontrol sisteminin blok diyagramı (Özbay, 1995).....	58
Şekil 4.12	a) Hata, hatanın değişimi ve kontrol sinyali için seçilen üyelik fonksiyonları b)Grafik olarak gösterilen netleştirme .....	59
Şekil 5.1	Sinirsel bulanık mantık sisteminin birinci yapısı (Elmas, 2003) .....	68
Şekil 5.2	Sinirsel bulanık mantık sisteminin ikinci yapısı (Elmas, 2003) .....	68
Şekil 5.3	Berenji nin ARIC mimarisi (Elmas, 2003) .....	69
Şekil 5.4	Basit bir sinir ağı.....	70
Şekil 6.1	Temel elemanlarla sinyalleri içeren proses kontrol diyagramı (Johnson, 2003).....	74
Şekil 6.2	Kuvvetli asitin kuvvetli baz ile titrasyonu (25.00 ml 0.100 M HCl ve 0.100 M NaOH) (Petrucci, 2002).....	81
Şekil 6.3	Zayıf asitin kuvvetli baz ile titrasyonu (0.100 M NaOH ve 25.00 ml 0.100 M HC <sub>2</sub> H <sub>3</sub> O <sub>2</sub> ) (Petrucci, 2002) .....	83
Şekil 6.4	Zayıf asit kuvvetli baz titrasyon eğrisini oluşturulması (Petrucci, 2002) .....	84
Şekil 6.5	Aç-kapa tipi karışımli nötralizasyon (Akkaya, 2001).....	84
Şekil 6.6	İki giriş akımlı gerikarışımli reaktör (Söğüt, 2004).....	85
Şekil 6.7	Titrasyon eğrisinin kazançlara göre dilimlenmesi .....	87
Şekil 7.1	Simulasyon ortamında modeline denemesine ilişkin yapı .....	88

Şekil 7.2	a) Gerçek bir pH nötralizasyon prosesinden elde edilen sonuçlar, b) Matematiksel modeli içeren simulasyon sisteminden elde edilen çıkış işaretinin grafiksel gösterimi.....	88
Şekil 7.3	PID kontrolörün uygulandığı simulasyon devresi .....	91
Şekil 7.4	PID kontrolöre ait parametrelerin sisteme girilmesi.....	91
Şekil 7.5	PID kontrolörün uygulandığı pH prosesinin çıkış değerlerinin oluşturduğu grafik .....	92
Şekil 7.6	YSA-Bulanık mantık tabanlı kontrol sisteminin tamamı.....	95
Şekil 7.7	İki bölümden oluşan YSA yapısı .....	96
Şekil 7.8	Bulanık mantık denetleyici ve giriş-çıkış parametreleri .....	98
Şekil 7.9	Referans giriş değişkeni için üyelik fonksiyonları.....	100
Şekil 7.10	“e” giriş değişkeni için üyelik fonksiyonları .....	100
Şekil 7.11	$F_b$ çıkış değişkeni için üyelik fonksiyonları.....	100
Şekil 7.12	YSA ve Bulanık mantık tabanlı kontrol sisteminin prosese uygulandığı simulasyon yapısı.....	102
Şekil 8.1	Projede kullanılacak olan giriş-çıkış portları ve portların durumları.....	105
Şekil 8.2	Bulanık mantık kontrolöre ait modelin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı .....	109
Şekil 8.3	YSA1 modelinin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı ( $F_a$ , $C_a$ , $C_b$ , $X_a(t-1)$ ve $X_b(t-1)$ gerçek girişler değildir).....	110
Şekil 8.4	YSA2 modelinin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı.....	111
Şekil 8.5	Prosesine ait modelin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı ( $F_a$ , $C_a$ , $C_b$ , $X_a(t-1)$ ve $X_b(t-1)$ gerçek girişler olmayıp, sabit değerleri bellekte yer almaktadır) .....	112

## ÇİZELGE LİSTESİ

	Sayfa
Çizelge 4.1	Bulanık mantık denetimin endüstriyel uygulamaları (Elmas, 2003) ..... 61
Çizelge 4.2	Bulanık mantık kontrol ile klasik kontrolün karşılaştırılması..... 64
Çizelge 6.1	Asitlerin çözünme sabitleri (Mahan, 1989) ..... 80
Çizelge 6.2	Lineer çalışma bölgeleri ve her bir bölgeye ait doğru denklemi ( $x$ : Asit/Baz) ..... 86
Çizelge 7.1	Kontrolöre ait bulanık kural çizelgesi..... 102

## **ÖNSÖZ**

Bu çalışmayı gerçekleştirmem sırasındaki yardımlarından dolayı, Danışman Hocam Yrd. Doç. Dr. Kayhan GÜLEZ e, Anabilim Dalı Başkanımız Prof. Dr. Galip CANSEVER Hocama, sabırla bana katlanan çalışma arkadaşlarıma ve aileme teşekkürü bir borç bilirim.

## ÖZET

Bilindiği üzere prosesler, belli girdilerin yine belli kurallar doğrultusunda işlenerek istenilen çıktılar elde edildiği yapılardır. Doğada kendiliğinden gerçekleşen prosesler de olduğu gibi bizim istediğimiz proseslerin gerçekleşmesi için ise, genelde zorlamalı yapılar gerekli olmaktadır. Bu durum ise, zorlamalı işlemin gerçekleşebilmesi için prosesin adım adım izlenerek kontrol altında tutulmasını mecbur kılmaktadır. İşte bu çalışmada da, özellikle atık suların arıtılıp istenilen PH değerinde doğaya geri verilebilmesi için gerekli olan nötrleştirme prosesine bir örnek teşkil edecek olan, asetik asidin (asetik asit) sodyum hidroksit (baz) ile nötralizasyonu prosesinin kontrol altına alınması için gerekli olan kontrolörün geliştirilmesine çalışılmıştır.

Aslında kontrol sistemleri denildiğinde akla ilk gelen yapılar; P, PI, PD veya PID kontrolör yapılarıdır. Ancak PH nötralizasyon prosesi gibi nonlineer (lineer olmayan) proseslerin kontrolünde geleneksel kontrol sistemlerinin pek de başarılı olmadıkları görülmüştür.

Son yıllarda yapay zeka ürünü olan kontrol sistemlerinden, bulanık mantık kontrol ve yapay sinir ağları ile geliştirilen kontrol yapıları ön plana çıkmaktadır. Ancak bu iki kontrol yapısının ayrı ayrı kullanılması yerine, her iki kontrolör tipinin de yapılarından kaynaklanan üstünlüklerinin birleştirilmesi ile geliştirilen kontrolörlerin başarımının çok daha yüksek olduğunu bu çalışmada görmek mümkün olacaktır. Özellikle de lineer olmayan sistemlerin kontrolünde, geliştirilen bu yapılar geleneksel kontrolör yapılarına göre çok daha iyi sonuçlar vermişlerdir.

Öte yandan geliştirilen bu yeni kontrolörün bilgisayar bağımlı kullanım ortamından çıkarılabilmesi için ise, mikrokontrolörlerin kullanımı ile sistemin çalışma ortamında kendi başına çalışabilmesine olanak sağlanmış olacaktır. Bu sayede de, yapay zeka ürünü olan bu kontrolör yapılarının bilgisayara olan bağımlılığı ortadan kaldırılarak, kullanım kolaylığı ile maliyet düşürümü sağlanmış olacaktır.

Yukarıdaki açıklamalarda göstermektedir ki, bu çalışma iki ana bölüme ayrılabilir. Birinci bölüm, bilgisayar ortamında yapılan Matlab simülasyonlarını içeren ve gerçekleştirilecek olan donanım tabanlı sisteme model sağlayacak olan ana kısımdır. Burada sistemin yapılabirliği ile donanım ortamına aktarım işlemi sırasında karşılaşılabilecek olan kilit noktaların tespiti amaçlanmıştır. Ayrıca geleneksel kontrol teknikleri ile kıyaslama için de bir fikir sağlamış olacaktır. Zira bu sistemlerin Matlab'da tasarımı ve simülasyonu oldukça kolay bir şekilde gerçekleştirilebilmektedir. İkinci ana bölümü ise, donanım tabanlı kontrol sistemi ile yine prosesin matematiksel modelini içeren donanımsal yapılar oluşturmaktadır. Bu yapının başarımı için de, ilk bölümde yer alan simülasyonlar ile kıyaslamalar yapılmış ve tatmin edici sonuçlara ulaşılmıştır.

Sonuç olarak; bulanık mantık-yapay sinir ağı tabanlı kontrol sistemleri, bilgisayar bağımlılığından kurtarılarak, kendi başına çalışabilen mikrokontrolör tabanlı sistemlere aktarılabilmiştir. Bu sayede de, bu tip kontrol sistemlerinin yaygınlaşmasına bir örnek teşkil ederek katkı sağlanmış olmaktadır. Öte yandan geleneksel kontrol tekniklerinin, lineer olmayan sistemleri kontrol etmede, yapay zeka ürünü olan kontrol tekniklerine nazaran son derece başarısız kaldıkları bu çalışmada bir defa daha görülmüştür.

**Anahtar kelimeler:** Yapay zeka, Bulanık mantık tabanlı kontrol, Yapay sinir ağları, PH nötralizasyon prosesi, Asit (asetik asit), Baz (sodyum hidroksit), Mikrokontrolör.

## ABSTRACT

Certain input arguments are changed by certain operations in processes. There are some natural processes, which operates by stand alone, but some processes, which is wanted to operates by us, needs to a controller system and forced power for operating. These types processes need to a controller system, which controls the operations of process step by step for operating truely. In this study, we tried to develop a controller system for taking under the controll to acedic acid (acid) and sodium hydroxide (base) pH neutralization process, which can be tought a sample of waste water PH neutralization process. In this way, waste water will be given to nature back again at neutral position.

Actually, well known controller systems, which are called conventional controller systems, are P, PI, PD and PID. But conventional controller systems are not successful enough for controlling to non-linear systems, which likes PH neutralization process.

In a few last year, fuzzy logic controllers and neural network basis controllers, which are built on artificial intellegence basis, are placed first in importance. Instead of using these types controllers lonely, dominant aspects of these controllers can be combined in a controller body, that can be seen in this project. According to traditional controller systems, new type ones have been more succesfull for controlling to non-linear systems.

On the other way, we used to microcontrollers, that one of them is running to like a cell of this controller, for transferring neuro-fuzzy controller software from computer to operation area. In this way, we set to free to this controller from computer and now it can operate to stand alone in the system. Another assistance of this project to world of controllers is that these types controllers (artificial intelligence based controllers) will be used to more effectively than in the past. Because microcontrollers will reduce to cost of artificial intelligence based controller systems.

Statements, which are placed rest of writing, make clear that two main parts are placed in this study. The first main part includes Matlab/Simulink workings, which will be basis products of study for hardware system design. With the way of this one, we could find easily to code points of the project and we could transfer the controller data from computer to hardware environment. And we need to learn that this project is feasibil or not. For the reason of this one, computer simulations were done. In addition, that conventional control techniques and new controller will be compared on each other with the way of simulations. We used Matlab for simulations of the study, because controllers can be projected and simulated easily at this soft tool. The second main part of project includes hardware components, which constitute to body of new controller. Finally, simulations and result of hardware runnings were compared on each other. And successful results were obtained.

That results were encountured; -artificial intelligence based controllers were carried from computers to microcontrollers, -in this way, usage of new type controllers will be increased, - well known result, that artificial intellegence based controllers are more succesful than the conventional controllers on controlling to non-linear systems, were encountered again in this study.

**Keywords:** Artificial intellegence, Fuzzy logic control, Neural Networks, PH neutralization process, Acid (acetic acid), Base (sodium hydroxide), Microcontroller.

## 1. GİRİŞ

Özellikle ikinci dünya savaşından sonra kontrol sistemlerinin insan hayatı içerisindeki kullanım payında küçümsenemeyecek artışlar meydana gelmiştir. Elektronikten-tıpa, çevreden-üretim proseslerine ve elektrik makinelerinden-mekanik elemanlara varıncaya kadar artık hayatımızın her alanında kontrol sistemleri kullanılmaktadır. Burada kontrol sistemleri denildiğinde, hem elektronik kontrol sistemleri hem de mekanik kontrol sistemleri göz önüne alınmalıdır. Ancak günümüzde kontrol sistemlerinin elektronik elemanlara doğru hızla bir yöneliş içerisinde olduğu da bir gerçektir. Bunda temel etmenler olarak ise, PLC'ler (programlanabilir lojik kontroller) ve kontrol sistemlerindeki uygulamalara yönelik olarak geliştirilen mikro kontrolörler gösterilebilir. Nitekim bu çalışmada da basit 8 bitlik mikro kontrolörler kullanılacaktır. Bu elemanları ön plana çıkaran etmen ise, tabii ki performans/maliyet parametrelerinin yüksekliğidir.

Aslında bu açıklamaların bizi götürdüğü sonuç, geleneksel kontrol sistemlerinin (analog P, PI, PD ve PID kontrol sistemleri) yerini almaya başlayan dijital kontrol sistemlerindeki artıştır. Öte yandan, günümüz kontrol sistemleri içerisinde, yüksek işlem gücü isteyen yapay zeka ürünü olan kontrol sistemlerinin (bulanık mantık kontrol, yapay sinir ağı tabanlı kontrol sistemleri vs.) ön plana çıkışları 1900'lü yılların sonundan itibaren karşımıza çıkmaktadır. Kontrol sistemlerindeki bu baş döndürücü gelişme, tabii ki de teknolojiye paralel olarak meydana gelmiştir ve de daha önemlisi artan bir ivme ile devam etmektedir. Zira DSP'ler (dijital sinyal işleme) yüksek hız ve paralel veri işleme mantığı çerçevesinde kontrol sistemlerinde kullanılmaya başlanmıştır. Burada, DSP'lerin yüksek hızlarının (mikro kontrolörler 1~10 MHz civarındaki frekanslarda çalışırken, DSP'ler 50~200 MHz'lere kadar ulaşabilmektedir) yanı sıra paralel veri işleyebilme (mikro kontrolörler seri veri işlerken, DSP'ler paralel veri işlerler) gibi özellikleri lokomotif olmuşlardır. İşin daha elim yönü ise, komplike sistemlerin kontrolünde bilgisayar yazılımları ile bu yazılımları kontrol edilecek sistem ile haberleştiren ara yüz kartlarının kullanılmasıdır. Bu gibi sistemler vasıtası ile de bir üretim bandının, fabrikanın veya uzay aracının kontrolü gibi işlemler gerçekleştirilebilmektedir. Ayrıca bu sistemler vasıtası ile çok hassas ayarlamaların istendiği cihazların kontrolü de son derece başarılı olarak gerçekleştirilebilmektedir.

Kontrolün günümüzde her alana yayıldığı gibi bir durumdan daha önce bahsetmiştim ve bu bağlamda kontrol sistemlerinin kimyasal proseslerin kontrolünde de kullanıldığı durumu, bir gerçektir. Bu çalışmada kontrolü amaçlanacak olan, pH nötralizasyon prosesi de bu kimyasal proseslere örnek olarak verilebilir. Ancak buradaki kontrol işleminde, geleneksel kontrol

sistemlerini kullanmak yerine yapay zeka ürünü olan kontrol sistemlerinden bulanık mantık kontrol ve yapay sinir ağı tabanlı kontrol sistemlerinin bir bileşimi kullanılacaktır. Burada amaç her iki yapının da, teorisinden kaynaklanan baskın yönlerini tasarımı yapılacak olan kontrol sistemine aktarmaktan ibarettir.

Çalışma iki ana bölüm altında ele alınabilir. İlk bölüm olarak, bilgisayar ortamında Matlab yazılımı kullanılarak tasarlanacak olan sistemin simülasyonu gerçekleştirilecektir. Bu sayede geliştirilen yeni yapının başarımı hakkında genel bir fikir elde edilmiş olacaktır. Yine bu konu ile ilgili yapılmış önceki çalışmalarda yer alan hem geleneksel hem de yapay zeka ürünü kontrol sistemlerine nazaran bu yeni yapının başarımının da değerlendirilmesine olanak sağlanacaktır. Bu simülasyonun gerçekleştirilmesinin bir diğer nedeni ise, burada elde edilecek değerlerden hareketle, çalışmanın ikinci bölümünü oluşturacak olan donanımsal uygulamaya temel teşkil etmesidir. Zira burada, sistemin matematiksel modeli bir mikro kontrolöre aktarılarak fiziksel sistem kurma ile kaybedilecek olan zamandan kazanıma gidilecektir. Yine bilgisayar ortamında geliştirilmiş olan sisteme ait değerler hücrenel bazda mikro kontrolörlere aktarılarak DSP kullanmadan daha düşük frekanslarda da olsa paralel veri işleme amacına ulaşılmış olacaktır. Zaten ele alınan prosesin karakteristiği de, yüksek frekanslarda çalışacak bir kontrolöre ihtiyaç duyulmayacak bir yapıdadır.

Yukarıdaki açıklamalar ışığında şu kazanımların elde edilmesinin amaçlandığı söylenebilir;

- ✓ Yapay zeka ürünü olan kontrol sistemlerinin uygulanabilirliğinin bilgisayar bağımlılığından kurtarılması,
- ✓ Maliyet düşüşünün sağlanması ile bu yeni tip kontrol sisteminin kullanımının artışının sağlanması,
- ✓ Lineer olmayan sistemlerin kontrolünde, yapay zeka ürünü olan kontrol sistemlerinin geleneksel kontrol sistemlerine nazaran çok daha başarılı olduklarının gösterilebilmesi.

Kontrol sisteminin tasarımına geçmeden önce tabii ki de, burada kullanılacak olan kontrol teknikleri ve de genel olarak kontrol tekniği ile ilgili gerekli açıklamalar ve kontrol edilecek olan prosese ait veriler sunulacaktır. Çalışmanın akışına uygun olarak da, daha sonraki bölümlerde hedeflenen kontrol sisteminin elde edilmesine çalışılacaktır.

## 2. KONTROL SİSTEMLERİ

Çalışmanın amacı, pH nötralizasyon prosesinin kontrolü olduğuna göre; kontrol kavramından, kontrolün kısaca tarihçesinden, en genel (klasik ve en fazla bilinen) kontrol tekniklerinden, yapay zeka ürünü kontrol teknikleri ile günümüzde kontrol tekniklerinin kullanım alanlarından yeterli altyapıyı verecek kadar bahsetmek yerinde olacaktır.

### 2.1 Kontrol Teorisi (Tarihi Geçmiş)

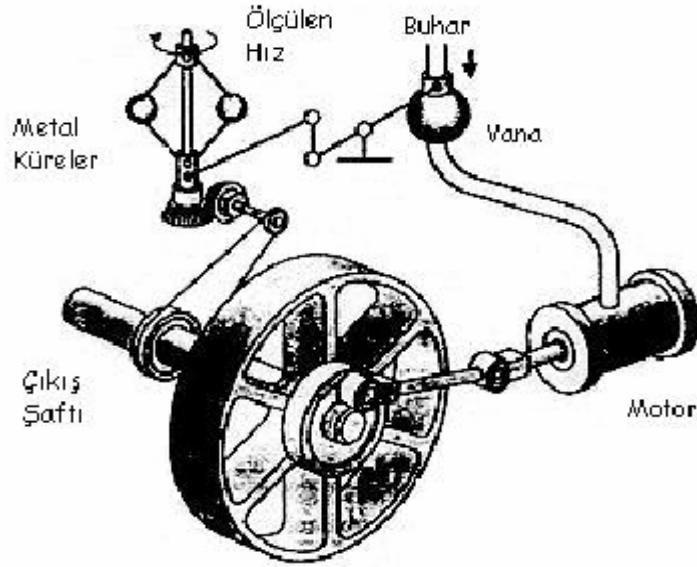
Çalışmanın amacının kontrolün tarihi geçmişi hakkında uzun uzadıya bilgi vermek olmadığı açıktır, bu nedenle de kontrolün tarihi geçmişine birkaç satırlık bir bilgi ile değinmek yeterli olacaktır.

Tarihin eski dönemlerinde dahi mekanik temelli kontrol sistemleri söz konusudur (eski islam alimlerinin yaptıkları mekanik, özellikle hidrolik temelli robotlar dahi bu gruba yerleştirilebilir). Ancak günümüzde mikroişlemci, mikrokontrolör, dsp ve hatta bilgisayar temelli kontrol sistemleri kullanılmaktadır. Tabi ki de unutulmaması gereken nokta, burada adı geçen sistemlerin (mikroişlemci, dsp vs...) geliştirilmesinden önceki kontrol sistemleri mekanik temelli idi (hatta günümüzde bile mekanik temelli kontrol sistemleri söz konusudur, buna en basit örnek ise otomobillerin vites kutuları verilebilir).

Modern kontrole ilk örnekler ise;

- ✓ James Watts'ın 1788'de buhar motorunun hız kontrolü için geliştirdiği "fly-ball governer",
- ✓ 1868'de Maxwell'in geribeslemeli kontrol yapısının matematiksel ifadesini analizi ile kontrolde sistematik çalışmanın önünün açılması,
- ✓ 1877'de ise E. J. Routh tarafından "routh kararlılık kriteri"ni geliştirmesi,
- ✓ 1960'larda dijital bilgisayar sistemlerinin geliştirilmesi ile de yapay zeka ürünü olan kontrol tekniklerinin açılmış olması,

olarak verilebilir.



Şekil 2.1 James Watts'ın 1788'de buhar motorunun hız kontrolü için geliştirdiği "fly-ball governor"

İkinci Dünya Savaşı'ndan sonra özellikle de günümüzde kontrol sistemleri, uygarlığın gelişme ve ilerlemesinde önemli rol oynayan bir bilim dalı halini almıştır. Savaşlar, aslında insanlık açısından çok acı durumlar oluşturmakla birlikte, teknolojik ve bilimsel açıdan sıçramaların gerçekleştiği olaylar olarak karşımıza çıkmaktadır. Örneğin ikinci dünya savaşının teknolojide meydana getirdiği sıçramanın, 50 yıllık bir süreye tekabül ettiği belirtilmektedir.

Proses kontrolü açısından, ilk kontrolçüler kimyacılar. Günümüzde ise, kontrol kendi başına bir bilim dalı halini alma yolunda hızla ilerlemektedir. Bunun da temel nedeni ise, günümüzün kontrol yapılarında ana elemanlar olan bilgisayarların ve mikroişlemcilerin gelişiminin 20. yüzyılın son çeyreğine dek gecikmesidir.

## 2.2 Kontrol Sistemlerine Genel Bakış

Kumanda ve kontrol sistemleri genellikle aşağıdaki teknolojiler ile gerçekleştirilmektedir.

✓ **Elektronik:** gerek ölçme ve gerekse bilgi iletme ve işleme elemanlarının çeşitliliği, kontrol devresinin gerçekleştirilmesinde çok geniş olanaklar sağlaması bakımından en çok uygulanan teknolojidir. Kontrol teçhizatı, ölçme-yükseltme-kalibrasyon, kontrol ve kumanda-güç elektroniği katlarından oluşur. Geometrik boyutları küçüktür. Gösterge ve kaydetme

elemanlarına kolayca bağlanabilir. Elektronik teknolojisi de kendi içinde iki önemli kola ayrılır:

1. Sürekli (Analog) Elektronik: ölçme devresi çıkışı genellikle -10 V, +10V standart aralığındadır. Bilgi iletimi uzak mesafelere yapıyorsa (30 m den fazla), akım çıkışlı ölçme devreleri kullanılır. Bunların standart değişim aralığı 4 mA - 20 mA dir. Kontrol elemanı ve devresi de seçilen ölçme elemanına göre gerilim girişli (—10,—10 V) ya da akım girişli (4—20mA) seçilir. Ölçme ve kontrol devrelerinde tüm devre elemanlar kullanılır. Kumanda devresi, kullanılacak kumanda (son kontrol) elemanına ya da kontrol edilen sisteme göre ayarlanabilir güçte bir güç (yükseltici devre) niteliğindedir.

2. Sayısal (Dijital) Elektronik: ölçme elemanı ve devresi sayısal ya da sürekli çıkış verebilir. Sürekli değişimler bir analog - dijital çevirici ile sayısal biçime dönüştürülür. Güç düzeyi genellikle (+5V,—5V) referans gerilimleri ile mA mertebesindedir. Kontrol elemanı ve devresi ya tüm devre mantık elemanlarından ya da, genellikle bir mikroişlemci sisteminden oluşur. Bu durumda programlanabilme kolaylığı uygulamalar için büyük bir esneklik sağlar. Sistem, dışındaki bilgi işlem sistemlerine, sayısal gösterge ve kaydedicilere kolaylıkla bağlanabilir. Kumanda devresi, kumandanın sürekli olduğu hallerde kontrol devresi ile bağlantıyı sağlayan bir dijital - analog çeviriciden ve son kontrol elemanını süren bir güç elektroniği devresinden oluşur.

✓ **Elektro - Mekanik:** yüksek güçlü sistemlere ve son kontrol elemanlarına özellikle sıralamalı işlemler için kumanda edildiğinde röleler, kontaktörler gibi elektro- mekanik elemanlar kullanılabilir. Bu türden uygulamalarda ölçme sistemi çıkışı, elektro-mekanik elemanlara doğrudan kumanda edebilecek düzeydedir. Kontrol sistemi, bu elemanlardan kurulu bir mantık devresi niteliğindedir. Gerek ölçme, gerek kontrol, gerekse kumanda elemanları için 220 V tek fazlı, 380 V üç fazlı alternatif akım veya 24 V doğru akım elektrik gücü kullanılır.

Son kontrol elemanına kumanda eden rölelerin en büyük akım değerleri, gerekli güce göre 16 Amper'e kadar yükselirken, ölçme ve kontrol sisteminde genellikle daha küçük (0,1 - 1,6 A) akım değerleri kullanılır.

Elektro-mekanik elemanlar genellikle aç-kapa türü süresiz kontrol sistemleri için kullanılırlar.

✓ **Pnömatik (Basınçlı Hava):** pnömatik teknolojisi, elektronik ve elektro-mekanik elemanların yangın tehlikeleri nedeniyle kullanılmadığı uygulama alanlarında, veya basınçlı havanın ölçme tekniği, kumanda tekniği bakımından daha elverişli olduğu durumlarda benimsenir. Eğer kullanma yerinde basınçlı hava besleme devresi zaten kurulmuş ise, bu da bir tercih nedenidir.

Basınçlı hava ile (pnömatik) kontrolde bilgi taşıyan değişken basınçtır. Standart değişme aralığı 0,2 bar -1 bar arasındadır. Pnömatik ölçme ve kontrol elemanlarına, bu elemanlar içindeki kayıpları da karşılamak üzere 1,6 bar besleme basıncı uygulanır. Kumanda elemanları da standart 0,2 bar -1 bar değişme aralığında gelen kontrol değişmelerini arzulanan kumandaya çevirecek biçimde tasarlanırlar. Ölçme elemanlarının giriş (ölçme) aralığı ayarlanabilir olduğu halde çıkışları yine 0,2 bar -1 bar standart aralığındadır. Ölçme veya kontrol değişmelerinin uzak mesafelere iletilmesi gerektiğinde (30m den fazla) ya pnömatik ileticilerle (transmitter) genliği azalan basınç değişmeleri standart aralığa yükseltilir veya yeni gelişmekte olan sayısal pnömatik teknolojisi kullanılır.

Pnömatik kontrol teknolojisi elektrik-pnömatik ve pnömatik-elektrik çeviriciler yardımı ile elektronik veya elektromekanik teknolojisi ile birlikte kullanılmaktadır. Sıralamalı işlemlerin gerçekleştirilmesinde pnömatik valfler ve mantık elemanları kullanılır. Bu elemanlarla gerçekleştirilen pnömatik güç uygulamalarında, sürekli değişmelerin kontrolü söz konusu değilse, kullanılan besleme basıncı genellikle 6 bar, kumanda basıncı ise 4 bar mertebesindedir.

✓ **Hidrolik (Basınçlı Yağ):** küçük hacimlerde yüksek güçlerin kontrolünü gerektiren uygulamalarda hidrolik kumanda teknolojisi kullanılır. Genellikle kumanda elemanlarının girişi elektronik, çıkışı hidroliktir. Yine de, değişik teknolojilerde ölçme yapmamak amacı ile hidrolik ölçme, hidrolik geri besleme yapılabilir. Hidrolik kontrol devrelerinde besleme ve kumanda basıncı 300 bar'a kadar çıkarılabilir. Bilgi taşıma ve işleme değişkeni, yine basınçtır. Ölçme-kontrol devresinde kumanda basıncından çok daha düşük, en fazla 20 bar mertebesinde basınçlarla çalışılır. Son kontrol elemanı bir hidrolik silindir veya hidrolik motordur (Özdaş, 1998).

Görüldüğü farklı alanlardan temel alan kontrol sistemleri, bir arada kullanılarak melez sistemler geliştirilebileceği gibi her bir sistem tek başına kullanılarak da tekil kontrol sistemleri geliştirilebilir. Bu çalışmada, elektronik kontrol sistemi (artificial intelligence based) tek başına kullanılarak pH prosesi kontrol altına alınmaya çalışılacaktır.

### 2.2.1 Kontrol Sistemlerinin Sınıflandırılması

Kontrol sisteminin türü burada sadece çıkış işaretinin kontrol ya da yönetilmesi açısından ele alınacaktır. Çıkışın ya da kontrol edilen büyüklüğün kumanda edilmesi bakımından kontrol sistemleri iki türe ayrılır: 1. Açık Çevrim Kontrol Sistemleri; 2. Kapalı Çevrim Kontrol sistemleri. Şimdi bu kontrol sistemlerinin tanımlarını verelim:

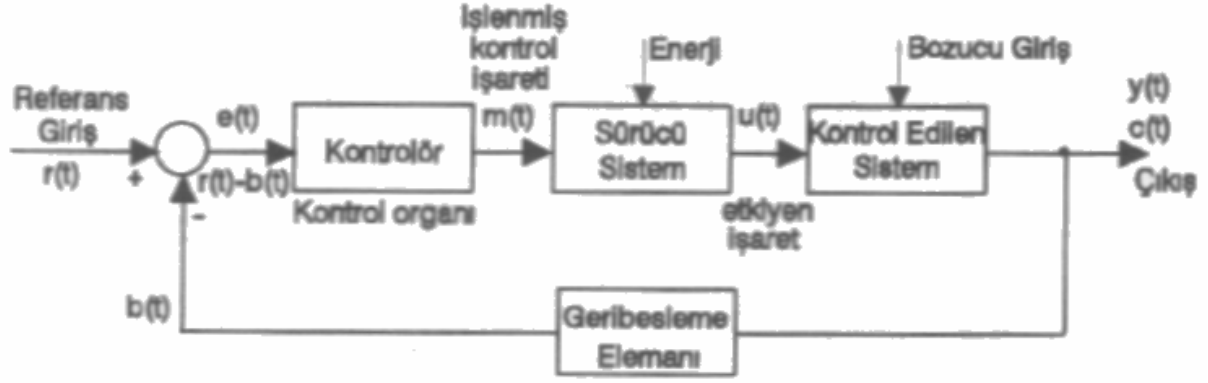
1. **Açık Çevrim Kontrol Sistemleri:** girişindeki kumanda ya da kontrol işareti çıkıştan (kontrol edilen büyüklükten) bağımsız olan bir kontrol sistemidir. Fakat açık çevrim kontrol sistemlerinde çıkış, giriş işaretinin bir fonksiyonudur.

2. **Kapalı Çevrim Kontrol Sistemleri:** girişindeki kumanda ya da kontrol işareti, çıkış işaretine ya da, çıkış işaretinden üretilen bir işaretle bir referans işaret arasındaki farka ya da bunların toplamına bağlı olan bir kontrol sistemidir. Bu sistemlere geri beslemeli kontrol sistemleri de denir (Sarıoğlu, 1995).

Burada her iki kontrol sistemine birer örnek vermek yerinde olacaktır.

Yollarda trafik kontrolü yapan ışıkları düşünelim. Bunlar, bir zaman ayarlayıcı (timer) ile yanıp sönerler. Belli bir  $t_k$  zamanı süresince kırmızı yanar, trafiği durdurur; diğer belli bir  $t_y$  zamanı süresince yeşil yanarak trafiği açar. Böyle bir kontrol sistemi trafiği iyi düzenleyemez ve kontrol edemez. Trafik olmasa ya da çok yoğun olsa da bu ışıklar aynı ( $t_k$  ve  $t_y$ ) zaman aralıkları ile kırmızı ve yeşil yanar. Burada kontrol sistemine giriş, ( $t_k$  ve  $t_y$ ) kırmızı ve yeşil yanma süreleri, çıkış ise trafik yoğunluğudur. Kontrol sisteminin girişi ya da kumanda işareti, örneğin  $t_y$  lambanın yeşil yanma süresi olarak alınırsa, çıkış büyüklüğü olan trafik yoğunluğu,  $t_y$  yeşil yanma süresine etkimez; başka bir deyimle, lambaların  $t_y$  yeşil yanarak trafiğe geçiş izini verme süresi trafik fazla aksın diye uzatılmaz. Buradan anlaşılmaktadır ki, kumanda işareti olan,  $t_y$  yeşil yanma süresi, çıkışı oluşturan trafik yoğunluğundan bağımsızdır. O halde, bu sistem bir açık çevrim kontrol sistemini oluşturur. Eğer bu sistemde trafik yoğunluğu (sistem çıkışı) ölçülür ve bir referansla karşılaştırılır, aradaki farka göre yeşil ve kırmızı yanma sürelerini ayarlanır ise böyle bir sistem kapalı çevrim kontrol sistemini oluşturur.

Kapalı çevrim otomatik kontrol sisteminin tanımından basit kapalı çevrim ya da geribeslemeli otomatik kontrol sisteminin blok diyagramı Şekil 2.2 deki gibi gösterilebilir. Burada kontrol işareti bir referans ile karşılaştırma işleminden sonra elde edilmektedir. Bu nedenle literatürde, kontrol işareti için hata ya da sapma deyimleri de kullanılmaktadır.



Şekil 2.2 Kapalı çevrim kontrol sistemi blok diyagramı (Sarıoğlu, 1995).

Bu noktada bizim için önemli olan, kapalı çevrim kontrol (geribeslemeli kontrol) olduğuna göre, geribeslemeli kontrol sistemlerinin yapısı hakkında kısaca da olsa bir veritabanı oluşturmak yerinde olacaktır.

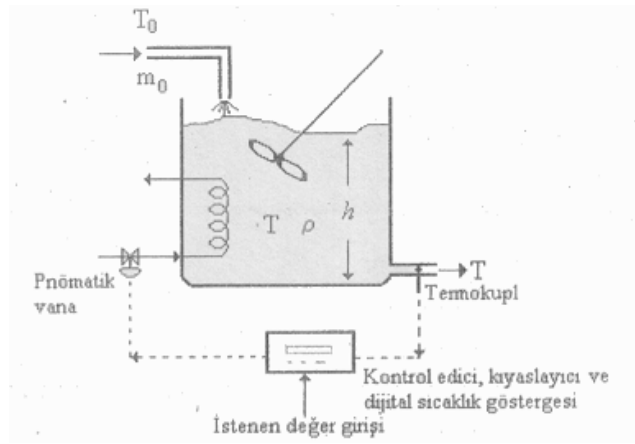
### 2.2.1.1 Geri Beslemeli Kontrol

Kontrolde, sistemi kontrol etmek amacıyla, sistemin istenen halden sapmasına ilişkin bilgiden yararlanılmasına geri beslemeli kontrol denir. Geri beslemeli bir kontrol devresi ölçme, kıyaslama, hesaplama ve düzeltme olarak adlandırılan temel kontrol işlevlerinin art arda yaptırıldığı kapalı bir çevrimdir. Kontrolü istenen, prosesi en iyi karakterize eden kontrol değişkeni seçilerek ölçülür. Sonra ölçülen bu değer, istenen durumu temsil eden istenen (referans) değerden sapması tespit edilir. Bu saptamaya hata sinyali büyüklüğü denir. Hatanın küçültülmesi veya tamamen ortadan kaldırılması için kontrol mekanizması tarafından sistemin belirli bir büyüklüğüne (ayar değişkeni) gerekli düzeltme işlemi yaptırılır (Söğüt, 2004).

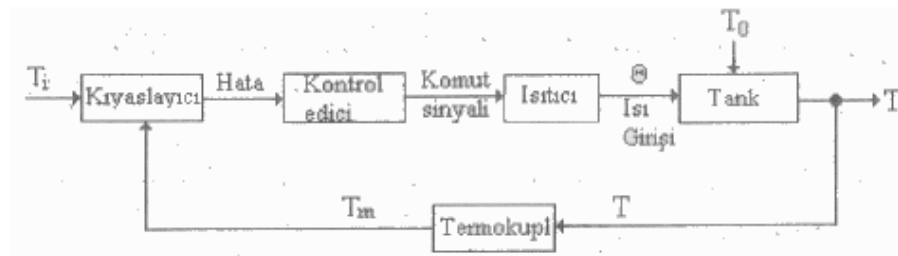
Kontrol devresi elemanlarından son kontrol vanası, proses elemanı ve ölçme elemanı, prosesin olduğu yerde veya çok yakınında bulunur. Çoğunlukla kontrol mekanizması olarak adlandırılan kıyaslayıcı ve kontrol edici ikilisi ise, prosesin uygun görülen değişik yerlerinde, tesisler için kontrol odasında bulunur. Ölçme elemanından alınan sinyali kıyaslayıcıya ve kontrol elemanından alınan sinyali son kontrol vanasına bağlayan elektriksel veya pnömatik sinyal iletim hatları ve sinyal dönüştürücüleri vardır ve bazen 30-60 m mesafelerde bulunabilirler. Hatların çok uzun olduğu durumlarda sinyal kuvvetlendirici aygıtlar (pilot / booster) kullanılmaktadır (Şaşmaz, 1990).

Geri beslemeli kontrolün bir diğer üstünlüğü, kontrol edilen değişkenin sürekli referans değer ile kıyaslanmasıdır. Böylece tek bir ayar değişkeni ile bilinen veya bilinmeyen bütün yük değişkenlerinin olumsuz etkisi görece ucuz bir düzenele giderilmiş olur (Akgün, 2003).

Bu tip kontrolün ana sakıncası kontrol hareketinin, referans değerden sapma olana kadar başlamamasıdır. Ancak sapma belirlendiği anda kontrol düzeneği bunu yok etmek için etki eder. Bu durum, en küçük değişimlere bile tahammülü olmayan bazı sistemlerde büyük zarara neden olabilir, zira müdahalede çok geç kalınmış olabilir. Yük değişkeninin meydana geldiği andan itibaren, yük değişkenlerinin etkisinin tespit edilip kontrol edici tarafından bir düzeltme komutunun verilmesine kadar geçecek olan bu gecikmeye, ölü zaman denilmektedir. Bu kontrol yöntemi deneme-yanılma yöntemiyle çalıştığından denge konumundan pozitif veya negatif ayrılmalar görülür. Geri beslemeli kontrolün tipik bir dalgalanma eğilimi vardır (Alpbaz, 1993).



Şekil 2.3 Geribeslemeli sıcaklık kontrolünün proses üzerindeki gösterimi (Akgün,2003).



Şekil 2.4 Geribeslemeli sıcaklık kontrolünün blok diyagramı (Akgün,2003).

Yine geribeslemeli kontrol sistemlerinin değişmez geleneksel kontrol sistemleri olan; oransal (P kontrol), türevsel (PD kontrol), integral (PI kontrol) ve oransal-integral-türevsel (PID kontrol) tekniklerinden kısaca bahsetmek gerekmektedir.

✓ **Oransal (P) kontrol edici:** Oransal kontrol edicinin çıkışı yalnızca hata sinyalinin değişimi halinde değişir. Oransal kontrol edici genellikle yatışkın hal hatası, yani ofset taşır. Bu durum oransal kontrolün olumsuz yanıdır ve bu yüzden gerektiğinde integral etkisi eklenir.

Ofsetin büyüklüğü yük etkisinin ve kontrol edici kazancının büyüklüğüne bağlıdır. Kazanç arttıkça ofset azalır. Yüksek kazanç değerlerinde ise döngü kararsızlaşır, aç/kapa kontrol edici gibi davranmaya başlar. Yatışkın hal hatası tümüyle istenmeyen bir özellik değildir, çoğu seviye kontrol sistemlerinde önemli olan taşma ya da boşalmadan kaçınmak olduğu için en uygun seçim oransal kontroldür (Luyben, 1997).

Oransal geribeslemeli kontrol edici; çıkış sinyalini (CO), transmitterden aldığı, proses değişkeninin gerçek değeriyle set değeri (SP) arasındaki fark olan hata sinyaliyle (E) doğru orantılı şekilde değiştirir.

$$CO = Bias \pm K_c (SP - PV) \quad (2.1)$$

Buradaki bias sinyali, bir sabittir ve değeri, hata olmadığı zamanki kontrol edici çıkış değerine eşittir.  $K_c$ , kontrol edici kazancıdır. Kazanç değeri arttıkça, belirli bir hata halindeki kontrol edici çıkış değişkeninin değeri büyür. Örneğin kazancın "1" olduğu durumda %10'luk bir hata, kontrol edici çıktısını % 10 değiştirir.

Bazı cihaz üreticileri, kazanç yerine oransal bant (PB, proportional band) adıyla bir başka terim kullanmaktadırlar. Bu iki terim aşağıdaki bağıntıyla birbirine bağlanır:

$$Pb = 100 / K_c \quad (2.2)$$

Oransal bant büyüdükçe, ya da genişledikçe kazanç azalır. Kontrol edicinin ters etkili veya doğrudan etkili olmasıyla, kazanç terimi pozitif veya negatif değerler alır. Proses değişkeni olması gereken değeri alınca kontrol edici, ilgili cihaza göndereceği sinyali azaltıyorsa ters; aksi halde doğrudan etkili bir kontrol edicidir. Kontrol edicinin ters etkili mi, yoksa düz etkili mi olacağına karar verilirken ilk olarak vanalara bakılır.

Örneğin bir ısı değiştirici için, arıza halinde buhar vanasının kapanması tercih edilir. Böyle bir sistemde, ancak açılması için hava gönderilen pnömatik vanalar (AO, air to open; normalde kapalı) gerekir.

Bundan sonra sıcaklık transmitterine bakılır. Proses sıcaklığı yükseldiğinde transmitter çıkış sinyali de büyüyeceği için sıcaklık transmitteri doğrudan etkilidir. Bu durumda sisteme daha az buhar gönderilmesi gerekir. Vana havayla açıldığı için kontrol edici çıkış değeri azalmalıdır. Buradaki kontrol edici ters etkili ve pozitif kazançlı olmalıdır.

Başka bir örnek olarak soğutmalı bir tepkime kabının soğutma suyu vanası incelenirse, bir

arıza halinde sıcaklığın aşırı artmaması için vananın tamamen açık olması gerektiği görülür. Böyle bir sistemde hava gönderilerek kapanan vanalar seçilir. Proses sıcaklığı istenen değerinin üzerine çıktığında vanaya gönderilen sinyal onu açmak suretiyle soğutma suyu debisini artırır. Bu durumda da ters etki söz konusudur.

Son örnek olarak bir distilasyon kolonunun alt akım vasıtasıyla seviye kontrolü irdelendiğinde, arıza halinde kolonun boşalmaması için vananın kapanması tercih edilir. Bu durumda vana havayla açılmalıdır. Seviye transmitteri sinyali seviye ile doğru orantılı olarak değişir. Seviye yükseldiğinde alt akımın hızlanması gerekir. Burada doğrudan etki söz konusudur.

✓ **İntegral etki (Reset):** Oransal etki, kontrol vanasını hatanın büyüklüğüyle doğru orantılı olarak hareket ettirir. İntegral etki kontrol vanasının hatanın zamana bağlı integraline bağlı miktarda hareket ettirir.

$$CO = Bias + \frac{1}{\tau_I} \int E(t) dt \quad (2.3)$$

İntegral kontrol edicide iletim fonksiyonu aşağıdaki gibidir;

$$G_c(s) = \frac{1}{\tau} \frac{1}{s} \quad (2.4)$$

burada  $\tau_I$ , integral zaman sabitini (ing. *reset time*) temsil eder. İntegral etkinin temel amacı oransal kontrol edicinin dezavantajı olan yatışkın hal hatasının ortadan kaldırılmasıdır. Genellikle sistemin dinamik yanıtının mertebesini artırır ve kararlılığı azaltarak bazı durumlarda daha salınımlı hale getirir. Ancak ofsetin istenmediği durumlarda kullanılması zorunludur.

✓ **Türevsel etki:** Türevsel etki kontrol edilen değişkenin istenen değerden uzaklaşmasını daha etkin bir şekilde engeller.

$$CO = Bias + \tau_D \frac{dE}{dt} \quad (2.5)$$

Türevsel etki birçok kontrol çevriminde dinamik yanıtını iyileştirir. Ancak sinyallerde gürültü varsa türevsel etki tercih edilmez.

Türevsel kontrol edicide iletim fonksiyonu aşağıdaki gibidir:

$$G_C(s) = \tau_D s \quad (2.6)$$

Ancak ideal türevsel etki fiziksel olarak imkansızdır, çünkü aygıtlar türevi mükemmelen hesaplayamazlar. Bu yüzden türevsel hesaplamalar yaklaşık olarak yapılır.  $\alpha$  ticari kontrol edicilerde 1/6 ile 1/20 arasında değişen bir sabit olmak üzere bu yaklaşık hesaplamamanın Laplace transfer fonksiyonu aşağıdaki gibidir (Luyben, 1988):

$$G_C(s) = \frac{\tau_D s + 1}{\alpha \tau_D s + 1} \quad (2.7)$$

Zaman boyutuna çevrildiğinde:

$$\alpha \tau_D \frac{dCO}{dt} + CO = \tau_D \frac{dE}{dt} + E \quad (2.8)$$

$$CO = Bias + \tau_D \frac{d(E - \alpha CO)}{dt} \quad (2.9)$$

✓ **Oransal-Integral (PI) kontrol edici:** Oransal etki, kontrol vanasını hatanın büyüklüğüyle doğru orantılı şekilde hareket ettirir. Çoğu kontrol döngüsünde PI kontrol edici kullanılır. Integral etkisi ofseti ortadan kaldırır. Integral zaman sabiti  $\tau_I$  azaldıkça hata daha çabuk yok edilir. Ancak çok küçükse sistemin kararsızlığı artar.

Oransal-integral kontrol edicinin denklemleri aşağıdaki gibidir:

$$CO = Bias + K_C \left( E + \frac{1}{\tau_I} \int E dt \right) \quad (2.10)$$

Transfer fonksiyonu ise şu şekildedir:

$$G_C(s) = K_C \left( 1 + \frac{1}{\tau_I} \frac{1}{s} \right) = K_C \frac{\tau_I s + 1}{\tau_I} \quad (2.11)$$

✓ **Oransal-Integral-Türevsel (PID) kontrol edici:** Eğer sinyallerde gürültü yoksa ve sıkı sıkıya bir dinamik yanıt gerekiyorsa PID kontrol ediciler kullanılır. Türev terimi, set değerine yaklaşmayı hızlandırır. Reaktörlerin sıcaklık kontrolünde genellikle PID kontrol edici kullanılır. Kontrol aygıtlarıyla fiziksel olarak mükemmelen gerçekleştirilemese de ideal PID kontrol edicinin denklemleri aşağıdaki gibidir:

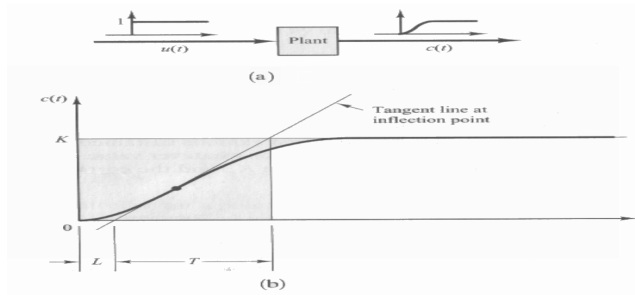
$$CO = Bias + K_c \left( E + \frac{1}{\tau_I} \int E dt + \tau_D \frac{dE}{dt} \right) \quad (2.12)$$

Gerçek cihazlarda ise aşağıdaki iletim fonksiyonu geçerlidir:

$$G_c(s) = K_c \left( \frac{\tau_I s + 1}{\tau_I s} \right) \left( \frac{\tau_D s + 1}{\alpha \tau_D s + 1} \right) \quad (2.13)$$

Burada  $\alpha$ , cihaz üreticilerinin belirledikleri bir sabittir (Luyben, 1988).

Öte yandan uygulamalarda PID kontrolörün katsayıları, sistemin birim basamak referans işaretine vermiş olduğu cevaptan hareketle de üretilebilmektedir. Şekil 2.5 ten bu katsayıların elde edilmesi için gerekli olan parametrelerin, geometrik olarak elde edilişi görülebilir.



Şekil 2.5 PID kontrolörün parametrelerinin atanmasında kullanılan, sistem birim basamak referans işaretin uygulandığı (a) ve sistemin bu işarete vermiş olduğu cevap (b).

Burada sistemin birim basamak referans işarete vermiş olduğu cevap, iki parametre ile karakterize edilebilir. İlk olarak bir gecikme zamanı olan  $L$  ve ikinci olarak ise, bir zaman sabiti olan  $T$ . Bu iki parametre ile çizilecek olan tanjant doğrusu ile sistem karakterize edilmektedir. Sistemin birim basamak cevabı üzerine kurulmuş olan , Ziegler-Nichols ayarlama değerleri Tablo 2.1'de görülebilir (Ogata, 2002).

Tablo 2.1 Sistemin birim basamak cevabı üzerine kurulmuş olan Ziegler-Nichols ayarlama değerleri

Kontrolör Tipi	$K_p$	$T_i$	$T_d$
P	$T/L$	$\infty$	0
PI	$0.9 T/L$	$L/0.3$	0
PID	$1.2 T/L$	$2L$	$0.5L$

Ziegler-Nichols kurallarına göre ayarlanacak olan sistemin PID kontrolörün transfer fonksiyonu aşağıdaki şekilde elde edilir.

$$G_C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) = 1.2 \frac{T}{L} \left( 1 + \frac{1}{2Ls} + 0.5Ls \right) = 0.6T \frac{\left( s + \frac{1}{L} \right)^2}{s} \quad (2.14)$$

Buradan da görüldüğü gibi PID kontrolör, orijinde bir ve  $s = -1/L$  noktasında iki adet olmak üzere toplamda üç adet sifıra sahip olan bir eğri ile ifade edilebilir.

### 2.2.2 Kontrol Sistemlerinin Karakteristiği

Kapalı çevrim bir kontrol sisteminde referans büyüklüğünde bir birim basamak değişmesi durumunda, cevap ve diğer bir deyimle çıkışın zamana bağlı olarak değişmesi Şekil 2.6 ta gösterilmiştir. Kararlı bir kontrol sistemi için elde edilmiş bulunan böyle bir cevap fonksiyonunun veya davranışının uygunluğunu belirtmek üzere genellikle aşağıdaki karakteristik değerler kullanılmaktadır ( Sarıoğlu, 1991),

1. **Aşma (Overshoot):** çıkış büyüklüğünün ilk yükselmede son (kalıcı hal) değeri aşma miktarıdır ( $C_{\max} - C$ ).

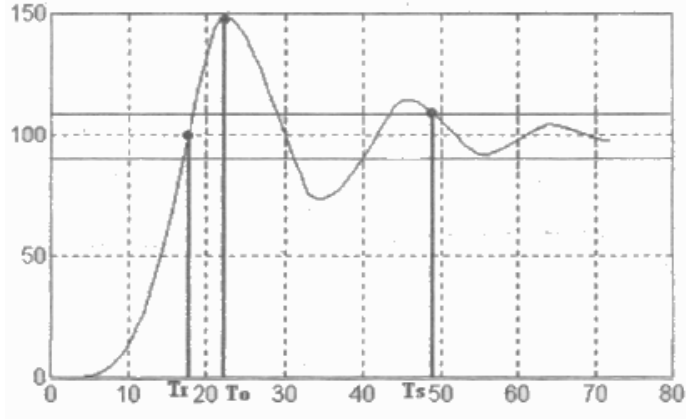
2. **Aşma yüzdesi:** aşma miktarının son değere oranıdır ve yüzde olarak tanımlanır.

$$\frac{C_{\max} - C}{C} 100\% \quad (2.15)$$

3. **Maksimum değere ulaşma zamanı ( $T_{\max}$ ) :** çıkışın  $C_{\max}$  değerine ulaşması için geçen zamandır.

4. **Yükselme zamanı (Rise time,  $T_r$ ) :** cevabın referans değerinin 10%' indan 90% a kadar çıkması veya cevabın başlangıçtan referans değerine kadar erişmesi için geçen zamandır.

Verilen her iki tanım da belirtilmek şartıyla kullanılmaktadır.



Şekil 2.6 Kapalı çevrim kontrol sistemi performans parametreleri.

5. **Yerleşme zamanı (  $T_s$  )** : cevap eğrisinde titreşimlerin son değerin belirli bir yüzdesine kadar küçülmesi için gereken zamandır. Genellikle  $T_s$  hatanın %2 veya %5 i geçmemesini sağlayacak şekilde tanımlanır ve değişimin hata için kabul edilen yüzde ile sınırlanmış bir tolerans bölgesi içinde kalmaya başlamasına kadar geçen süreden ibarettir.

6. **Genlik oranı**: cevap eğrisinin referans eğrisine nazaran birbirini takip eden titreşimlerinin genliklerinin oranıdır.

7. **Sönümlü titreşim periyodu veya frekansı  $w_1$** : kararlı bir sistemde cevap eğrisinin sönümlü titreşimlerinin periyodu veya frekansıdır.

Bilindiği gibi, sistemlerin başlangıç şartlarına bağlı kalarak çıkış fonksiyonu, aynı giriş için değişir. Dolayısıyla sistemlerin davranışlarını karşılaştırmak için sükunetteki sistemlerin girişine bir birim basamak girişi uygulamak yoluyla yükselme zamanı, aşma yüzdesi, yerleşme zamanı v.s gibi karakteristik değerler karşılaştırılabilir. Söz konusu karakteristik değerler bir veya birkaçı birlikte kullanılarak optimum performans tanımlanması gereklidir.

Herhangi bir sistemin referans girişine bir birim basamak uygulandığı da en mükemmel çıkış şüphesiz ki aynı birim basamak fonksiyonu elde ederek her an hatasız çalışma sağlayabilmesidir. Fiziksel sistemlerde böyle bir çıkışın imkan dahilinde olmamasından dolayı Şekil 2.6 daki çıkışa ait karakteristik değerler tek tek veya birlikte göz önüne alınarak en uygun sistem davranışı bulunmaya çalışılmaktadır. En uygun sistem cevabını belirlemede yaygınca kullanılan metodlar;

a. Hatanın karesinin integrali (ISE) kriteri;

$$J = \int_0^{\infty} e^2(t) dt \quad (2.16)$$

ISE kriterine göre sistem performansının kalitesi yükseltilmesi, hatanın karesinin integralinin minimize edilmesiyle mümkün olmaktadır. Bu kriter ile tasarlanan sistemlerde büyük başlangıç hatası hızla azalacak şekilde bir davranış görülmektedir. Buna göre sistemin cevabı hızlı fakat osilasyonlu olmaktadır.

b. Mutlak hatanın zaman çarpımlı integrali (ITEA) kriteri;

$$J = \int_0^{\infty} t |e(t)| dt \quad (2.17)$$

Bu kriter gere göre optimum sistem, yukarıdaki “J” kriterini minimize eden sistemdir. ITAE ye göre tasarlanan sistemlerde geçici cevapta aşım miktarı küçük ve osilasyonlar bastırılmıştır (Bulut, 2001).

c. Hatanın karesinin zaman çarpımlı integrali (ITSE);

$$J = \int_0^{\infty} t e^2(t) dt \quad (2.17)$$

Çalışmada tasarımı ve uygulaması gerçekleştirilecek olan, kontrolör sistemlerinin performanslarının kıyaslanmasında bir parametre teşkil edecek ITSE kriterinin kullanılması uygun olacaktır.

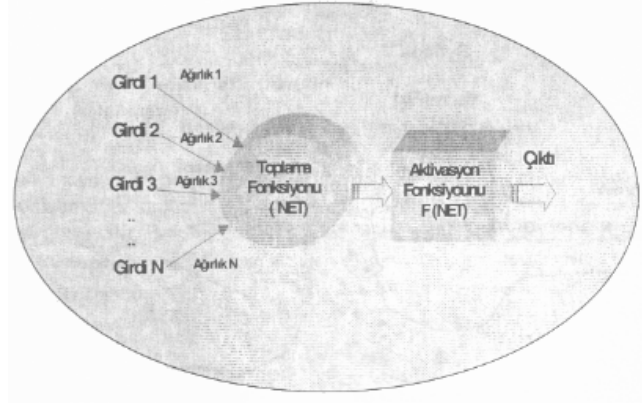
### **3. YAPAY SİNİR AĞLARI ile KONTROLÜN TEMEL ELEMANLARI**

Yapay Sinir Ağları (YSA), insan beyninin çalışma sisteminin yapay olarak taklit edilmesi sonucu ortaya çıkmıştır. YSA, insan beynindeki nöronlara karışık gelen yapay basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanmasından oluşan karmaşık bir sistemdir. Önceleri temel tıp bilimlerinde insan beynindeki nöronların matematiksel modelleme çabaları ile başlayan çalışmalar, geçtiğimiz onbeş-yirmi sene içerisinde disipline bir şekil almıştır. YSA bugün fizik, matematik, elektrik ve bilgisayar mühendisliği gibi çok farklı bilim dallarında araştırma konusu haline gelmiştir. Yapay sinir ağlarının pratikte kullanımı genelde değişik yapıda ve formlarda bulunabilen verileri hızlı bir şekilde sınıflandırma, tanımlama ve algılaması üzerinedir. Aslında mühendislik uygulamalarında YSA'nın geniş çaplı kullanımının en önemli nedeni, klasik tekniklerle çözümü zor problemler için etkin bir alternatif oluşturmasıdır. Yapay sinir ağları, geniş bir alana yayılmış bilim ve mühendislik problemlerinin çözümü için uygulanmaya devam etmektedir. Yapay sinir ağlarının matematiği, biyolojik sistemlerin yararlı özellikleri kullanılarak elde edilmiştir. Biyolojik sinir sistemlerinde olduğu gibi, yapay sinir ağları da kendi çözüm tarzlarını bulmak için eğitilebilirler.

Nitekim bu çalışmada da, YSA'nın, kimya ile kontrol mühendisliği alanlarının kesimindeki bir çalışmada kullanılması amaçlanmıştır. Burada amaç, YSA'nın karmaşık sistemleri kontrol etmede geleneksel kontrol sistemlerine (bu çalışmada geleneksel kontrol sistemlerini PID kontrol sistemi temsil etmektedir) olan üstünlüğünü göstermek ve de aynı zamanda yapay zeka ürünü olan bu tür kontrol sistemlerinin de basit (8 bitlik) mikrokontrolörler ile gerçekleştirilebileceğini göstermektir.

#### **3.1 Yapay Sinir Hücresi**

Biyolojik sinir ağlarının sinir hücreleri olduğu gibi yapay sinir ağlarının da yapay sinir hücreleri vardır. Yapay sinir hücreleri mühendislik biliminde proses elemanları olarak da adlandırılmaktadır. Her proses elemanının 5 temel elemanı vardır (Şekil 3.1).



Şekil 3.1 Yapay sinir hücresinin yapısı.

Yapay sinir hücresinin 5 temel elemanı:

- Girdiler:** bir yapay sinir hücresine (proses elemanına) dış dünyadan bilgilerdir. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir. Yapay sinir hücresine dış dünyadan olduğu gibi başka hücrelerden veya kendi kendisinden de bilgiler gelebilir.
- Ağırlıklar:** ağırlıklar bir yapay hücreye gelen bilginin önemini ve hücre üzerindeki etkisini gösterir. Şekildeki Ağırlık 1, Girdi 1 in hücre üzerindeki etkisini göstermektedir. Ağırlıkların büyük yada küçük olması önemli veya önemsiz olduğu anlamına gelmez. Bir ağırlığın değerinin sıfır olması o ağ için en önemli olay olabilir. Eksi değerler önemsiz demek değildir. O nedenle artı veya eksi olması etkisinin pozitif veya negatif olduğunu gösterir. Sıfır olması ise herhangi bir etkinin olmadığını gösterir. Ağırlıklar değişken veya sabit değerler olabilirler.
- Toplama fonksiyonu:** bu fonksiyon, bir hücreye gelen net girdiyi hesaplar. Bunun için değişik fonksiyonlar kullanılmaktadır. En yaygın olanı ağırlıklı toplamı bulmaktır. Burada her gelen girdi değeri kendi ağırlığı ile çarpılarak toplanır. Böylece ağa gelen net girdi bulunmuş olur. Bu şu şekilde formülize edilmektedir.

$$NET = \sum_i^n G_i A_i \quad (3.1)$$

Burada G girdileri, A ise ağırlıkları, n ise bir hücreye gelen toplam girdi (proses elemanı) sayısını göstermektedir. Yalnız yapay sinir ağlarında daima bu formülün kullanılması şart değildir. Uygulanan yapay sinir ağı modellerinden bazıları kullanılacak toplama fonksiyonunu belirleyebilmektedir. Literatürde yapılan araştırmalarda toplama fonksiyonu olarak değişik formüllerin kullanıldığı görülmektedir. Ancak bu çalışmada ağırlıklı genel toplam

kullanılacağından, bu formülün verilmesi ve genel yapısı yeterli olacaktır. Konunun ana amacından sapmaması adına doğru bir karar olacaktır.

d. **Aktivasyon fonksiyonu:** bu fonksiyon, hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonu olarak da çıktıyı hesaplamak içinde değişik formüller kullanılmaktadır. Bazı modeller (mesela çok katmanlı algılayıcı) bu fonksiyonun türevinin alınabilir bir fonksiyon olmasını şart koşturmaktadır. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonunda da ağın proses elemanlarının hepsinin aynı fonksiyonu kullanması gerekmez. Bazı elemanlar aynı fonksiyonu diğerleri farklı fonksiyonları kullanabilirler. Bir problem için en uygun fonksiyonda yine tasarımcının denemeleri sonucunda belirleyebileceği bir durumdur. Uygun fonksiyonu gösteren bir formül bulunmuş değildir.

Günümüzde en yaygın olarak kullanılan çok katmanlı algılayıcı modelinde genel olarak aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmaktadır. Bu fonksiyon şu formül ile gösterilmektedir.

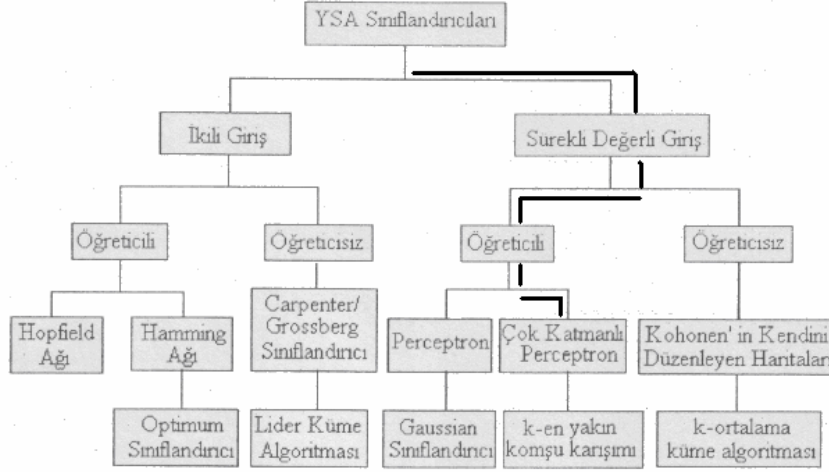
$$F(NET) = \frac{1}{1 + e^{-NET}} \quad (3.2)$$

Burada NET proses elemanına gelen NET girdi değerini göstermektedir. Bu değer toplama fonksiyonu kullanılarak belirlenmektedir.

e. **Hücrenin çıktısı:** aktivasyon fonksiyonu tarafından belirlenen çıktı değeridir. Üretilen çıktı dış dünyaya veya başka bir hücreye gönderilir. Hücre kendi çıktısını kendisine girdi olarak da gönderebilir. Bir proses elemanının birden fazla çıktısı olmasına rağmen sadece bir çıktısı olmaktadır. Ağ şeklinde gösterildiğinde bir proses elemanının birden fazla çıktısı varmış gibi görülmektedir. Bu sadece gösterim amacıyladır. Aslında bir proses elemanından çıkan tek bir çıktı değeri vardır. Aynı değer birden fazla proses elemanına girdi olarak gitmektedir (Öztemel, 2003).

### 3.2 Yapay Sinir Ağı Mimarileri

Bütün yapay sinir ağları sinirler, bağlantılar ve aktarım işlevlerine bağlı olduğu için, farklı mimariler, yapılar ya da sinir ağları arasında bir benzerlik bulunmaktadır. Çeşitliliğin çoğunluğu farklı öğrenme kurallarından ve bu kuralların bir ağın yapısını nasıl değiştirdiğinden kaynaklanmaktadır (Elmas, 2003). Öğrenme yöntemlerine ve girdi parametresinin yapısına göre ağ yapıları Şekil 3.2 de görülmektedir.



Şekil 3.2 Yapay sinir ağları sınıflandırıcıları.

Öte yandan ağlar verinin akış yönüne bağlı olarak da iki farklı gruba ayrılabilirler.

a. **İleri beslemeli ağlar;** ileri beslemeli ağlarda, işlem elemanları birbiri ardı sıra sıralanırlar. Girdiler dışından ya da önceki nörondan alınır; nöronların çıktıları da bir sonraki nörona veya dışarıya verilir Bu ağlarda, nöronun önceki veya aynı katmandaki herhangi bir nörona bağlantısı ve veri iletmesi söz konusu değildir. Bu mimarilerde ağın çıktısı, tamamen ağa giren girdilere bağlıdır. İleri beslemeli ağlar herhangi bir dinamiklik özelliği taşımazlar ve gösterdikleri özellik bakımından doğrusal ve doğrusal olmayan kararlı problem alanlarına uygulanabilirler.

b. **Geribeslemeli ağlar;** geri beslemeli ağlarda işlem elemanlarının çıktıları daha gerilerdeki işlem elemanlarına girdi olarak kabul edilir. Geri beslemeli ağları ileri beslemeli ağlardan ayıran temel özellik, bu tip mimariyle oluşturulan ağlar dinamik doğrusal olmayan sistemleri, özellikle doğrusal olmayan diferansiyel denklemleri modelleyebilmeleridir (Cichocki, 1993).

### 3.3 Yapay Sinir Ağının Eğitilmesi

Öğrenme kavramı, "zaman içinde yeni bilgilerin keşfedilmesi yolu ile davranışların iyileştirme süreci" olarak tanımlanmaktadır. Bundan yola çıkarak, makine öğrenmesi, "bilgisayarın bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek, benzeri olaylar hakkında karar verebilmesi ve problemlere çözüm üretebilmesidir" denilebilir. Bilgisayarların öğrenebilmesi, ilgili olaylarla ilgili bilgiler ile donatılmasına bağlıdır (Simon, 1983). Bu noktada iki parametre ön plana çıkmaktadır; öğrenme stratejileri ve öğrenme kuralları. Şimdi sırası ile bu iki konuyu ele alalım.

### 3.3.1 Öğrenme Stratejileri

Yapay sinir ağları gibi örneklerden öğrenen sistemlerde değişik öğrenme stratejileri kullanılmaktadır. Öğrenmeyi gerçekleştirecek olan sistem ve kullanılan öğrenme algoritması bu stratejilere bağlı olarak değişmektedir. Genel olarak 3 öğrenme stratejisinin uygulandığı görülmektedir (Öztemel, 2003). Bunlar:

✓ **Öğretmenli (supervised) öğrenme;** bu tip stratejide öğrenen sistemin olayı öğrenebilmesine bir öğretmen yardımcı olmaktadır. Öğretmen sisteme öğrenilmesi istenen olay ile ilgili örnekleri girdi/çıkı seti olarak verir. Yani, her örnek için hem girdiler hem de o girdiler karşılığında oluşturulması gereken çıktılar sisteme gösterilirler. Sistemin görevi girdileri öğretmenin belirlediği çıktılara haritalamaktır. Bu sayede olayın girdileri ile çıktıları arasındaki ilişkiler öğrenilmektedir. Çok katmanlı algılayıcı ağı, bu stratejiyi kullanan ağlara örnek olarak verilebilir.

✓ **Destekleyici (reinforcement ) öğrenme;** bu tür stratejide de öğrenen sisteme bir öğretmen yardımcı olur. Fakat öğretmen her girdi seti için olması gereken (üretilmesi gereken) çıktı setini sisteme göstermek yerine sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Sistem, öğretmenden gelen bu sinyali dikkate alarak öğrenme sürecini devam ettirir. LVQ ağı bu stratejiyi kullanan sistemlere örnek olarak verilebilir.

✓ **Öğretmensiz (unsupervised) öğrenme;** bu tür stratejide sistemin öğrenmesine yardımcı olan herhangi bir öğretmen yoktur. Sisteme sadece girdi değerleri gösterilir. Örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendisine öğrenmesi beklenir. Bu, daha çok sınıflandırma problemleri için kullanılan bir stratejidir. Yalnız sistemin öğrenmesi bittikten sonra çıktıların ne anlama geldiğini gösteren etiketlendirmenin kullanıcı tarafından yapılması gerekmektedir. ART ağları bu stratejiyi kullanan sistemlere örnek olarak verilebilir.

✓ **Karma stratejiler;** yukarıdaki 3 stratejiden birkaçını birlikte kullanarak öğrenme gerçekleştiren ağlarda vardır. Burada kısmen öğretmenli, kısmen ise öğretmensiz olarak öğrenme yapan ağlar kastedilmektedir. Radial tabanlı yapay sinir ağları (RBN) ve olasılık tabanlı ağlar (PBNN) bunlara örnek olarak verilebilir.

### 3.3.2 Öğrenme Kuralları

Yapay sinir ağları gibi öğrenen sistemlerde öğrenme, yukarıda anlatılan, stratejilerden hangisi uygulanırsa uygulansın bazı kurallara göre gerçekleştirilmektedir. Bu kuralların bazıları

çevrimiçi (on-line) bazıları ise çevrimdışı (off-line) çalışmaktadır (Öztemel, 2003).

✓ **Çevrimiçi (on-line) öğrenme kuralları;** bu kurallar gerçek zamanlı çalışabilmektedir. Bu kurallara göre öğrenen sistemler gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte diğer taraftan ise öğrenmeye devam etmektedir. ART ağının öğrenme kuralı ve Kohonen öğrenme kuralı bu sınıfta bulunan öğrenme kurallarına örnek olarak verilebilir.

✓ **Çevrimdışı (off-line) öğrenme kuralları;** çevrimdışı öğrenme kurallarına dayalı öğrenen sistemler kullanıma alınmadan önce örnekler üzerinde eğitilirler. Bu kuralları kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. Sistemin öğrenmesi gereken yeni bilgiler söz konusu olduğunda sistem kullanımdan çıkarılmakta ve çevrimdışı olarak yeniden eğitilmektedir. Eğitim tamamlanınca sistem tekrar kullanıma alınmaktadır. Yapay sinir ağlarında yaygın olarak kullanılan Delta Öğrenme Kuralı bu tür öğrenmeye örnek olarak verilebilir.

✓ **Öğrenme kurallarından bazıları;** öğrenme sistemlerinde kullanılan değişik öğrenme kuralları vardır. Yapay sinir ağlarında bu öğrenme kurallarının çoğu Hebb kuralına dayanmaktadır. Bu çalışmada kullanılacak olan, çok katmanlı algılayıcı ve burada kullanılacak olan öğrenme kuralı detaylı olarak ele alınacaktır (bkz. 3.4). Bu bölümde ise kurallara genel bir bakış yapılacaktır.

1. Hebb kuralı; bilinen en eski öğrenme kuralıdır. Diğer öğrenme kurallarının temelini oluşturmaktadır. 1949 yılında geliştirilen bu kurala göre, bir hücre (yapay sinir ağı elemanı) diğer bir hücreden bilgi alırsa ve her iki hücrede aktif ise (matematik olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantı kuvvetlendirilmelidir. Diğer bir deyişle bu kural şu şekilde özetlenebilir. Bir hücre kendisi aktif ise bağlı olduğu hücreyi aktif yapmaya pasif ise pasif yapmaya çalışmaktadır. Diğer öğrenme kurallarının çoğu bu felsefeyi baz alarak geliştirilmiştir.

2. Hopfield kuralı; bu kural Hebb kuralına benzemektedir. Yapay sinir ağı elemanlarının bağlantılarının ne kadar kuvvetlendirilmesi veya zayıflatılması gerektiği belirlenir. Eğer beklenen çıktı ve girdiler ikisi de aktif/pasif ise öğrenme katsayısı kadar ağırlık değerleri kuvvetlendir/zayıflat denmektedir. Yani, ağırlıkların kuvvetlendirilmesi veya zayıflatılması öğrenme katsayısı yardımı ile gerçekleştirilmektedir. Öğrenme katsayısı genel olarak 0-1 arasında kullanıcı tarafından atanan sabit ve pozitif bir değerdir.

3. Delta kuralı; bu kural Hebb kuralının biraz daha geliştirilmiş şeklidir. Bu kurala göre

beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak için yapay sinir ağının elemanlarının bağlantılarının ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanarak geliştirilmiştir. Ağın ürettiği çıktı ile üretilmesi gereken (beklenen) çıktı arasındaki hatanın karelerinin ortalamasını enazlamak hedeflenmektedir.

4. Kohonen kuralı; bu kurala göre ağın elemanları (hücreleri) ağırlıklarını değiştirmek için birbirleri ile yarışır. En büyük çıktıyı üreten hücre kazanan hücre olmakta ve bağlantı ağırlıkları değiştirilmektedir. Bu, o hücrenin yakınındaki hücelere karşı daha kuvvetli hale gelmesi demektir. Hem kazanan elemanların hem de komşuları sayılan elemanların (hücrelerin) ağırlıklarını değiştirmesine izin verilmektedir.

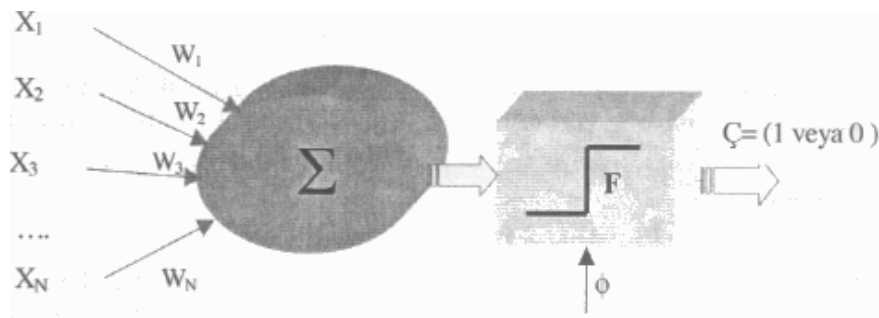
### 3.4 Perceptron ve MLP (Multi Layer Perceptron)

#### 3.4.1 Perceptron (Basit Algılayıcı Modeli)

İlk defa 1958 yılında Rosenblat tarafından örüntü (şekil) sınıflandırma amacı ile geliştirilmiştir.

Model iki aşamalı olarak ele alınıp incelenecektir, böylece bu temel konu üzerinde daha detaylı bir çalışma sağlanmış olacaktır.

**1. Basit algılayıcıların yapısı:** Perceptron bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanmaktadır. Ağın çıktısı bir veya sıfırdan oluşan mantıksal (boolean) değerdir. Çıktının değerinin hesaplanmasında eşik değer fonksiyonu kullanılır. Perceptron'un yapısı Şekil 3.3 de gösterildiği gibidir;



Şekil 3.3 Bir basit algılayıcı yapısı (Öztemel, 2003).

Şekilden de görüldüğü gibi, perceptron eğitilebilen tek bir yapay sinir hücresinden (proses elemanından) oluşur. Eğitilebilirden kasıt ağırlıkların (W) değiştirilebilir olması demektir. Girdiler proses elemanına gösterilirler. Her girdi setine karşılık gelen çıktı değerleri de ağa gösterilir. Daha sonra öğrenme kuralına göre ağın çıktı değeri hesaplanır. Eğer ağın çıktısı

olması gereken çıktıdan farklı ise ağırlıklar ve eşik değerleri değiştirilir. Değişikliğin nasıl yapılacağını ise öğrenme kuralı belirler. Girdilere karşılık gelene çıktı değerleri bir veya sıfırdan oluşmaktadır.

Yukarıda belirtildiği gibi eşik değeri, aktivasyon fonksiyonunun konumunu belirlemek için kullanılır, Şekil 3.4 eşik değerinin sıfır ve “a” pozitif değerleri alması durumunda aktivasyon fonksiyonunun konumunu göstermektedir.



Şekil 3.4 Eşik değerinin aktivasyon fonksiyonunun konumuna etkisi (Öztemel, 2003).

**2. Basit algılayıcı öğrenme kuralı:** Basit algılayıcıların öğrenme kuralı adım adım aşağıda açıklanmıştır:

✓ **Adım 1:** ağa girdi setini ve ona karşılık olarak beklenen çıktı gösterilir (X,B). Burada birden fazla girdi değeri olabilir. Yani  $X = (x_1, x_2, x_3, \dots, x_N)$  demektir. Çıktı değeri ise “1” ve “0” değerlerinden birisini alır.

✓ **Adım 2:** perseptron ünitesine gelen net girdi şu şekilde hesaplanır;

$$NET = \sum_{i=1}^m w_i x_i \quad (3.3)$$

✓ **Adım 3:** perseptron ünitesinin çıktısı hesaplanır. Net girdinin eşik değerinden büyük veya küçük olmasına göre çıktı değeri “0” ve “1” değerlerinden birisini alır. Yani;

$$C = \begin{cases} 1 & \text{Eğer } NET > \phi \\ 0 & \text{Eğer } NET \leq \phi \end{cases} \quad (3.4)$$

Eğer gerçekleşen çıktı ile beklenen çıktı aynı olursa ağırlıklarda herhangi bir değişiklik olmaz. Ağ, beklenmeyen bir çıktı üretmiş ise o zaman iki durum söz konusudur:

1. Ağın beklenen çıktısı “0” değeridir. Fakat NET girdi eşik değerinin üstündedir. Yani ağın gerçekleşen çıktısı “1” değeridir. Bu durumda ağırlık değerleri azaltılmaktadır. Ağırlıkların değişim oranı girdi değerlerinin belirli bir oranı kadardır. Yani vektörel olarak;

$$W_n = W_0 - \lambda X \quad (3.5)$$

olur. Burada “ $\lambda$ ” öğrenme katsayısıdır. Ağırlıkların değişim miktarlarını belirlemekle ve sabit bir değer olarak alınmaktadır.

2. Beklenen çıktının “1” olması ve ağırlık gerçekte çıktısının “0” olması durumudur. Yani net girdi eşik değerinin altındadır. Bu durumda ağırlıkların değerinin artırılması gerekmektedir. Yani vektörel olarak;

$$W_n = W_0 + \lambda X \quad (3.6)$$

olacaktır.

✓ **Adım 4:** yukarıdaki adımları bütün girdi setindeki örnekler için doğru sınıflandırmalar yapılıncaya kadar ilk uç adımdaki işlemler tekrarlanır.

### 3.4.2 MLP (çok Katmanlı Algılayıcı)

Bir yapay sinir ağının öğrenmesi istenen olayların girdi ve çıktıları arasındaki ilişkiler doğrusal olmayan ilişkiler olursa, bu tür olayların öğrenilmesi için daha gelişmiş modellere ihtiyaç vardır. Bu bölümde anlatılan Çok Katmanlı Algılayıcı modeli bunlardan birisidir. Olayın doğrusal olup olmaması ne demektir? Bu konuyu iyi anlayabilmek için ünlü XOR problemine bakmak gerekir. Bu problemin özelliği doğrusal olmayan bir ilişkiyi göstermesidir. Yani çıktıların arasına bir doğru veya doğrular çizerek onları iki veya daha fazla sınıfa ayırmak mümkün değildir. Bu problem Tablo 3.1 de gösterildiği gibidir. Basit algılayıcı ile bu problemi çözmek mümkün değildir.

Tablo 3.1 XOR problemi

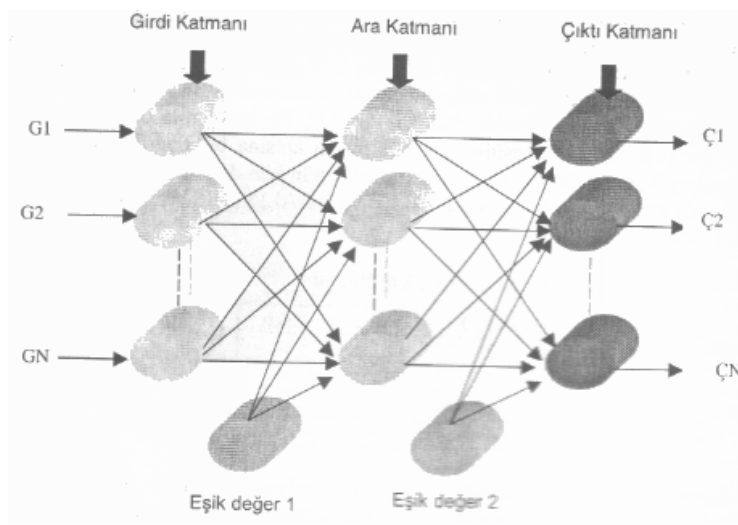
Girdi 1:	Girdi 2:	Çıktı:
0	0	0
0	1	1
1	0	1
1	1	0

Bu problem, hemen hemen her yapay sinir ağını anlatan her kitapta örnek olarak

verilmektedir. Bundan çok yaygın olarak bahsedilmesinin nedeni şöyle açıklanabilir. Minsky özellikle basit algılayıcı (perseptron) modelinin bu probleme çözüm üretmediğini göstermiş ve yapay sinir ağlarının doğrusal olmayan problemlere çözüm üretmediğini iddia ederek bilimsel araştırmaların durmasına neden olmuştur. Çünkü günlük olayların çoğu (hemen hemen hepsi) doğrusal olmayan bir nitelik taşımaktadır. XOR probleminin çözülememesinden sonra neredeyse bütün çalışmalar durmuş sadece birkaç araştırmacı çalışmalara devam etmiştir. Bu problemi çözerek yapay sinir ağlarına tekrar dikkatleri çekmeyi başarmışlardır. O nedenle bu problem yapay sinir ağı araştırmalarında bir kilometre taşı olarak görülmektedir.

XOR problemini çözmek amacı ile yapılan çalışmalar sonucu çok katmanlı algılayıcı modeli ÇKA geliştirilmiştir. Rumelhart ve arkadaşları tarafından geliştirilen bu modele hata yayma modeli veya geriye yayım modeli (backpropagation network) de denilmektedir. ÇKA modeli yapay sinir ağlarına olan ilgiyi çok hızlı bir şekilde artırmış ve yapay sinir ağları tarihinde yeni bir dönemin başlamasına neden olmuştur. Bu model günümüzde mühendislik problemlerinin hemen hemen hepsine çözümler üretebilecek bir güce sahiptir. Özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektiren problemler için çok önemli bir çözüm aracıdır. Bu model Delta Öğrenme Kuralı denilen bir öğrenme yöntemini kullanmaktadır. Bu kural aslında ADALINE ve basit algılayıcı modellerinin öğrenme kurallarının geliştirilmiş bir şeklidir. Temel amacı ağın beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmektir. Bunu hatayı ağı yayarak gerçekleştirdiği için bu ağı hata yayma ağı da denmektedir.

**a. ÇKA modelinin yapısı;** ÇKA ağlarının yapısı Şekil 3.5 de gösterildiği gibidir. Şekilden de görüldüğü gibi ÇKA ileriye doğru bağlantılı ve 3 katmandan oluşan bir ağıdır. Bunlar;



Şekil 3.5 ÇKA modeli.

✓ **Girdi katmanı:** dış dünyadan gelen girdileri (G1, G2, ..., GN) olarak ara katmana

gönderir. Bu katmanda bilgi işleme olmaz. Gelen her bilgi geldiği gibi bir sonraki katmana gider. Birden fazla girdi gelebilir. Her proses elemanının sadece bir tane girdisi ve bir tane çıktısı vardır. Bu çıktı bir sonraki katmanda bulunan bütün proses elemanlarına gönderilir. Yani, girdi katmanındaki her proses elemanı bir sonraki katmanda bulunan proses elemanlarının hepsine bağlıdır.

✓ **Ara katmanlar:** ara katmanlar girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Bir ÇKA ağında birden fazla ara katman ve her katmanda birden fazla proses elemanı olabilir. Ara katmandaki her proses elemanı bir sonraki katmandaki bütün proses elemanlarına bağlıdır.

✓ **Çıktı katmanı:** ara katmandan gelen bilgileri işleyerek ağa girdi katmanından verilen girdilere karşılık ağın ürettiği çıktıları (Ç1, Ç2, ..., ÇN) belirleyerek dış dünyaya gönderir. Bir çıktı katmanında birden fazla proses elemanı olabilir. Her proses elemanı bir önceki katmanda bulunan bütün proses elemanlarına bağlıdır. Her proses elemanının sadece bir tane çıktısı vardır.

ÇKA ağında bilgiler girdi katmanından ağa sunulur ve ara katmanlardan geçerek çıktı katmanına gider ve ağa sunulan girdilere karşılık ağın cevabı dış dünyaya iletilir.

ÇKA ağı öğretmenli öğrenme stratejisini kullanır. Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar (beklenen çıktı) verilmektedir. Ağ kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir.

**b. ÇKA ağının öğrenme kuralı;** ÇKA ağları öğretmenli öğrenme stratejisine göre çalışırlar. Yani; bu ağlara eğitim sırasında hem girdiler hem de o girdilere karşılık üretilmesi gereken (beklenen) çıktılar gösterilir. Ağın görevi her girdi için o girdiye karşılık gelen çıktıyı üretmektir. ÇKA ağının öğrenme kuralı en küçük kareler yöntemine dayalı Delta Öğrenme Kuralının genelleştirilmiş halidir. O nedenle öğrenme kuralına *Genelleştirilmiş Delta Kuralı* da denmektedir. Ağın öğrenebilmesi için eğitim seti adı verilen ve örneklerden oluşan bir sete ihtiyaç vardır. Bu set içinde her örnek için ağın hem girdiler hem de o girdiler için ağın üretmesi gereken çıktılar belirlenmiştir. Genelleştirilmiş Delta Kuralı iki safhadan oluşur.

1. *İleri doğru hesaplama:* ağın çıktısını hesaplama safhasıdır.

2. *Geriye doğru hesaplama:* ağırlıkları değiştirme safhasıdır.

1. İleri doğru hesaplama; bu safhada bilgi işleme eğitim setindeki bir öreğin Girdi Katmanından ( $G_1, G_2...$ ) ağa gösterilmesi ile başlar. Daha önce belirtildiği gibi, girdi katmanında herhangi bir bilgi işleme olmaz. Gelen girdiler hiç bir değişiklik olmadan ara katmana gönderilir. Yani girdi katmanındaki  $k$ . proses elemanının çıktısı şu şekilde belirlenir.

$$C_k^i = G_k \quad (3.7)$$

Ara katmandaki her proses elemanı girdi katmanındaki bütün proses elemanlarından gelen bilgileri bağlantı ağırlıklarının ( $A_1, A_2...$ ) etkisi ile alır. Ara katmandaki proses elemanlarına gelen net girdi, şu formül kullanılarak hesaplanır.

$$NET_j^a = \sum_{k=1}^n A_{kj} C_k^i \quad (3.8)$$

Burada  $A_{kj}$   $k$ . girdi katmanı elemanını  $j$ . ara katman elemanına bağlayan bağlantının ağırlık değerini göstermektedir.  $j$ . ara katman elemanın çıktısı ise bu net girdinin aktivasyon fonksiyonundan (genellikle sigmoid fonksiyonundan) geçirilmesiyle hesaplanır. Uygulamada genellikle bu fonksiyon kullanılmakla beraber, kullanılması zorunlu değildir. Önemli olan burada türevi alınabilir bir fonksiyon kullanmaktır. Geriye doğru hesaplamada burada kullanılan fonksiyonun türevinin alınacağını unutmamak gerekir. Sigmoid fonksiyonu kullanılması halinde çıktı,

$$C_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}} \quad (3.9)$$

şeklinde olacaktır. Burada  $\beta_j$ , ara katmanda bulunan  $j$ . elemana bağlanan eşik değer elemanın ağırlığını göstermektedir. Bu eşik değeri ünitesinin çıktısı sabit olup 1 e eşittir. Ağırlık değeri ise sigmoid fonksiyonunun oryantasyonunu belirlemek üzere konulmuştur. Eğitim esnasında ağ bu değeri kendisi belirlemektedir.

Ara katmanın bütün proses elemanları ve çıktı katmanının proses elemanlarının çıktıları aynı şekilde kendilerine gelen NET girdinin hesaplanması ve sigmoid fonksiyonundan geçirilmesi sonucu belirlenirler. Çıktı katmanından çıkan değerler, yani çıktıları, ( $\Ç_1, \Ç_2, \dots$ ) bulununca ağın ileri hesaplama işlemi tamamlanmış olur.

2. Geriye doğru hesaplama; ağa sunulan girdi için ağın ürettiği çıktı ağın beklenen çıktıları ( $B_1, B_2, \dots$ ) ile karşılaştırılır. Bunların arasındaki fark hata olarak kabul edilir. Amaç bu hatanın düşürülmesidir. O nedenle geriye hesaplamada bu hata ağın ağırlık değerlerine

dağıtılarak bir sonraki iterasyonda hatanın azaltılması sağlanır. Çıktı katmanındaki m. proses elemanı için oluşan hata (Em),

$$E_m = B_m - C_m \quad (3.10)$$

olacaktır. Bu bir proses elemanı için oluşan hatadır. Çıktı katmanı için oluşan toplam hatayı (TH) bulmak için bütün hataların toplanması gerekir. Bazı hata değerleri negatif olacağından toplamın sıfır olmasını önlemek amacı ile ağırlıkların kareleri hesaplanarak sonucun karekökü alınır. ÇKA ağının eğitilmesindeki amaç bu hatayı en azlamaktır. TH şu formül ile bulunur.

$$TH = \frac{1}{2} \sum_m E_m^2 \quad (3.11)$$

Toplam hatayı enazlamak için bu hatanın kendisine neden olan proses elemanlarına dağıtılması gerekmektedir. Bu ise proses elemanlarının ağırlıklarını değiştirmek demektir. Ağırlıkların ağırlıklarını değiştirmek için iki durum söz konusudur.

- ✓ Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi,
- ✓ Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi.

#### a. Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi

Ara katmandaki j. proses elemanını çıktı katmanındaki m. proses elemanına bağlayan bağlantının ağırlığındaki değişim miktarına  $\Delta A$  denirse; herhangi bir t zamanında (t iterasyonda) ağırlığın değişim miktarı şöyle hesaplanır.

$$\Delta A_{jm}^a(t) = \lambda \delta_m C_j^a + \alpha \Delta A_{jm}^a(t-1) \quad (3.12)$$

Burada  $\lambda$  öğrenme katsayısını,  $\alpha$  momentum katsayısını göstermektedir. Öğrenme katsayısı ağırlıkların değişim miktarını, momentum katsayısı ise ÇKA ağının öğrenmesi esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlarlar. Bu konu aşağıda tekrar tartışılacaktır. Eşitlikteki  $\delta_m$  ise m. çıktı ünitesinin hatasını göstermektedir. Şu şekilde hesaplanır.

$$\delta_m = f'(NET)E_m \quad (3.13)$$

Buradaki  $f'(NET)$  aktivasyon fonksiyonunun türevidir. Sigmoid fonksiyonunun kullanılması durumunda;

$$\delta_m = C_m(1 - C_m)E_m \quad (3.14)$$

olacaktır. Değişim miktarı hesaplandıktan sonra ağırlıkların t. iterasyondaki yeni değerleri şöyle olacaktır.

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t) \quad (3.15)$$

Benzer şekilde eşik değer ünitesinin de ağırlıklarını değiştirmek gerekmektedir. Onun için öncelikle değişim miktarını hesaplamak gerekir. Eğer çıktı katmanında bulunan proses elemanlarının eşik değer ağırlıkları  $\beta^c$  ile gösterilirse; bu ünitenin çıktısının sabit ve t olması nedeni ile değişim miktarı,

$$\Delta\beta_m^c(t) = \lambda\delta_m + \alpha\Delta\beta_m^c(t-1) \quad (3.16)$$

olacaktır. Esik değer t. iterasyondaki ağırlığının yeni değeri ise,

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta\beta_m^c(t) \quad (3.17)$$

şeklinde olacaktır.

### **b. Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi**

Dikkatli incelenirse, ara katman ile çıktı katmanı arasındaki ağırlıkların değişiminde her ağırlık için sadece çıktı katmanındaki bir proses elemanının hatası dikkate alınmıştır. Bu hataların oluşmasında girdi katmanı ve ara katman arasındaki ağırlıkların (varsa iki ara katman arasındaki ağırlıkların) payı vardır. Çünkü, en son ara katmana gelen bütün bilgiler girdi katmanı veya önceki ara katmandan gelmektedir. O nedenle girdi katmanı ile ara katman arasındaki (veya iki ara katman arasındaki) ağırlıkların değiştirilmesinde çıktı katmanındaki proses elemanlarının hepsinin hatasından payını alması gerekir. Bu ağırlıklardaki değişimi (mesela girdi katmanı ile ara katman arasındaki ağırlıkların değişimi)  $\Delta A^i$  ile gösterilirse değişim miktarı;

$$\Delta A_{kj}^i(t) = \lambda\delta_j^a C_k^i + \alpha\Delta A_{kj}^i(t-1) \quad (3.18)$$

olacaktır. Buradaki hata terimi  $\delta^a$  ise şöyle hesaplanacaktır.

$$\delta_j^a = f'(NET) \sum_m \delta_m A_{jm}^a \quad (3.19)$$

Aktivasyon fonksiyonu olarak sigmoid fonksiyonu düşünülürse bu hata değeri şu şekilde hesaplanacaktır.

$$\delta_j^a = C_j^a (1 - C_j^a) \sum_m \delta_m A_{jm}^a \quad (3.20)$$

Hata değeri hesaplandıktan sonra yukarıda verilen eşitlik ile değişim miktarını bulmak mümkün olur. Ağırlıkların yeni değerleri ise,

$$A_{kj}^a(t) = A_{kj}^a(t-1) + \Delta A_{kj}^a(t) \quad (3.21)$$

şeklinde olacaktır. Benzer şekilde, eşik değer ünitesinin yeni ağırlıkları da yukarıdaki gibi hesaplanır. Ara katman eşik değer ağırlıkları  $\beta^a$  ile gösterilirse değişim miktarı,

$$\Delta \beta_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1) \quad (3.22)$$

olacaktır. Ağırlıkların yeni değerleri ise t. iterasyonda şöyle hesaplanacaktır.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta \beta_j^a(t) \quad (3.23)$$

Böylece ağırlıklarının hepsi değiştirilmiş olacaktır. Bir iterasyon hem ileri hem de geriye hesaplamaları yapılarak tamamlanmış olacaktır. İkinci bir örnek verilerek sonraki iterasyona başlanır ve aynı işlemler öğrenme tamamlanıncaya kadar yinelenir.

Eğitme işlemini de içine alacak şekilde ÇKA ağıının genel çalışma prosedürü ise şu şekilde sıralanabilir:

- ✓ Örneklerin toplanması; ağıın çözmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağıın eğitilmesi için örnekler toplandığı gibi (eğitim seti) ağıın test edilmesi için de örneklerin (test seti) toplanması gerekmektedir. Ağıın eğitilmesi sırasında test seti ağa hiç gösterilmez. Eğitim setindeki örnekler tek tek gösterilerek ağıın olayı öğrenmesi sağlanır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağıın performansı ölçülür. Hiç görmediği örnekler karşısındaki başarısı ağıın iyi öğrenip öğrenmediğini onaya koymaktadır.
- ✓ Ağıın topolojik yapısının belirlenmesi; öğrenilmesi istenen olay için oluşturulacak olan ağıın topolojik yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane proses elemanı ve kaç tane çıktı elemanı olması gerektiği bu adımda belirlenmektedir.

- ✓ Öğrenme parametrelerinin belirlenmesi; ağıın öğrenme katsayısı, proses elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
- ✓ Ağırlıkların başlangıç değerlerinin atanması; proses elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin atanması yapılır. Başlangıçta genellikle rasgele değerler atanır. Daha sonra ağı uygun değerleri öğrenme sırasında kendisi belirler.
- ✓ Öğrenme setinden örneklerin seçilmesi ve ağı gösterilmesi ağıın öğrenmeye başlaması ve yukarıda anlatılan öğrenme kuralına uygun olarak ağırlıkları değıştirmesi için ağı örnekler (Girdi/Çıktı değerleri) belirli bir düzeneğe göre gösterilir.
- ✓ Öğrenme sırasında ileri hesaplamaların yapılması; yukarıda anlatıldığı şekilde sunulan girdi için ağıın çıktı değerleri hesaplanır.
- ✓ Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması; ağıın ürettiğı hata değerleri bu adımda hesaplanır.
- ✓ Ağırlıkların değıştirilmesi; yukarıda anlatıldığı gibi geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değıştirilmesi yapılır.

Yukarıdaki adımlar ÇKA ağıının öğrenmesi tamamlanıncaya, yani gerçekleşen çıktılar ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye, kadar devam eder. Ağıın öğrenmesi için bir durdurma kriterinin olması gerekmektedir. Bu ise genellikle üretilen hatanın belirli bir düzeyin altına düşmesi olarak alınmaktadır (Öztemel, 2003).

### 3.5 Yazılımsal ve Donanımsal Yapay Sinir Ağı Tasarım Elemanları

Günümüzde yapay sinir ağıları uygulamalarının çoğı yazılım teknolojisi olarak görülmektedir. Belirli bir modelin yazılımı gerçeklenmekte ve seri bilgisayarlarda çalıştırılarak sorunların çözülmesi istenmektedir. Yapay sinir ağılarının paralellik gibi bazı özelliklerinin gösterilebilmesi için özel donanım teknolojisine ihtiyaç vardır. Yapay sinir donanımları ticari olarak genellikle şu alanlarda kendini göstermektedir;

- Optik karakter tanıma,
- Ses tanıma,
- Trafik izleme,

- Veri madenciliği ve filtreleme.

Yapay sinir ağları için özel donanım geliştirilmesinin birçok yararı vardır. Bunlar arasında şunları saymak mümkündür.

- Hız; birçok uygulamada sistemin karar verme hızı çok önemlidir. Günümüzdeki en hızlı seri işlemciler bile özellikle çok sayıda girdi parametresinin gerektiği ve ağı boyutlarının çok büyük olması durumunda gerçek zamanlı kullanıma ve öğrenmeye uygun olmamaktadırlar. Sistemin gerçek zamanlı kullanılması için özel bir donanım ile sistemin hızının artırılması istenebilir. Örneğin birbirine paralel bağlanan işlemciler yardımıyla uygulamanın hızı artırılabilir.
- Güvenirlilik; özel donanım sayesinde sistemin güvenirliliği artırılabilir. Özellikle donanım hatalarının kontrol altına alınması sağlanacağından sisteme olan inanç artacaktır.
- Özel çalışma koşulları; donanımın probleme uygun bir şekilde tasarlanması özellikle sistemin boyutları, ağırlığı gibi konularda en uygun durumun seçilmesini sağlar.
- Güvenlik; özel donanım ile sistemin güvenliği ve koruması da daha rahat kontrol altına alınmaktadır.

Yapay sinir ağlarının verimli bir şekilde kullanılması için geliştirilecek olan özel donanımların geliştirilmesi çok kolay olmamaktadır. Özellikle resimlerin metin haline dönüştürülmesi ve girdi olarak sisteme tanıtılması zordur. Ayrıca donanımı geliştirmek için gereken zaman da oldukça uzundur. O nedenle, bir donanım geliştirilmeden veya satın almadan düşünülmesi gereken sistemin donanım maliyeti, bu donanım üzerinde çalışacak olan yazılımların maliyeti ve geliştirilme zamanlarıdır. Satın alınacak veya geliştirilecek sistemin bu açıdan katlanılabilir olması ve neticesinde ondan elde edilecek olan yarara değmesi gerekir. Değişik şekillerde yapay sinir ağı donanımlarının geliştirildiğini görmek mümkündür. Hangi donanımın daha iyi olacağı ağı öğreneneği probleme göre değişmektedir. Yapay sinir ağlarını geliştirmeye yönelik çalışmaların merkezinde VLSI yongaları (chips) bulunmaktadır. Yapay sinir ağları donanımlarının iki yönlü geliştirildiği görülmektedir:

- ✓ Genel amaçlı donanımlar; bunlar nörobilgisayarlar gibi her problem için uygulanabilir donanımlardır.
- ✓ Özel amaçlı donanımlar; bunlar ise belirli bir spesifik problemi çözmek için geliştirilmiş ve belirli bir amaç için kullanılan donanımlardır.

Piyasada bu iki yaklaşıma dayanarak geliştirilmiş ve kullanılmış donanım çeşitlerine örnekleri şu şekilde sıralamak mümkündür.

- **Nörobilgisayarlar;** bu bilgisayarlar tamamen yapay sinir ağı teknolojisine dayanarak geliştirilmiştir. Genellikle çok hız gerektiren ve büyük boyutlardaki problemlerin çözülmesi için geliştirilmişlerdir. Siemens tarafından geliştirilmiş *Synapse 1* nörobilgisayarı bir ana bilgisayara ethernet kartı ile bağlanıp çalışabilen bir bilgisayardır ve 8 tane *MA-16* sistolik dizi yongasını kullanmaktadır. Bu bilgisayarın performansı 25 MHz ile saniyede 3.2 milyar çarpım (16 bit \* 16 bit) ve toplama işlemi yapabilecek bir güce sahiptir.
- **PC hızlandırıcılar ve kartları;** bunlar bilinen bilgisayarlara takılabilen kartlardır. Bunlar sistemin hızını artırmakla beraber daha çok küçük boyutlardaki problemlerin çözülmesinde tavsiye edilmektedirler. Nörobilgisayarlardan daha ucuzdurlar. IBM'in geliştirdiği *ZISC PCI* kartları saniyede her birisi 8 bitlik 64 elemandan oluşan 165000 örüntüyü işleme yetenekleri vardır.
- **Yongalar (Chips);** bunlar yukarıdaki iki tür donanımı oluşturmak için kullanılan sistemler olabilecekleri gibi probleme özel makinelerin (örneğin makinelerin yanında kullanılacak aparatların) oluşturulmasında kullanılırlar.
- **Gömülü mikroişlemciler;** bunlar belirli bir yapay sinir ağı uygulamasını koştan fakat klavye, ekran, disk gibi aksamaları olmayan bilgisayarlar olarak görülebilir.

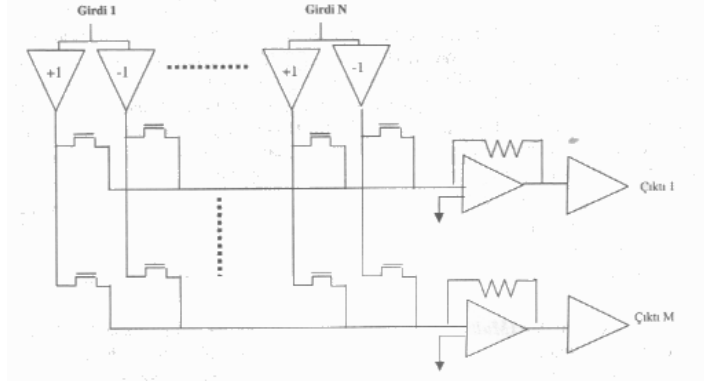
Yukarıdaki donanımların uygulamalarında ise 3 grupta toplamak mümkündür. Her kategoride de öğrenme ve tanıma fonksiyonlarını gerçekleştirmek için değişik donanım yapıları görülmektedir. Bunlar;

- i. Dijital uygulamalar,
- ii. Analog uygulamalar,
- iii. Karma sistemler.

### 3.5.1 Analog Yapay Sinir Ağı Donanımları

Analog sistemlerin özellikle hızları onları cazip hale getirmektedir. Fakat çarpma ve toplama işlemlerinde duyarlılıkları dijital sistemler kadar iyi değildir. Sıcaklık ve ısı şartlarına, imalat sırasında kullanılan toleranslara bağlı olarak çok küçük sinyalleri sınırlamaktadır. Diğer bir problem ise, ağırlık değerlerinin uzun süreli saklanması zorluğudur. Analog ağların tasarlanmasında kullanılan yaklaşım nöromorfik tasarımıdır. Bu tasarımda, elektronik devre

daha önce anlatılan biyolojik sinir hücreleri ve sinapsların davranışlarını mümkün olduğunca taklit etmeye çalışır. Şekil 3.6 temel bir analog yapay sinir ağı yonga yapısını göstermektedir

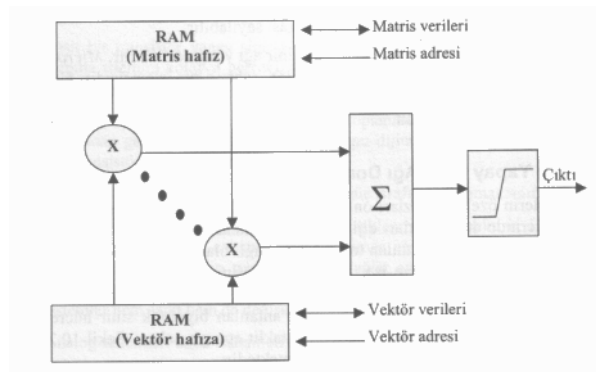


Şekil 3.6 Temel analog yapay sinir ağı yongası (Öztemel, 2003).

Analog yapay sinir ağı donanımları, doğrusal ve doğrusal olmayan aygıtlardan oluşan elektronik devrelerdir. Transistörler, resistörler ve kapasitörler bu aygıtlarda kullanılmaktadır. Bu şekilde geliştirilmiş *Sinaptik Silikon Retina* sistemi resim işlemede biyolojik retinanın fonksiyonlarını taklit ederek başarılı sonuçlar göstermiştir. Bu konuda verilebilecek diğer bir örnek ise, *Intel 8017NW ETANN* eğitilebilen bir donanım olup 64 hücre ve 10280 ağırlık değerini saklayabilecek güçteki nörobilgisayardır (Öztemel, 2003).

### 3.5.2 Dijital Yapay Sinir Ağı Donanımları

Dijital yapay sinir ağı donanımlarında, ağ (network) üzerindeki bütün hesaplamalarda kullanılan değerler büyüklüğü belirlenmiş ikili (binary) vektörlerdir. Dijital teknolojinin verilerin gürültüden ayrılmış olması, ağırlıkları saklamak için RAM kullanılması, çarpma ve toplama işlemlerinde duyarlılığın (doğruluk derecesinin) artması ve özellikle de mevcut sistemlere entegrasyonunun sağlanmasındaki kolaylıklar gibi olumlu yanları vardır. Şekil 3.7 da bir dijital yapay sinir ağı donanımının yapısı gösterilmektedir.



Şekil 3.7 Dijital yapay sinir ağı yonga yapısı (Öztemel, 2003).

Dijital yapay sinir ağı, ağa girdi olarak verilen bilgilerdeki gürültüye karşı analog sistemlere göre daha toleranslıdır. Hem sabit hem de değişken ağırlık matrisleri saklayabilirler. Programlanabilir elemanları içerebilirler. Bunların yanında bazı olumsuz yanları da vardır. Özellikle, çarpma ve ekleme işlemlerindeki yavaşlık ve dış dünyadan gelen bilgiler genellikle analog bilgiler olması ön plana çıkar. Onların dijital bilgilere dönüştürülmesi için dönüştürücülere ihtiyaç olması dezavantajlara örnek olarak verilebilir. Uygulanan dijital teknolojiler arasında şunları saymak mümkündür (Öztemel, 2003).

Dilim yapıları (*slice architectures*); dilim yapısı geleneksel dijital işlemcilerdeki bit dilimlerine benzeyen yapay sinir ağı yapılarıdır. İstenilen boyutta yapay sinir ağı oluşturacak geliştirme blokları sunarlar. Bu konuda “*Philips Lneuro chip*”, “*The Micro devices*” *MDI20* ve *Neurologic NLX-420* işlemcileri örnek olarak gösterilebilir.

Çoklu-işlemciler (Multi-Processors); bir yonga üzerine birden fazla işlemcinin konulması sonucu ortaya çıkan işlemcilerdir. Genel olarak bu kapsamda iki grup prosesor görülmektedir. Bunlardan birisi *SIMD* (*Single Instruction Multiple Data*) olarak bilinen yongalar ve *sistolik dizilerdir* (*systolic arrays*). *SIMD* yongalarında, bütün işlemciler aynı komutu farklı veriler kullanarak işletirler. *Sistolik* dizilerde ise her işlemci hesaplamanın bir kısmını gerçekleştirir ve sonuçlarını dizideki bir sonraki işlemciye göndermektedir. *SIMD* yongalarına *Inova N64000* (64 işlem elemanı olan), *the HNC 100NAP* (4 proses elemanı) ve *Siemens MA-16* (hızlı matris operasyonları yapabilen) örnek olarak verilebilir.

Radyal temelli fonksiyon (RBF) ağı; RTF ağı genel olarak verilere dayanarak belirlenmiş etki alanlarını belirleyen vektörleri kullanırlar. Her bir vektör çok büyük boyutlu (*hyper-dimentional*) bir uzayın bir boyutu olarak düşünülebilir. Geliştirilen donanım bu vektörleri işleyebilecek işlemcilerden oluşmaktadır. Geliştirilmiş RTF donanımlarına örnek olarak, *IBM ZISC* (*Zero Instruction Set Computer*) yongası ve *Nestor Ni 1000* yongası sayılabilir.

Bu sınıflara girmeyen bazı dijital yapay sinir ağı yongaları vardır. *Micro Circuit Engineering* Şirketinin *MT 19003 NISP RISC* tabanlı bir işlemcisidir. ÇKA için geliştirilmiştir. *Hitachi* tarafından geliştirilmiş Hopfield ve ÇKA ağı için geliştirilmiş yongalar da vardır.

### 3.5.3 Hibrid Yapay Sinir Ağı Tasarımları

Bu donanım sistemleri hem dijital hem de analog sistemlerin birleştirilmesi ile elde edilen donanımlardır. Bu sistemlerin amacı, her iki yaklaşımında avantajlarını bir araya

getirmektedir. Genel olarak dış dünya ile ilişki dijital sinyaller ile gerçekleştirilmektedir. Dahili prosesler ile ya tamamen ya da kısmen analog devreler ile sağlanmaktadır. Karma sistemlere örnek, *Bellcore CLNN-32* ve *AT&T ANNA* yongalarıdır.

### 3.5.4 Yapay Sinir Ağı Donanımlarının Performanslarının Ölçülmesi ve Karşılaştırılması

Yapay sinir ağlarının donanımlarını karşılaştırmak ve performanslarını değerlendirmek için şu ana kadar geliştirilmiş bir kıyaslayıcı henüz geliştirilmemiştir. Çünkü ağın sahip olduğu topoloji, modelin yapısı ve öğrenme kuralı donanımın da performansını etkilemektedir. Buna rağmen, yapay sinir ağlarına dayalı donanımların performanslarını ölçmek için kullanılan en yaygın ölçüt saniyede gerçekleştirilen işlem (*connection per second*) sayısıdır. Bu ağa bilgiler sunulduktan sonra saniyede gerçekleştirilen işlem sayısını göstermektedir. Bir problemi öğrenme zamanı da yine ölçülebilecek ve performansı gösteren bir faktör olarak görülebilir.

Yapay sinir ağı donanımlarını birbirleri ile karşılaştırmak için öncelikle üretilen ürünlerin fabrikasyon özelliklerine bakmak gerekir. Fakat bu verileri bulmak kolay değildir. O nedenle, burada donanımları birbirleri ile karşılaştırmaya neden olacak faktörler daha çok ürünlerin bilinen özellikleri açısından olacaktır. Bunlar:

Teknoloji; analog, dijital, karma,

Girdiler; giriş sayısı, girdi değerlerinin duyarlılığı,

Çıktılar; çıktı eleman sayısı, çıktı değerlerinin duyarlılığı,

Matris hafıza; kapasite, organizasyon, teknoloji,

Vektör hafıza; kapasite, organizasyon, teknoloji,

Fonksiyon işleme; matematik işlemlerin sabit veya programlanabilir şekilde gerçekleştirilmesi,

Saniyede bağlantı; her saniyede gerçekleştirilen işlem sayısı,

Run-time adresleme; matris ve vektör hafızalarının yonga çalışırken nasıl adreslendiği,

Transfer fonksiyonu; hesaplamaların nasıl güncellendiği (Öztemel, 2003).

### 3.6 Yapay Siniri Ağlarının Genel Kullanım Alanları

Yapay sinir ağlarının uygulamaları gözden geçirildiğinde binlerce uygulamanın yapıldığı ve

başarılı sonuçların elde edildiği görülebilir. Uygulamalar o kadar yaygındır ki, bunların listesini çıkartmak hemen hemen mümkün değildir. Örneğin, 1997 yılında pazarlanan, *Caere Inc.* tarafından üretilen optik karakter okuma sisteminin yılda 3 milyon \$ dan fazla para kazandırmaktadır. Benzer şekilde aynı yıl, HNC firması tarafından pazarlanan ve kredi kartlarının haksız yere kullanımlarını ortaya çıkartan *Falcon* isimli yapay sinir ağı sistemi yılda 23 Milyon \$ dan fazla para kazandırmıştır. Wall Street Journal 1998 yılında *Sensory Inc.* Firması tarafından geliştirilen ses tanıma sistemindeki yonganın (chip) 5 \$ a mal olduğunu ve bir milyondan fazla sayıda satıldığını rapor etmiştir.

Çalışmalar eskiden laboratuarlarda yürütülmekte ve veriler benzetim yolu ile elde edilmekte iken artık yapay sinir ağları günlük hayatımızın vazgeçilmez bir parçası haline gelmiştir. Ağların eğitilmesinde gerçek örnekler kullanılmaktadır. Örnek bulmak eskisi kadar zor olmamaktadır. Evimizdeki aletlerden elimizdeki cep telefonlarına kadar birçok alanda yapay sinir ağlarının uygulamalarını görmek mümkündür. Burada her türlü uygulamadan örnek vermek imkansızdır. O nedenle burada, yapay sinir ağları uygulamalarının bir sınıflandırılması yapılacaktır. Bu kapsamda, yapay sinir ağları uygulamaları;

- ✓ *Endüstriyel uygulamalar*
- ✓ *Finansal uygulamalar*
- ✓ *Askeri ve savunma uygulamaları*
- ✓ *Sağlık uygulamaları*
- ✓ *Diğer alanlardaki uygulamalar*

Şeklinde sınıflandırılarak incelenebilir. Bu alanlardaki uygulamalar incelendiğinde yapay sinir ağlarının genel olarak şu fonksiyonları gerçekleştirmek için uygulandıkları görülmektedir:

- ✓ Tahmin; bu amaçla kullanılan yapay sinir ağları, ağa sunulan bilgilerden yararlanılarak karşılık gelen çıktı değerini tahmin ederler. Hava tahmini, borsada hisselerin değerlerinin tahmini, döviz kurların tahmini gibi örnekler vermek mümkündür.
- ✓ Sınıflandırma; bu amaçla kullanılan yapay sinir ağları kendilerine sunulan bilgileri kategorize etmek görevini üstlenirler. Bir makine üzerinde görülen hataların sınıflandırılması buna örnek olarak verilebilir.
- ✓ Veri ilişkilendirme; bu amaçla eğitilen ağlar ağa sunulan verilerin hatalı ve eksik olup olmadığını belirlerler. Öğrendikleri bilgiler ile eksik olan bilgileri tamamlarlar. Eksik bir

resmin tamamlanması bu konuda örnek olarak verilebilir.

✓ Veri filtreleme; bu amaçla eğitilen ağlar, birçok veri arasından uygun verileri belirleme görevini yerine getirirler. Telefon konuşmalarındaki gürültüleri asıl konuşmalardan ayıran ağlar bu konudaki uygulamalara örnek olarak verilebilir.

✓ Tanıma ve eşleştirme; değişik şekil ve örüntülerin tanınması, eksik, karmaşık, belirsiz bilgilerin işlenerek eşleştirme ve tanıma fonksiyonları gerçekleştirilebilir.

✓ Teşhis; bu amaçla geliştirilen ağlar sistemlerin olumsuzluklarının ortaya konulması ve problemlerin teşhis edilmesi işlemini yerine getirirler. Makinelerin, süreçlerin arazi durumlarının ve hatalarının teşhis edilmesi buna örnek olarak verilebilir. Tıp alanında da bu tür sistemler yaygın olarak geliştirilmektedir.

✓ Yorumlama; bir olay hakkında toplanan örneklerden elde edilen ve eğitim sonucu oluşturulan bilgileri kullanarak yeni olayların yorumlanması işlemleri bu kapsamda düşünülmektedir. Bir olay hakkında toplanan verilerin yorumlanarak istatistiksel dağılımlarının belirlenmesi bu konuda örnek olarak verilebilir.

Yapay sinir ağlarının yukarıdaki açıklamalar ışığında değişik uygulama alanlarına örnekler verilmiş olup, bunlardan kontrol mühendisliği açısından önemli olan bir bölümü aşağıda sunulmuştur.

✓ Endüstriyel uygulamalar; yapay sinir ağlarının sayısız endüstriyel uygulaması vardır. Bunlardan bazıları şu şekilde sıralanabilir:

- Yapay sinir ağları bir endüstriyel proseste fırınların ürettiği gaz miktarını tahmini,
- İmalatta, ürün tasarımı, proses ve makinelerin bakımı ve hataların teşhisi görsel kalite kontrolü,
- Kimyasal proseslerin dinamik modellenmesi,
- Otomobillerde otomatik rehber sisteminin geliştirilmesi,
- Robotlarda görme sistemleri ve manipulatörlerin kontrol edilmesi,
- Cep telefonlarında ses ile çalışabilme,
- Araba pistonlarının üretim şartlarının belirlenmesi,
- Elektronik yonga hata analizleri,

- Optimizasyon çalışmaları (üretim planlama ve kontrol çalışmalarında),
- Müşteri tatmini ve pazar verilerinin değerlendirilmesi ve analiz edilmesi,
- Kömür güç istasyonları için çevrimiçi (on-line) karbon akımı ölçülmesi,
- İşlerin makinelere atanması ve çizelgeleme,
- Gezgın satıcı problemini

bu listeyi uzatmak mümkündür. Görüldüğü gibi endüstriyel uygulamalar oldukça yaygın bir alanı kapsamaktadır. İmalattan güç istasyonlarına kadar birçok alanda bu ağlar kullanılmaktadır (Öztemel, 2003).

✓ Askeri uygulamalar; yapay sinir ağlarının sivil hayattaki uygulamaları kadar askeri alandaki uygulamaları da dikkatleri çekmektedir. Bunlar arasında şunlar örnek olarak sayılabilir:

- Hedef tanıma ve takip sistemleri,
- Yeni sensörlerin performans analizleri,
- Radar ve görüntü sinyalleri işleme,
- Sensör fizyonu,
- Askeri uçakların uçuş yörüngelerinin belirlenmesi (optimizasyonu),
- Mayın detektörleri.

✓ Sağlık uygulamaları; yapay sinir ağlarının insan beyni çalışmaları ile yakın ilişki içinde olması tıp ve sağlık alanında da uygulamaların gelişmesine neden olmuştur. Bunlardan bazı örnekler ise şöyle sıralanabilir:

- Solunum hastalıklarının teşhisi,
- EEC ve ECG analizleri,
- Transplant zamanlarının optimizasyonu,
- Hastalıkların teşhisi ve resimlerden tanınması,
- Kardiovasküler sistemlerin modellenmesi ve teşhisi,
- Tıbbi resim işleme,

- CTG izleme,
- Hamile kadınların karınlarındaki çocukların kalp atışlarının izlenmesi,
- Yumurtalık kanserinin immunoterapik izlenmesi,
- Üroloji uygulamaları (prostat analizleri, sperm analizleri).

#### 4. BULANIK MANTIK KONTROLÜN TEMELLERİ

İnsan beyni dünyanın en karmaşık makinesi olarak kabul edilebilir. İnsan beyni sayısal bir işlemi birkaç dakikada yapabilmesine karşın; idrak etmeye yönelik olayları çok kısa bir sürede yapar. Örneğin yolda giden bir şoför, yolun kayganlık derecesini, önündeki tehlikeden ne kadar uzak olduğunu, sayısal olarak değerlendiremezse dahi geçmişte kazanmış olduğu tecrübeler sayesinde aracın hızını azaltır. Çünkü o saniyelerle ölçülebilecek kadar kısa bir sürede tehlikeyi idrak etmiş ve ona karşı koyma gibi bir tepki vermiştir. Bu noktada akla gelen ilk soru şu olmaktadır; acaba bir bilgisayar yardımı ile böyle bir zeka üretmek mümkün olabilir mi? Bilgisayarlar çok karmaşık sayısal işlemleri anında çözümleyebilmelerine karşın, idrak etme ve deneyimlerle kazanılmış bilgileri kullanabilme noktasında çok yetersizdirler. Bu olayda insanı ya da insan beynini üstün kılan temel özellik, sinirsel algılayıcılar vasıtası ile kazanılmış ve görelî olarak sınıflandırılmış bilgileri kullanabilmesidir. *Uzman Sistemler (US)*, *Bulanık Mantık (BM)*, *Genetik Algoritma (GA)* ve *Yapay Sinir Ağları (YSA)* gibi yapay zeka alt dalları özellikle son yıllarda, geniş bir araştırma ve uygulama alanı bulmaktadırlar (Elmas, 2003).

Uzman sistem temelde insan düşüncelerini gerçekleştirmek amacıyla bilgisayar tarafından işlenen bir yazılımdır. Uzman sistem geliştirilirken, uzmanların belli bir konudaki bilgi ve deneyimlerini bilgisayara aktarılması amaçlanmaktadır.

Bulanık mantık, bulanık küme teorisine dayanan bir matematiksel disiplindir. Bulanık mantık insan mantığında olduğu gibi, Uzun-Kısa, Sıcak-Soğuk, Hızlı-Yavaş, Siyah-Beyaz yerine Çok Uzun-Uzun-Orta-Kısa-Çok Kısa, Sıcak-Ilık-Az Soğuk-Soğuk-Çok Soğuk vb. gibi ara değerlere göre çalışmaktadır.

Genetik algoritmalar yapay zekanın gittikçe genişleyen bir kolu olan evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadır. Genetik algoritma Darwin'in evrim kuramı doğada en iyinin yaşaması kuralından esinlenerek oluşturulan, bir veri öbeğinden özel bir veriyi bulmak için kullanılan bir arama yöntemidir.

Genetik algoritma geleneksel yöntemlerle çözümü zor veya imkansız olan problemlerin çözümünde kullanılmaktadır. Herhangi bir problemin genetik algoritma ile çözümü, problemi sanal olarak evrimden geçirerek yapılmaktadır.

Genel olarak yapay sinir ağları, insan beyninin sinir ağlarını taklit eden bilgisayar programlarıdır. Yapay sinir ağları bir anlamda paralel bilgi işleme sistemi olarak da

düşünülebilir. Yapay sinir ağlarına bu bilgiler ilgili olaya ait örnekler üzerinde eğitilerek verilir. Böylece, örnekler sayesinde açığa çıkarılmış özellikler üzerinde çeşitli genelleştirmeler yapılarak daha sonra ortaya çıkacak ya da o ana kadar hiç rastlanmamış olaylara da çözümler üretilmektedir.

#### 4.1 Tarihi Geçmiş

Bulanık mantık, insan düşünme ve algılamasını modellemek için kullanılan güçlü bir araçtır. İki değerli önerme yerine bulanık sistemler, çok-değerli kümelerle sonuç verir. Bulanık sistemler kuralları saklayıp dilsel girişten dilsel çıkışa örneklenmiş fonksiyonları belirler. Klasik mantığın dayandığı temel varsayım, “her önerme doğru veya yanlıştır”. Bu Aristo’dan beri tartışma konusu olmuştur. Aristo, “temel varsayım” adlı tezinde gelecek şartlara bağlı olarak olayların şüpheli doğruluk durumlarından bahseder. Bahsettiği gelecek olaylar hakkındaki önermeler aslında ne doğru ve ne de yanlıştır. Fakat iki durumun da olması imkan dahilindedir. Bulanık kümeler konusu ilk defa Lotfi A. Zadeh tarafından 1965 yılında ortaya atılmıştır. Zadeh’in bu çalışması şu gerçeği kuvvetlendirmiştir ki, insanların bazı sistemleri, makinelerden daha iyi kontrol edebilmelerinin bir sebebi, insanların kesinlik ifade etmeyen bir takım bilgileri kullanarak karar verebilme özelliğine sahip olmalarındandır. Dolayısıyla eğer bu özellik sistemlerin modellenmesinde kullanılırsa dizayn edilen kontrolörlerin performansının artırılması mümkün olacaktır. Bulanık mantık kümelerinin Bulanık Mantık Kontrol (*Fuzzy Logic Control-FLC*) olarak pratik uygulamalarda kullanılması ise, 70 li yıllarda gerçekleşmiştir. Bunlar bazı küçük çaplı endüstriyel uygulamalardır. İlk bulanık çipi ise, 80 lerin başında *AT&T Bell* laboratuvarlarında Togai ve Watanabe tarafından geliştirilmiştir (Atmaca, 1995).

Temel olarak geleneksel veya konvansiyonel bir PID kontrolör sistemi, matematiksel olarak bir dizi diferansiyel eşitlik ile sistemin modellenmesi ve bunların çözümü ile sistem davranışının kontrol edilmesine dayanır. Fakat bir FLC sistemi ise bir insanın bir sistemi kendisi kontrol ediyorken yapacağı düşünce işlemlerinin mantıksal olarak modellenmesine dayanır. Bu mantık, yapay sinir ağları (neural networks) ve genetik algoritmalar (Genetic Algorithms) ile beraber “akıllı” kontrol sistemleri kuşağının ilk adımlarını oluşturmakta olduğu bugün kabul edilmektedir (Kruse, 1995).

FLC nin dünyaya tanıtılmasında önemli olaylardan biri FLC nin Sendai (Japonya) metrosunun otomatik kontrolü için kullanılması olmuştur (1987). Bu uygulamada FLC nin bir çok parametre açısından konvansiyonel bir PID kontrolörden daha üstün olduğu gösterilmiştir

(örneğin, istasyonda yavaşlayıp durma, yolcu konforu ve yakıt tüketimi gibi...). Bu uygulamanın başarısı sebebiyle FLC kontrollü bu sistem yeni Tokyo metrosunda da kullanılmaktadır (Bulut, 2001).

Özellikle Japonya da 90 lı yıllarda FLC kullanılan tüketici ürünleri pazarda sıkça görülmeye başlanmıştır. Genellikle bu tüketici ürünlerinde (FLC çiplerinin yüksek hızlarına gerek olmadığından dolayı) FLC çipleri yerine, standart dijital çipler üzerinde tablo göz atmaları (table look-up) vasıtasıyla simülasyonlar kullanılmıştır. Tüketici ürünleri örneklerini şöyle sıralayabiliriz; çamaşır makineleri, elektrik süpürgeleri, klimalar, fanlar, ısıtıcılar, mikrodalga fırınlar, çamaşır kurutucuları. Bunun yanında birçok endüstriyel uygulamalarda da FLC başarıyla kullanılmıştır (King, 1977).

#### **4.2 Bulanık Mantık Nedir?**

Bulanık Mantık, problem çözmek için kullanılan, bir kontrol sistemi metodolojisidir. Donanıma, yazılıma veya her ikisine birden uygulanabilir. Bulanık Mantık; belirsiz, muallak, gürlütülü, eksik giriş verileri ile kesin bir sonuca varmayı sağlar. Bulanık Mantık, kontrol problemlerine, bir insanın karar verme şekliyle yaklaşır, sadece daha hızlı karar verir (Bulut, 2001).

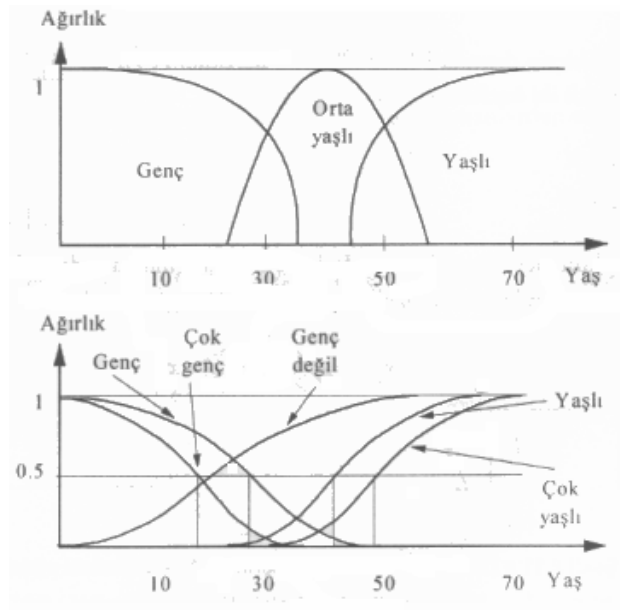
Doğal dilde kullandığımız birçok cümlede “az”, “çok”, “orta” gibi dilsel niceleyiciler kullanırız. Bu tür cümleleri bulanık mantığın gösterimi ile ifadelendirmek daha kolay olmaktadır. Bulanık mantıkta “Ahmet yaşlıdır” ve “Bugün hava sıcaktır” cümlelerindeki “yaşlı” ve “sıcak” ifadelerine iki değerli mantıktaki gibi “doğru” veya “yanlış” yerine 0 ile 1 arasında değer verilebilmektedir.

Bunu bir örnek ile açıklamamız gerekirse. Hafta içi günleri ve hafta sonu günleri diye bir ayırım yapacak olursak, kesin mantıkta bu günler bellidir. Cumartesi, Pazar hafta sonu günleri diğerleri ise hafta içi günleridir. Oysaki insanlar, Cuma gününü de biraz hafta sonuymuş gibi düşünürler ve de Pazar gününü, Pazartesi hafta içi olduğundan dolayı, Cumartesi kadar hafta sonu olarak düşünmezler. Sonuç olarak, Cumartesi en yüksek derecede hafta sonu değerini alırken Cuma ve pazar günleri daha az seviye de hafta sonu değeri taşımaktadır. Kümelerin kesişmesine ve küme elemanlarının üyelik değerlerine sahip olmasına olanak sağlayan bu metodolojiye bulanık mantık diyoruz [The MathWorks Fuzzy Logic Toolbox for Use with MATLAB].

### 4.3 Bulanık Küme Teorisi

Bilindiği gibi klasik (klasik matematik) kümeler kuramında birleşme, kesişme, tümlenme gibi işlemler tanımlanmıştır. Bunlara karşılık gelecek bulanık işlemler de tanımlıdır.

Bulanık küme keskin kümedeki açık/kapalı, soğuk/sıcak gibi ikili (binary) denetim değişkenlerinden oluşan keskin dünyayı, Az açık/Az kapalı, Serin/Ilık, Biraz hızlı/Biraz yavaş gibi yumuşak (soft) niteleyicilerle gerçek dünyaya benzetir. Yani klasik kümelerdeki gibi bir değişken verilen kümenin ya elemanı yada elemanı değildir yaklaşımının tersine her değişken verilen kümede belirli bir üyelik derecesine sahiptir. Bu yaş konusuna uygulandığında 35 yaşındaki bir insana pek Orta yaşlı denemeyeceği gibi, o kişi pek gençte sayılmaz, duruma göre belki genç tanımı belki de orta Yaşlı tanımı daha uygun düşer. İşte bulanık kümeler Şekil 4.1 de gösterildiği gibi böyle esnek bir düşünüşe imkan sağlar. Kümelerin birbirinden keskin olarak ayrılmamış olması, aralarında belirli bir örtüşüm (Overlap) olması 35 yaşın bir oranda hem Orta Yaşlı hem Genç olarak düşünülmesine imkan verir (Elmas, 2003).

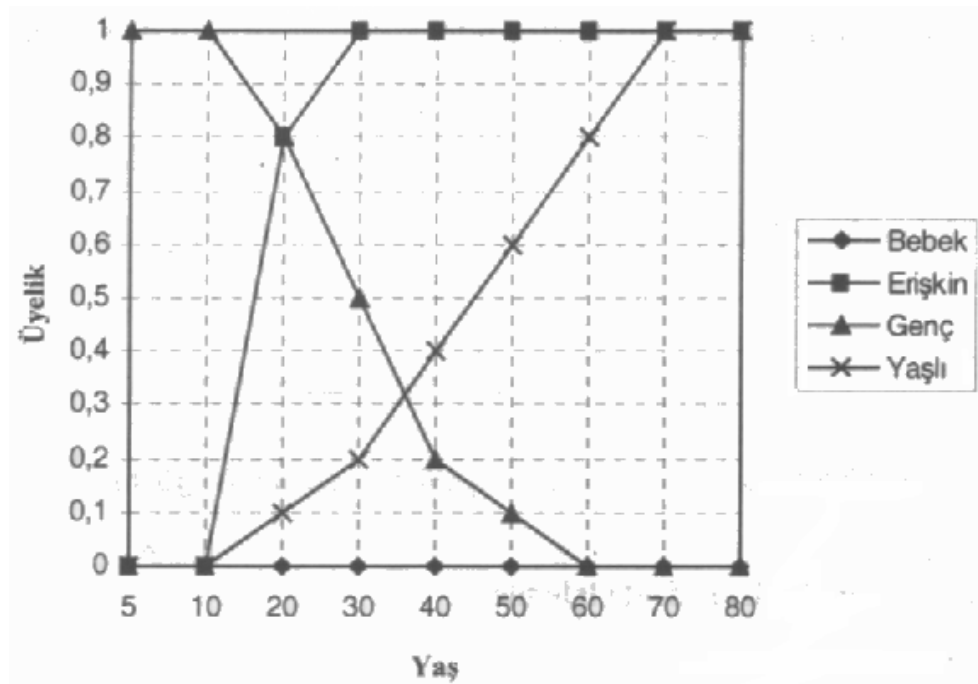


Şekil 4.1 Bulanık kümeler; a) Yaşın üç gruba ayrıldığı kümeler, b) Diğer bir kümelendirme yapısı (Elmas, 2003).

Bulanık kümelerde kullanılan semboller ve ifadeler ile keskin kümelerde kullanılan ifadelerin büyük bir bölümü benzemektedir. Küçük bir keskin evrensel kümenin elemanlarının dört farklı bulanık kümeye üyelik dereceleri Tablo 4.1 de ve eğrisi Şekil 4.2 de gösterilmiştir. Burada  $X = \{5, 10, 20, 30, 40, 50, 60, 70, 80\}$  bütün yaşların kümesini ve Bebek, Erişkin, Genç, Yaşlı bulanık kümeleri de X evrensel kümesinden seçilen değerlerin üyelik derecelerini göstermektedir.

Tablo 4.1 Bulanık kümeler (Elmas, 2003).

Yaş	Bebek	Erişkin	Genç	Yaşlı
5	0	0	1	0
10	0	0	1	0
20	0	0,8	0,8	0,1
30	0	1	0,5	0,2
40	0	1	0,2	0,4
50	0	1	0,1	0,6
60	0	1	0	0,8
70	0	1	0	1
80	0	1	0	1



Şekil 4.2 Tablo 4.1 de tanımlana bulanık kümelerin eğrisi.

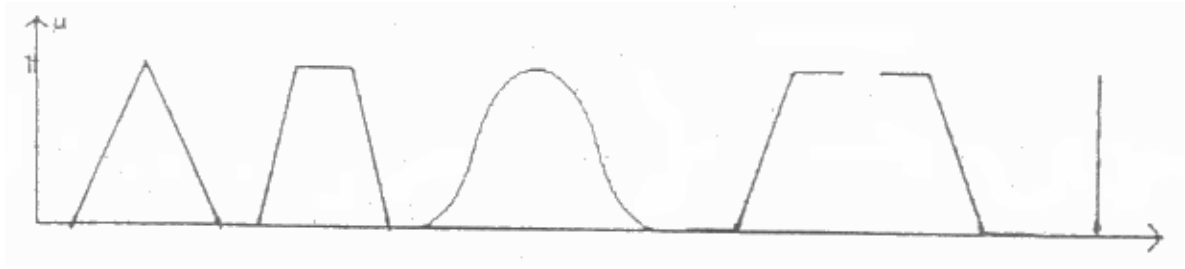
Görüldüğü gibi bulanık kümelerde önemli olan nokta bir değer bir kümeyle olan üyelik derecesidir. Bu dereceyi belirlemede etkili olan eleman yada birim ise, üyelik fonksiyonu olarak adlandırılır. Bu sebepten dolayı, bulanık kümelerde kullanılan üyelik fonksiyonlarının ele alınması çalışma açısından yerinde bir karar olacaktır. Sonuç olarak, bulanık mantık karar sisteminde bu fonksiyonlardan hareketle elde edilecek olana veriler kullanılacaktır.

#### 4.3.1 Üyelik Fonksiyonları

Bilinen klasik lojikte bir elemanın bir kümenin elemanı olması "1", olmaması "0" ile ifade edilir. Yani bir eleman bir kümeyle aitse "1", ait değilse "0" ile gösterilir. İşte bu "1" veya "0"

a üyelik ağırlığı (*grade of membership*) denir. Bulanık lojikte üyelik ağırlığının alabileceği en büyük değer “1” en küçük değer “0” dır. Fakat üyelik ağırlığı bulanık mantıkta “0” ile “1” arasında çeşitli değerler olabilir (0.1, 0.25, 0.9 ... gibi)

Üyelik ağırlığı belirli bir değer için bir bulanık küme içerisinde yer almasının güvenilirliğinin (*confidence*) veya eminliğinin (*certainty*) bir işaretidir. Üyelik işlevleri bilimsel olarak, denetlenen sürecin özelliklerine göre değişik şekillerde olabilir. Üçgen, yamuk, çan v.b. biçimlerde olabilir.



Şekil 4.3 Çeşitli üyelik fonksiyonları.

Farklı uygulamalarda çeşitli sayıda ve isimde etiket kullanılmaktadır. Fakat genelde 7 (yedi) ayrı etiket (*label*) kullanılır. Bunlar;

NB: Negatif Büyük (*NB: Negative Big*),

NO: Negatif Orta (*NM: Negative Medium*),

NK: Negatif Küçük (*NS: Negative Small*),

SF: Sıfır (*Z: Zero*),

PK: Pozitif Küçük (*PS: Positive Small*),

PO: Pozitif Orta (*PM: Positive Medium*),

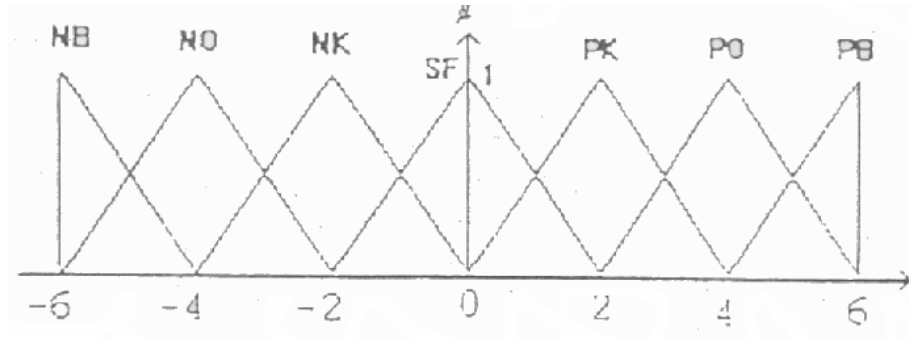
PB: Pozitif Büyük (*PB: Positive Big*).

Bu 7 etiketin dışında PVB (*positive very big*), PVS (*positive very small*), NVB (*negative very big*), NVS (*negative very small*) etiketleri de bazı uygulamalarda kullanılırlar (Özbay, 1995).

Etiketlerin maksimum ve minimum değerleri -6 ile +6 arasında derecelendirilebilirler. Bu değerler giriş ve çıkış büyüklüğünün sınır değerlerine göre belli bir katsayı ile çarpılır. Üyelik fonksiyonları değişken parametreleri olan bir fonksiyonu veya bir tablo olarak ifade edilebilir. Aşağıda çeşitli üyelik fonksiyonları ve üyelik değerlerini gösteren şekiller ve tablolar

verilmiştir.

✓ **Üçgen üyelik fonksiyonu;** sistem kontrolünde üçgen şeklindeki üyelik fonksiyonu hem giriş hem de çıkış büyüklüğünü tanımlamak için kullanılır Şekil 4.4 de görüldüğü gibi NB (*negative big*) etiketini taşıyan üyelik fonksiyonu -6 ile -4 aralığında tanımlanmıştır. NO (*negatif orta*) etiketini taşıyan üyelik fonksiyonu -6 ile -2 aralığında tanımlanmıştır. Dolayısıyla -5 değeri hem NB hem de NO üyelik fonksiyonlarına karşılık gelir. Çıkış kural tablosunda kullanılacak giriş kuralları NB ve NO için hesaplanacaktır (Şekil 4.4). Tablo olarak gösterimi Tablo 4.2 deki gibidir.

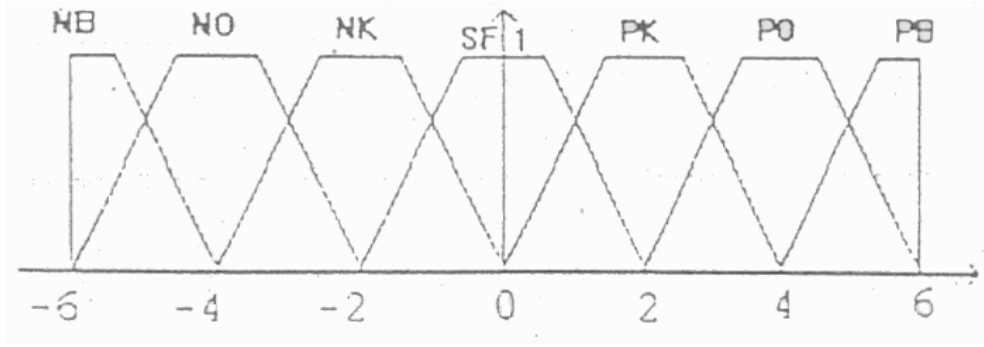


Şekil 4.4 Üçgen üyelik fonksiyonu.

Tablo 4.2 Üçgen üyelik fonksiyonu tablosu.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PB	0	0	0	0	0	0	0	0	0	0	0	0.5	1
PO	0	0	0	0	0	0	0	0	0	0.5	1	0.5	0
PK	0	0	0	0	0	0	0	0.5	1	0.5	0	0	0
SF	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
NK	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
NO	0	0.5	1	0.5	0	0	0	0	0	0	0	0	0
NB	1	0.5	0	0	0	0	0	0	0	0	0	0	0

✓ **Yamuk üyelik fonksiyonu;** yamuk şeklindeki üyelik fonksiyonu da yine üçgen üyelik fonksiyonu gibi giriş ve çıkış büyüklüğünü tanımlamak için kullanılabilir. Şekil 4.5 ile Tablo 4.3 bu fonksiyona ait gerekli bilgileri vermektedir.

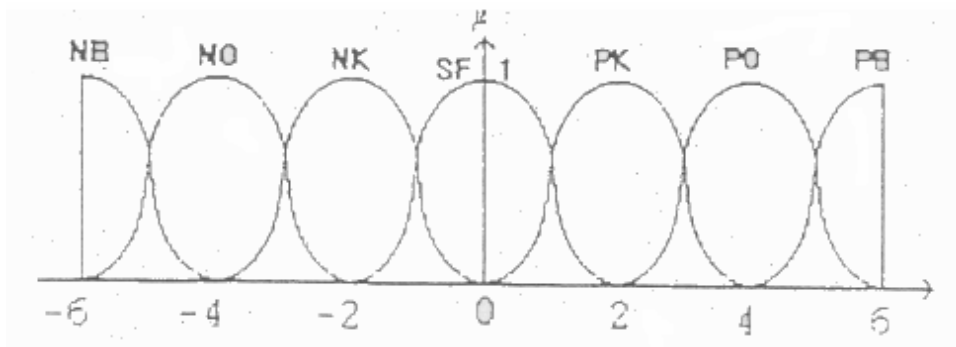


Şekil 4.5 Yamuk üyelik fonksiyonu.

Tablo 4.3 Yamuk üyelik fonksiyonu tablosu.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PB	0	0	0	0	0	0	0	0	0	0	0	0.66	1
PO	0	0	0	0	0	0	0	0	0	0.66	1	0.66	0
PK	0	0	0	0	0	0	0	0.66	1	0.66	0	0	0
SF	0	0	0	0	0	0.66	1	0.66	0	0	0	0	0
NK	0	0	0	0.66	1	0.66	0	0	0	0	0	0	0
NO	0	0.66	1	0.66	0	0	0	0	0	0	0	0	0
NB	1	0.66	0	0	0	0	0	0	0	0	0	0	0

✓ **Çan üyelik fonksiyonu;** çan şeklindeki üyelik fonksiyonu giriş/çıkış büyüklüklerini tanımlamada kullanılan en iyi üyelik fonksiyonudur. Bu durum pek çok makale ve çalışmada ifade edilmektedir. Şekil 4.6 ile Tablo 4.4 bu fonksiyona ait gerekli bilgileri vermektedir.

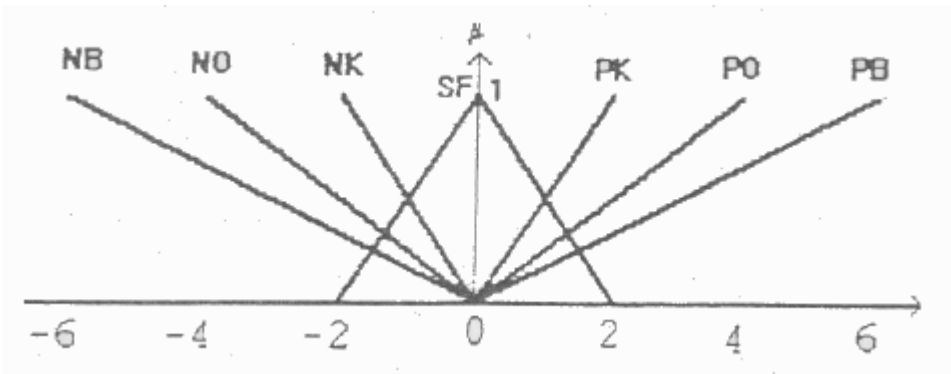


Şekil 4.6 Çan üyelik fonksiyonu.

Tablo 4.4 Çan üyelik fonksiyonu tablosu.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PB	0	0	0	0	0	0	0	0	0	0	0	0.66	1
PO	0	0	0	0	0	0	0	0	0	0.66	1	0.66	0
PK	0	0	0	0	0	0	0	0.66	1	0.66	0	0	0
SF	0	0	0	0	0	0.66	1	0.66	0	0	0	0	0
NK	0	0	0	0.66	1	0.66	0	0	0	0	0	0	0
NO	0	0.66	1	0.66	0	0	0	0	0	0	0	0	0
NB	1	0.66	0	0	0	0	0	0	0	0	0	0	0

✓ **Monolitik üyelik fonksiyonu;** monolitik üyelik fonksiyonu giriş ve çıkış değişkenlerini tanımlamada kullanılır. Her üyelik fonksiyonu doğru parçaları şeklinde tanımlanmıştır. Yalnızca SF (sıfır) üyelik fonksiyonu üçgen şekline benzer. Fakat iki doğru parçasının birleşimi olarak düşünülür. Şekil 4.7 ile Tablo 4.5 bu fonksiyona ait gerekli bilgileri vermektedir.

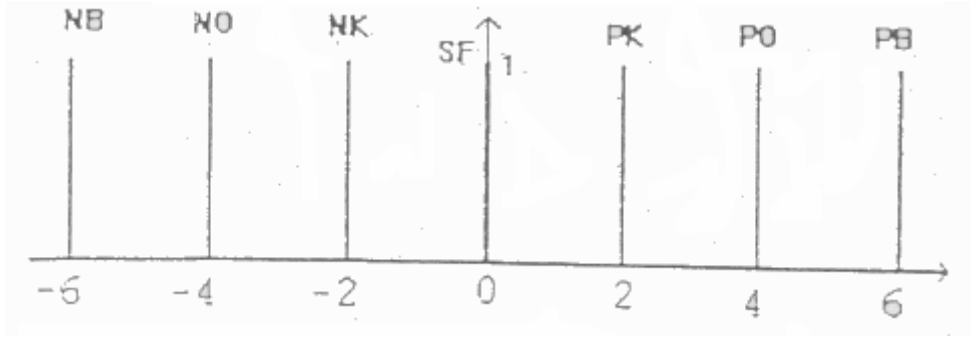


Şekil 4.7 Monolitik üyelik fonksiyonu.

Tablo 4.5 Monolitik üyelik fonksiyonu tablosu.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PB	0	0	0	0	0	0	0	0.16	0.33	0.5	0.66	0.83	1
PO	0	0	0	0	0	0	0	0.25	0.5	0.75	1	0	0
PK	0	0	0	0	0	0	0	0.5	1	0	0	0	0
SF	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
NK	0	0	0	0	1	0.5	0	0	0	0	0	0	0
NO	0	0	1	0.75	0.5	0.25	0	0	0	0	0	0	0
NB	1	0.83	0.66	0.5	0.33	0.16	0	0	0	0	0	0	0

✓ **Monotonik üyelik fonksiyonu;** monotonik üyelik fonksiyonu yalnızca çıkış değişkenlerini tanımlamada kullanılır. Her üyelik fonksiyonu doğru parçaları şeklinde tanımlanmıştır. Üyelik fonksiyonları birbirlerini örtmezler. Yani üyelik fonksiyonlarının merkez noktaları haricindeki değişkenin üyelik ağırlığı "0" dır. Bu nedenle çıkış değişkenlerinin tanımlanmasında kullanılır. Şekil 4.8 ile Tablo 4.6 bu fonksiyona ait gerekli bilgileri vermektedir.



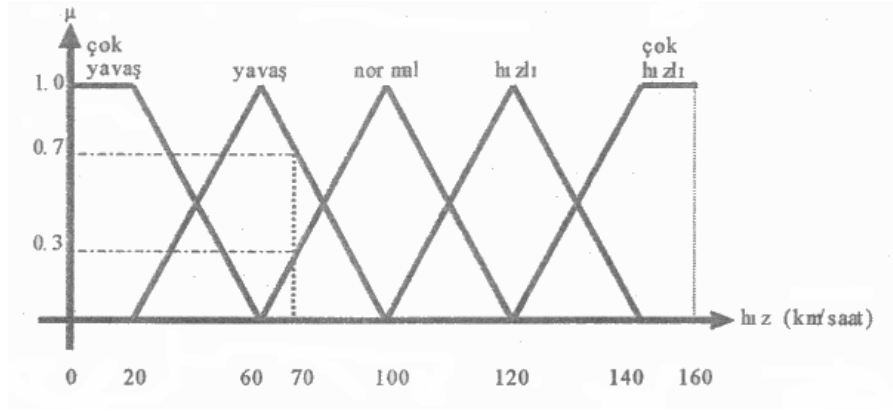
Şekil 4.8 Monotonik üyelik fonksiyonu.

Tablo 4.6 Monotonik üyelik fonksiyonu tablosu.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PB	0	0	0	0	0	0	0	0	0	0	0	0	1
PO	0	0	0	0	0	0	0	0	0	0	1	0	0
PK	0	0	0	0	0	0	0	0	1	0	0	0	0
SF	0	0	0	0	0	0	1	0	0	0	0	0	0
NK	0	0	0	0	1	0	0	0	0	0	0	0	0
NO	0	0	1	0	0	0	0	0	0	0	0	0	0
NB	1	0	0	0	0	0	0	0	0	0	0	0	0

#### 4.4 Dilsel Niceleyiciler

Bulanık kümelere ait en önemli dilsel niceleyiciler "dilsel değişken" ve "dilsel değerler" kavramlarıdır. Bunları bir örnekte ifade edecek olursak;



Şekil 4.9 Dilsel niceleyicilerin ifade edilmesi (Dalcı, 2000).

Şekil 4.9 da bir otomobile ait hız ifade edilmiştir. Burada “hız” kelimesi 0-160 km/saat sayısal değerlerini alabilen bir dilsel değişkendir. Üyelik fonksiyonlarının üzerinde yazılı değerler ise, “çok yavaş”, ... , “çok hızlı” olarak isimlendirilen dilsel değerlerdir. Şekil 4.9 üzerinde 70 km/saat değerini ele alacak olursak bu değer hem “yavaş” ve hem de “normal” üyelik fonksiyonlarını keser ve her bir üyelik fonksiyonuna belirli bir üyelik derecesi ile katılır.

Dilsel değere ait keskin bir değer bulanıklaştırma adı verilen yöntemle bulanık olarak işlem yapılabilecek bir duruma dönüştürülür. Bulanık kümelerle ait dilsel değerler kesin değerlerin üyelik derecelerinden meydana gelir. Burada ifade edilen “bulanıklaştırma” terimi, bir sonraki bölümde ayrı başlık halinde ele alınacaktır.

#### 4.5 Bulanık Mantık Kontrol Sistemleri

Bulanık kontrolörler, diğer kontrolörlerden farklı olarak kontrol edilecek sisteme ait önermelere dayanarak işlem yapan bir yapıya sahiptir. Burada belirtilen önermeler en genel haliyle aşağıdaki şekilde verilebilir.

Eğer (IF) .... Öyleyse (THEN) .....

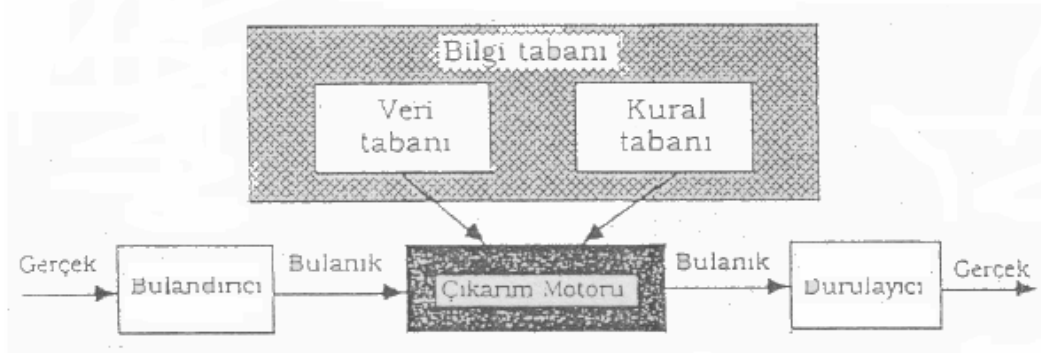
Bu önermelerin kontrol edilecek sistemin operatörlerden elde edilmesi, kontrolörün istenilen yapıda çalışma olanağını arttıracaktır. Operatör tarafından verilen bu önermeler kontrol kurallarını oluşturmaktadır. Bulanık kontrolörler, bu özellikleriyle dilsel önermeler işleyebilen bir yapıya sahiptirler.

Bulanık kontrolörler, klasik kontrol yapılarına alternatif olarak geliştirilmiştir. Bunun için klasik kontrol teorisinin tersine, sistemin modeliyle ilgilenilmeyen bir yol tutulmuştur. Buna göre sistemi kontrol eden bir operatörden yararlanılmış ve operatörün sistemi kontrol ederken

uyguladığı kuralların tümü bulanık kontrolörün temelini oluşturmaktadır.

Bu yöntem ilk yıllarda oldukça kabul görmüş ve hatta pek çok sayıda makaleye konu olmuştur. Ancak ilerleyen yıllarda, bulanık kontrolörlerin yapısına, kontrol edeceği sistemle ilgili bilgileri eklemekle daha verimli sonuçlara ulaşılacağı görülmüş ve çalışmalar da bu doğrultuya yöneltmiştir (Özby, 1995).

### Bulanık kontrolörün yapısı;



Şekil 4.10 Bir bulanık kontrolörün yapısı (Özby, 1995).

Klasik matematik pek çok probleme uygulandığında yetersiz kalır. Karmaşık işlemlerin kontrolünde operatörler devreye girmiştir. Bu operatörlerin karşılaştığı durumları ve bu durumlarda nasıl hareket ettiğini dilsel kurallarla açıklayabilir. Bu dilsel kuralları bir makineye yaptırmak bilinen sistemler içinde oldukça güçtür. Böyle bir otomatik kontrol işlemi, sözel tarif sistemini kullanan bulanık küm (*Fuzzy Set*) teorisini kullanmayı gerektirir.

Bir bulanık denetleyici, Şekil 4.10 dan da görüldüğü gibi 4 ana bölümden oluşur.

**Bulandırıcı (*Fuzzifier*):** bu bölüm giriş değişkenlerini (gerçek sayılar) ölçer, onlar üzerinde bir ölçek değişikliği yapar ve bulanık kümelere dönüştürülür. Yani onlara birer etiket vererek dilsel nicelik kazandırır.

**Çıkarım motoru (*Inference Machine*):** burada kurallar üzerinde bulanık mantık yürütülür (*Fuzzy Reasoning*). Yani insan beyninin düşünüş şeklinin bir benzetimi yapılmaya çalışılır.

**Veri tabanı (*Data Base*):** çıkarım motoru, kural tabanında kullanılan bulanık kümelerin, üyelik işlevlerini bu bölümden alır.

**Kural tabanı (*Rule Base*):** denetim amaçlarına uygun dilsel denetim kuralları burada bulunur ve çıkarım motoruna buradan verilir.

Durulandırıcı (*Defuzzifier*): çıkarım motorunun bulanık küme çıkışları (birden fazla çıkış olabilir) üzerinde ölçek değişikliklerini yapar ve bunları gerçek sayılara dönüştürür.

Bir bulanık denetleyicinin gerçekleştirilmesinde denetlenecek sistemin bir matematiksel modelinden daha çok o sistemi çalıştıran operatörün sistemin davranışı konusunda sahip olduğu bilgiler önemlidir. Tasarım sırasında genellikle bu tür bilgilerden yararlanır. Böyle bir yaklaşım uzun yıllar boyunca kullanılan deneyimlerin (rules of thumb, heuristics) denetleyici içerisine yorumlanmış halde kolaylıkla yerleştirilmesine olanak sağlar. Bu yararın yanında getirdiği sakınca denetleyici tasarımında kullanılacak belirli ölçütlerin olmaması ve tasarımın belirli bir otomasyona bağlanamamasıdır. Bununla birlikte bulanık denetleyicinin en önemli kısmını oluşturan kural tabanının oluşturulması için kullanılacak çeşitli yaklaşımlar literatürlerde açıklanmıştır. Bunlardan bazıları:

- Bir uzmanın bilgi ve deneyimlerine dayanır,
- Sürecin bulanık bir modelinin kullanılmasına dayanır,
- Operatörün süreç üzerinde yaptığı işlemlere dayanır,
- Öğrenen algoritmalar kullanılır.

Bulanık kontrol sistemlerinin oluşturulmasındaki ilk adım kontrol işlemindeki IF-THEN kurallarının oluşturulmasıdır. Bu kural oluşturulurken genellikle kontrol işleminde uzman sistemin-uzman kişi veya kontrol ekipmanının hangi durumda müdahale yaptığı sözel ifadelerle anlatılır (Özbay, 1995).

Bir sistem için formüle edilmiş dört bulanık kural düşünelim. Şöyle ki:

Eğer hata (e) sıfırsa ( SF) ve hatanın değişimi (ce) küçük pozitifse (PK) o zaman kontrol çıkışı (u) negatif küçüktür (NK).

Eğer hata ve hatanın değişimi sıfır ise o zaman kontrol girişi sıfırdır.

Eğer hata küçük negatif (NK) ve hatanın değişimi küçük negatif (NK) ise o zaman kontrol girişi küçük pozitifdir (PK).

Eğer hata küçük negatif (NK) ve hatanın değişim sıfırsa (SF) o zaman kontrol girişi büyük pozitifdir (PB).

Bu kurallar daha sonra bulanık kontrolöre uygun bir karar tablosu oluşturmak için birleştirilir.

#### 4.5.1 Bulanıklaştırma

Bulanık kontrol sadece bulanık değişkenler ile işlem yapabilecek şekilde tasarlanmıştır. Dolayısıyla tüm “crisp” giriş değerleri kullanılmadan önce bulanık kümelere çevrilmelidir. Bu işleme bulanıklaştırma denilir. En sık kullanılan bulanıklaştırma algoritmasında, fiziksel büyüklüğe karşı düşen nokta “1”, diğerleri “0” alınmaktadır. Örneğin uzayı {1,2,3} olan ayrık bulanık kümenin, 2 değerine karşı düşeni aşağıdaki gibi olacaktır.

$$A = (0/1) + (1/2) + (0/3) \quad (4.1)$$

#### 4.5.2 Bulanık Çıkarım

İkili mantık (*Binary Logic*) kullanılan sistemlerde veriler kurallardan bir sonuca varmaya, yani çıkarma bir örnek aşağıda verilmiştir.

Kural i: bebekler konuşamazlar,

Kural ii: Selma bir bebektir,

Çıkarım: Selma konuşamaz.

Görüldüğü gibi kurallar açık ve kesindir. Çıkarım ise ancak kuralların birbirleri ile tam olarak uyduğu durumlarda mümkündür. Bulanık sistemlerde girişler; sıcak, ılık, orta v.b. gibi dilsel değişkenlerden (*Linguistic Variables*) oluşur. Dilsel değişkenler duruma esneklik kazandıracaktır. Bu girişler üzerinde muhakeme ve bir sonuca varma, EGER - O HALDE (IF-THEN) türünden kuralların kullanımı ile gerçekleştirilir.

Bilgi: su çok sıcaksa soğuk su musluğunu biraz aç,

Gerçek: su biraz sıcak,

Çıkarım: soğuk su musluğunu biraz aç.

Görüldüğü gibi bulanık çıkarımda, eldeki gerçeğin, verilen bilgidен biraz farklı olduğu durumlar bir sorun oluşturmaz. Çıkarım o yönde gelişir.

Bulanık mantık kuramı, makinelere insanların öznel (*subjective*) verilerini işleyebilme ve onların deneyimlerinden ve önsezilerinden yararlanarak çalışabilme yeteneğini verir. Böylece onların kullanıcıya daha yakın olması sağlanır. EGER - O HALDE yapısını kullanan yapay zekadan farkı, sembolik bir süreç olmaması ve kullandığı kural sayısının göreceli olarak daha az olmasıdır (Özbay, 1995).

#### 4.5.3 Netleştirme

Bulanık kontrolörlerin çıkışında sistemin ihtiyaç duyduğu duru (*crisp*) değere çevirmeye

ihtiyaç vardır. Bu işlem durulaştırma olarak adlandırılır. Durulaştırma için değişik metodlar kullanılabilir.

Ağırlık merkezi metodu (*The Center of Area Method*): bu methodda bulanık kümenin ağırlık merkezine düşen değer kesin değer olarak ele alınmaktadır.

$$Z = \frac{\sum_{j=1}^q z_j \mu_c(z_j)}{\sum_{j=1}^q \mu_c(z_j)} \quad (4.2)$$

q: çıkış kuantalama sayısı,

z<sub>j</sub>: kuantalama seviyesinde kontrol çıkışının miktarıdır,

M<sub>c</sub>(z<sub>j</sub>): z<sub>j</sub> nin “c” üyelik fonksiyonundaki üyelik ağırlığı.

Maksimum Metodu (*The mean of Maximum Method*): bu algorithmada, bulanık küme içinde en büyük üyelik değerine sahip olan eleman kesin değer olarak kabul edilir.

$$Z = \sum_{j=1}^l \frac{z_j}{l} \quad (4.3)$$

l: maksimum üyelik değerlerine ulaşan kuantalanmış “z” değerlerinin sayısıdır.

Tsukomoto durulaştırma metodu:

$$Z = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} \quad (4.4)$$

n: üyelik ağırlığı sıfırdan büyük olan kural sayısıdır,

x: z. kuralın gerektirdiği kontrol hareketi miktarıdır.

Kuralların çıkışının girişin fonksiyonu olma durumunda durulaştırma:

Kural i: Eğer x = A<sub>i</sub> ve y = B<sub>i</sub> ise z = f<sub>i</sub>(x,y),

Eğer α<sub>i</sub>, i. kuralın üyelik derecesi ise;

$$Z = \frac{\sum_{i=1}^n \alpha_i f_i(x_i, y_i)}{\sum_{i=1}^n \alpha_i} \quad (4.5)$$

olur. Burada “n” üyelik ağırlığı sayısıdır (Özbay, 1995).

#### 4.5.4 Bulanık Kontrol Algoritması

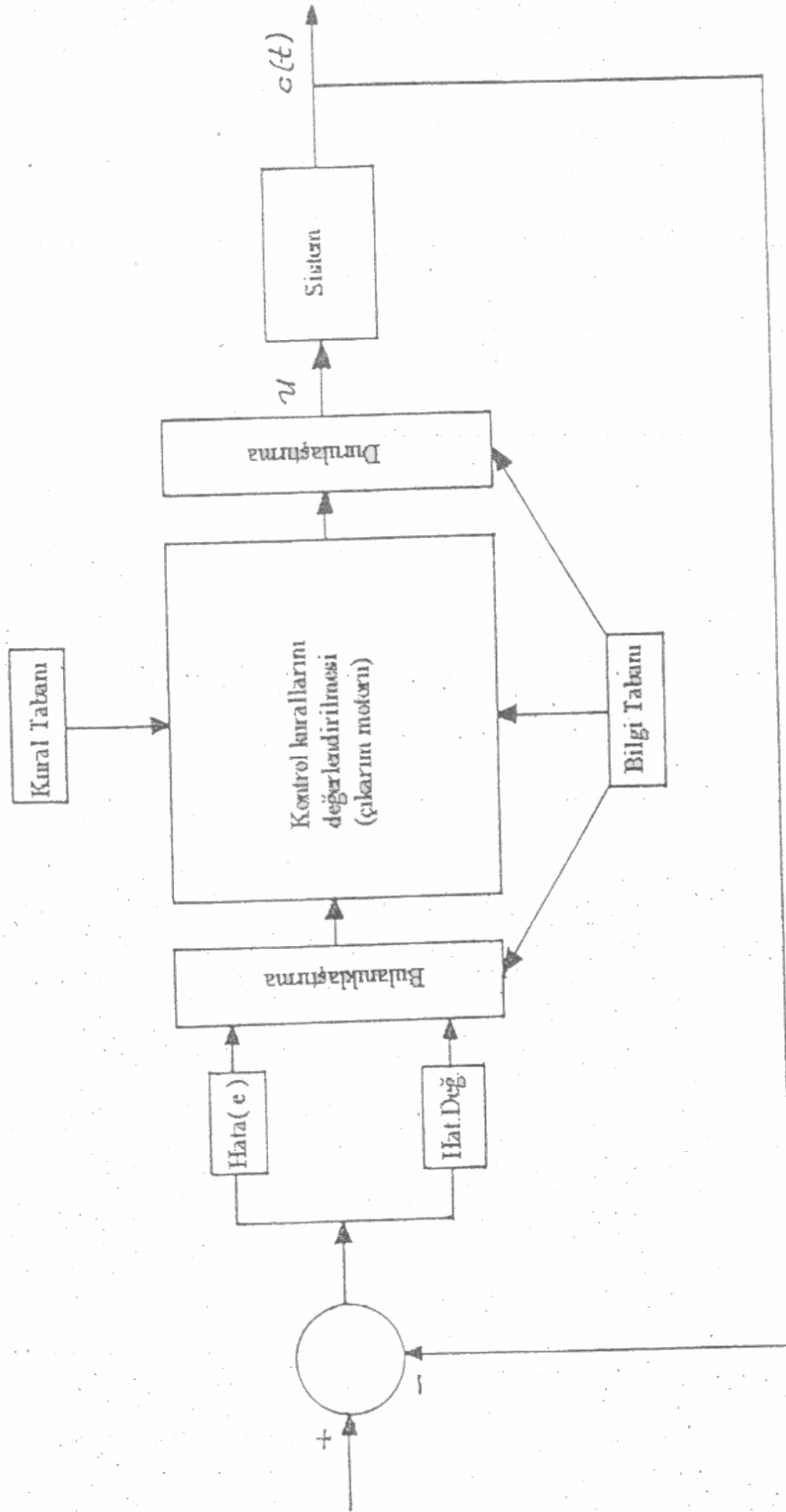
Bir bulanık kontrol sisteminin blok diyagramı Şekil 4.11 de gösterilmiştir. Kontrol sinyali  $u$ , hata  $e$ , hatanın değişimi  $ce$ , örnekleme aralığındaki  $de/dt$  dir. “ $e$ ” ve “ $ce$ ” gerçek “ $E$ ” ve “ $CE$ ” sinyallerini kazanç faktörlerine bölerek elde edilen, birim başına (*per-unit*) sinyallerdir (Özbay, 1995).

Hata ( $e$ ) = Referans Konum ( $R$ ) - O andaki konum ( $c$ ),

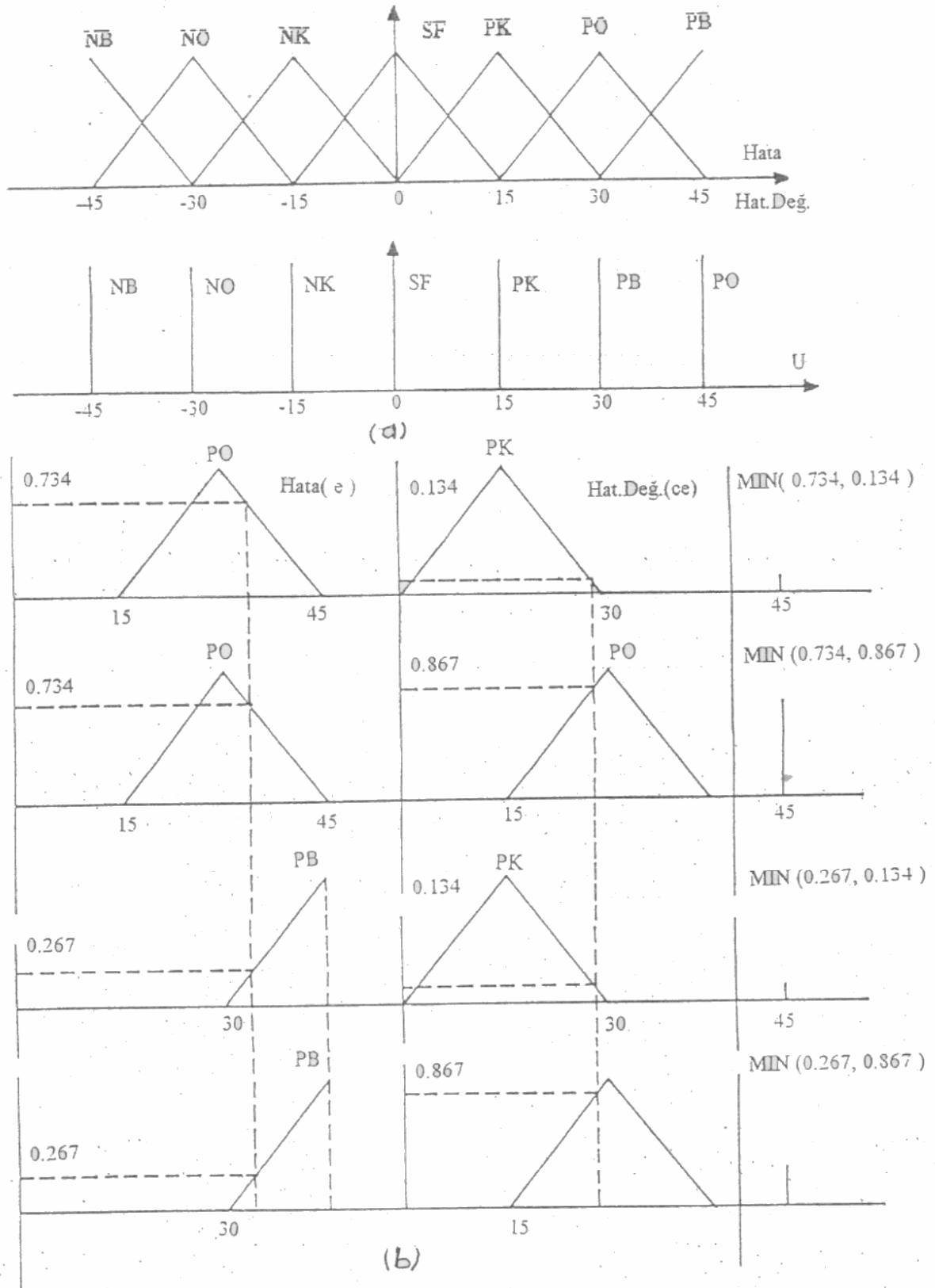
Hatadaki Değişim ( $ce$ ) = Şimdiki Hata ( $ek$ ) – Önceki hata ( $ek-i$ ).

Denetim kurallarımızın sonuçsal kısmı denetim çıktımız ( $u$ ) olup, bu değer ( $e$ ) ve ( $ce$ ) nin  $u=f(e,ce)$  şeklindeki herhangi bir fonksiyonu olacaktır. O halde bulanık denetleyicimizi bir, doğrusal olmayan oransal-türevsel (PD) denetleyici olarak düşünebiliriz.

Şimdi de hata ( $e$ ) ve hatanın değişimi ( $ce$ ) için PB, PO, PK, SF, NK, NO, NB olmak üzere 7 bulanık küme tanımlayalım. Burada P: Pozitif, N: Negatif, B: Büyük, K: Küçük ve SF: Sıfır olarak kullanılmış olup, örneğin PO, pozitif orta anlamına gelir. Hata ( $e$ ) ve hatanın değişimi ( $ce$ ) için üçgen üyelik fonksiyonu, kontrol sinyali ( $u$ ) için monolitik üyelik fonksiyonu kullanılmıştır (Şekil 4.12).



Şekil 4.11 Bir bulanık kontrol sisteminin blok diyagramı (Özbay, 1995).



Şekil 4.12 a) Hata, hatanın değişimi ve kontrol sinyali için seçilen üyelik fonksiyonları

b) Grafik olarak gösterilen netleştirme.

Şimdi de denetim kurallarımızı belirleyelim. Bulanık denetleyicilerde kurallar Tablo 4.7 de gösterildiği üzere bir tablo şeklinde yazılır. Bu kurallar aynı sistemden elde edilen deneyimler sonucu elde edilebilir. Tablodan da görüldüğü gibi kurallar bulanıktır. Bu kurallarla sistem kontrolü yapılmak istendiğinde öncelikle bu kuralların durulaştırılması gerekir. Aşağıda ağırlık merkezi metodu kullanılarak yapılan bir durulaştırma görülmektedir.

Tablo 4.7 Bulanık denetleyici için kural tabanı.

$\Delta e^c$	NB	NO	NK	SF	PK	PO	PB
PB	NB	PO	PB	PB	PB	PB	PB
PO	NB	*	PB	PB	PB	PB	PB
PK	NB	*	*	PK	PB	PB	PB
SF	NB	NB	NK	SF	PK	PB	PB
NK	NB	NB	NB	NK	*	*	PB
NO	NB	NB	NB	NB	NB	*	PB
NB	NB	NB	NB	NB	NB	NB	PB

$e = 34$  ve  $ce = 28$  için  $u$  çıkışını bulalım.  $e = 34$  için PO ve PB üyelik fonksiyonları bir ağırlığa sahiptirler.  $e = 34$  için, PO üyelik fonksiyonu ağırlığı  $\mu_{PO}(34) = 0.734$ , PB üyelik fonksiyonu ağırlığı  $\mu_{PB}(34) = 0.267$  dir. Benzer şekilde  $ce = 28$  için üyelik fonksiyonu PO ve PK üyelik fonksiyonları bir ağırlığa sahiptir. Aynı şekilde  $\mu_{PO}(28) = 0.134$  ve  $\mu_{PK}(28) = 0.867$  dir. Dolayısıyla hata ( $e$ ) ve hatanın değişimi ( $ce$ ) arasında 4 kural oluşur ve sonuçta da “ $u$ ” için 4 sonuç mevcuttur. Oysa bize yalnız bir “ $u$ ” değeri gereklidir. “ $u$ ” değerini bulmak için ağırlık merkezi metodu ile aşağıdaki durulaştırma (netleştirme) yapılır.

$$u = \frac{(0.134 * 45 + 0.734 * 45 + 0.134 * 45 + 0.267 * 45)}{(0.134 * +0.734 + 0.134 + 0.267)} = 45 \quad (4.6)$$

#### 4.6 Kontrol Uygulamalarında Bulanık Mantık

Bugün elektronik pazarında, pek çok üretim bulanık mantık temeline dayanmaktadır. Bulanık mantık denetim sistemlerinin pek çoğu tüketiciler için *SEA/Japonya* da üretilmektedir. Bulanık mantığa dayanan pek çok otomotiv ürünleri piyasaya sunulmuştur. Çizelge 4.1 de bulanık mantık yaklaşımının kullanıldığı birkaç örnek görülmektedir.

Bulanık mantık uygulamaları, ısı, elektrik akımı, sıvı gaz akımı denetimi, kimyasal ve fiziksel süreç denetimlerinde kullanılmaktadır (Elmas, 2003).

Bulanık mantık yaklaşımı uygulandığında öncelikle problemin özellikleri tanımlanır. Bulanık mantık yaklaşımlarının kullanıldığı sistemler klasik sistemlere göre daha etkin ısı ve hız denetimi yapabilmektedir. Ayrıca, enerji tasarrufu sağlanmakta ve aygıt ömrü uzamaktadır.

Çizelge 4.1 Bulanık mantık denetimin endüstriyel uygulamaları (Elmas, 2003).

ÜRÜN	ŞİRKET
Çamaşır makinesi	AEG, Sharp, Goldstar
Pirinç fırını	Goldstar
Fırın/Kızartıcı	Tefal
Mikrodalga fırın	Sharp
Elektrikli Tıraş Makinesi	Sharp
Buzdolabı	Whirlpool
Batarya şarj cihazı	Bosch
Elektrikli Süpürge	Philips, Siemens
Camcorder	Canon, Sanyo, JVC
Klima Denetimi	Ford
Isı Denetimi	NASA inspace shuttle
Kredi Kartı	GE Corporation

Bulanık sistemlerde denetim kurallarının tanıtımı genellikle daha kolay ve basittir. Genel olarak bulanık mantık denetleyiciler daha az kural gerektirmekte ve daha yüksek performans sağlamaktadırlar.

Bulanık mantık işlemleri problemin analiz edilmesi ve tanımlanması, kümelerin ve mantıksal ilişkilerin oluşturulması, mevcut bilgilerin bulanık kümelere dönüştürülmesi ve modelin yorumlanması aşamalarından oluşmaktadır. Birçok önkoşul kullanılarak bulanık mantığın tutarlılık oranı ve verilerin belirlilik ölçüleri de dahildir.

Öncelikle çözülecek problem için bulanık mantık yaklaşımının doğru bir seçenek olup olmadığına karar verilir. Eğer uygulanacak sistemin davranışı kurallarla ifade edilebiliyorsa veya karmaşık bir matematiksel işlem gerektiriyorsa, bulanık mantık yaklaşımı uygulanabilir. Aksi takdirde bulanık mantık ile elde edilen sonuçlar büyük olasılıkla istenilen değerleri vermeyecektir.

Sistemin her bir çıkış ve giriş değişkenleri için üyelik işlevi tanımlanmalıdır. Üyelik işlevinin

sayısı sistemin davranışına bağlı olmakla birlikte, aynı zamanda tasarımcı seçimine de bağlıdır. Kaç tane kural gerektiğine tasarımcı karar verir.

Bulanık mantık çok değişkenli mantıktır. Yani bu mantıkta küme üyeleri derecelendirilebilir. Bu basit bir örnek ile açıklanacak olursa; bilgisayar dünyasında büyük önemi olan ikili sayılarda, sayı “0” yada “1” olabilir, bilgisayar mantığına uygulanırsa ya doğru yada yanlış olabilir.

Bulanık mantık kuramının en büyük özelliği “klasik” bilgide olduğu gibi sayılardan çok sembolik bilgilerin kullanılmasıdır. Bu bilgi kavramları nesnelere düşünürken bir insanın göz önünde bulundurduğu olguların aynılarını temsil eder. Bu sayısal işlem yöntemlerinin kullanılmasını dışlamaz, ancak sonuçların incelenmesi genellikle sembole dayalı olarak yapılır. Bulanık mantıkta bulunan ikinci bir kavramda klasik algoritma metotlarının tersine “tecrübeye dayalı bilgi” metotları kavramıdır.

Bulanık mantığın bir başka özelliği de istenen verilerin ve bilgilerin belirsiz, eksik, yanlış ve hatta çelişkili olduğu durumlarla yetinmesidir. Bulanık mantık çok karmaşık bir problemi tamamen çözmese de etkili metotlar geliştirir.

Bulanık mantık ile tasarlanan ürünlerin kullanımı, tasarlanması, denenmesi daha kolay ve standart sistemlere göre daha iyi bir denetim sağlamaktadır. Ayrıca bulanık mantığın uygulamaya geçirilişi kolay, hızlı ve ekonomiktir.

#### **4.7 Klasik Kontrol ile Bulanık Mantık Tabanlı Kontrolün Karşılaştırılması**

Bugün kullanılan konvansiyonel kontrol sistemlerinin çoğu geribesleme (*feedback*) üzerine kurulmuştur. Geribesleme, çıkıştan alınan sinyalin istenen sinyal ile karşılaştırılıp aradaki fark sinyalini (*error signal*) çıkış sinyalindeki hatayı düzelterek şekilde sistem girişine vermek olarak tanımlanabilir. Geribesleme, bu tanım ile insana algısal olarak uygulaması kolay gibi görünüyorsa da, uygulamada tamamiyle otomatik bir kontrol sisteminin geri besleme ile dizayn edilmesi bir mühendis için gerçekten zor bir olaydır. Bunun en büyük sebebi ise, kontrol mekanizmalarındaki (sensörler, aktivatörler gibi...) zaman gecikmeleridir.

Bir geribeslemeli otomatik kontrol sisteminde etkin bir kontrol için birçok sistem parametresinin bilinmesi lazımdır. Fakat bu bahsedilen zaman gecikmesi yüzünden bu parametrelerin aynı anda bilinmesi her zaman mümkün olmayabilir. Eğer sistemde, dışarıdan gelen olağandışı bir gürültü yüzünden ani bir değişiklik olursa, kontrol sistemi çıkışları mümkün olduğu kadar çabuk düzeltmeye çalışacaktır. Fakat eğer sistemi çok kuvvetli olarak

bu ani deęişikliğe cevap verirse, sistem davranışının kararsız olma ihtimali yükselir. Bu gibi problemler yüzünden sadece basit bir geri besleme ile etkin bir tam otomatik kontrol sistemi gerçekleştirmek mümkün olmayabilir. Fakat, günümüz kontrol teorisi bu problemlere bir takım çareler bulmuştur. Birisi konvansiyonel kontrolde dizayn ve ayarlama metodlarının tamamen gelişmiş matematiksel modellemelere dayandırılmasıdır. İkinci olarak, PID kontrol sistemlerinin geliştirilmesidir (Tremaine, 1992).

Bulanık kontrole baktığımızda da tamamına yakın örneklerde geribesleme yapısı üzerinde kurulduğunu görmekteyiz. Ayrıca bulanık kontrolde geleneksel kontrol kuramındaki PID geribesleme yapısı da kullanılabilir. Bu özellikler bulanık kontrol teorisine geleneksel kontrolü bazı güzel ve gelişmiş taraflarını alma imkanını vermektedir. Geleneksel kontrolde doğrusal olmayan kontrol sistemleri ileri bir konu olarak düşünülmektedir. Fakat bu düşünce bulanık kontrol için geçerli değildir. Bulanık kontrol deęişkenler arasında karmaşık ve doğrusal olmayan ilişkilerde bile aynı mantığa sahiptir ve pratik olarak uygulaması nispeten daha kolaydır.

FLC (*Fuzzy Logic Controller*) nin normal PID kontrolörlere karşı bazı avantajları vardır. FLC kontrolörlere ilgili sensör verilerinin zayıflamasına ve bozunumuna karşı daha kuvvetlidirler ve sistem parametrelerinin deęişmesine karşı tekrar dizayn etmeyi gerektirmezler. Doğrusal olmayan kontrolde de normal PID ye göre performansları daha yüksektir. FLC kontrolörlere dizaynı PID kontrol sistemlerine göre daha kolaydır ve az elektronik eleman gerektirir. Bu yüzden maliyetleri de düşüktür. Bu sebeple de PID kontrolünün rahatça kullanıldığı yerlerde bile FLC kullanılması yaygınlaşmıştır. Bulanık mantığın en iyi uygulama alanları doğrusal olmayan (*non-linear*), tam olarak bir matematiksel model ile tanımlanamayan zamanla deęişen sistemlerdir. Bulanık mantık kontrolün bahsedilen avantajlarına karşın bazı dezavantajlara da sahiptir;

- a. Bulanık kontrolde kullanılan kurallar deneyime çok baęlıdır,
- b. Bu durum sistemi yeterince tanımayan birinin yazacağı kurallar ile FLC nin sistemi istenen şekilde kontrol etmesini zorlaştırır,
- c. Üyelik fonksiyonlarının seçiminde kesin bir yöntem yoktur. En uygun fonksiyon deneme ile elde edilebilir,
- d. Bu da sağlam bir kontrolör gerçekleştirilmesini geciktirir,
- e. Yazılan kurallar ile sistemin bir kararlılık analizi yapılamaz. Sistemin nasıl cevap vereceęi

önceden kestirilemez.

Çizelge 4.2 Bulanık mantık kontrol ile klasik kontrolün karşılaştırılması.

<b>Klasik Kontrol</b>	<b>Bulanık Kontrol</b>
Proses için matematiksel modele ihtiyaç duyar	Kontrol için uzman deneyim ve tecrübelerine ihtiyaç duyar.
Proses değişkenlerinin ölçümleri doğru ve kesin olmalıdır.	Kesin olmayan bilgileri kullanabilir
Karmaşık sistemlerde denetleyici de karmaşık olacağından uygulamaya koymak ekonomik olmayabilir.	Ucuz sensörler sayesinde prosesin ölçümünde esneklik kazandırır.
	Hızlıdır
	Uygulamaya geçirilişi kolaydır.
	Prosesin matematiksel modeline gereksinimi yoktur.

Bu nedenle FLC ilk önce sistemin simülasyonu üzerinde uygulanmakta ve alınacak olumlu sonuçlardan sonra gerçek sisteme uygulanmaya başlanılmaktadır.

## 5. SİNİRSSEL BULANIK MANTIK SİSTEMLER

Burada temel amaç; hem yapay sinir ağı tabanlı sistemlerin hem de bulanık mantık tabanlı sistemlerin zayıf yönlerini ortadan kaldıracak şekilde entegre edilmiş sistemler geliştirmektir. Yani her iki yapının da baskın yönlerinin kullanılması, bu tür yapılar için temel ilkeyi teşkil etmektedir.

### 5.1 Giriş

Konuya tam bir giriş yapmadan önce gerek bulanık mantık ve gerekse de yapay sinir ağlarının temel özelliklerini Tablo 5.1 den gözlemek gelecek açıklamalar için çok faydalı olacaktır.

Tablo 5.1 Bulanık mantık ve yapay sinir ağlarının temel özellikleri (Karray, 2004)

	<i>Bulanık Mantık</i>	<i>Yapay Sinir Ağları</i>
<i>Tanımlama:</i>	Bilginin dilsel tanımlanması.	Bilgi hesaplama üniteleri (hücreler) arasında dağıtılmıştır.
<i>Adaptasyon (Uyarlama):</i>	Kısmi uyarlama.	Adaptif (değişken).
<i>Veri Tanımlama:</i>	Açık ve basit yorumlu.	Dolaylı ve yorumu zor.
<i>Öğrenme:</i>	Yapısında yok.	Öğrenmeyi açığa vurmak için iyi bir araç.
<i>Doğrulama:</i>	Basit ve etkili.	Açık değildir.

Bulanık mantık, sinir ağları, genetik algoritmalar ve uzman sistemler gibi bütün yapay zeka tekniklerinin her birinin kendisine özgü yetenekleri bulunmaktadır. Örneğin yapay sinir ağları öğrenme, örnekleri tanımlamada iyi iken, kararların nasıl alındığı konusunda iyi değildir. Bulanık mantık yaklaşımı karar almada çok iyi sonuçlar verir, fakat karar alma sürecindeki kural oluşturmayı kendiliğinden gerçekleştiremez.

Sinirsel bulanık mantık yaklaşımı, yapay sinir ağlarının öğrenme yeteneği, en uygunu bulma ve bağlantılı yapılar gibi, bulanık mantığın insan gibi karar verme ve uzman bilgisi sağlama kolaylığı gibi üstünlüklerinin birleştirilmesi fikrine dayanmaktadır. Bu yolla, bulanık denetim sistemlerine, sinir ağlarının öğrenme ve hesaplama gücü verilebilirken, sinir ağlarına da

bulanık denetimin insan gibi karar verme ve uzman bilgisi sağlama yeteneği kazandırılmaktadır (Elmas, 2003).

Sinirsel bulanık denetim sisteminin asıl amacı, sinirsel bulanık denetim sistemlerinin yapısını, değişkenlerini ayarlamak ve bulmak için sinirsel öğrenme tekniklerini uygulamaktır. Bulanık mantık denetleyicilerde yapısal ayarlama v değişken ayarlama olmak üzere iki önemli ayarlama gerekir. Yapısal ayarlama hesaplanacak değişkenlerin sayısı, kuralların sayısı, her bir giriş çıkış değişkeninin tanım uzaylarının bölümlenmesi gibi bulanık mantık kuralı yapılarının ayarlarından oluşur. Uygun kural yapısının elde edilmesinden sonra, denetleyici değişkenlerin ayarlanmasına ihtiyaç duyar. Değişkenlerin ayarlanması bölümünde üyelik işlevlerinin uygun merkezleri, eğimleri, genişlikleri, bulanık mantık kurallarının ağırlıkları hesaplanır.

Sinir ağları öğrenme yöntemlerinin üç kategorisi içinden, harici bir bilgi almaksızın giriş vektörlerindeki kurallara uygun olanları muhafaza ederek, dahili yapılar kuran danışmansız öğrenme, yapıyı öğrenmek için uygundur. Çıkış vektörlerini belirtmek için öğretmen gerektiren danışmanlı öğrenme ve çıkışın sadece tek ölçekli değerlendirmesini gerektiren yaparak öğrenme ise değişken öğrenme için uygundur.

Bir denetleyicinin kendi kendisini düzenleyebilme özelliği, doğrusal olmayan sistemler için kesin bilinmeyen sistem dinamiklerinin ve zamanla değişebilen değişkenlerin denetiminde çok etkili olmasını sağlar. Bulanık denetim sistemlerinin endüstriyel alanlarda uygulanmasıyla birlikte, yüksek performanslı bulanık sistem kurmanın her zaman kolay olmadığı görülmüştür. Literatürde çok gelişmiş tasarım yöntemlerinden bahsedilse de, uygun üyelik işlevlerinin ve bulanık kuralların bulunması için yapılan deneme yanılma işlemi çoğunlukla uzun zaman almakta ve en iyi değerler yine de bulunamayabilmektedir. Bu işlemleri kolaylaştırabilmek için bulanık mantık denetleyici sistemlerine öğrenme algoritmaları eklenmesi fikri doğmuştur ve eğitim veya uyum sağlama algoritmaları ile oluşturulan bu sistemler, uyarlamalı bulanık sistemler olarak bilinmektedir. Bu konudaki ilk tasarımlar 1979 yılında Procky ve Mamdani tarafından ortaya atılan uyarlamalı veya kendi kendisini düzenleyebilen bulanık denetleyiciler olarak bilinmektedirler. Bu tip uyarlamalı modeller genellikle mevcut bilgiye dayanan yöntemlerdir. Bununla birlikte bulanık sistemlerin değişkenlerinin yapay sinir ağları tarafından elde edilmesi de mümkündür. Bu alandaki ilk tasarımlar sinirsel bulanık sistemler (neuro-fuzzy systems veya fuzzy-neural systems) olarak adlandırılmıştır. Günümüzde bu yaklaşımlar çok geneldir ve denetim, veri analizi, karar verme gibi bir çok alanlara uygulanmaktadır. Sinirsel bulanık ağların

yapılarındaki bağlantıların ağırlıkları, yayılmaları ve transfer işlevleri klasik sinir ağlarından farklıdır. Bu tip bulanık denetim sistemlerinde, değişkenlerin öğrenilmesinde genellikle eğimli iniş yöntemi kullanılır.

Klasik uyarlamalı denetim literatüründe, uyarlamalı bulanık denetleyiciler doğrudan ve dolaylı denetim olmak üzere iki gruba ayrılır. Doğrudan uyarlamalı denetimdeki denetleyici değişkenleri, sistem çıkışı ve istenen değer arasındaki hata değerini küçültmek için ayarlanır. Dolaylı uyarlamalı denetimdeki sistem değişkenleri tahmin edilir ve denetleyici değişkenleri, sistem değişkenlerinin doğru değerlerini temsil ettiği kabul edilenlerin içerisinde seçilir.

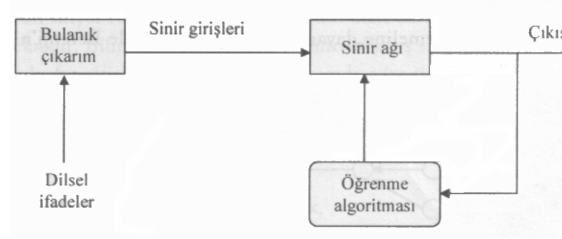
Modern sinirsel bulanık sistemler genellikle ileri beslemeli çok katmanlıdırlar. Son yıllarda bir çok araştırmacı tarafından yoğun olarak ANFIS, FALCON, FuNe, RuleNet, GARIC, NEFCLASS, NEFCON, NEFPROX diye adlandırılan sinirsel bulanık sistemler kullanılmaktadır. Bunun yanı sıra diğer ağ yapılarında da bulanıklaştırma çalışmaları yapılmaktadır.

Sinirsel bulanık ağlarda öğrenmeyle ilgili çok sayıda farklı yaklaşım bulunmaktadır. Jang'ın ANFIS modeli Sugeno benzeri bulanık sistemi içeren bir yapıdadır ve geri yayımlı öğrenmeyi kullanır. Sadece türevlenebilen üyelik işlevlerini kullanarak bunlara ait değişkenleri düzenler. GARIC modeli ise, türevlenebilen özel bir işlev kullanır. Bu tür modellerde Mamdani tipi bulanık sistem kullanılamamaktadır. NEFCON, NEFCIASS ve NEFPROX modelleri ise özel öğrenme algoritmaları ile eğitilirler ve Mamdani tipi bulanık sistemleri kullanırlar.

İleri beslemeli çok katmanlı ağların yanı sıra başka ağ yapıları da bulanık teknikleri kullanabilmektedir. Bunlara örnek olarak kendisini düzenleyebilen ağlar ve bulanık ilişkisel hafızalar verilebilir. Bunlar sadece bulanık sistemin değişkenlerini öğrenebilmektedir.

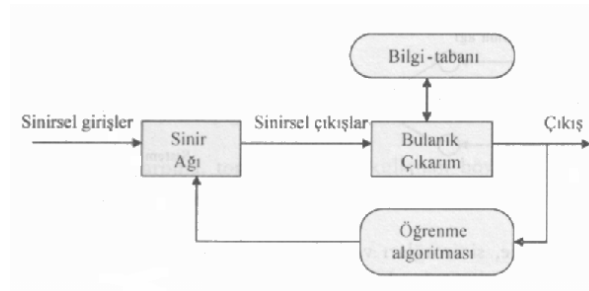
## **5.2 Sinirsel-Bulanık Mantık Ağ Yapıları**

Sinirsel bulanık mantık ağları temel olarak iki yapıdan oluşmaktadır. Birinci sinirsel bulanık mantık yapısında (Şekil 5.1), bulanık çıkarımın dilsel ifadelerle göre oluşturduğu çıkışlar çok katmanlı sinir ağına giriş vektörü olarak verilmektedir. Bu yapıda, sinir ağı eğitilmekte ve istenen çıkışları sağlamaktadır.



Şekil 5.1 Sinirsel bulanık mantık sisteminin birinci yapısı (Elmas, 2003).

İkinci sinirsel bulanık mantık yapısında (Şekil 5.2) ise, çok katmanlı sinir ağının çıktıları bulanık çıkarım mekanizmasını sürmektedir.

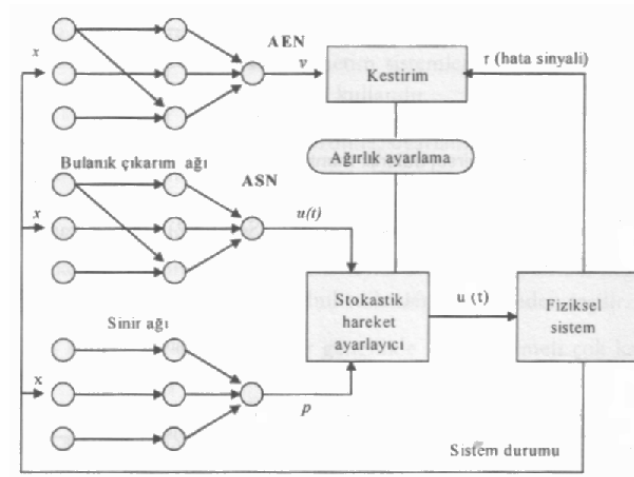


Şekil 5.2 Sinirsel bulanık mantık sisteminin ikinci yapısı (Elmas, 2003).

Bulanık mantık yaklaşımında uzman bilgilerinden oluşturulan kurallar dilsel ifadelerle etiketlenebildiği halde genellikle tasarım deneme yanılma yöntemiyle yapıldığından uzun zaman almaktadır. Sinir ağları kullanılarak bu kurallar oluşturulabilmektedir. Sinirsel bulanık mantık yaklaşımında sinir ağları bulanık mantık sistemlerin karar verme mekanizmasının üyelik işlevlerini uyarlamada kullanılmaktadır.

Temel bir sinirsel bulanık mantık sistemi Berenji'nin ARIC (Approximate Reasoning based Intelligent Control, yaklaşık çıkarım temelli zeki denetim) mimarisidir. ARIC hareket durum değer ağı (Action-state Evaluation Network, AEN) ve hareket seçme ağı (Action Selection Network, ASN) olmak üzere iki çift ileri besleme sinir ağı içermektedir.

ARIC, fiziksel sistemin davranışları ve denetim bilgi tabanından sağlanan kestirimlere göre yapılan uyarlamalarla bulanık mantık denetleyicinin yapay sinir ağının eğitilmesi temeline dayanmaktadır. Şekil 5.3 de Berenji'nin ARIC mimarisi görülmektedir.



Şekil 5.3 Berenji nin ARIC mimarisi (Elmas, 2003).

Bu mimaride, sinir ağları ve bulanık mantık denetimin üstünlüklerinin birleştirilmesi mümkün olmaktadır. Bu sistem öğrenilemekte ve sistem içindeki bilgiler “eğer o halde” kuralları yapısına sahip olmaktadır. Önceden tanımlanan bu kurallar nedeniyle sinirsel bulanık mantık sistemleri sinir ağlarından daha hızlı öğrenirler.

ASN, gerçekte iki ayrı ağdan oluşmaktadır. Ağın biri, bulanık çıkarım bölümünü, diğeri ise, “ $t$ ” ve sistem durumu ( $t+1$ ) zamanında ki ağırlıkları kullanarak bulanık çıkarım değeri  $u(t+1)$  bileşim ölçümü  $p[t, t+1]$  ve hesaplama yapan sinir ağıdır. Bir stokastik uyarlayıcı olarak bulanık çıkarım parçasının denetim değeri  $u(t)$  yi birleştirir ve olasılık değeri  $p$  ile anılır. Bu durumda ASN nin çıkış değerleri,

$$u(t) = o(u(t), p(t, t+1)), \quad (5.1)$$

ile saptanır.

### 5.3 Sinirsel-Bulanık Mantık Ağ Kuramı

Bütün sinyal ve ağırlıkları gerçek sayılar olan basit bir sinir ağı Şekil 5.4 te görülmektedir. Bilindiği gibi, ağın giriş sinirleri giriş sinyallerini değiştiremediğinden çıkışlar girişler ile aynıdır. Bu durumda çıkış ürünü  $p_i$ ,  $x_i$  giriş sinyalinin  $w_i$  ağırlığı ile çarpımından elde edilir.

$$p_i = w_i x_i, i = 1, 2, \dots \quad (5.2)$$

Çıkış düğümünde ürünler toplanarak,

$$ağ = p_1 + p_2 = w_1 x_1 + w_2 x_2 \quad (5.3)$$

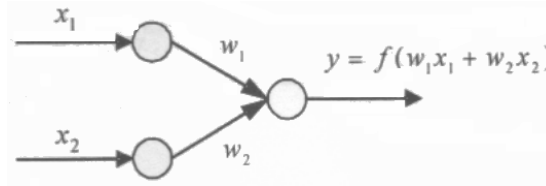
elde edilir. Bu toplam, etkinlik  $f$  eğrisel (sigmoid) işlevi,

$$f(x) = [1 + \exp(-x)]^{-1} \quad (5.4)$$

ile hesaplanırsa ağın çıkışı,

$$y = f(ağ) = f(w_1x_1 + w_2x_2) \quad (5.5)$$

elde edilmiş olur. Çarpma, toplama ve  $f$  i kullanan böyle sinir ağına basit (standart) sinir ağı denir.



Şekil 5.4 Basit bir sinir ağı.

Eğer ağda eğrisel (sigmoid) işlev yerine, t-norm (üçgen-norm) ve t-conorm (üçgen-conorm) gibi diğer işlevler kullanılırsa karma sinir ağları elde edilir.

Bu uyarlamalar bulanık aritmetik işlemlerine dayanan sinirsel bulanık mantık mimarisi fikrinin temelini oluşturmaktadır.

Bir sinirsel bulanık mantık ağında, bütün girişler, çıkışlar ve ağırlıklar gerçek sayılar olmak üzere  $[0, 1]$  arasında alınmalıdır. Genel olarak bir sinirsel bulanık mantık ağında;

- ✓  $x_i$  ve  $w_i$ , t-norm, t-conorm veya diğer sürekli işlevlerle oluşturulabilir.
- ✓  $p_1$  ve  $p_2$ , t-norm, t-conorm veya diğer sürekli işlevlerle oluşturulabilir.
- ✓  $f$  etkinlik işlevi girişten çıkışa her hangi bir sürekli işlev olabilir (Elmas, 2003).

#### 5.4 Sinirsel-Bulanık Mantık Ağ Yapılarında Çıkarım Yöntemleri

Sinirsel bulanık denetleyiciler kullanılan bulanık çıkarım yöntemlerine göre üçe ayrılabilir:

- ✓ Bu tip en çok bilinen  $R_1$ : EĞER  $x = A_1$  ve  $y = B_1$  ise, O HALDE  $z = C_1$  yapısındaki kuralları kullanır. Üyelik işlevleri çan eğrisi, testere dişli veya üçgen şeklinde olabilir. Her bir kuralın tepkisi ateşleme kuvveti ile çıkış üyelik işlevinin etkisidir ve yaklaşık çıkışın üretilmesi için tepkilerin merkezi hesaplanır.
- ✓ Bu tipte S-tip işlev gibi monoton yükselen üyelik işlevleri kullanılır. Her bir kuralın

tepkisi ateşleme kuvvetinin, çıkış üyelik işlevine ayarlanması olarak tanımlanabilir.

✓ Bu tip TSK (Takagi-Sugeno-Kant) bulanık kuralları olarak bilinen,  $R_1$ : EĞER  $x = A_1$  ve  $y = B_1$  ise, O HALDE  $z = f_i(x, \dots, y)$  yapısındaki kuralları kutlanır. Çıkarım yöntemi olarak dördüncü tip çıkarım yöntemi uygulanır. Tepkiler, sonuç; çıkışı hesaplamak için ağırlıkları ve toplamları alınarak ateşleme kuvvetlerine ayarlanır.

### 5.5 Bulanık Kuralların Öğrenilmesi

Geleneksel bilgi sınıflandırıcı yaklaşımları istenen kategoriler için eğitim örneklerinin kümeleri ve bu kümeler arasında ilişkileri içerir. Bulanık sınıflayıcılar ise, *eğer o halde* kurallarını kullanarak sınıflandırma yaparlar.  $K$  adet eğitim girişi,

$$x_p = (x_{p1}, \dots, x_{pn}), p = 1, \dots, K \quad (5.6)$$

olarak tanımlanan bir sistem için oluşturulan bulanık sınıflandırma kuralları şu şekilde belirlenebilir;

EĞER  $x_{p1}$  Küçük ve  $x_{p2}$  Çok Büyük O HALDE  $x_p = (x_{p1}, x_{p2})$   $C_1$  sınıfına aittir.

EĞER  $x_{p1}$  Büyük ve  $x_{p2}$  Küçük O HALDE  $x_p = (x_{p1}, x_{p2})$   $C_2$  sınıfına aittir.

Her bir kuralın etkinlik seviyesi  $\alpha_i$  ise,

$$\alpha_i = \min(f_1(x_{p1}), f_2(x_{p2})) \quad (5.7)$$

şeklinde ifade edilir. Burada  $f_1$ , ve  $f_2$  üyelik işlevleridir. Öğrenme kabiliyeti olan bir sistemde bulanık kural tabanı, kullanılmayan kuralların silinmesiyle iyileştirilebilir.

### 5.6 Üyelik İşlevlerinin Öğrenilmesi

Bulanık modelin performansı üyelik işlevlerin şekline ve aralığına göre değişmektedir. Bu yüzden bulanık denetimde en uygun üyelik işlevlerinin belirlenmesi çok önemlidir. Yapay sinir ağları ve genetik algoritmalar üyelik işlevlerinin belirlenmesi için kullanılabilir. Farklı değişkenlere bağlı olarak değişen üyelik işlevleri için uygun değerlerin yapay sinir ağları ile bulunmasına yönelik çalışmalar 1992 yılında Nomura ve arkadaşları tarafından yapılmıştır. Nomura ve arkadaşları yaptıkları bu çalışmada iki değişkenle tanımlanan simetrik üçgen işlevleri kullanarak en yüksek değer bulduğu noktayı ve genişliğini bulmuşlardır. Ichihashi radyal tabanlı işlevin özel bir şekli olan gauss üyelik işlevini yapay sinir ağlarına öğretmiştir. Üyelik işlevlerinin öğretilmesi için eğimli iniş kullanılarak geri yayımlı

algoritmaya dayanan bir çok yaklaşım ortaya konmuştur. Eğimli iniş yönteminin kullanılabilmesi için üyelik işlevleri ve kural tabanına ait işlevlerin türevlenebilir olması gerekmektedir. Üçgen ve yamuk gibi üyelik işlevleri kullanılamamaktadır. Ancak Sugeno modelinde gauss üyelik işlevi kullanılarak eğimli iniş kolaylıkla uygulanabilmektedir. Nomura ve arkadaşları basit bir Sugeno modelinde kuralların sonuç kısımlarını sabit olarak öğrenme işlemini gerçekleştirmişlerdir. Denetimli öğrenme ile eğimli iniş yöntemi kullanarak değişkenleri öğretmişlerdir. Ancak bu çalışmalarda üyelik işlevlerini üçgen olarak almışlar ve türevlenemez olduklarını ihmal etmişlerdir. Aynı zamanda her kurala ait bulanık kümeler diğerlerinden bağımsız olarak değiştirildiği için farklı bulanık kümelere ait aynı dilsel ifadeler elde edilmiştir. Öğrenme işlemi sonucunda oluşan dilsel ifadeler yorumlandığında kabul edilemez hatalar oluşmaktadır. Her bir dilsel ifadenin sadece bir bulanık küme ile ifade edilmesine yönelik çalışmalar Bersini, Nauck tarafından yapılmıştır.

1993 yılında Jang tarafından ortaya atılan ANFIS modeli çan eğrisi üyelik işlevleri kullanan Sugeno tipi bulanık sistem esas alınarak gerçekleştirilmiştir. Kuralların sonuç kısımları giriş değerlerinin doğrusal bir işlevi olarak tanımlanmıştır. ANFIS te kullanılan öğrenme işlemi ileri geçiş ve geri geçiş olmak üzere iki kısımdan oluşmaktadır. İleri geçişte bulanık küme değişkenleri sabit alınarak sonuç kısmındaki doğrusal işleve ait değişkenler en küçük kare tahmin etme yöntemiyle hesaplanır. Geri geçişte ise bulanık küme değişkenleri eğimli iniş kullanılarak bulunmuştur. Bu döngü tüm sistem hatası istenen hata değerinden küçük oluncaya kadar veya fazla değişim göstermeyinceye kadar devam etmektedir.

Nauck ve arkadaşları tarafından ortaya atılan NEFCLASS modelinde ise denetimli öğrenme kullanılarak sistem hatası istenen hata değerinden küçük oluncaya kadar veya değişim göstermeyinceye kadar eğitim işlemi devam etmektedir (Elmas, 2003).

## 6. pH NÖTRALİZASYON PROSESİ

### 6.1 Proses ve Proses Kontrol

Genel anlamı ile kontrol, canlılarda veya cansız fiziksel sistemlerde, davranışların bir amaca yöneltilmesidir.

Tabiatta, hemen hemen her yerde ve her zaman, birtakım tabiat kanunlarına göre meydana gelen kontrol olaylarına rastlamaktayız. Bu tip kontrole *doğal kontrol* adını vermekteyiz. Doğal kontrole örnek olarak insanın vücut ısısının kas basancı yoluyla sabit tutulması ve yine insanın kanındaki şeker konsantrasyonunun bir hormon vasıtasıyla sabit tutulması verilebilir.

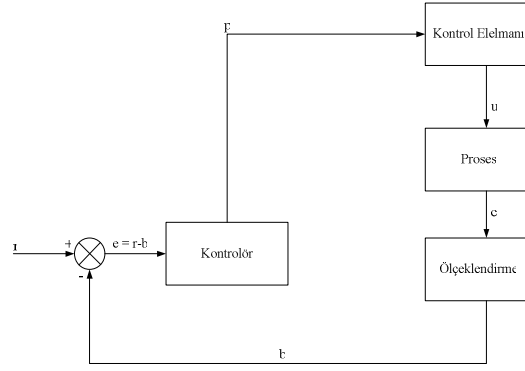
Doğal kontrol, vuku bulduğu ortamlardan ötürü, daha çok mühendislik dışı bilim dallarını ilgilendirmektedir. Örnek olarak verilen doğal kontrol olayları, mühendislerden ziyade fizyologların konusudur, Doğal kontrol, olaylarında, insan elemanı kontrolde aktif bir rol oynamaz. Ancak kontrol düzeneği bozulduğunda insan dışarıdan müdahale ederek düzelmeyi sağlayabilir.

İnsanın bizzat kendisinin veya zihinsel gayreti ile oluşturduğu cihazların yardımıyla gerçekleştirilen bir diğer kontrol şekli vardır ki, bu kontrole de *yapay kontrol* denilmektedir.'

Uygulandığı konular ve uygulama elemanlarına değişik olmasına rağmen, özellikle özellikle gerçekleştirme şekilleri bakımından doğal kontrol ile yapay kontrol arasında ilginç paralellikler bulunmaktadır. Yapay kontrol, sosyal, ekonomik ve özellikle teknik karakterdeki olaylar için söz konusudur. Mühendislerin ilgisi, konuları gereği *teknik yapay kontrol ve kimya mühendisi* olarak da bizlerin ilgisi *proses kontrol* üzerine yoğunlaşmaktadır (Şaşmaz, 1990). Fakat bugün *kontrol mühendisliği* kendi başına bir bilim dalı halini almış olup, hızlı bir gelişim ve profesyonelleşme içindedir.

Proses, maddelerde fiziksel ve kimyasal değişikliklerin oluşmasındaki işlemler zincirinin bütünüdür. Herhangi bir prosesin verimliliğine iç veya dış bir takım etkenler etki eder. Bu etkenler *proses değişkenleri* (proses parametreleri) olarak tanımlanır. Proses değişkenlerine örnek olarak örnek olara ise; sıcaklık, basınç, debi, **pH**, konsantrasyon, seviye vb. gibi birtakım parametreleri sayabiliriz (Söğüt, 2004).

Konu proses kontrol olunca, Şekil 6.1 bu konuda yeterince açıklayıcı olacaktır (genel yapı açısından).



Şekil 6.1 Temel elemanlarla sinyalleri içeren proses kontrol diyagramı (Johnson, 2003).

### Genel Olarak pH Kontrolü;

pH nötralizasyon prosesleri doğrusal olmayan matematik ifadelerle tanımlanırlar. Şimdiye kadar doğrusal sistemlerin kontrolü konusundaki kuramlar iyice anlaşılmiş ve oturmuştur. Bu durum, neredeyse tümü doğrusallıktan az veya çok sapan fiziksel sistemlerin doğrusal bağıntılarla incelenmesine dayalı bakış açısının haklılığını göstermektedir. Buna karşın birçok kimyasal sistem tepkime kinetiği ve termodinamik bağıntılar nedeniyle doğrusallıktan oldukça sapmaktadır. Birçok kimyasal ve biyokimyasal prosesin tabiatında doğrusallıktan sapma bulunur. Bunların çoğu doğrusal kontrol teorisi vasıtasıyla yeterince iyi kontrol edilebilmektedir. Çevresel kısıtlamalar, enerjinin idareli kullanımı, ürün kalitesinin korunması, üretim esnekliği gibi etmenler proses işlemlerini karmaşıktırılmış, daha geniş işletme aralıklarını gerektirmeye başlamıştır. Bu durumda doğrusal yöntemlerle kontrol tatmin edici sonuçlar vermekten uzaklaşmıştır. Kimyasal sistemlerin tasarımında doğrusallık varsayımı gerçekten çeşitli derecelerde saptmaya neden olmaktadır. Doğrusallık varsayımı yapabilmek için oldukça dar işletim aralıklarında çalışılması gerekmektedir. Üstelik kimyasal süreçlerin özelliklerinin zamana bağlı olarak değişmesi ve ciddi gecikmeler içerebilmeleri, işi iyice karmaşıktırılmaktadır. Bu etmenlerin ışığında, kimyasal süreçlerin doğrusal kontrolle etkin şekilde idare edilmesi genellikle mümkün olamamaktadır. Bu gelişmeler, doğrusal olmayan kontrol edicilere duyulan ilgiyi arttırmıştır.

pH sisteminin doğrusallıktan sapması, statik pH yanıtının “S” şeklinde olmasından kaynaklanmaktadır. Bu konuya çeşitli yaklaşımlar ileri sürülmüştür. Bunlardan bazıları *Jenerik model kontrol*, *dahili model kontrolü (internal model control, IMC)*, *tepkime sabiti kontrol (reaction invariant kontrol)* ve *kazanç ayarlamalı (seçimli) PI kontrolüdür*. Birçok çalışmada adaptif kontrol yöntemi pH kontrolünde uygun seçim olarak gösterilmiştir. Yine de

uyarlamalı (adaptif) kontrol düzeninin sürekli işletiminde tatmin edici bir uzun vadeli kontrol davranışı elde edilememiştir. Uyarlamalı kontrol yönteminin kullanımı bazı hallerde proses modelinin işaretini değiştirmek suretiyle vanaları yanlış bir seçimle tam açık hale getirerek sorunlara neden olmuştur. Bunun üzerine alışılmadık pH yanıtlarının gözlemlendiği durumlarda uyarlama seçeneği genellikle iptal edilir (Regunath ve Kadirkamanathan, 2001). Birçok araştırmacı doğrusallıktan sapmanın üstesinden gelmek için modele dayalı proses kontrol üzerine incelemeler yapmışlardır. Proseslerin modeli kontrol sistemi tasarımında önemli bir rol oynamaktadır. Model esaslı kontrolde bu rol ön planda, kontrol parametrelerinin ayarlanmasına dayalı yöntemlerde ise arka plandadır (Doherty, 1999). Regunath ve Kadirkamathan (2001), pH nötralizasyon sürecinin kontrolüne basit bir bakış açısı geliştirmiştir. Kontrol çalışmasının amacı, akımlardaki bileşim ve/veya debi değişimlerinden kaynaklanan yük değişimlerinin varlığında pH'ı istenen yörüngede tutabilmektir. Önerilen kontrol ediciyle iyileştirilmiş başarımın sağlandığı deneysel sonuçlarla gösterilmiştir. Kullanılan kontrol edici basit bir yapıda olup, sayısal hesaplama yükü de fazla olmamıştır. Doğrusal olmayan kontrol teorisindeki çeşitli ilerlemelere rağmen, kazanç seçmeli PI kontrol sanayide tercih edilen seçenek olmuştur. PI kontrol edicilerin standardı, geri karışmalı kapta geçerli pH değerine bağlıdır. pH değeri düşükse buna bağlı olarak kontrol edici kazancı da değişir. Kazanç tespitinde iki boyutlu *sugeno* tipi bulanık çıkarım (Sugeno type fuzzy inference system - FIS) kullanılmıştır. Bu kontrol düzeni performansta büyük ölçüde iyileşme sağlamıştır. Ayrıca getirdiği hesap yükünün hafifliğinin yanı sıra anlaşılıp uygulanması kolaydır. Tüm bu olumlu yanları bu tip kontrol anlayışının sanayide yaygınlaşmasını kolaylaştırmaktadır (Regunath, 2001).

Ancak bu çalışmada hem yapay sinir sisteminin hem de bulanık mantık kontrol sisteminin bir entegrasyonu ile ortaya çıkarılmak istenen kontrolör kullanılacaktır. Bu yol ile yapay zekanın non-lineer sistemleri kontrol etmedeki başarımı ortaya çıkarılmak istenmektedir.

## 6.2 pH Kavramı ve Nötralizasyon İşlemi

Çalışmanın bu bölümünde ise, genel olarak asit-baz yapıları ile nötralizasyon işlemi üzerinde durularak prosesin daha tanınabilmesi için bir bilgi (veritabanı) oluşturulacaktır.

### 6.2.1 Asitler ve Bazlar

Dengenin en büyük önem taşıdığı konular şüphesiz asit ve bazlarla ilgili olanlardır. Asit-baz

dengelerinin matematik problemlerine girişmeden önce, asit ve bazların isimlendirilmesini ve sınıflandırılmasını ele almamız gerekmektedir. Zira ele alınan prosesin yapısının daha iyi anlaşılabilmesi için bu gereklidir.

*Arrhenius a göre asitler ve bazlar;*

İlk zamanlarda tadı ekşi olan maddelere asit dendi (Latince acidus, ekşi, acetum, sirke). Asitlerin etkisini ortadan kaldıran, nötrleştiren maddelere ise alkaliler (Arapça al kali, bitki külü) ismi verildi. Önceleri her asidin belli bir element, oksijen (Yunanca oxus, ekşi; gennae, doğuran) ihtiva etmesini gerektiğine inanılırdı, fakat 1810 da Davy hidroklorik asidin yalnız hidrojenle klordan oluştuğunu gösterdi. Bu bulgudan hemen sonra bütün asitlerin temel bileşeninin hidrojen olduğu fikri kabul edildi.

1880 ile 1890 arasında Arrhenius tarafından geliştirilen iyonların ayrışması (dissosyasyonu) teorisinin en önemli sonuçlarından biri asitlerin niçin farklı kuvvette olduğunun açıklanabilmesi olmuştur. Asit çözeltilerinin kimyasal aktiflikleri ve iletkenlikleri, biri  $H^+$  olan iyonlarına ayrışmasının (dissosyasyonunun) bir sonucu olarak kabul edildi (Mahan, 1989):



Farklı asitlerin kuvvetlerinin farklı olması bunların ayrışma (dissosyasyon) derecelerinin de farklı olması sonucuyla açıklanıyordu.

Çözeltiye hidroksil iyonu verdikleri kabul edilen bazların davranışı için de buna benzer bir açıklama yapılmaktaydı:



Yani asit özelliklerinden proton, baz özelliklerinden ise hidroksil iyonu sorumluydu.

Bu görüş kimya teorisinin gelişmesine büyük katkıda bulunurken ortaya bazı güçlüklerin de çıkmasına yol açtı. Bunlardan birincisi sulu çözeltideki protonun niteliği, ikincisi ise hidroksil iyonuna sahip olmadıkları halde baz özelliği gösterebilen maddeler ile ilgili olarak ortaya çıkmıştır.

*Lowry-Brönsted e göre asitler ve bazlar;*

Yukarıda belirtilen güçlükler, 1923 te yeni bir asit ve baz kavramının ortaya çıkmasına yol

açmıştır. Lowry ve Brönsted in ismini taşıyan bu tanıma göre; proton verebilen bütün maddeler asit, proton alabilenler ise bazdır. Buna göre HCl in iyonlaşması HCl in suya bir proton vermesi (asit olarak davranması) suyun ise bu protonu alması (baz olarak davranması) şeklinde düşünülebilir:



Bu reaksiyon tersinir niteliktedir, zira  $Cl^-$  iyonu  $H_3O^+$  dan bir proton alarak tekrar HCl haline dönüşebilir. Öyleyse klorür iyonu bir baz,  $H_3O^+$  iyonu ise bir asittir. HCl ve  $Cl^-$  çifti birbirinden bir tek proton bakımından farklı olduğundan bunlara konjüge asit-baz çifti denmektedir (Mahan, 1989).

*Lewis e göre asitler ve bazlar;*

1923 te asit ve bazlar için G. N. Lewis tarafından başka bir genel tanımlama önerilmiştir, Lewis e göre elektron alan her madde asit, elektron veren her madde bazdır. Böyle asit-baz reaksiyonları için iki örnek aşağıda gösterilmiştir.



Buradan Lewis tanımının proton alışverişini içermeyen reaksiyonları da kapsadığını görüyoruz.

Lowry-Brönsted bazlarının da, protona elektron verdiklerinden, aynı zamanda Lewis bazı oldukları açıktır. Öte yandan bir Brönsted asidinin başka bir moleküle verebileceği bir protona sahip olması gereklidir. Halbuki Lewis asidi için bu şart yoktur. Bu sebeple, proton alışverişleriyle el ele giden sulu çözeltilerdeki asit baz reaksiyonları için Lowry-Brönsted, protonun mevcut olmadığı diğer maddeler için Lewis tanımı büyük önem taşır.

## 6.2.2 pH Ölçeği

Uğraştığımız çözeltiler genellikle çok seyreltik olduğundan, bunlardaki hidrojen iyonlarının konsantrasyonu da çok kere küçüktür. Bunun sonucu olarak hidrojen iyonları konsantrasyonu sık sık litrede mol olarak  $10^{-3}$  un negatif kuvvetleri cinsinden ifade olunur. Örneğin doymuş bir  $CO_2$  çözeltilinde hidrojen iyonu konsantrasyonu  $1,3 \cdot 10^{-3}$  M, 0,5 M asetik asit çözeltilinde  $3 \cdot 10^{-3}$  M dir. Böyle konsantrasyonları daha basit ve kısa olarak ifade etmek için bunların negatif logaritmalarını vermek daha uygundur. Buna göre pH;

$$pH = -\log[H_3O^+] \quad (6.5)$$

şeklinde tanımlanır.

Konsantrasyonu ifade etmek için negatif logaritma kullanılması yalnız hidrojen iyonlarına özge bir durum değildir. Örneğin pOH hidroksil iyonları konsantrasyonunun negatif logaritması, pAg aynı şekilde gümüş iyonları konsantrasyonunun negatif logaritmasıdır. Denge sabiti değerleri de böyle negatif logaritma cinsinden ifade edilebilir ve pK olarak gösterilir.

### 6.2.3 Suyun Kendi Kendine İyonlaşması

Suyun hem asit hem de baz olarak etkiyebileceği bilinmektedir. Bu bakımdan saf suda da;



reaksiyonunun az fakat ölçülebilir oranda ortaya çıkması bizim için şaşırtıcı değildir. Sabit konsantrasyonları, her zaman ki gibi, denge sabiti bağıntısının içine alırsak (Mahan, 1989);

$$K_{su} = [H_3O^+][OH^-] \quad (6.7)$$

denge sabiti eşitliğini elde ederiz.  $K_{su}$  suyun iyonlar çarpımı sabitidir ve değeri 25 °C de, konsantrasyonlar litrede mol olarak ifade edilirse  $10^{-14}$  e eşittir.

Bu tanımlamaya göre eşit konsantrasyonda  $H_3O^+$  ve  $OH^-$  iyonları ihtiva eden bir çözelti ne asitli ne de baziktir. Yani, 25 °C de nötral bir çözeltilde  $H_3O^+$  ve  $OH^-$  iyonları konsantrasyonu;

$$\begin{aligned} [H_3O^+] &= [OH^-] \\ [H_3O^+][OH^-] &= 10^{-14} \end{aligned} \quad (6.8)$$

olduğundan,

$$[H_3O^+] = [OH^-] = 10^{-7} M \quad (6.9)$$

olur. Yahut nötral bir çözelti için,

$$pH = pOH = 7 \quad (6.10)$$

diyebiliriz.

Suyun iyonlaşmasının bir çözeltildeki hidrojen ve hidroksil iyonları konsantrasyonuna şüphesiz bir katkısı vardır ama bu asit ve baz çözeltilerindeki hidrojen ve hidroksit iyonları konsantrasyonunun hesaplanmasını pek ender olarak zorlaştıran bir faktördür.

Çözünen asit veya bazdan gelen iyonlara oranla suyun iyonlaşmasından gelen  $H_3O^+$  ve  $OH^-$  iyonlarının önemi hakkındaki düşüncelerimiz, zayıf asit ve bazlar hesapları da

kolaylaştırmaktadır. Kısaca diyebiliriz ki çözünen bir asitten suya  $10^{-6}$  M kadar veya daha fazla  $H_3O^+$  iyonu geliyorsa, suyun dissosiyasyonundan ileri gelen  $H_3O^+$  iyonlarının konsantrasyonunu bunun yanında ihmal edebiliriz. Bazlar ve  $OH^-$  konsantrasyonu için de aynı şey söylenebilir (Mahan, 1989).

#### 6.2.4 Asitlerin ve Bazların Kuvveti

Lowry-Brönsted tanımına göre kuvvetli bir asit başka bir moleküle bir proton aktarmak için büyük bir eğilim, kuvvetli bir baz ise proton almak için büyük bir eğilim gösterir. Buna göre;



gibi bir reaksiyondaki asidin kuvvetini reaksiyona katılan maddelerin reaksiyon ürünlerine dönme yatkınlığı derecesine göre ölçebiliriz. Fakat bu konu üzerinde biraz düşünersek, reaksiyonun ürünler yönünde ilerlemesinin yalnız asit1 in kolayca proton kaybetmesine değil aynı zamanda baz2 nin bu protonu alma yatkınlığına da bağlı olduğunun farkına varmalıyız. Proton alış verişinin derecesi hem asit1 in hem de baz2 nin özelliklerine bağlı olduğuna göre çeşitli asitlerin kuvvetini karşılaştırabilmek için bunların aynı baza proton verme eğilimlerini bilmemiz gereklidir. Çeşitli asitlerin suya proton verme yatkınlıklarına göre bunları asit kuvveti yönünden sıralamak mümkündür. Bir asidin kuvvetinin nicel ölçüsü bu asidin dissosiyasyon sabiti yani;



reaksiyonu için,

$$k = \frac{[H_3O^+][A^-]}{[HA]} \quad (6.13)$$

denge sabitidir. Çizelge 6.1 de iyi bilinen bazı asitlerin dissosiyasyon sabitleri verilmiştir.

Çizelge 6.1 Asitlerin çözünme sabitleri (Mahan, 1989).

Asit	Reaksiyon	K
Asetik asit	$\text{CH}_3\text{COOH} + \text{H}_2\text{O} = \text{H}_3\text{O}^+ + \text{CH}_3\text{COO}^-$	$1,85 \times 10^{-5}$
Formik asit	$\text{HCOOH} + \text{H}_2\text{O} = \text{H}_3\text{O}^+ + \text{HCOO}^-$	$1,8 \times 10^{-4}$
Siyantür asidi	$\text{HCN} + \text{H}_2\text{O} = \text{H}_3\text{O}^+ + \text{CN}^-$	$4,8 \times 10^{-10}$
Hidrojen sülfat iyonu	$\text{HSO}_4^- + \text{H}_2\text{O} = \text{H}_3\text{O}^+ + \text{SO}_4^{2-}$	$1,2 \times 10^{-2}$
Fluorlu hidrojen	$\text{HF} + \text{H}_2\text{O} = \text{H}_3\text{O}^+ + \text{F}^-$	$6,8 \times 10^{-4}$

Son olarak, bir asidin kuvvetiyle konjüge bazı arasında da bir bağıntı bulunduğuna işaret edelim. Örneğin, proton verme yatkınlığı büyük olduğundan HCl kuvvetli bir asittir diyoruz. Fakat bu asidin konjüge bazı olan klorür iyonunun da proton alma yatkınlığı çok küçüktür. Yani  $\text{Cl}^-$  zayıf bir bazdır. Genellikle bir asit kuvvetli ise konjüge bazının zayıf, zayıfsa konjüge bazının kuvvetli olduğunu söyleyebiliriz.

Aslında asitlerin ve bazların çözünürlüğü üzerindeki bir diğer etmen de, *sıcaklık* olup, bu çalışmanın teorik bir çalışma olmasından dolayı bu etmen ön plana çıkarılmamıştır. Zira çalışmanın amacı; sistemin lineer olmayan yapısını kullanarak, yapay zeka ürünü olan kontrol sistemlerinin başarımını kanıtlamaktır.

### 6.2.5 Zayıf Asitler ve Zayıf Bazlar

Laboratuarda çok kullanılan asetik asit tipik bir zayıf asit olarak kabul edilir. Öyleyse uğraşacağımız reaksiyon şudur:



Kolaylık olmak üzere asetik asiti HOAc ve asetat iyonunu OAc simgeleriyle gösterelim. Buna göre denge sabiti ifadesi;

$$\frac{[\text{H}_3\text{O}^+][\text{OAc}^-]}{[\text{HOAc}]} = K = 1,85 \cdot 10^{-5} \quad (6.15)$$

Olacaktır. Bu denge sabitinin değeri zayıf asitler için kabul edilen sınırların aşağı yukarı ortasında olduğundan asetik asit tipik bir zayıf asit olarak kabul edilmektedir.

İçinde saf suda çözülmüş  $C_0$  mol asetik asit bulunan bir litrelik bir çözeltimiz olsun. Bu asidin bir kısmı  $\text{H}_3\text{O}^+$  ve  $\text{OAc}^-$  iyonlarına dissosiyeye olacak, miktarını bilmediğimiz bir kısmı da dissosiyeye olmadan geri kalacaktır. Denge sabitinden yararlanarak denge durumundaki  $\text{H}_3\text{O}^+$  ve  $\text{OAc}^-$  iyonlarının konsantrasyonlarının hesaplanması işlemi ise, *prosesin matematiksel*

*modelinin çıkarılması* bölümünde detaylı olarak ele alınacaktır.

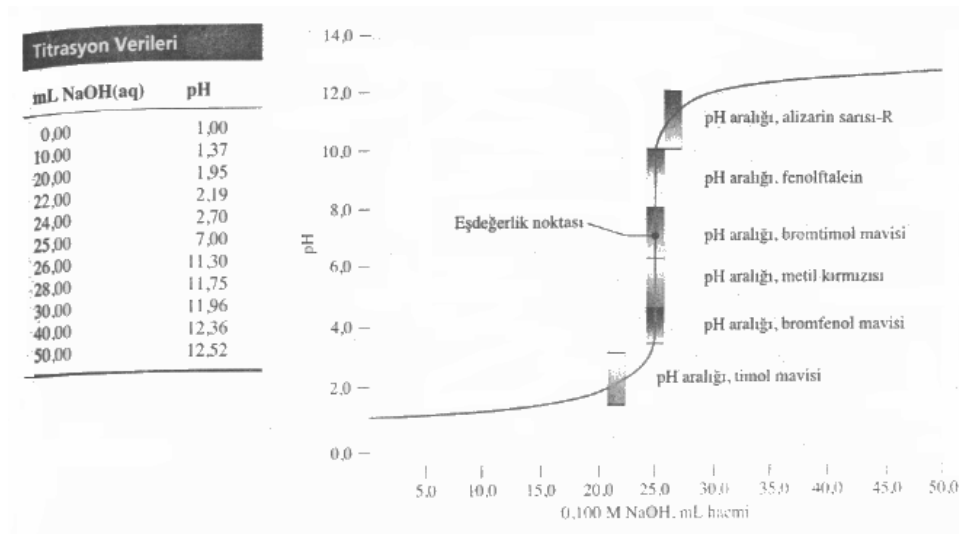
### 6.2.6 Asit-Baz Titrasyonları

Asit-baz titrasyonu analitik kimyanın en önemli tekniklerinden biri sayılır. İşlemin temeli, örneğin bir asidin miktarını tayin etmek için bir bazın eşdeğer ölçülü miktarını ilave etmek veya bunun tersidir.

Bu noktada bizim için bir temel teşkil etmesi açısından kuvvetli asit-kuvvetli baz titrasyonu ile çalışma konumuzu oluşturan zayıf asit-kuvvetli baz titrasyonunun temel bilgilerini edinmek yerinde olacaktır.

#### 6.2.6.1 Kuvvetli Asit-Kuvvetli Baz Titrasyonları

Bir beher ya da erlene 25,00 mL 0.100 M HCl (kuvvetli asit) koyduğumuzu ve bir büretten 0.1 M NaOH (kuvvetli baz) eklediğimizi düşünelim. Titrasyonun belli noktalarında pH değerlerini hesaplayabilir ve bu değerleri eklenen NaOH hacmine karşı grafiğe geçirebiliriz. Bu titrasyon eğrisinde eşdeğerlik noktası pH sını saptayabilir ve ona göre uygun bir indiktor seçebiliriz. Şekil 6-2 de HCl-NaOH titrasyon eğrisi verilmiştir.



Şekil 6.2 Kuvvetli asitin kuvvetli baz ile titrasyonu (25.00 ml 0.100 M HCl ve 0.100 M NaOH) (Petrucci, 2002).

Bu şekilden kuvvetli asitin kuvvetli baz ile titrasyonları eğrilerinin önemli özelliklerini görebiliriz.

- ✓ Titrasyon başlangıcında pH değeri küçüktür.
- ✓ Eşdeğerlik noktasının biraz öncesine kadar pH değişmesi yavaş olur.

- ✓ Eşdeğerlik noktasında pH hızla yükselir. Bu yükselme, örneğin 0.10 mL (2 damla) baz eklenmesiyle 6 pH birimi kadar olabilir.
- ✓ Eşdeğerlik noktasından sonra pH yükselmesi yine yavaş olur.
- ✓ Renk değişimi pH = 4 - 10 aralığına düşen bir indikatör bu titrasyon için uygundur (Petrucci, 2002).

Kuvvetli bir bazın kuvvetli bir asit ile titrasyon eğrisini de Şekil 6.2 deki gibi titrant hacmine karşı (kuvvetli asit) pOH çizerek elde edebiliriz. Bu durumda da yukarıdaki örnek için sıralana özellikler pH yerine pOH kullanılması dışında benzerlik gösterir.

### 6.2.6.2 Zayıf Asit-Kuvvetli Baz Titrasyonları

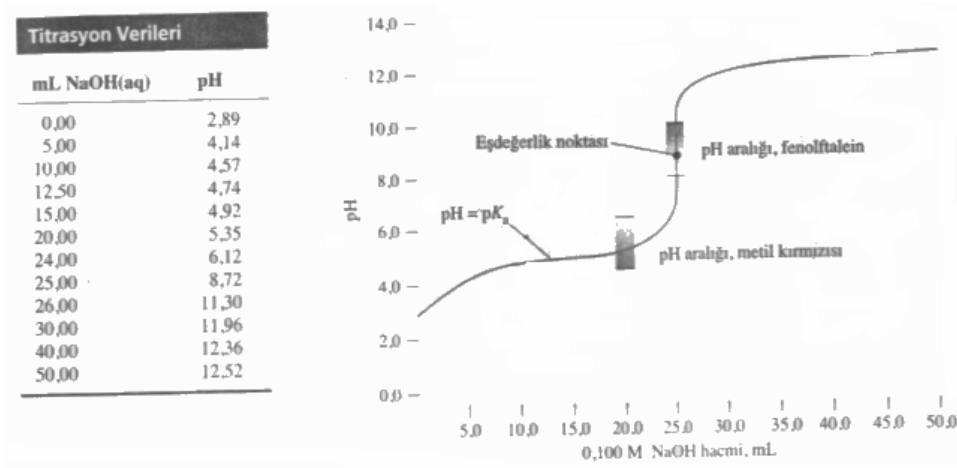
Kuvvetli asit ve zayıf asit titrasyonları arasında önemli bir çok fark vardır. Ancak bu ikisi arasındaki ortak nokta şudur:

*Aynı molaritedeki asit çözeltilerinin eşit hacimdeki miktarlarını eşdeğerlik noktasına ulaştıracak bazın hacmi, asidin kuvvetli ya da zayıf olmasına bağlı değildir.*

Örneğin, HC<sub>2</sub>H<sub>3</sub>O<sub>2</sub> gibi zayıf bir asidin nötrleşmesini, HC<sub>2</sub>H<sub>3</sub>O<sub>2</sub> moleküllerindeki protonların doğrudan doğruya OH<sup>-</sup> iyonlarına geçişi gibi düşünebiliriz. Oysa, kuvvetli asitle OH<sup>-</sup> iyonuna geçen proton, daha önce kuvvetli asit tarafından suya proton verilerek oluşturulan H<sub>3</sub>O<sup>+</sup> iyonundan gelir. Ancak her iki durumda da asit ve bazın mol oranları 1:1 dir.



Zayıf asitin kuvvetli baz ile titrasyonunda, zayıf asitin kısmi iyonlaşmasını dikkate almak için hesaplama stokiyometrik ve denge olarak ikiye ayrılır. Ancak hesaplama tampon çözeltiliye kuvvetli baz eklemeye benzer. Şekil 6.3 de 25,00 mL 0,100 M HC<sub>2</sub>H<sub>3</sub>O<sub>2</sub> nin 0.100 M NaOH ile titrasyonu verilmektedir.



Şekil 6.3 Zayıf asidin kuvvetli baz ile titrasyonu (0.100 M NaOH ve 25.00 ml 0.100 M  $\text{HC}_2\text{H}_3\text{O}_2$ ) (Petrucci, 2002).

- ✓ Başlangıç pH sı kuvvetli bir asidin titrasyonunda görülenden daha yüksektir (daha az asidik). Zayıf asit kısmen iyonlaşmıştır.
- ✓ Titrasyon başlangıcında pH da biraz keskin bir artma görülür (zayıf asidin nötrleşmesi ile oluşan anyon, ortak iyon olduğu için. asidin iyonlaşmasını biraz engeller).
- ✓ Eşdeğerlik noktasının öncesinde, titrasyon eğrisinin önemli bir bölümünde, pH yavaş bir artış gösterir (eğrinin bu bölümlerinde tampon çözelti söz konusudur).
- ✓ Yarı nötrleşme noktasında  $[\text{HA}] = [\text{A}^-]$  olacağı için  $\text{pH} = \text{pK}_a$  dır.
- ✓ Eşdeğerlik noktasında pH 7 den büyüktür (zayıf asidin eşlenik bazı hidroliz olarak  $\text{OH}^-$  oluşturur).
- ✓ Eşdeğerlik noktasından sonra titrasyon eğrisi kuvvetli asit-baz titrasyon eğrisine benzer (titrasyon eğrisinin bu bölgelerinde pH tepkimeye girmemiş fazla  $\text{OH}^-$  kullanılarak hesaplanır).
- ✓ Eşdeğerlik noktası dolayında titrasyon eğrisindeki dik yükselme dar bir pH (7-10) aralığında olur.
- ✓ Titrasyon için indikatör seçimi kuvvetli asit-baz titrasyonlarından daha sınırlıdır (pH 7 nin altında renk değişimi gösteren indikatörler kullanılamaz) (Petrucci, 2002).

Şekil 6.4 de gösterildiği gibi, zayıf asit-kuvvetli baz titrasyon eğrilerinin hesaplamaları dört farklı yaklaşımla yapılır. Bu hesaplamalar eğrinin değişik bölümlerine göre değişir



Şekil 6.4 Zayıf asit kuvvetli baz titrasyon eğrisini oluşturulması (Petrucci, 2002).

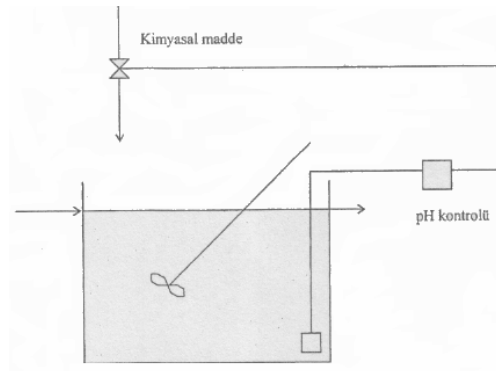
## 6.2.7 Nötralizasyon İşlemi

### 6.2.7.1 Atık Sularda pH Nötralizasyon Prosesi

Bu işlem atık suyun pH sınırın ayarlanması işlemidir. Bu işlem, atık suyun alıcı ortama veya kanalizasyona deşarjından önce deşarj standartlarını sağlamak için pH ın nötr pH civarına getirilmesi, arıtma düzenlerinde biyolojik arıtmaya girişten önce pH ayarlanması ve kimyasal çöktürme için uygun pH sağlanması amacıyla yapılır.

Bilindiği gibi suların ne asit ne de baz özelliği gösterdikleri tek nokta pH ın 7 olduğu noktadır. Bu noktanın yakalanması su kalitesinin en önemli unsurlarından birisidir ve 7 ve 8 nolu bölümlerde detaylarıyla geliştirilen nöro-fuzzy kontrolörün performansı ve başarımı ele alınacaktır.

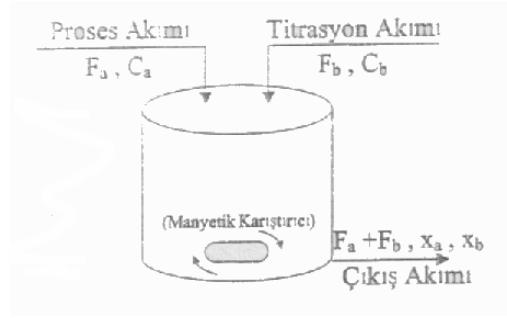
Aşağıda tipik bir aç-kapa tipi karışımli nötralizasyonun diyagramı gösterilmiştir.



Şekil 6.5 Aç-kapa tipi karışımli nötralizasyon (Akkaya, 2001).

### 6.2.7.2 Prosesin Matematiksel Modelinin Çıkarımı

pH prosesi asidik ve bazik iki tepkenin tam karışma varsayımı ile nötralizasyonu olarak düşünülebilir. Buradaki pH prosesinde bunlardan birincisi bir zayıf asit olan asetik asit ( $\text{CH}_3\text{COOH}$ ), ikincisi ise bir kuvvetli bir baz olan sodyum hidroksit ( $\text{NaOH}$ ). Geri karışmalı reaktörün (GKR) akımları proses giriş akımı ve titrasyon giriş akımı ve bunların kütlece toplamından ibaret olan çıkış akımıdır (Şekil 6.6).



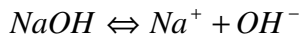
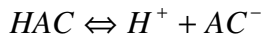
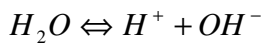
Şekil 6.6 İki giriş akımlı geri karışmalı reaktör (Söğüt, 2004).

Aşağıdaki matematik model 0.1 M asetik asit ve 0.1 M sodyum hidroksit için çıkarılmıştır. Karışım dinamiği iki denklemlle ifade edilir;

$$V \frac{dx_a}{dt} = F_a C_a - (F_a + F_b) x_a \quad (6.17)$$

$$V \frac{dx_b}{dt} = F_b C_b - (F_a + F_b) x_b \quad (6.18)$$

Yine asetik asit ile sodyum hidroksit arasındaki reaksiyon;



(6.19)

Elektronötrallik gereği iyonik yükler toplamı sıfır olmalıdır;

$$[\text{Na}^+] + [\text{H}^+] = [\text{AC}^-] + [\text{OH}^-] \quad (6.20)$$

Yine asetik asitin ve suyun çözünme sabitleri;

$$K_a = \frac{[\text{AC}^-][\text{H}^+]}{[\text{HAC}]} = 1,778 \cdot 10^{-5} \quad (6.21)$$

$$K_{su} = [\text{H}^+][\text{OH}^-] = 1,0 \cdot 10^{-14}$$

olup, eşitlik 6.20 den aşağıdaki denklem elde edilir.

$$[H^+]^3 + [H^+]^2 \{K_a + x_b\} + [H^+] \{K_a(x_b - x_a) - K_{su}\} - K_{su} K_a = 0 \quad (6.22)$$

$pK_a = -\log K_a$ , olarak tanımlanırsa;

$$x_b + 10^{-pH} - 10^{pH-14} - \frac{x_a}{1 + 10^{pK_a - pH}} = 0 \quad (6.23)$$

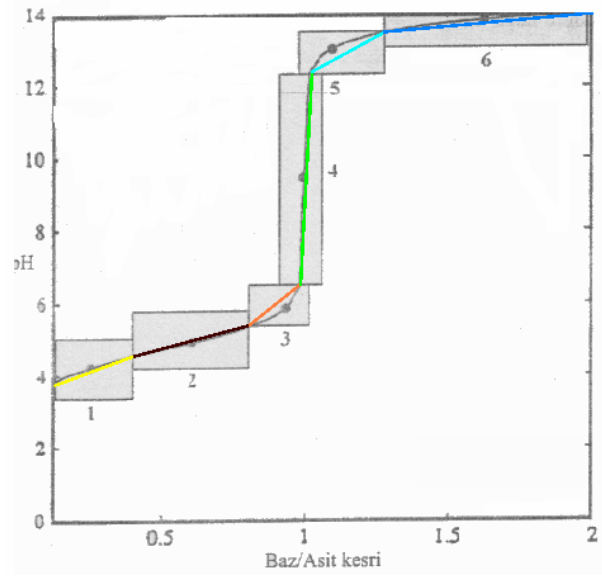
eşitliği elde edilir (Leondes, 1998).

Bu eşitlikten pH değeri rahatlıkla elde edilebilir. Yine bilgisayar simülasyonlarında eşitlik rahatlıkla kullanılabilir. Ancak “8” bitlik basit mikroişlemcili çalışma düzeneği için sistemin lineer çalışma bölgelerine ayrılması gerekmektedir. Bunun için titrasyon eğrisinin şu lineer bölgelere ayrılması ve her bir lineer bölgenin de doğrusal fonksiyonunun elde edilmesi gerekmektedir. Adı geçen tüm bu bilgiler Çizelge 6.2 de bulunabilir.

Çizelge 6.2 Lineer çalışma bölgeleri ve her bir bölgeye ait doğru denklemi (x: Asit/Baz).

<i>Lineer Çalışma Bölgesi (Baz/Asit)</i>	<i>Karşılık Düşen pH Aralığı</i>	<i>Elde Edilen Lineer Doğru Denklemi</i>
$0 < x \leq 0.4$	$3.7 < \text{pH} \leq 4.5$	$y = 2x + 3.7$
$0.4 < x \leq 0.8$	$4.5 < \text{pH} \leq 5.5$	$y = 2.5x + 3.7$
$0.8 < x \leq 0.95$	$5.5 < \text{pH} \leq 7.0$	$y = 10x - 2.5$
$0.95 < x \leq 1.05$	$7.0 < \text{pH} \leq 12.5$	$y = 70x - 61$
$1.05 < x \leq 1.3$	$12.5 < \text{pH} \leq 13.5$	$y = 4x + 8.3$
$1.3 < x \leq 2.0$	$13.5 < \text{pH} \leq 14.0$	$y = 0.7x + 12.6$

Bu formüller oluşturulurken bir doğrunun denklemi olan,  $y = ax + b$  yapısı ile alt ve üst değerler kullanılmıştır. Artık bu ifadelerin basit bir mikroişlemci ile bile ele alınabileceği durumu ortadır. Görsellik açısından, Şekil 6.7 çalışma bölgelerinin daha iyi görülmesini sağlayacaktır.



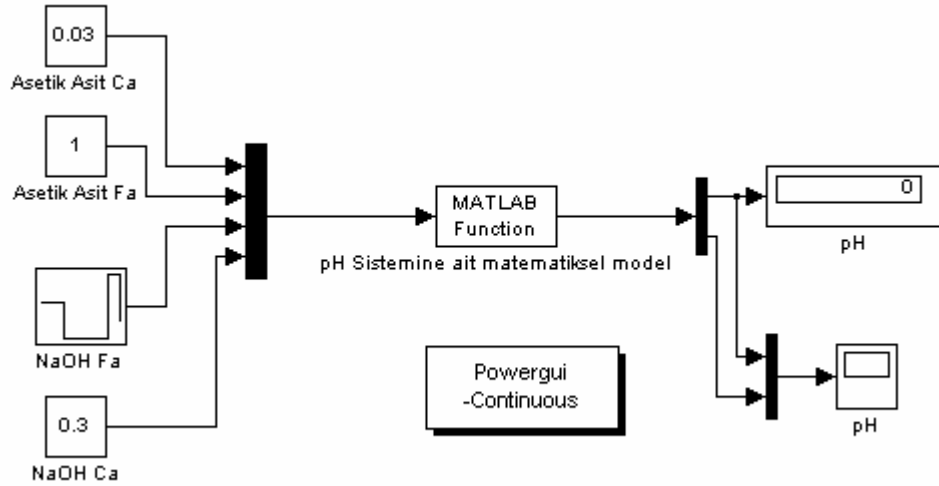
Şekil 6.7 Titrasyon eğrisinin kazançlara göre dilimlenmesi.

## 7. BİLGİSAYAR ORTAMINDA SİSTEMİN SİMÜLASYONU

Çalışmanın bu bölümünde amaçlanan, hem sistemin matematiksel modelinin doğruluğunu görebilmek hem de uygulamada tasarımı ve uygulaması gerçekleştirilecek olan sistemin performansına kriter teşkil edebilecek verileri oluşturmaktır. Zira sistemlerin performanslarının kendi başına değerlendirilmesi bilimsel olarak bir anlam ifade etmemektedir. Öte yandan, soft ortamda hesaplamaların yapılabilirliği daha kolay olup, eldeki hesaplama gücü ise 8 bitlik mikro elemanlar ile kıyaslanamayacak derecede güçlüdür. Bütün bunlar bu bölümünü oluşmasına temel teşkil eden etmenlerdir.

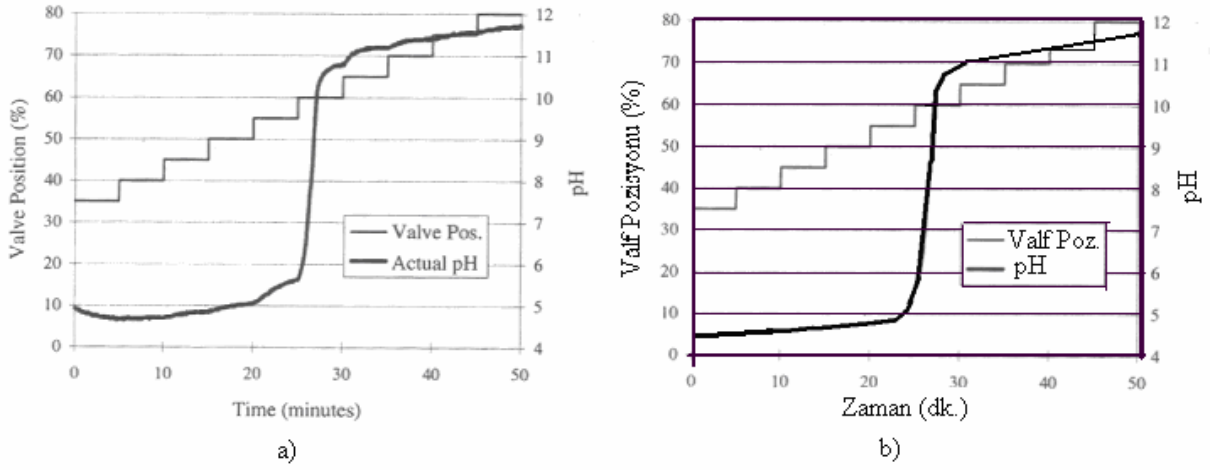
### 7.1 Sistemin Matematiksel Modelinin Doğruluğu

Sistemin matematiksel modelinin doğruluğunun incelenmesinde izlenebilecek en özet ve de yeterli olacak olan yol, simülasyon ortamında matematiksel modelin denenmesi olacaktır. Şekil 7.1 bu konuda yeterli olacak, aydınlatıcı fikrin elde edilmesine öncülük edebilir.



Şekil 7.1 Simülasyon ortamında modeline denemesine ilişkin yapı.

Yapılacak olan işlen sabit asetik asit akışına (1 l/dk) karşılık değişken NaOH akışları ile elde edilecek olan sonuçların gerçek bir proses ile karşılaştırılması olacaktır. Burada asetik asit derişimi (Ca) 0.03 g mol/l ve NaOH derişimi ise 0.3 g mol/l olarak alınmıştır (karşılaştırması yapılacak olan gerçek proses ile aynı değerlerdir). Simülasyon sonuçları ile gerçek prosese ait sonuçlar Şekil 7.2 de aynen verilmiştir.



Şekil 7.2 a) Gerçek bir pH nötralizasyon prosesinden elde edilen sonuçlar, b) Matematiksel modeli içeren simülasyon sisteminden elde edilen çıkış işaretinin grafiksel gösterimi.

Grafiklerden rahatlıkla görüldüğü üzere, modele ait çıkışlar ile gerçek prosese ait çıkışlar yaklaşık olarak aynı değerleri vermektedir. Ancak başlangıç aşamasında bir farklılık göstermektedir. Bu noktada ise, iki faktörün etkisi söz konusudur. İlk olarak gerçek sistemlerde sıcaklığın (hem ortamın sıcaklığı hem de nötralizasyon prosesinin ürettiği sıcaklık) asit-baz reaksiyonları üzerindeki etkisi tartışılmaz. Öte yandan, gerçek sistem ile simülasyonda ele alınan sistemin başlangıç değerlerinin birebir uyumunun 100% olamayacağı ortadır. Bütün bunlar grafikteki ufak farklılıkların temelini teşkil eder. Önemli olan ise, sistemin genel hareketlerinin (reaksiyon sürecindeki hareketler) uyumudur ki, bu da burada tatmin edici derecede yakalanmıştır. Bu sayede, simülasyon değerleri ile uygulama değerleri arasında bir kıyaslama yapabilmek (uygulama devresinin başarımı açısından) ve de gerçek bir prosese ihtiyaç duyulmadan çalışmanın gerçekleştirilmesine olanak sağlanmış olacaktır. Zira bu model mikrokontrolöre aktarılarak, mikrokontrolörün proses gibi davranması sağlanacaktır.

## 7.2 pH Nötralizasyon Prosesine PID Kontrolör Tasarımı ve Uygulanması

Bu bölümde ilk olarak prosesin matematiksel yapısına uygun PID (oran-integral-türev) kontrolörün tasarımı gerçekleştirilecektir. Daha sonra bir klasik kontrol tekniği olarak prosesin bu kontrolör üzerinden kontrolüne çalışılacaktır. Böylece daha sonra tasarımı ve uygulaması gerçekleştirilecek olan bulanık mantık-yapay sinir ağı tabanlı kontrolörün performansına bir kriter olarak ele alınabilecektir. Zira kıyaslama ve karşılaştırma işlemi yapılmadan bir sistemin başarımlarından bahsedilemez.

### 7.2.1 pH Nötralizasyon Prosesine PID Kontrolör Tasarımı

Proseste ve simulasyon işlemlerinde kullanılacak olan değerler 7.1 de verildiği gibidir.

$$\left\{ \begin{array}{l} Fa = 1l / dk \\ Fb = 0.1l / dk \\ Ca = 0.03g.mol/l \\ Cb = 0.3g.mol/l \\ V = 2l \end{array} \right\} \quad (7.1)$$

Burada ilk olarak göze çarpan iki ifadeden birincisi, Fa'nın Fb'ye göre oldukça büyük bir değer olduğudur ( $Fa \gg Fb$ ). Diğeri ise, Cb'nin Ca'ya göre oldukça büyük olduğudur ( $Cb \gg Ca$ ). Nu ikinci etmen NaOH akışındaki küçük değişikliklerin bile pH değerine büyük etkilerde bulunacağı anlamına gelecektir. İlk yaklaşım ise, oldukça lineer bir sistemin oluşmasına mahal vermektedir (Leondes, 1998). Bu durumda ;

$$G_m(s) = \frac{0.05}{s + 0.05} \quad (7.2)$$

olacak şekilde bir referans model bile ele alınabilir.

Yine bir PID kontrolörün matematiksel ifadesi,

$$K_p \left[ (\beta pH_{set} - pH) + T_d \frac{de}{dt} + \frac{1}{T_i} \int edt \right] \quad (7.3)$$

olacağından, kontrolör tasarımında  $K_p$ ,  $\beta$ ,  $T_d$  ve  $T_i$  katsayılarının belirlenmesi yeterli olacaktır. Katsayıların atanması için ise, en sık kullanılan yöntemlerden birisi olan Ziegler-Nichols yöntemi kullanılacaktır. İfade 7.3 de belirtilen değerlerden birisi de, e değeridir ki,

$$e = pH_{set} - pH \quad (7.4)$$

olarak netleştirilmelidir.

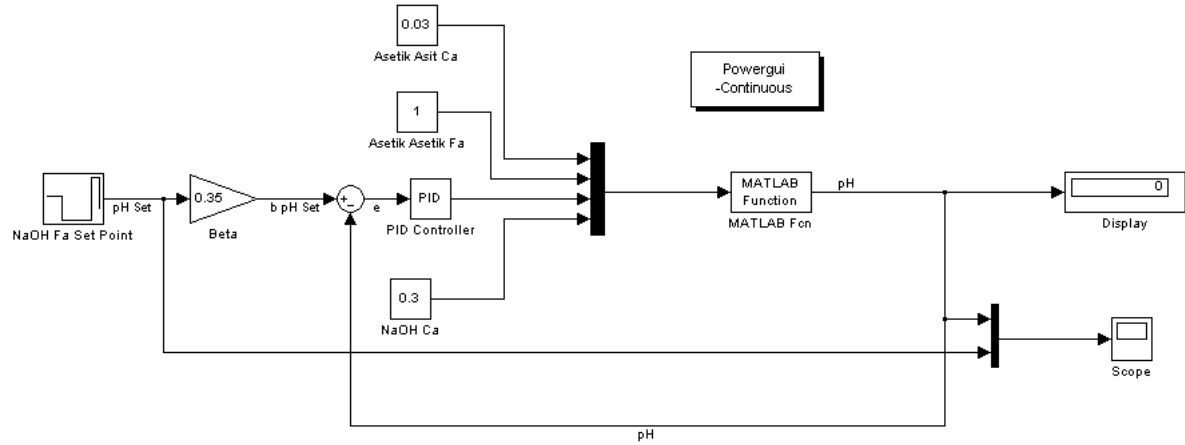
Tüm bu etmenler bir araya getirilirse Ziegler-Nichols kurallarından hareketle katsayılar aşağıdaki gibi elde edilecektir (Leondes, 1998).

$$K_p = 11.78, T_i = 15, T_d = 3.6, \beta = 0.35 \quad (7.5)$$

Normalde PID denkleminde yer almayan bir değer olan  $\beta$ , referans değer ağırlığıdır ve birim basamak cevabındaki üst aşımaları (overshoots) azaltmak amacıyla kullanılmıştır.

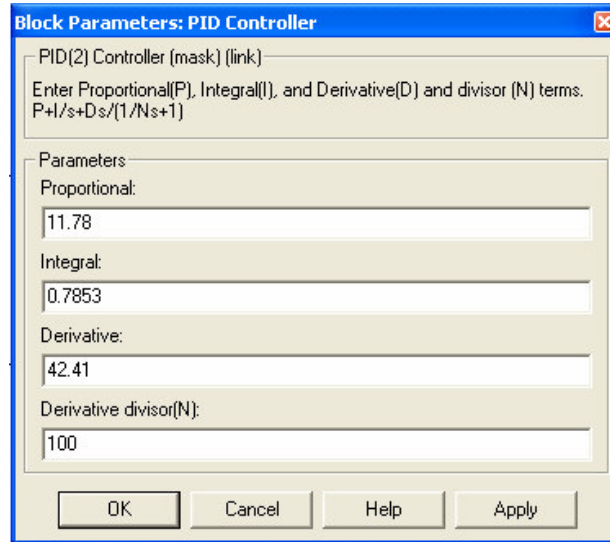
## 7.2.2 pH Nötralizasyon Prosesine PID Kontrolör Uygulanması

MATLAB ortamında gerçekleştirilecek olan simülasyon işlemi için kurulan yapı Şekil 7.3 te verilmiştir.



Şekil 7.3 PID kontrolörün uygulandığı simülasyon devresi.

Bu yapıda ayarlanması gereken asıl bölümü, PID kontrolör bloğu oluşturmaktadır. Zira bir önceki bölümde belirtilen değerlerin (kontrolör parametrelerinin) uygulamaya geçirilmesi gerekmektedir. Buna dair veriler ise, Şekil 7.4 te görülmektedir.



Şekil 7.4 PID kontrolöre ait parametrelerin sisteme girilmesi.

Şekil 7.3 te bütün giriş değerleri ve her bir yolun anlamı net olarak belirtilmiştir. "Matlab Function" bloğu, prosesin matematiksel modelini içermektedir. Tüm bu bileşenler bir araya getirildiğinde ve simülasyon işlemi 100 dakika için çalıştırıldığında Şekil 7.5 teki grafiksel değişim elde edilmiştir. Burada ilk 15 dakikası sistemin oturması için ayrılmıştır ve daha sonra her 5 dakikada bir referans değer değiştirilmiştir. Elde edilen sonuçlara ilişkin verilerin



sorgulama ya da açıklama gerekmemektedir.

- ✓ YSA ların genelleştirebilme özellikleri vardır. Bir örnekten hareketle, diğer örneklerdeki benzerlikleri doğru olarak anlayabilirler ve görmedikleri örnekler hakkında bilgiler üretebilirler.
- ✓ YSA lar non-lineerdirler. Matematiksel olarak non-lineer davranışlar hissedilir, algılanır, anlaşılabilir. Ancak bu davranışları ya da problemleri çözmek zordur. YSA ile karmaşık problemler lineer tekniklerden daha kolay çözülebilir.
- ✓ YSA lar paralel çalışabilirler. Bağımsız işlemleri aynı anda çok hızlı olarak yürütebilirler. Paralel donanımlara olan uygunlukları sebebiyle, YSA lar diğer yapay zeka metotlarından daha çok tercih edilirler.
- ✓ YSA lar hata toleransına sahiptirler. YSA ların eksik bilgilerle çalışabilme yetenekleri hatalara karşı toleranslı olmalarını sağlamaktadır. Ağın bazı hücrelerinin bozulması ve çalışamaz duruma düşmesi halinde ağ çalışmaya devam eder.

### 7.3.1.2 YSA ile Klasik Yazılımlar Arasındaki Farklar

YSA, önceden tahmin edebilme, örnekleri değerlendirme ve gruplama işlemlerinde etkili bir yöntemdir. Aynı işlemleri klasik bir bilgisayar programı ile de yapmak da mümkündür. Ancak klasik programlar belirli bir görev için yazıldıklarından, hep aynı işi yaparlar ve yeni bir durum için veri elde edilmesi zordur. YSA lar ise, kuralları açık bir şekilde belli olmayan veya anında bazı sınırlamalar getirilen uygulamalar için idealdir. Özellikle endüstriyel kontrol uygulamalarında çok sık kullanılır. Çünkü bu uygulamalarda kurallar çok sık değişmez ve başka çalışma koşullarına ait veriler kolaylıkla elde edilebilir.

YSA ların uygulamadaki dezavantajları:

- ✓ Bir problemin çözümünde çok uygun bir çözüm bulamayabilirler ve çözümde hata yapabilirler. Bunun sebebi ise, ya eğitilecek bir fonksiyonun bulunamaması ya da fonksiyon bulunsa bile yeterli veri sağlanamamasıdır.
- ✓ Probleme uygun ağ yapısının belirlenmesinin genellikle deneme yanılma yolu ile yapılması önemli bir dezavantajdır. Çünkü eğer uygun bir ağ yapısı oluşturulamazsa çözümü olan bir problem çözülememiş veya performansı düşük bir çözüm elde edilmiş olur.
- ✓ YSA kabul edilebilir çözümler üretebilir ama optimum (en iyi) çözümü garanti edemez.

- ✓ Sonuç almak, yüzlerce giriş örneğinin hesaplanmasına bağlı olabilir. Ayrıca hangi ağırlığın sonucu nasıl etkileyeceğini tahmin etmek zordur.
- ✓ YSA larla bir dizi işlem yapmak, bunları eğitmek yavaş ve pahalı olabilir. Çünkü eğitme verilerinin toplanması, değerlendirilmesi ve doğru değerleri bulmak için deneylerin yapılması gerekebilir.
- ✓ YSA nın kalitesi ve kapasitesi uygulamadaki hızı ile orantılıdır. Nöron sayısındaki küçük bir artış, eğitmenin çok daha fazla bir zamanda yapılmasına sebep olabilir ve böylece ağıın çıkışından alınan cevap süresi uzayabilir.
- ✓ YSA larda kurulan ağıın davranışları açıklanamaz. Bir probleme çözüm üretildiği zaman bunun nasıl ve neden üretildiği konusunda bilgi bulmak mümkün değildir. Bu ise, ağıın sonucuna olan güveni azaltmaktadır.

Görüldüğü gibi, YSA kullanımı her zaman bir avantaj değil, duruma göre bir dezavantaja da dönüşebilir. Bu noktada hesaplamaların doğru yapılması ve probleme uygun ağ yapısının seçimi gibi parametrelerin etkisi ön plana çıkmaktadır.

### 7.3.1.3 MLP ve Fuzzy Kullanım Nedenleri?

Aslında burada derin teorik yaklaşımlar aramanın fazla bir anlamı yoktur. Sistem dizaynında en geçerli parametreler en etkilisi, bildiğiniz sistemler içerisinde en etkili olanı kullanılmaktadır. Ticarete de denildiği gibi, “*en iyi iş bildiğiniz iştir*”. Yukarıda YSA kullanım nedenleri belirtilmiştir. Bunlara ek olarak MLP seçim nedenlerin de şu şekilde sıralanan etkenlerde etkili olmuştur.

- ✓ Bir yapay sinir ağıının öğrenmesi istenen olayların girdileri ve çıktıları arasındaki ilişkiler doğrusal olmayan ilişkiler ise, gelişmiş YSA modellerine ihtiyaç duyulur (Elmas, 2003). MLP yapısı da bu gelişmiş ve non-linear sistemleri öğrenme konusunda yetenekli olan YSA modellerinden birisidir.
- ✓ Uygulama kolaylığı ve hesaplama olanakları açısından uygun YSA modellerindedir.
- ✓ Ayrıca MLP kontrol sistemleri geliştirme adına uygun YSA yapılarının başında gelmektedir.
- ✓ Gerek eğitimi ve gerekse de tasarım kolaylığı da bu etkenler arasındadır.

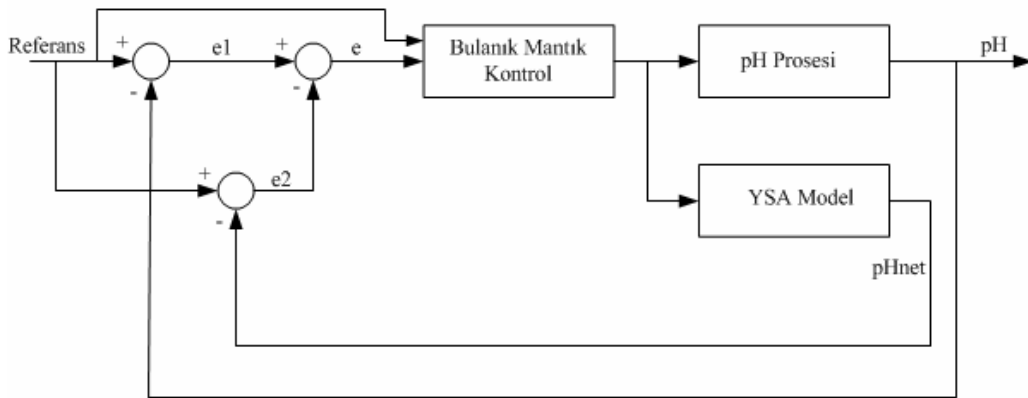
Bulanık mantık kullanımında etkili olan ve daha önceki belirtilen bulanık mantık kontrol

sistemlerinin üstün yanlarına ilave olan temel etmen ise, şu şekilde açıklanabilir. Eğer dikkat edilirse, nötr noktası civarında baz akışındaki küçük değişimler bile pH değerinde büyük değişimler neden olabilmekte sistemi denge noktasından uzaklaştırmaktadır. İşte bu adı geçen büyük değerli baz akışlarını ve nötr noktası civarında sisteme keskin ve/veya yüksek değerli kontrolör etkilerini engellemek adına bulanık mantık kontrolör devreye alınacaktır. Daha öncede belirtildiği gibi bulanık mantık kontrol sistemi burada, YSA modelinin çıkış sinyallerini yumuşatmada kullanılacaktır. Yani sisteme aşırı tepki şeklinde geri dönüşe sebep olacak olan, kontrol sinyallerinin engellenmesi amaçlanmaktadır. Zira bu, bulanık mantık kontrol sistemlerinin yapısından ileri gelen durumdur.

Bu durum şu şekilde bir dezavantaja sebebiyet verebilir. Nötr noktasından uzak değerlerde sisteme zayıf etkilere sebebiyet vererek, nötr noktasına (referans değere) ulaşmayı geciktirebilir. Ancak bu durum, bir önceki durumda sağlanan faydanın yanında göz ardı edilebilecek kadar küçük olup, bu belirtilen noktalardaki zayıflatıcı etkileri ise, beklendiği gibi yüzde olarak büyük değerlere ulaşmamaktadır. Çalışmanın devamında simulasyon ve uygulama sonuçları elde edildikçe, bu tercihin ne denli doğru olup olmadığı daha net olarak ortaya çıkacaktır.

### 7.3.2 pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Tasarımı

Görüldüğü gibi kontrolör iki farklı kısımdan oluşmaktadır; YSA tabanlı proses modeli ve bulanık mantık tabanlı kontrolör. Tasarımı gerçekleştirilecek olan yapının tamamı Şekil 7.6 da görülebilir.



Şekil 7.6 YSA-Bulanık mantık tabanlı kontrol sisteminin tamamı.

Şekilden de görüldüğü gibi, yapılmak istenen, YSA çıkışı ile proses çıkışının ele alınarak, bu iki değerlerin farkları aracılığı ile sistemi kontrol etmektir. Aşağıda bu yapının formülasyonu

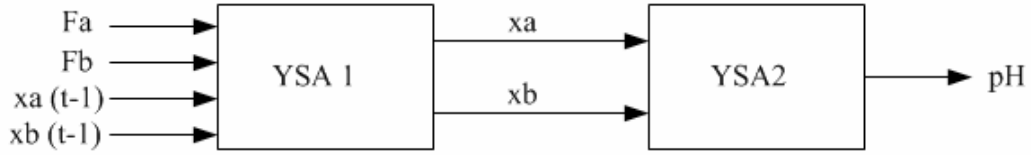
verilmiştir.

$$\begin{aligned}
 e &= e_1 - e_2, \\
 e_1 &= \text{Re } f_{\text{erans}} - pH, \\
 e_2 &= \text{Re } f_{\text{erans}} - pH_{\text{net}}, \\
 e &= pH_{\text{net}} - pH
 \end{aligned}
 \tag{7.6}$$

Burada temel olarak, YSA'nın karmaşık sistemleri tanıma özelliğinden yararlanılacaktır. Şimdi sırası ile her bir yapının nasıl geliştirildiğini görelim.

### 7.3.2.1 YSA Tabanlı Modelin Tasarımı

İlk olarak yapılması gereken tek bir yapı yerine sistemin, dinamik ve statik bölümlerini ayrı ayrı iki ağ yapısı ile ele almaktır. Şekil 7.7 bu yapıların ayrımını ele almaktadır.



Şekil 7.7 İki bölümden oluşan YSA yapısı.

YSA1 6.17 ve 6.18 formüllerinde yer alan dinamik değerleri kullanarak  $x_a$  ve  $x_b$  değerlerini elde edecektir. Bu bölüm YSA'nın dinamik bölümüdür. YSA2 ise, 6.23 eşitliğini kullanarak (modelleyerek) sistemin çıkış değeri olan PH değerini elde edecektir. Burada her bir bölümü ayrı ayrı ele alıp inceleyelim.

✓ YSA1;

YSA larda temel sorunlardan ilkinin giriş-çıkış parametreleri oluşturmaktır. YSA1 in modellemeye amaçladığı eşitlikler birinci dereceden ve her biri tek değişkenli olduğundan dolayı, burada her bir değişken için geçmiş son değerlerin sisteme giriş olarak alınması yeterli olacaktır.

Diğer önemli olan nokta ise, ağın yapısıdır. Burada Leondes in ağ yapısı esas alınarak, 4/10/2 olacak şekilde bir ağ yapısı amaçlanmıştır (k/m/n; her bir katmandaki hücre sayısıdır). YSA2, 4 hücreli giriş, 10 hücreli tek saklı katman ve 2 hücreli çıkış katmanı olmak üzere 2 katmanlı 16 hücreli bir yapıdır.

Aktivasyon fonksiyonu olarak, rahatlıkla sigmoid fonksiyonu kullanılabilir.

Ağ yapısı ise, MLP olup eğitim ve onaylama işlemleri bölüm 3.4 te ele alındığı gibi gerçekleştirilmektedir.

✓ YSA2;

YSA1 in yapısında da MLP ve bu yapıya ait ilgili yaklaşımlar kullanılmıştır (eğitim ve onaylama için).

Eşitlik 6.23 diferansiyel ifadeler içermediğinden dolayı, giriş parametreleri olarak sadece değişkenlerin kendilerinin ele alınmaları yeterli olacaktır.

Burada aktivasyon fonksiyonu olarak, sisteme uygun bir fonksiyon ele alınabilir. pH prosesinin grafiksel değişimleri tanjant hiperbolik fonksiyonu ile kayda değer bir benzerlik göstermektedir. Buradan hareketle aktivasyon fonksiyonu,

$$f(s) = 0.41 \tanh(0.1s) + 0.59 \tanh(2s) \quad (7.7)$$

olarak elde edilebilir (Leondes, 1998).

Ağın yapısı, 2/20/1 olarak alınabilir (Leondes,1998). Ağ 2 katmanlı olup, 23 hücreden oluşmaktadır.

Görüldüğü gibi YSA yapısı ile ilgili olarak bütün değerler açıklığa kavuşturulmuştur.

### 7.3.2.2 Bulanık Tabanlı Kontrolörün Tasarımı

Bir bulanık mantık kontrolörün tasarımı aşağıda liste halinde verilen her bir bölümün ayrı ayrı ele alınması esasına dayanır.

- ✓ Giriş-çıkış parametrelerinin atanması,
- ✓ Bulandırma aşaması,
- ✓ Giriş-çıkış parametreleri için üyelik fonksiyonlarının belirlenmesi,
- ✓ Bulanık Çıkarım,
- ✓ Durulama,

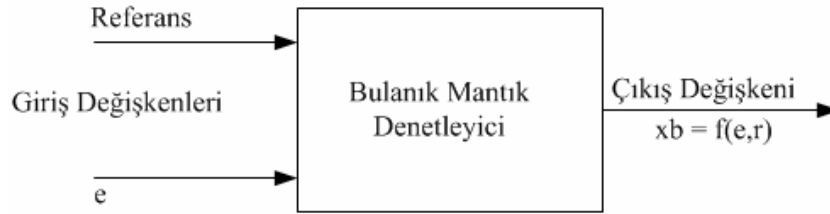
Biz de bu adımlara uyarak sistemimize uygun olan kontrolörü geliştirelim.

- ✓ Giriş-Çıkış parametrelerinin atanması,

Giriş parametrelerinin belirlenmesinde etkili olan düşünce, kontrolörün amacının sadece

$e = \text{pH}_{\text{net}} - \text{pH}$  değerini sıfırlamak olmayıp, aynı zamanda referans değeri de yakalamak olduğu durumudur. Bu nedenle bulanık mantık kontrolöre giriş parametreleri olarak;  $e$  ve referans değerleri alınmıştır.

Çıkış için ise, bizim amacımız girdi değerlerinden hareketle  $F_b$  (baz akış değeri) değerini elde etmek olduğuna göre çıkış parametresi olarak,  $F_b$  alınabilir.



Şekil 7.8 Bulanık mantık denetleyici ve giriş-çıkış parametreleri.

✓ Bulandırma aşaması;

Bu aşamada, sayısal giriş ve çıkış değişkenleri sembolik değerlere dönüştürülür. Bunun için ilk olarak giriş değişkenlerinden referansa ait bulanık küme tanımlanmıştır (Tablo 7.1). Tablo değerlerinin atanmasında pH'ın 0-14 aralığında değişmesi hali göz önüne alınmıştır.

Tablo 7.1 Referans giriş değişkenine ait bulanık küme tanımlaması.

NB	Negatif Büyük	0-3.0
NO	Negatif Orta	2.5-6.0
NK	Negatif Küçük	3.0-7.0
S	Sıfır	6.0-8.0
PK	Pozitif Küçük	7.0-10.0
PO	Pozitif Orta	8.0-12.0
PB	Pozitif Büyük	10.0-14.0

Şimdi de diğer bir giriş değişkeni olan  $e$  işareti için bulanık küme tanımlamalarını verelim (Tablo 7.2). Burada hatanın (-14)-14 aralığında değiştiği durumu göz önüne alınmıştır. Çünkü pH maksimum 14.0 değerini alabilir.

Tablo 7.2 “e” giriř iřareti için bulanık küme tanımlaması.

NB	Negatif Büyük	(-14.0)-(-6.0)
NO	Negatif Orta	(-7.0)-(-1.5)
NK	Negatif Küçük	(-6.0)-(-0.5)
S	Sıfır	(-1.5)-1.5
PK	Pozitif Küçük	(-0.5)-4.0
PO	Pozitif Orta	1.5-7.0
PB	Pozitif Büyük	4.0-14.0

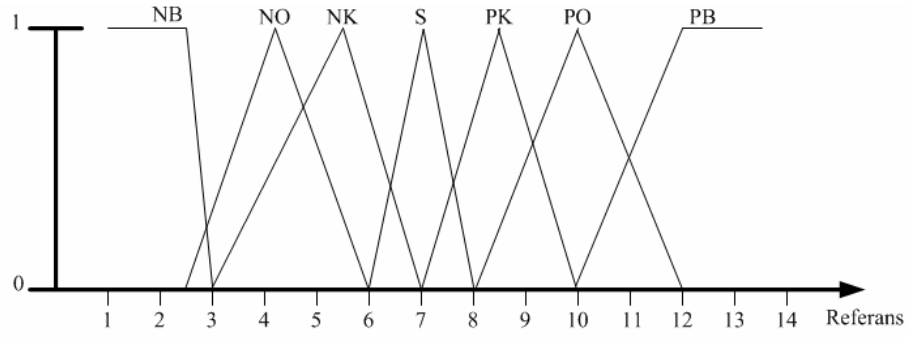
Çıkış deęiřkeni için ise ( $F_b$ ), Şekil 6.7 de yer alan grafik ve bu grafięin ayrıştırıldığı lineerleştirilmiş bölgeler bulanık küme atamalarında kullanılabilir (Tablo 7.3). Bu şekilde Baz/Asit oranının en fazla 2 olabileceęi belirtilmiştir. Buradan hareketle,  $F_b/F_a = 2$  olabilir ve  $F_a = 1$  l/dk olduęuna göre,  $F_{bmax} = 2$  l/dk olabilir.

Tablo 7.3 Çıkış deęiřkeni olan,  $F_b$  için bulanık küme atamaları.

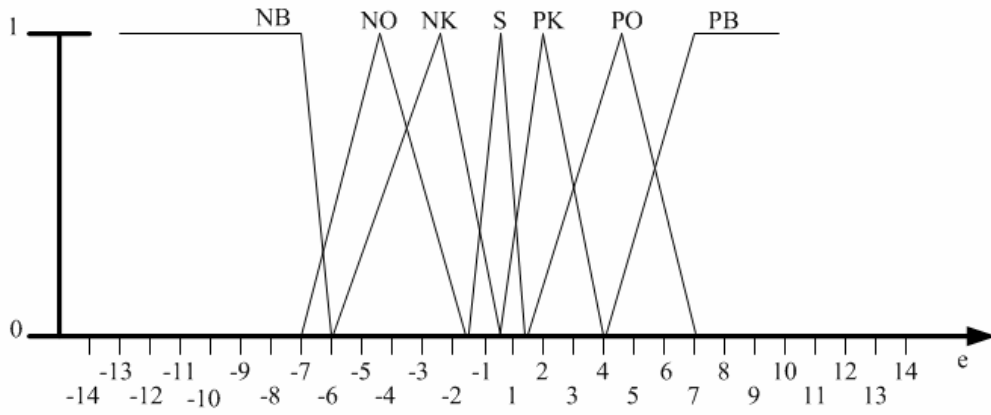
NB	Negatif Büyük	0-0.4
NO	Negatif Orta	0.3-0.9
NK	Negatif Küçük	0.4-1.0
S	Sıfır	0.9-1.1
PK	Pozitif Küçük	1.0-1.4
PO	Pozitif Orta	1.1-1.7
PB	Pozitif Büyük	1.4-2.0

✓ Giriř ve Çıkış için üyelik fonksiyonlarının atanması;

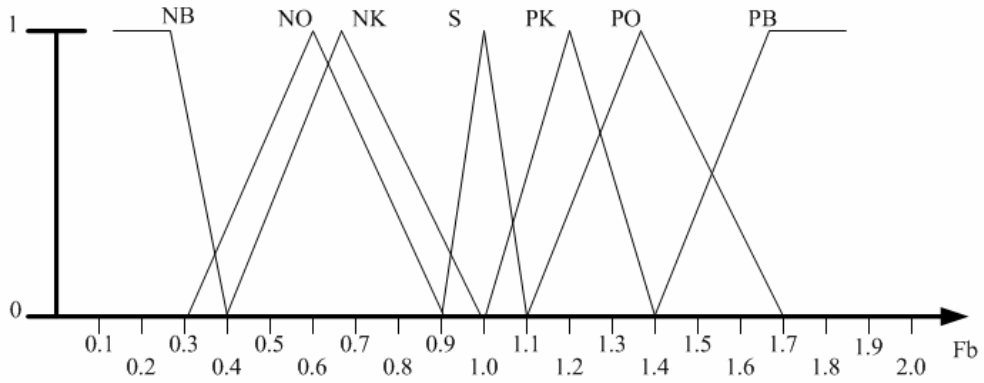
Burada üçgen üyelik fonksiyonları kullanılacaktır. Buradan hareketle; sırası ile referans, e ve  $F_b$  için üyelik fonksiyonlarının grafiksel gösterimlerini Şekil 7.9-10-11 de bulabiliriz.



Şekil 7.9 Referans giriş değişkeni için üyelik fonksiyonları.



Şekil 7.10 "e" giriş değişkeni için üyelik fonksiyonları.



Şekil 7.11  $F_b$  çıkış değişkeni için üyelik fonksiyonları.

Üyelik fonksiyonları tamamlandığına göre, bulanık çıkarım için düzenleme yapılabilir.

✓ Bulanık Çıkarım;

Bulanık kurallar üzerine bulanık mantık uygulanarak elde edilen ifadelere bulanık çıkarım denir. Şimdi bu iki giriş değişkeni ve bunlara ait üyelik fonksiyonları doğrultusunda oluşan bulanık kurallar dizisi Çizelge 7.1 de verilmiştir.

Çizelge 7.1 Kontrolöre ait bulanık kural çizelgesi.

Referans <i>e</i>	NB	NO	NK	S	PK	PO	PB
NB	NB	NB	NO	NK	PO	PB	PB
NO	NB	NO	NK	NK	PK	PO	PB
NK	NO	NK	NK	NK	PK	PK	PB
S	NB	NO	NK	S	PK	PO	PB
PK	NB	NO	NK	PK	PK	PK	PO
PO	NO	NO	NK	PK	PK	PO	PB
PB	NO	NK	NO	PK	PO	PB	PB

✓ Durulama;

Bu noktaya gelindiğinde elimizdeki çıktı verisi hala bulanık bir veridir ve gerçek sistemimize uygulanabilirliği yoktur. Bu sebeple bu verinin gerçek ve kesin değerlere dönüştürülmesi gerekmektedir. Durulama işleminde, her kural için üyelik ağırlık değerleri bulunarak, giriş değerlerinin en az üyelik ağırlığı ve buna göre çıkış üyelik değerleri tespit edilir. Tasarlanan bulanık mantık denetleyicide durulama işlemi için genelde en çok kullanılan yöntemlerden ağırlık ağırlık merkezi yöntemi kullanılmış ve durulanmış çıkış işareti;

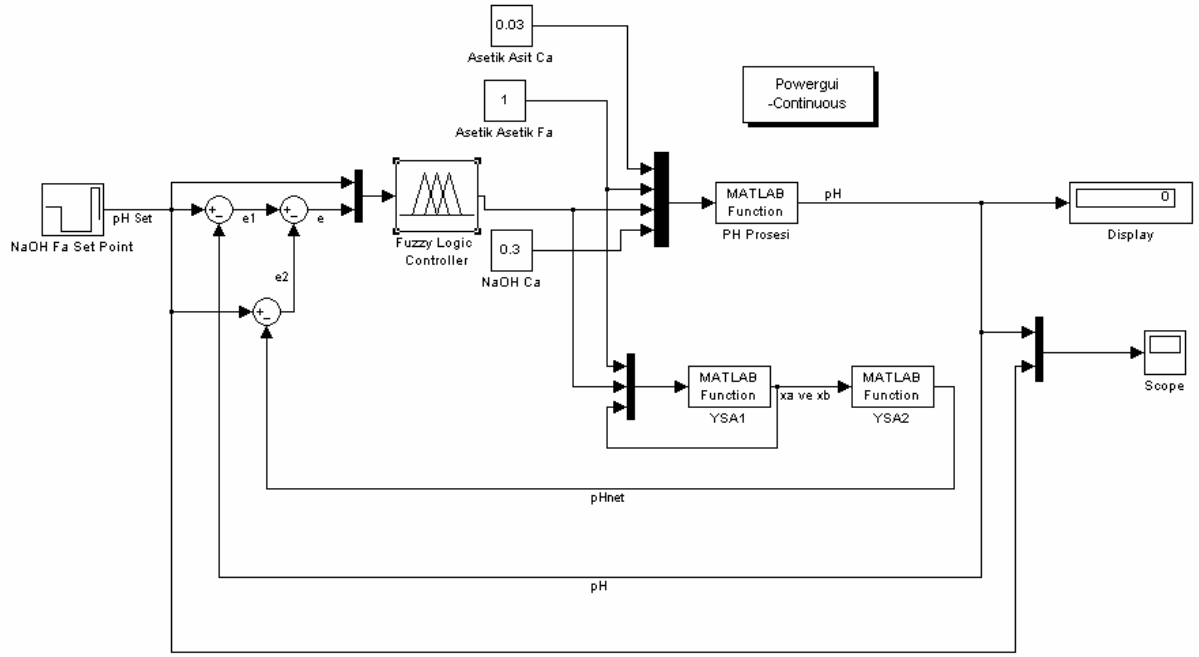
$$U = \frac{\sum_{i=1}^n u_i \cdot \mu(u_i)}{\sum_{i=1}^n \mu(u_i)} \quad (7.8)$$

olarak elde edilir.

### 7.3.3 pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Uygulanması

Şimdi tasarımı yapılan kontrolör sisteminin sisteme uygulanmasını ve bunun sonucunda elde edilen çıkış işaretlerini inceleyelim.

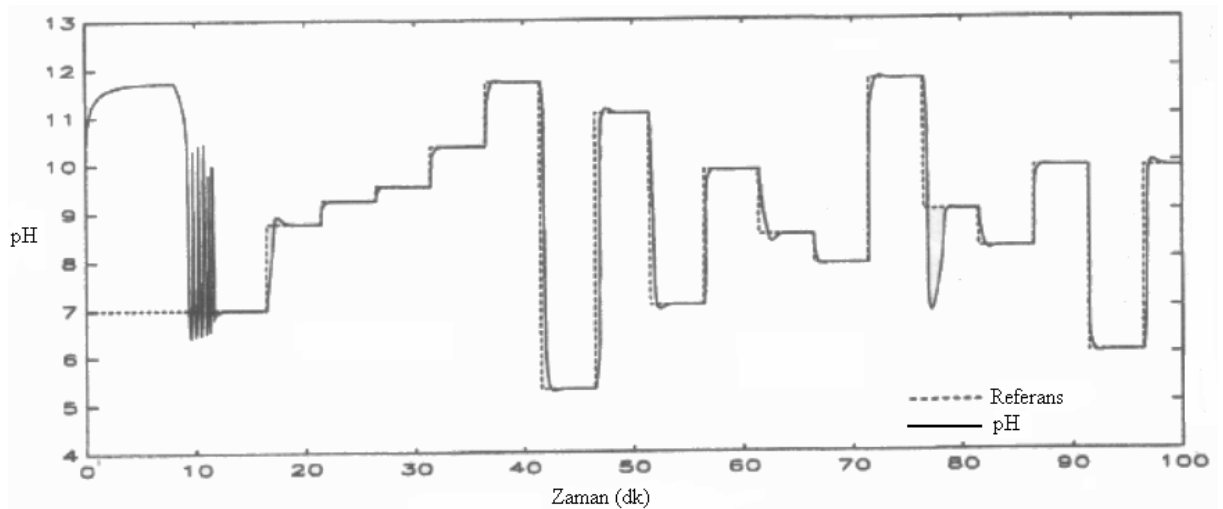
İlk olarak Şekil 7.12 de, uygulamanın gerçekleştirildiği simülasyon devresinin görebiliriz.



Şekil 7.12 YSA ve Bulanık mantık tabanlı kontrol sisteminin prosese uygulandığı simulasyon yapısı.

Burada verilen yapı, Şekil 7.6 verilen yapının Matlab ortamına aktarılmış halidir.

Tasarımı gerçekleştirilen kontrolörün bu simulasyonda uygulanmaya alınması sonucunda Şekil 7.13 te verilen grafiksel gösterim elde edilmiştir. Geliştirilen kontrolörün başarımı ile ilgili yorumlar ve sonuçlar son bölümde ayrıntılı olarak ele alınacaktır.



Şekil 7.13 Geliştirilen YSA-Bulanık mantık tabanlı kontrolörün prosese uygulanması sonucunda elde edilen sonucun grafiğe aktarılmış hali.

## 8. YAPILAN UYGULAMA

Aslında çalışmanın en can alıcı bölümünü bu kısım oluşturmaktadır. Zira sıradan sekiz (8) bitlik mikrokontrolörlere, YSA ve bulanık mantık gibi yapay zeka ürünü tekniklerin aktarımı hem zor hem de pek fazla denenmemiş çalışmalardır. Bu açıdan çalışmanın bu bölümü tam olarak sayılmasa da, yine de bir yenilik olarak düşünülebilir.

Burada yapılacak olan işlemler özetlenecek olursa;

- ✓ İlk olarak kullanılacak olan mikrokontrolör hakkında kısa da olsa, özet bilgi verilecektir.
- ✓ YSA1 e ait model mikrokontrolöre aktarılacaktır.
- ✓ YSA2 ye ait model mikrokontrolöre aktarılacaktır.
- ✓ Bulanık mantık kontrolöre ait model mikrokontrolöre aktarılacaktır.
- ✓ pH nötralizasyon prosesine ait model mikrokontrolöre aktarılacaktır.
- ✓ Bütün bileşenlerin bir araya getirilmesi bir bütün olarak sistemin elde edilmesi ve uygulamanın gerçekleştirilmesi.

Görüldüğü çalışma planlı olarak adım adım gerçekleştirilecek olan, bir parçalar bütünüdür. Aşağıda bu adımların teker teker alındığı detaylı bölümlere yer verilmiştir.

### 8.1 Mikrokontrolör ile pH Nötralizasyon Prosesine Nöro-Fuzzy Kontrolör Tasarımı ve Uygulanması

İlk olarak yapılması gereken, gerekli yazılımların (YSA1, YSA2, bulanık mantık kontrolör ve prosese ait matematiksel model) mikrokontrolöre aktarılmasıdır. Ancak bu adımlara geçmeden önce yapılması gereken asıl ve önemli adım, sistem için doğru mikrokontrolörün seçilmesi işlemidir. İşte ilk olarak buna değinilecek ve neden GP32 nin seçildiği izah edilmeye çalışılacaktır.

Bunu takiben ise, adı geçen her bir modele ait gerekli işlemler ve gerekli açıklamalar ele alınacaktır.

Göz ardı edilmemesi gereken en önemli nokta olarak, kullanılan mikrokontrolörün sıradan bir mikrokontrolör olduğu ve hem matematiksel işlemleri gerçekleştirirmede bilgisayara göre yetersizliği hem de kısıtlı (dar-8 bitlik) bir hafıza yapısına sahip olduğu gerçekleridir.

### 8.1.1 Kullanılan Mikrokontrolörün Teknik Özellikleri

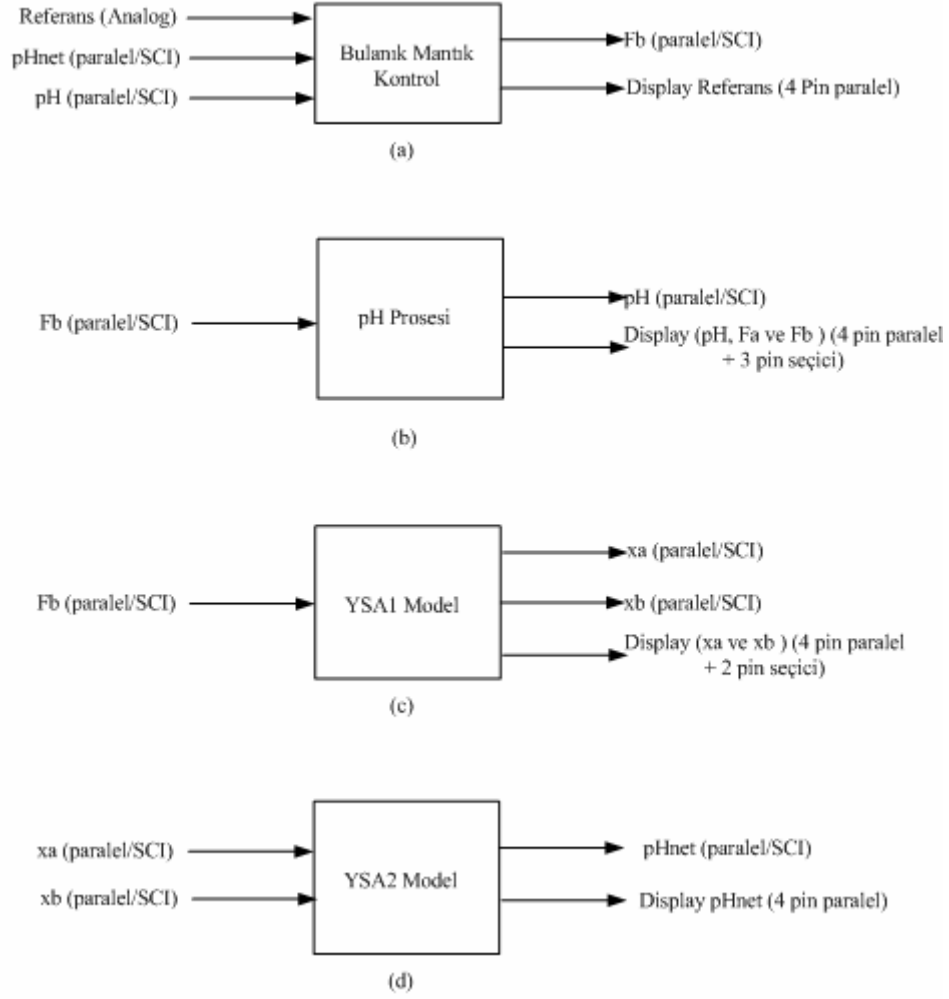
Aslında bu noktada olayı tek bir yönden ele almak yerine, hem eleman seçimine etkileri hem de seçilen elemanın temel özelliklerini ele almak daha doğru bir yaklaşım olacaktır. Şimdi sırası ile bu adımları teke tek ele alalım.

a. Mikrokontrolör seçimine etkide bulunan temel parametreler:

Kullanılan mikrokontrolörün bu çalışmada kullanılmasında sadece bu işlemcinin teknik özellikleri değil, hem teknik hem de diğer yan etmenler etkili olmuştur. Aşağıda bu etmenlerin genel bir listesi bulunabilir.

- ✓ Kullanılan mikrokontrolörün giriş-çıkış port sayısı,
- ✓ Analog giriş-çıkış portlarının sayısı ve dönüştürme hızı,
- ✓ Mikrokontrolörün çalışma saat hızı,
- ✓ Adresleme ihtiyaçlarına cevap verebilmesi için yeterli EEPROM a sahip olma durumu,
- ✓ İşlemlerde geçici bellek ihtiyacına cevap verebilmesi için yeterli RAM a (geçici bellek) sahip olma durumu,
- ✓ İstenilen matematiksel işlemleri gerçekleştirebilecek komut desteğine sahip olması,
- ✓ Bu mikrokontrolörün piyasadan temininin mümkün olup olmadığı durumu,
- ✓ Devreyi geliştirecek olan dizayncının ve programcının ilgili mikro eleman üzerindeki uzmanlık derecesi... gibi sıralanabilir.

Bunlar içerisinde bizim için en önemlilerinden ilki giriş çıkış portlarının sayısı ve bu portların özellikleridir. Aşağıda bu projede kullanılacak olan giriş-çıkış portları ile bunların durumları belirtilmiştir.



Şekil 8.1 Proje de kullanılacak olan giriş-çıkış portları ve portların durumları.

Eğer seçilecek olan işlemcinin SCI (Serial Comunication Interface) (Seri Haberleşme Arayüzü) özelliği yok ise, 23 adete varan giriş-çıkış portu ihtiyacı ortaya çıkacaktır. Buda haberleşmede sorunlar anlamına gelmektedir. Bu sebeple seri haberleşme arayüzü ve analog giriş destekleme özelliği gibi harici özellikler dışında genel amaçlı giriş-çıkış protlarına sahip olan bir işlemci bu çalışma için en azından bu şarta göre yeterlidir.

YSA yapılarına dikkat edilecek olursa, YSA2 iki katmanlı ve 21 hücreden oluşan bir yapıdır. Buda çalışma sırasında çok fazla miktarda geçici bellek (RAM – Random Access Memory) ihtiyacı ile birlikte en azında 4K (4 Kbytes) dahili bellek ihtiyacı anlamını taşımaktadır. Bu sebeple de, 512 bytes üstünde Ram a sahip ve 4K üstü değerde EEPROM a sahip bir işlemci bu şart açısından yeterli olacaktır.

Yine hem YSA hem de bulanık mantık kontrolör için çok sayıda çarpma ve bölme işlemi yapmak gerekecektir. Keza prosesin matematiksel modeli için de aynı bölme ve çarpma

işlemleri gerekli olacaktır. bu nedenlerle de çarpma ve bölme yapabilen (komut setinde bu komutları içeren) işlemciler bizim gerekli olacaktır.

Öte yandan geçmiş tecrübelerim (Freescale ve Microchip Firmalarının işlemcileri ile geliştirmiş olduğum projeler) doğrultusunda, Freescale firmasının 68HC908GP32 mikrokontrolörü ile bu projenin gerçekleştirilmesi yerinde bir karar olacaktır. zira bu işlemci ile ilgili hem teorik hem de pratik altyapıya sahip olmam bu kararı almaktaki en büyük etkenlerdendir.

**b.** Seçilen mikro elemana ait temel parametreler:

İşlemcinin bütün özelliklerinin burada verilmesi sadece zaman kaybı olacaktır. ancak işlemciye ait bütün veriler

[http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=68HC908GP32&nodeId=016246844976638055](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=68HC908GP32&nodeId=016246844976638055) internet adresinden edinilebilir. Yine işlemcinin komut seti (instruction set) verileri de bu adresten edinilebilir.

Şimdi sırası ile önce bu işlemcinin standart özelliklerini ve daha bu işlemcinin kullandığı CPU08 çekirdeğinin özelliklerini ele alalım.

✓ MC68HC908GP32 Mikrokontrolörünün genel özellikleri;

Bu özellikler şu şekilde sıralanabilir.

- C derleyiciler için geliştirilmiş yüksek performanslı MC68HC908 çekirdeği.
- M6805, M146805 ve M68HC05 ile uyumlu kod yapısı.
- 8 MHZ lik çalışma frekansı.
- FLASH program hafızası koruma.
- Yüksek değerli EEPROM programlama gerilimine ihtiyaç duymadan programlanabilme.
- Sistem içi programlama.
- Sistem koruma özellikleri;
  - Bilgisayar tarafından düzenlene opsiyonel Reset (COP),
  - Alçak gerilim gerilimi tanıma ve resetleme,
  - İlegal kod tanıma ve resetleme,

- İllegal adres tanıma ve resetleme.
- Düşük güç dizaynına olanak sağlama (bekleme ve duraklama).
- Standart durma ve bekleme özellikleri.
- Master reset ve power-on-reset (POR).
- 32 K Flash bellek (dahili programlanabilme özelliği ile).
- 512 bytes geçici bellek (RAM).
- Seri çevresel arayüz (SPI).
- Seri haberleşme arayüzü (SCI).
- 2 adet 16 bit, 2 kanallı Zamanlayıcı (Timer), giriş-kontrol ve çıkış-karşılaştırma özellikleri ve her bir kanal için PWM (Dalga Genişlik Modülasyonu).
- 8 kanallı ve 8 bit (ADC).
- Debug işlemi için tek nokta kesmelerine olanak sağlayan, BREAK modülü.
- Sistem maliyetini azaltmak için dahili Pullup lar, IRQ ve RESET.
- Saat işareti üretme modülü (PLL).
- 33 adet genel amaçlı giriş-çıkış portu.
- PORTA-C-D için seçilebilir Pullup lar.
- 10 mA lik giriş veya çıkış akımı.
- PTC0-4 için yüksek değerli 15 mA lik giriş-çıkış akımı.
- Düşük değerli (frekans olarak) işaret üretme amaçlı, Timebase modülü.
- Osilatör durdurabilme ve çalıştırabilme özellikleri.
- 8 bit lik tuş takımı girişi.
- Tuş takımı için 5 mA lik giriş koruma akım sınırı.
- 40, 42 veya 44 pinlik paket yapıları (opsiyonellik için).
- 40 pinlik paket yapısına (*bu projede kullanılacak olan yapı*) özgü özellikler;

- Port C sadece 5 bit dir (PTC0-PTC4),
- Port D 6 bit dir (PTD0-PTD5).

✓ CPU08 Çekirdeğinin genel özellikleri;

Aşağıdaki sıralama yeterli olacaktır.

- HC05 ile benzer programlama modeli.
- Geniş çevrim kontrol fonksiyonu.
- 16 farklı yapıda adresleyebilme özelliği.
- 16 bit lik index hafızası ve yığın noktalama.
- Hafızadan hafızaya veri transferi.
- Hızlı 8 bitlik iki veriyi çarpabilme özelliği.
- Hızlı 16 bitlik veriyi 8 bitlik veriye bölme özelliği.
- İkili kodlanmış onluk sistemde sayı özelliği (BCD).
- Kontrol uygulamaları için optimizasyon.
- Etkili C programlama dili desteği.

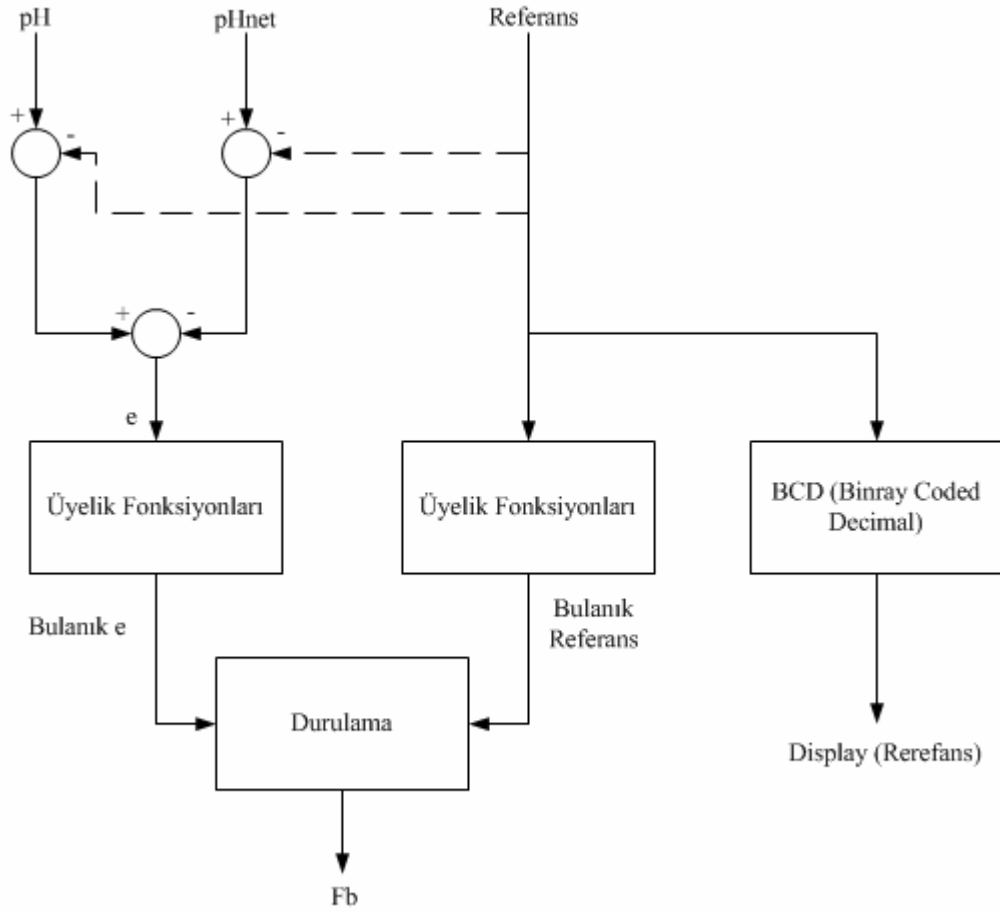
### 8.1.2 Mikrokontrolöre Nöro-Fuzzy Kontrolörün Uyarlanması

Aslında bu bölüm üç ayrı konu olarak ele alınabilir. Bunlar;

- ✓ Bulanık mantık kontrolörün GP32 ye aktarımı.
- ✓ YSA1 modelinin GP32 ye aktarımı.
- ✓ YSA2 modelinin GP32 ye aktarımı.

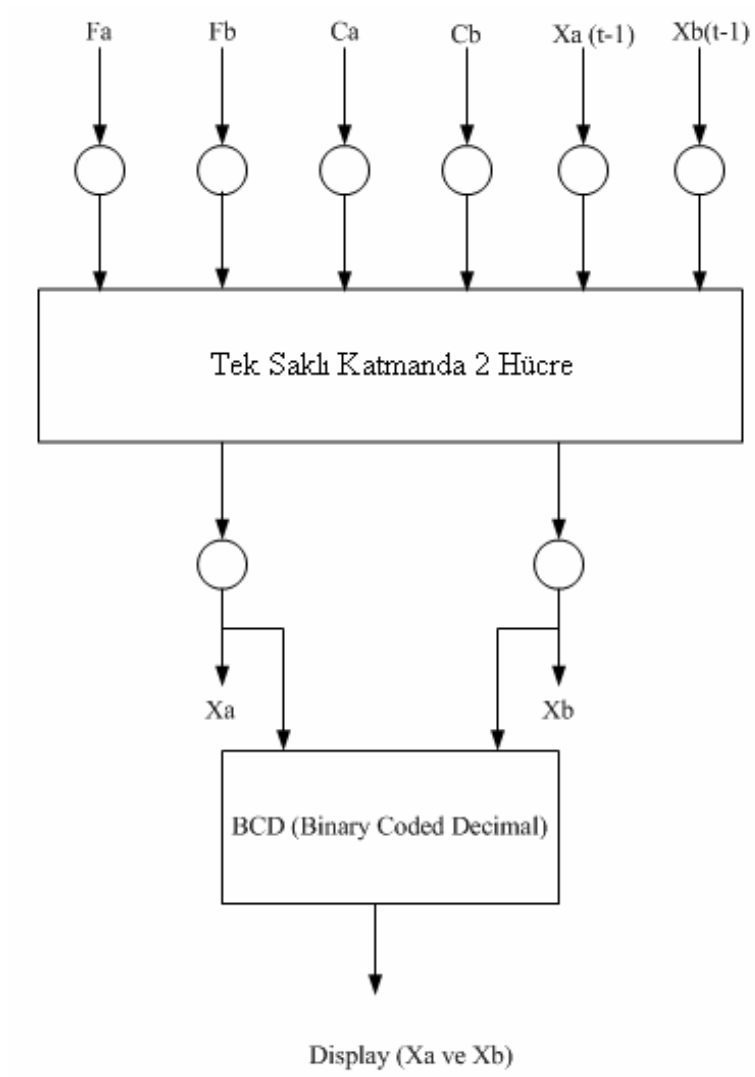
Sırası ile her bir bölüm ele alınacak ve yazılacak olan assembler programları, akış diyagramı şeklinde karakterize edilecektir. Yine her bir bölüme ait giriş-çıkış çiftleri bu akış diyagramları üzerinde rahatlıkla görülebilir. Ayrıca buradaki her bir giriş elamanı mikrokontrolöre dışarıdan bir pin aracılığı ile giriş olmayıp, hafızada bulunan sabit bir değerdir (örneğin; Fa ve Ca gibi...). Yine bu giriş olmayan her bir değer, her bir akış diyagramının sonunda belirtilecektir.

- ✓ Bulanık mantık kontrolörün GP32 ye aktarımı;



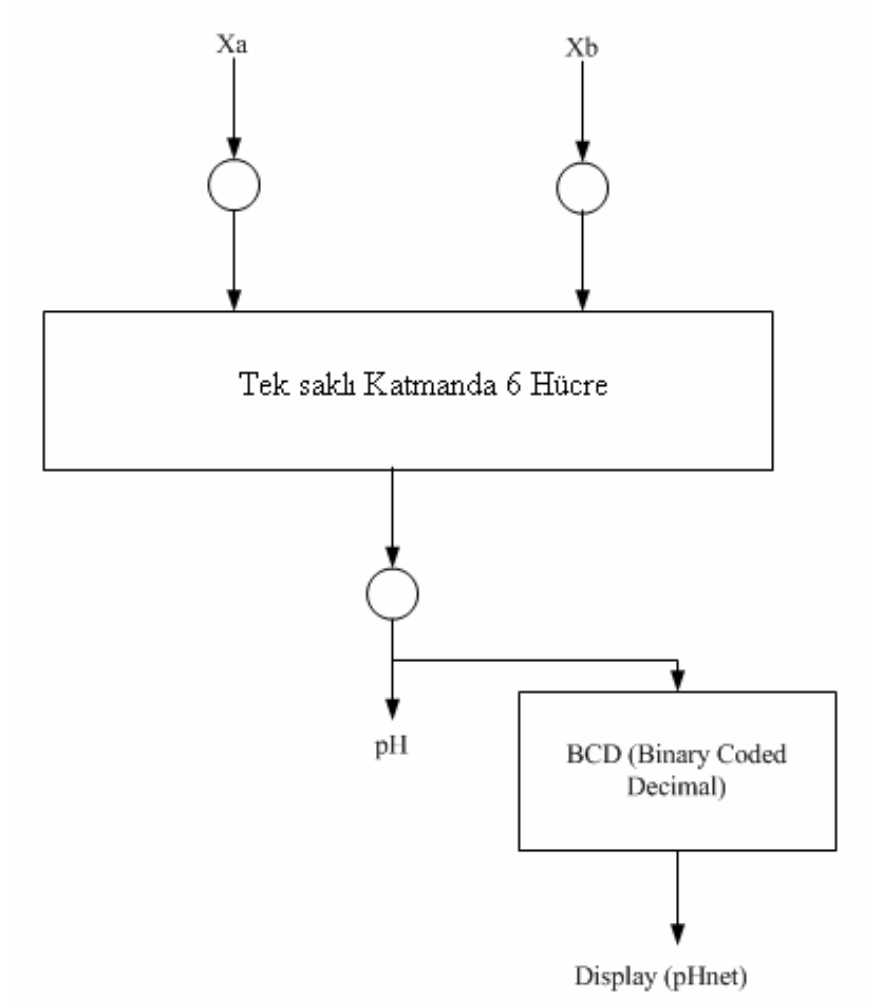
Şekil 8.2 Bulanık mantık kontrolöre ait modelin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı.

- ✓ YSA1 modelinin GP32 ye aktarımı;



Şekil 8.3 YSA1 modelinin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı (Fa, Ca, Cb, Xa(t-1) ve Xb(t-1) gerçek girişler değildir).

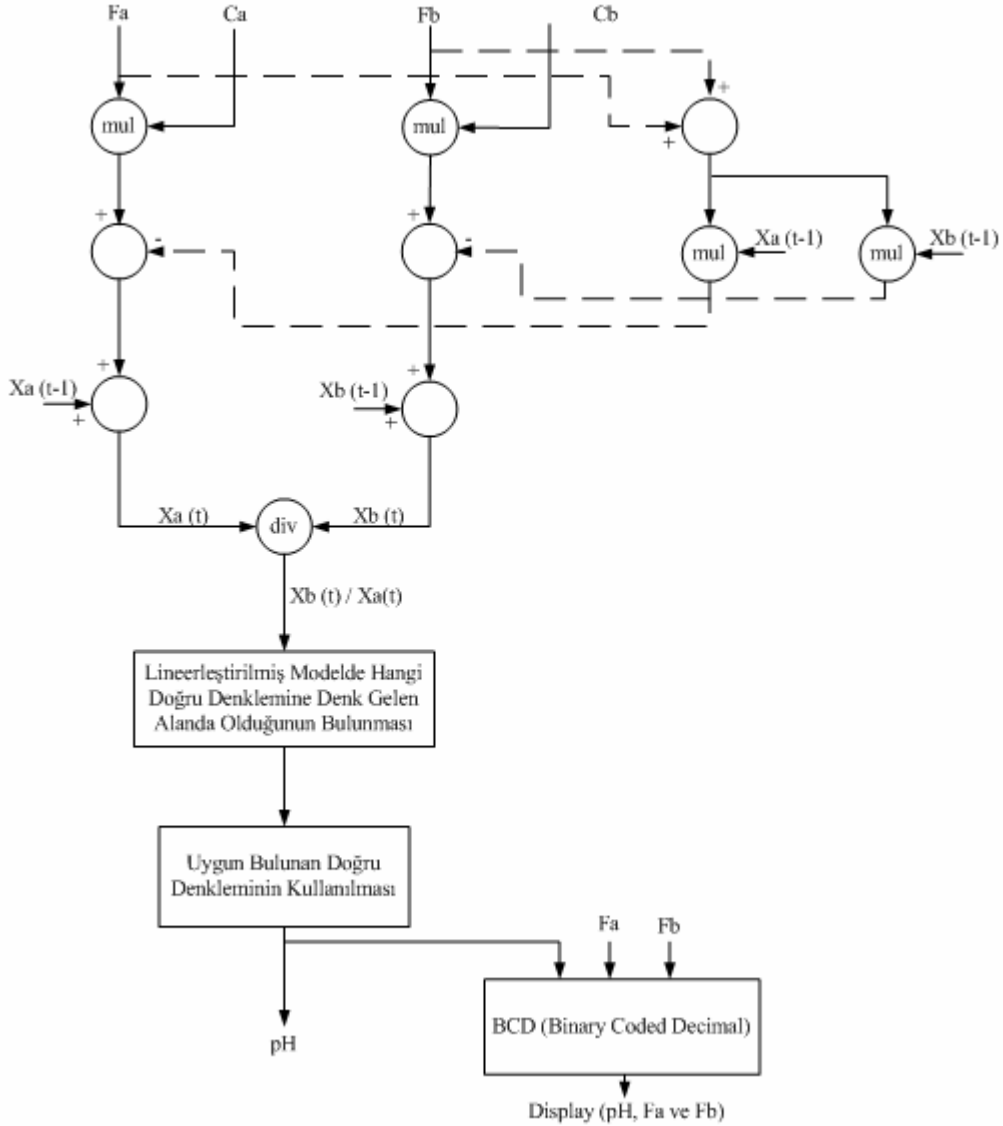
- ✓ YSA2 modelinin GP32 ye aktarımı;



Şekil 8.4 YSA2 modelinin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı.

### 8.1.3 pH Nötralizasyon Prosesine Ait Matematiksel Modelin Mikrokontrolöre Uyarlanması

Yapılacak işlem gayet basit olup, akış diyagramını kullanmak yerinde bir karar olacaktır. Yine girişlerle ilgili gerekli bilgiler, diyagramın devamında verilecektir.



Şekil 8.5 Prosesine ait modelin, mikrokontrolöre aktarılmasına ilişkin akış diyagramı ( $F_a$ ,  $C_a$ ,  $C_b$ ,  $X_a(t-1)$  ve  $X_b(t-1)$  gerçek girişler olmayıp, sabit değerleri bellekte yer almaktadır).

Görüldüğü gibi tek tek komutlar düzeyinde bilgi vermek yerine, akış diyagramlarını kullanmak daha büyük kolaylıklar sağlamaktadır. Böylece araştırmacının olayı daha geniş bir boyutta görmesine olanak sağlanmıştır.

Öte yandan bu modele (prosesine ait model) has olarak, doğrudan prosesin matematiksel modeli kullanılmamıştır. Zira elimizdeki işlemci, 8 bitlik kısıtlı işlemleri gerçekleştirebilen ve yine

kısıtlı bir hafıza yapısına sahip modeldir (bilgisayar işlemcilerine göre, unutulmamalıdır ki günümüzde hem 32 bitlik hem de 64 bitlik bilgisayar işlemcileri rahatlıkla bulunmaktadır ve kişisel bilgisayarlarda kullanılmaktadır). Bu nedenden dolayı işlemciye modeli aktarmada, Şekil 6.7 ve Çizelge 6.2 de verilen lineerleştirilmiş model kullanılmıştır. Bu durum Şekil 8.5 de rahatlıkla görülmektedir.

## 8.2 Uygulama Sonuçlarının Elde Edilmesi

Sonuçların elde edilmesi için gerekli ve yeterli olan, sadece verilen programların yazılması değildir. Bu programların bir de uygulama ortamı bulması gerekmektedir. İşte bunu için yapılacak olanlar iki bölüme ayrılabilir.

- ✓ Yazılımların geliştirilmesi;

Yazılımlar bilgisayarda, “*WinIDE Development Enviroment*” programında assembler dilinde geliştirilmiştir. Geliştirilen ve derlenen programlar, *GP32EVB* geliştirme kiti kullanılarak mikrokontrolöre aktarılmaktadır. Aktarılan kodlar tamamı ile *hexadecimal* (onaltılık sayı düzeni) formattadır.

Önemli olan temel noktalardan birisi de, kullanılan işlemcinin 8 bitlik olup yine çarpma ve bölme işlemlerini gerçekleştirebildiğidir.

Bütün bunlar yazılım geliştirmede ele alınmış noktalardır.

- ✓ Donanımsal ortamın geliştirilmesi;

Yazılımların çalıştırılması ve sonuçların elde edilebilmesi için gerek ve yeter koşullar donanımsal altyapının sağlanması ve göstergeleri kullanarak da görsel olarak çıktuların izlenebileceği ortamı oluşturmaktır.

Bu koşullardan ilki olan, donanımsal altyapıdan kasıt, sistemin kendi başına (*stand alone*) çalıştırılmasıdır. Bunun için de, her bir işlemci (her bir modele ait 4 adet işlemci için ayrı ayrı) için kendi çalışma ortamı oluşturulmuştur. Veri haberleşmesi için portların yeterli olmasından hareketle, paralel haberleşme hatları kullanılmıştır. Devrenin güç ihtiyacı ise, basit doğrultucu devreleri ve basit gerilim regülatörleri (7805 gibi) kullanılarak sağlanmıştır.

Son olarak sonuçların izlenebilmesi için 7 kollu (*seven segment*) göstergeler kullanılmıştır. Böylece dışarıdan ve bilgisayar arabağlantısı kullanmaya gerek kalmadan sistemin çalışması izlenebilmiştir.

## 9. SONUÇLAR ve ÖNERİLER

Son olarak yapılacak olan, her bir çalışma durumuna ilişkin verileri incelemek, karşılaştırmak ve bu verileri kullanarak makul bir sonuca ulaşmaktır. İşte bu noktayı göz önüne alarak, her bir duruma ilişkin verileri ele alalım.

### a. Prosele Ait “Stand Alone” Çalışma Deneysel Sonuçları (Simulink)

Amaçlanan gayet basit olup, elde edilen matematiksel modelin ne kadar bir doğrulukta sistemi temsil ettiğinin elde edilmesi olarak belirtilebilir. Elimizde belirtilen asit-baz reaksiyonuna (Asetik Asit-Sodyum Hidroksit) ait ve gerçek bir uygulamadan alınmış grafik bulunmaktadır (Şekil 7.2a). Burada baz akışını kontrol eden valfin yüzde (%) olarak pozisyon değişimine karşılık düşen pH değişimi grafiğe aktarılmıştır. Doğru bir karşılaştırma amacı ile aynı işlemin bizim tarafımızdan da yapılması gerekmektedir. Nitekim Şekil 7.1 de belirtilen simülasyon devresi kurulmuştur. Burada baz akışının değişimini “NaOH Fa” bloğu sağlamaktadır. Yüzde değişim ise, “MATLAB function” bloğu içinde hesaplanmıştır. Yine, sisteme ait model de bu blok içerisinde yer almaktadır. “Powergui” bloğu ise, istenilen grafiklerin elde edilmesine olanak sağlamaktadır.

Çalışma sonucunda elde edilen grafik, Şekil 7.2b de sunulmuştur. Gerçek prosele ait grafik de, karşılaştırmada kolaylık sağlama adına simülasyon sonucu ile aynı şeklin içinde yan yana verilmiştir. Görüldüğü gibi her iki grafikte de aynı baz akışı artışına karşılık gelen pH değeri değişim miktarları, hemen hemen aynıdır. Bu da geliştirilen modelin doğruluğunu ortaya koymaktadır.

### b. Prosele Ait PID Kontrolörlü Yapının Deneysel Sonuçları (Simulink)

Her şeyden önce belirtmelidir ki, burada amaç; bu prosesi kontrol etmek amacı ile yeni bir PID kontrolör tasarlamak değildir. Amaç; geliştirilecek olan YSA-bulanık mantık tabanlı kontrol sisteminin başarımını test etmede bir kriter olarak, geleneksel bir kontrol tekniği olan PID kontrolörün sistemi kontrol etmedeki başarımını göz önüne almaktır. Nitekim, Leondes 1998 de bu proses için tasarlanan PID kontrolörün katsayıları burada, doğrudan kullanılmıştır. Stand alone çalışmadaki bloklardan farklı olarak, PID kontrol bloğu sisteme eklenmiştir (Şekil 7.3). Bloğun parametrelerinin atanması işlemi de, Şekil 7.4 te görülebilir.

Şekil 7.5 te, PID kontrolörün sistemi kontrol etmedeki başarımı görülebilir. İlk 10 dakikalık bölümdeki salınımların sebebi, kontrolörün otomatik ayarlamaları gerçekleştirmesidir (Leondes, 1998). Bu noktadan itibaren kontrolörün sistemi istenilen, referans değere

ulaştırmayı başardığı görülmektedir. Ancak performans konusunda, oturma zamanı ve üst aşım gibi parametreler kıyaslamada kullanılacaktır. Her şeye rağmen hem dizayn kolaylığı hem de kabul edilebilir başarımından dolayı, PI ve PID kontrolörler bu tip proseslerin (Zayıf Asit-Kuvvetli Baz) kontrolünde piyasada yaygın olarak kullanılmaktadırlar.

**c. Proses Ait Nöro-Fuzzy Kontrolörlü Yapının Deneysel Sonuçları (Simulink)**

Geliştirilen yapay sinir ağı-bulanık mantık tabanlı kontrolöre ait hesaplamalar ve teorik bilgiler önceki bölümler de verilmiştir. Buna göre; yapay sinir sistemin modellenmesinde ve bulanık mantık ise, kontrol işleminde kullanılacaktır. Şekil 7.12 geliştirilen bu yeni yapının bütün bloklarını görmemizi sağlamaktadır. Daha sonra uygulamaya geçildiğinde, her bir Matlab Function Bloğu (YSA1, YSA2 ve Proses) ve Fuzzy Logic Controller (bulanık mantık kontrol) Bloğu ayrı bir işlemciye aktarılacaktır. Burada PID kontrol için geliştirilen simulasyondan farklı olarak; kontrol için bulanık mantık ve yapay sinir ağı blokları sisteme eklenmiş ve PID kontrolöre ait blok sistemden çıkarılmıştır.

PID kontrol işlemi ile aynı girdi (referans) değerleri için sistem simüle edilmiştir. Simulasyon sonucunda oluşan grafiksel yapı da, Şekil 7.13 te sunulmuştur. PID kontrol işleminde olduğu gibi burada da ilk 10 dakikalık bölüm, otomatik ayarlamalardan kaynaklanan salınımların gerçekleştiği kısımdır. Ancak bundan sonrasında kontrol sisteminin prosesi kontrol edebildiği yani referans değere ulaştırabildiği rahatlıkla görülmektedir.

Karşılaştırma boyutuna gelince, ilk 10 dakikalık kontrol sürecindeki performans bakımından çok büyük bir fark olmadığı ortadadır. Simulasyon işleminin sonraki 90 dakikalık bölümü için ise, oturma zamanları açısından her iki kontrolörde birbirine yakın performans sunmaktadır. Fakat üst aşım konusunda, her iki grafiğinde aynı ölçütle çizildiği de göz önüne alınırsa, geliştirilen kontrolörün PID kontrolöre göre daha başarılı sonuçlar ürettiği gözle görülebilmektedir. Her ne kadar, çok büyük bir performans farkı elde edilememiş olsa da, yapay zeka ürünü bu iki farklı tekniğin (yapay sinir ağları ve bulanık mantık) bir arada kullanılabilmesi ve istenilen hedefe doğru bir adım daha atılabilmesi, bu çalışma açısından bir başarımlar olarak görülebilir.

**d. Proses Ait Nöro-Fuzzy Kontrolörlü Yapının Deneysel Sonuçları (Mikrokontrolörlü Yapı)**

Bu kısım grafiksel bir yapı içermemekte olup, tamamen gözlemsel olarak değerlendirilebilir. Bundan dolayı da, diğer bölümlere nazaran tam bir kıyaslama söz konusu değildir. Amacın; yapay sinir ağı ve bulanık mantık kontrolörün, 8 bitlik sıradan bir mikrokontrolöre aktarmak

olduğu unutulmamalıdır. Başarı da amacı gerçekleştirme derecesi olduğuna göre, değerlendirmelerimizi ele alalım.

✓ Yapay sinir ağının mikrokontrolöre aktarılması;

Her şeyden önce, YSA1 ve YSA2 mikrodenetleyiciye aktarılabilmiştir. Ancak hem işlemsel boyutta hem de veri boyutunda (veri uzunluğu 8 bit) kayıpların olduğudur. Uygulama kolaylığı açısından, dinamik bir yapay sinir ağı yapısı kullanılmış olup, çalışma sırasında da ağırlıklar güncellenmektedir. Böylece ağ tarafından oluşturulan modelin optimizasyonu sağlanmıştır. Bu da daha önceki gösterimlerde yer almayan; procesten  $x_a$  ile  $x_b$  çıktıları, YSA1 için  $x_a$  ile  $x_b$  girişleri ve YSA2 için pH girişi anlamına gelmektedir. Bu veriler normal çalışma şartları ile ilgili olmadığı ve daha çok eğitimle ilgili olduğu için gösterimlere eklenmemiştir.

Matematiksel yetersizliklere düşük çalışma frekansı (YSA yapısı için) da eklenince tam bir başarıml elde edilememesi sonucu ortaya çıkmıştır. Ancak aktarımın gerçekleştirilebilmesi bile bir açılım noktası oluşturacağından dolayı çalışma başarılı kabul edilebilir.

✓ Bulanık mantık kontrolörün mikrodenetleyiciye aktarılması;

Asıl sorun üyelik fonksiyonlarının aktarımında ortaya çıkmıştır. Diğer matematiksel işlemler, kısıtlı da olsa, gerçekleştirilebilmektedir. Ancak gelen verinin bulanıklaştırılması ve hangi fonksiyona ne derece üye olduğu gibi noktalar assembler ortamında oldukça zor olmaktadır. Tabi ki de, bir bilgisayar kadar başarılı olmasa da mikrodenetleyicili yapıda bulanık mantık kontrolör çalıştırılabilmiştir.

✓ Prosesin mikrodenetleyiciye aktarımı;

Burada tek sorun, sistemin non-lineer fonksiyonunun doğrudan denetleyiciye aktarılamamasıdır. Zira logaritmik ifadeler ile üçüncü dereceden fonksiyonlara ait denklemlerin kökleri böyle bir denetleyici ile gerçekleştirilememektedir. Bunun yerine lineerleştirilmiş denklemler sistemi (doğrusal ifadeler) kullanılmıştır. Zira çarpma ve bölme işlemleri bu denetleyici ile yapılabilmektedir. Bunun yol açacağı hatalar çok büyük olmasa da, tam bir sonucun elde edilememesine etkenlerden birisidir.

✓ Sistemin başarımı;

Denetleyicinin yapay zeka için gerekli paralel işlemleri gerçekleştirilememesi, nispeten düşük frekanslarda (8 MHz  $\ll$  150 MHz lik DSP çalışma hızına göre) çalışması ve daha önce belirtilen lineerleştirmeler tam bir sonuca ulaşamamasına temel etkenlerdir.

### e. Sonuçlar ve Öneriler

Tek bir yorum yapmak yerine bu bölümü, avantajlar ve dezavantajlar olmak üzere iki bölüme ayıralım. Tabii ki de, tercihi sistem dizayncısına bırakmak en doğrusu olacaktır. İlk olarak böyle bir mikrodenetleyicili yapay zeka ürünü kontrol tekniği kullanmanın dezavantajlarını ele alalım.

#### Sistemin dezavantajları;

- ✓ İşlemsel kuvvet kaybından dolayı, çok hassas kontrol gereken noktalarda bu sistem kullanılmamalıdır.
- ✓ Nispeten düşük hız nedeni ile dinamik ve hızlı değişim gösteren sistemlerin kontrolünde bu sistem kullanılmamalıdır.
- ✓ Maliyeti, basit bir PI veya PID kontrolöre göre çok daha fazla olup, bunun kontrol edilecek sisteme ve genel yapıya göre göz önüne alınması gerekmektedir.
- ✓ Veri boyutunun yetersizliği (8 bit yani 0-255 arası değerler) nedeniyle, aralığın dışına çıkan durumlarda istemin kullanılması yerinde olacaktır.
- ✓ Olabilecek yazılımsal hatalar, sistemin tamamen kontrolden çıkmasına veya kontrolcünün devre dışı kalması sistemin kontrolsüz duruma düşmesine sebebiyet verebilir.

#### Sistemin Avantajları;

- ✓ Hata ile birlikte referansı da dikkate aldığından, klasik kontrolde (PI ve/veya PID) olduğu gibi hata ile orantılı olarak değişen bir kontrol sinyali yerine, farklı değerlerde ve sistemi daha hızlı toplayabilecek sinyaller üretebilir.
- ✓ Klasik kontrolde sıfır hata tehlikelidir. Zira hata yoksa kontrol sinyali de yoktur, bu da sistemin kontrolsüz bırakılması anlamına gelmektedir. Oysa yapay zeka ürünü kontrol tekniklerinden olan yapay sinir ağlarında ve bulanık mantık kontrolde birden fazla sistem değişkeni dikkate alınarak (referans ve hata gibi) sistemin kontrolsüz bırakılması engellenebilir.
- ✓ YSA1 ve YSA2 prosesin modelini içerdiğinden dolayı, bu model bir çevrim içerisinde birden fazla işletilerek, prediktif bir yapı kurulabilir ve böylece hata daha oluşmadan buna uygun kontrol sinyali sisteme uygulanabilir. Oysa klasik kontrolde önce hata oluşur ve kontrolör bu hatadan sonra hatayı durdurup, yok etmeye çalışır.

**KAYNAKLAR**

- Akgün, M., (2003), “Proses Dinamiği ve Ders Notları”, Yayınlanmamış Ders Notları.
- Akkaya, B.D., (2001), “Endüstriyel Atıksu Arıtma Sistemlerinin Scada Temelli Otomasyonu, RTU/PLC Kontrolü ve Bulanık Mantık Kullanılarak pH Kontrolünün Gerçekleştirilmesi”, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- Alpbaz, M., (1993), “Proses Kontrol”, A.Ü.F.F. Döner Sermaye İşletmesi Yayınları 12, Ankara.
- Atmaca, H. ve Bulut, M., (1995), “Bulanık Mantık Denetleyiciler ve Uygulamaları”, Kara Harp Okulu Öğretim başkanlığı, 1. Sistem Mühendisliği ve Savunma Uygulamaları, Ankara.
- Bulut, M., (2001), “Doğru Akım Motorunun Genetik Algoritmalar Yardımıyla Bilgisayar Temelli PI Tip Bulanık Mantık Kontrolü”, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- Cichocki, A. ve Unbehaven, R., (1993), “Neural Network for Optimization and Signal Peocessing”, John Wiley andq Sons, Gb. Stutgart, pp. 39-40.
- Dalcı, K.B., (2000), “Parametreleri Bulanık Mantık Kullanılarak Ayarlanan Oran ve İntegral Kontrolörü ile Daimi Miknatıslı Doğru Akım Motorunun Kalkış Annındaki Performansının Arttırılması”, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- Doherty, S.K., (1999), “Control of pH in Chemical Processes Using Artificial Neural Networks”, Doktora Tezi, School of Engineering, Liverpool John Moores University, İngiltere.
- Elmas, Ç., (2003), “Yapay Sinir Ağları”, Seçkin Yayıncılık, Ankara.
- Elmas, Ç., (2003), “Bulanık Mantık Denetleyiciler”, Seçkin Yayıncılık, Ankara.
- Johnson, C.D., (2003) , “Process Control Instrumentation Technology”, Prentice Hall, New Jersey.
- Karray, F. ve Silva, C., (2004), “Soft Computing and Intelligent System Design”, Addison Wesley, Harlow.
- King, P.J. ve Mamdani, E.H., (1977), “The Application of Fuzzy Control Systems to Industrial Processes”, Automatica, Vol. 3, pp. 235-242.
- Kruse, R. ve Nauck, D., (1995), “Learning Methods for Fuzzy Systems”, Proc. 3 rd German GI-Workshop Neuro-Fuzzy Systems, Almanya.
- Leondes, C.T., (1998), “Industrial and Manufacturing Systems”, Academic Press, San Diego.
- Luyben, W.L., (1988), “Process Modeling Simulation and Control for Chemical Engineers”, Mc-Graw Hill, Singapur.
- Luyben, W.L., (1997), Essentials of Process Control”, Mc-Graw Hill, Singapur.
- Mahan, B.H., (1989), “Genel Kimya”, Hacettepe Üniversitesi Yayınları, Ankara.
- Minsky, M. ve Papert, S., (1969), “Perceptrons”, The MIT Press.
- Ogata, K., (2002), “Modern Control Engineering”, Prentice Hall, New Jersey.

Özbay, E., (1995), “Bulanık Mantık ile Motor Kontrolünün İncelenmesi”, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

Özdaş, N., Dinibütün, T., Kuzucu, A. (1998), “Otomatik Kontrol Temelleri”, Birsen Yayınevi, İstanbul.

Öztemel, E., (2003), “Yapay Sinir Ağları”, Papatya Yayıncılık, İstanbul.

Petrucci, R.H., Harwood, W.S. ve Herring, G., (2002), “Genel Kimya: İlkeler ve Modern Uygulamalar 2”, Palme Yayıncılık, Ankara.

Regunath, S. ve Kadiramanathan, V., (2001), “Design of pH Control System Using Fuzzy Non-Uniform Grid Scheduling”, Applied Soft Computing 1, pp. 91-104.

Sarıoğlu, M.K., (1991), “Otomatik Kontrol I-II”, Sistem Yayıncılık, İstanbul.

Sarıoğlu, M.K., (1995), “Otomatik Kontrol I-II”, Sistem Yayıncılık, İstanbul.

Simon, H.A., (1983), “Why Should Machine Learn? In: Machine Learning-An Artificial Intelligence Approach”, Tioga Publishing.

Söğüt, O.Ö., (2004), “PID ve Değişken Kazançlı PID Kontrol Algoritmalarıyla Bilgisayar Tabanlı pH Kontrolü”, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

Şaşmaz, D.A., (1990), “Proses Kontrol”, İTÜ Kimya-Metalurji Fakültesi Offset Atölyesi, İstanbul.

Tremaine, B. P., (1992), “Comparison of Fuzzy and Classical Control: Applied to Disc Drive Spindle Servos”, Seagate Technology, Scotts Valley, CAY.

#### **INTERNET KAYNAKLARI**

The MathWorks Fuzzy Logic Toolbox for Use with MATLAB.

[www.ieee.org](http://www.ieee.org)

[www.freescale.com](http://www.freescale.com)

[www.sciencedirect.com](http://www.sciencedirect.com)

**ÖZGEÇMİŞ**

Doğum tarihi	09.02.1982	
Doğum yeri	Trabzon	
Lise	1995-1998	Beylerbeyi Lisesi
Lisans	1999-2003	Yıldız Teknik Üniversitesi Elektrik-Elektronik Fak. Elektrik Mühendisliği Bölümü
	2000-2004	Yıldız Teknik Üniversitesi Makine Fak. Endüstri Mühendisliği Bölümü
Yüksek Lisans	2003-2006	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik Müh. Anabilim Dalı, Kontrol ve Kumanda Sistemleri Programı

**Çalıştığı kurum(lar)**

2005-2006	Tüm Elektronik Mühendislik San. ve Tic. Ltd Şti.
2006-Devam ediyor	YTÜ Fen Bilimleri Enstitüsü Araştırma Görevlisi