

**T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**PIC MİKRODENETLEYİCİ İLE YAPAY SİNİR AĞI
DONANIM MODÜLÜ**

Ulvi BAŞPINAR
(Teknik Öğretmen)

**YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
ELEKTRONİK-HABERLEŞME EĞİTİMİ PROGRAMI**

**DANIŞMAN
Yrd.Doç.Dr. H.Hüseyin ÇELİK**

İSTANBUL 2006

**T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**PIC MİKRODENETLEYİCİ İLE YAPAY SİNİR AĞI
DONANIM MODÜLÜ**

Ulvi BAŞPINAR
(Teknik Öğretmen)
(141101020020081)

YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
ELEKTRONİK-HABERLEŞME EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd.Doç.Dr. H.Hüseyin ÇELİK

İSTANBUL 2006

**T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

KABUL VE ONAY BELGESİ

**PIC MİKRODENETLEYİCİ İLE YAPAY SİNİR AĞI DONANIM
MODÜLÜ**

Ulvi BAŞPINAR'ın PIC Mikrodenetleyici İle Yapay Sinir Ağı Donanım Modülü isimli Lisansüstü tez çalışması, M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 02.01.2006 tarih ve 2006/01-36 sayılı kararı ile oluşturulan jüri tarafından Elektronik-Haberleşme Anabilim Dalı Elektronik-Bilgisayar Programında YÜKSEK LİSANS Tezi olarak Kabul edilmiştir.

Danışman : (Yrd.Doç.Dr H.Hüseyin ÇELİK)(M.Ü.)
Üye : (Prof.Dr. Murat DOĞRUEL)(M.Ü)
Üye : (Yrd.Doç.Dr.A.Emin KUZUCUOĞLU)(M.Ü.)
Tezin Savunulduğu Tarih :20.01.2006

ONAY

M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararı ile Ulvi BAŞPINAR'ın Elektronik-Haberleşme Anabilim Dalı Elektronik-Bilgisayar Programında Y.Lisans (MSc.) / Doktora (Dr, PhD.) derecesi alması onanmıştır.

Marmara Üniversitesi
Fen Bilimleri Enstitüsü
Müdürü

TEŞEKKÜR

Tez çalışmamda vermiş olduğu maddi ve manevi desteklerinden dolayı başta tez danışmanım Yrd. Doç. Dr. H.Hüseyin ÇELİK'e, deney düzeneğinin kurulmasında bize yardımcı olan ve gösterdiği sabır için Yrd. Doç. Dr. Hasan ERDAL'a ve laboratuvarında çalışma izni verdiği için Yrd. Doç. Dr. Mustafa ONAT'a ayrıca desteklerinden dolayı Yrd. Doç.Dr. Nazmi EKREN'e, her zaman yanımda olan aileme, arkadaşlarım Araş.Gör. Oğuzhan GÜNDÜZ'e, Araş. Gör. Önder DEMİR'e, Araş. Gör. Mustafa DEMETGÜL'e, Cengiz ŞAFAK'a ve önerileriyle bana yardımcı olan Araş. Gör. Kenan TOKER'e , TEŞEKKÜRLERİMİ borç bilirim.

Ocak,2006

Ulvi BAŞPINAR

İÇİNDEKİLER

	SAYFA
ÖNSÖZ VE TEŞEKKÜR.....	I
İÇİNDEKİLER.....	II
ÖZET	V
ABSTRACT	VII
SEMBOL LİSTESİ.....	VIII
KISALTMALAR.....	IX
ŞEKİL LİSTESİ	X
TABLO LİSTESİ	XIII
BÖLÜM I GİRİŞ VE AMAÇ	1
BÖLÜM II YAPAY ZEKA KAVRAMLARI VE TEKNİKLERİ..	3
II.1 AKIL VE ZEKA.....	3
II.2 YAPAY ZEKA KAVRAMI	4
II.3 YAPAY ZEKA TARİHÇESİ.....	5
II.4 YAPAY ZEKA TEKNİKLERİ.....	6
II.4.1 Genetik Algoritma.....	7
II.4.2 Bulanık Mantık.....	9
II.4.3 Uzman Sistemler.....	9
II.4.4 Karınca Algoritmalar.....	11
BÖLÜM III YAPAY SİNİR AĞLARI.....	13
III.1 BİYOLOJİK NÖRON.....	14
III.2 YAPAY NÖRON.....	20
III.2.1 Aktivasyon Fonksiyonları.....	22
III.3 YSA MODELLERİ.....	24

III.3.1	ADALINE.....	24
III.3.2	Çok Katlı Perseptronlar.....	25
III.3.2.1	ÇKA Ağı'nın İşletim Aşamaları.....	26
III.3.3	Kohonen Ağı.....	27
III.3.4	ART Ağlarının Yapısı.....	28
III.3.4.1	ART Ağlarının İşletim Aşamaları.....	29
III.3.5	Hopfield Ağı.....	31
III.4	YSA ÖĞRENME KURALLARI.....	32
III.4.1	Temel Öğrenme Kuralları.....	32
III.4.1.1	Hebb Kuralı.....	33
III.4.1.2	Hopfield Kuralı.....	33
III.4.1.3	Delta Kuralı.....	33
III.4.1.4	Kohonen Öğrenme Kuralı.....	33
III.4.2	Öğrenme Kurallarının Sınıflandırılması.....	34
III.4.2.1	Danışmanlı Öğrenme.....	34
III.4.2.2	Danışmansız Öğrenme.....	39
III.4.2.3	Takviyeli Öğrenme.....	40
III.4.3	Uygulamaya Göre Öğrenme Yöntemlerinin Sınıflandırılması.....	44
III.4.3.1	Off-line Öğrenme.....	44
III.4.3.2	On-line Öğrenme.....	44
III.5	YAPAY SİNİR AĞI DONANIMLARI.....	45
III.5.1	Analog Olarak Oluşturulan YSA Donanımları.....	45
III.5.1.1	Değişken Lineer İletkenli Cihazlar.....	46
III.5.1.2	Aktivasyon Fonksiyon Üretici.....	47
III.5.1.3	Sinaptik Bağlantıların Gerçekleştirilmesi.....	51
III.5.1.4	Analog Sistemlerin Avantajları ve Dezavantajları.....	53
III.5.2	Sayısal Yapay Sinir Ağı Donanımları.....	54
III.5.3	Analog-Sayısal YSA Donanımları.....	55
BÖLÜM IV MİKRODENETLEYİCİLER.....		57
IV.1	PIC MİKRODENETLEYİCİLERİ.....	58
IV.1.1	PIC Mikrodenetleyicilerinin İç Yapısı.....	58
IV.1.2	PIC 16F877 Mikrodenetleyicisi.....	59
IV.1.3	PIC 16F877'nin Bazı Özel Fonksiyonları.....	61
IV.1.3.1	USART.....	61
IV.1.3.2	Senkronize Seri Port(MSSP).....	61
IV.1.3.3	Analog/Sayısal Çevirici Modülü.....	62
IV.2	I2C SERİ HABERLEŞMESİ VE BUS ÖZELLİKLERİ....	62
IV.2.1	I2C Bus Kavramı.....	63
IV.2.2	I2C Çalışma Karakteristiği.....	64
IV.2.2.1	Bit Transferi.....	64
IV.2.2.2	Veri Geçerliliği.....	64
IV.2.2.3	START ve STOP Şartları.....	65
IV.2.2.4	BYTE Formatı.....	65
IV.2.2.5	Bilgi Alındı Sinyali (Acknowledge).....	66
IV.3	PIC MİKRODENETLEYİCİ TABANLI YSA DONANIMININ OLUŞTURULMASI.....	68
IV.3.1	Tek Mikrodenetleyicili YSA Donanımı.....	68

IV.3.1.1 Devrenin Çalışması.....	70
IV.3.1.2 Tek Mikrodenetleyici Kullanan Modelin Program Akış Diyagramı Açıklaması	73
IV.3.2 Çoklu Mikrodenetleyicili YSA Donanımı.....	73
IV.3.2.1 Devrenin Çalışması.....	75
IV.3.2.2 Çoklu Mikrodenetleyicili YSA Donanımında Kullanılan Programların Akış Diyagramı Açıklaması.....	75
IV.4 MİKRODENETLEYİCİLER İLE ETKİN YSA DONANIMI OLUŞTUMA.....	79
IV.4.1 Kullanılan Programlama Dili.....	79
IV.4.2 Kullanılan Sayı Formatı.....	80
IV.4.2.1 Özel Sayılar.....	82
1.a Sıfır Sayıların Temsili.....	82
1.b Sonsuz Sayıların Temsili.....	82
1.c Tanımsız İfadelerin Temsili.....	82
IV.4.2.2 PICC Derleyicisinde Kullanılan 24 Bit Kayan Nokta Aritmetiğinin İncelenmesi.....	82
IV.4.3 Kullanılan Aktivasyon Fonksiyonu.....	83
IV.4.3.1 Sigmoid Fonksiyon Uygulamaları.....	84
1.a Kural Tabanlı Yaklaşımlar.....	84
1.b Alippi ve Storti Gajani Yaklaşım.....	84
1.c PLAN Yaklaşımı.....	85
IV.4.4 Kullanılan Algoritma.....	86
IV.5 PIC MİKRODENETLEYİCİSİNDE NÖRON DEĞİŞİMİNİN ETKİLERİ.....	86
BÖLÜM V DENEYSEL ÇALIŞMALAR.....	90
V.1 DOĞRUSALLAŞTIRMA UYGULAMASI.....	90
V.2 DC MOTOR KONTROL UYGULAMASI.....	94
BÖLÜM VI SONUÇ VE DEĞERLENDİRME.....	102
KAYNAKLAR.....	104
EK-A PROGRAM KODLARI CD.....	106
EK-B BASKI DEVRE ŞEMALARI.....	107
ÖZGEÇMİŞ.....	114

ÖZET

PIC MİKRODENETLEYİCİ İLE YAPAY SİNİR AĞI DONANIM MODÜLÜ

Yapay sinir ağları (YSA) bir çok alanda akıllı çözüm gerektiren problemler üzerinde başarıyla uygulanmaktadır. Özellikle karmaşık sınıflandırma ve doğrusal olmayan modeller için problemler üzerinde etkin çözümler sağlayan bir yaklaşım olarak güncelliğini korumaktadır.

YSA gelişim sürecine bakıldığında, programlama yoluyla bilgisayar ortamında gerçekleştirilen uygulamalar yanında, özel elektronik YSA donanımları ile çözümler oluşturma çalışmalarında hızla geliştiğini görmekteyiz. Bu alandaki çalışmalar sonucu çoğunlukla bilgisayara göre daha yüksek maliyet taşıyan ve genel amaçlı YSA çipleri üretilmiştir.

Bu tez çalışmasında, PIC denetleyici tabanlı bir elektronik YSA donanımı gerçekleştirilmiştir. Etkin uygulamalarda gerekli olan sürekli değerler ile çalışma göz önünde bulundurularak, YSA donanımı Analog giriş/çıkış özelliğine yönelik olarak tasarlanmıştır. Uygulamada temel zorlukları taşıyan tamamen Analog elemanlar ile tasarım yerine, bu çalışmada sayısal kontrollü ve Analog veri işleyen bir YSA modeli gerçekleştirilmiştir. Tasarlanan YSA donanımı eğitilmiş bir YSA yapısının gerçek donanımsal uygulamaya aktarılmasını sağlamaktadır. Yapısal çalışmada tekli ve çoklu mikrodenetleyici yapısı ile gerçekleştirilen alternatif YSA donanımları üretilmiştir. Çoklu mikrodenetleyici YSA yapısı, artan hücre sayısına karşılık hızlı veri işlemeyi hedef alan bir çözüm olarak oluşturulmuştur.

Tasarlanan YSA donanımının test edilmesi amacıyla iki uygulama gerçekleştirilmiştir. İlk uygulamada farklı giriş fonksiyonlarına karşılık doğrusal bir

ıkıř fonksiyonu retme problemi temel alınmıřtır. İkinci uygulamada ise bir DC motorun basit kontrolne ynelik bir YSA kontrolc oluřturularak, YSA donanımının iřleyiři gzlenmiřtir.

Deneyssel alıřmalar, retilen YSA donanım modlnn bařarılı sonular rettiğini gstermektedir. Tasarlanan donanım, ok yksek hcre sayısı ve hız gerektirmeyen YSA problemlerinde kullanılabilecek bir uygulama alt yapısı saęlanmaktadır.

ABSTRACT

ARTIFICIAL NEURAL NETWORK HARDWARE MODULE USING PIC MICROCONTROLLERS

Artificial Neural Networks(ANN) are being used successfully in wide range of area especially for the solution of problems such as classifying and linearization.

Fistly, ANN's are implemented using computers programs but according to developments in semiconductor technology and the advantages of analog hardware attract more attention to the analog neural networks. As a result of researchs, neural chips which are expensive than PC, are produced.

In this thesis, a PIC microcontroller based electronic ANN hardware has been implemented. The hardware is designed to allow analog input and output because most applications in real world use analog signals. Digitally controllable analog processing hardware was chosen because it is very difficult to implement ANN hardware using analog components. For this study, two different hardware module is designed. First hardware include one PIC microcontroller the other one include multi PIC so the second one is expandable. Expandable architecture provide more speed and more processing capability to the hardware.

In order to test the designed hardware two different application have been done. First application is linearization of three nonlinear function. Second application is controlling a DC motor.

Experimental studies have shown that designed ANN hardware has good results. This hardware is suitable for applications that needs not so much neuron and speed .

January,2006

Ulvi BAŞPINAR

SEMBOL LİSTESİ

G	:İletkenlik
N	:İşlemci Eleman Sayısı
P	:Sınıf sayısı
U	:Giriş Voltajı
V	:Çıkış Voltajı
p	:Nöron Giriş Sayısı
x	:Yapay Sinir Ağı Giriş Sayısı
w	:Yapay Sinir Ağı Ağırlık Katsayıları
β	:İletkenlik parametresi(JFET'ler için)
θ	:Aktivasyon Fonksiyonu Eşik Seviyesi
ε	:Hata miktarı

KISALTMALAR

ADALINE	:Adaptif Lineer Eleman
ADC	:Analog-Sayısal Çevirici(Analog-Digital Converter)
ART	:Adaptif Rezonans Teorisi(Adaptive Resonance Theory)
ÇKA	:Çok Katlı Ağlar
DAC	:Sayısal-Analog Çevirici(Digital-Analog Converter)
DLVQ	:Dinamik Vektör Öğrenme Katsayısı
GA	:Genetik Algoritma
İE	:İşlemci Eleman
KDH	:Kısmi Zamanlı Hafıza
KKO	:Karınca Kolonisi Optimizasyonu
LVQ	:Vektör Öğrenme Katsayısı
RBF	:Radyal Tabanlı Fonksiyon
RISC	:Azaltılmış komut seti(Reduced Instruction Command Set)
SCI	:Seri iletişim arabirim(Serial Communication Interface)
SCL	Seri saat sinyali(Serial Clock)
SDA	:Seri Veri (Serial Data)
SL1	:1.Köle
SL2	:2.Köle
SL3	:3.Köle
UDH	:Uzun Dönemli Hafıza
US	:Uzman Sistem
USART	:Evrensel seri asenkron alıcı-verici
VLSI	:Çok Büyük Ölçekli Entegrasyon
YSA	:Yapay Sinir Ağları
YYM	:Yeniden Yerleştirme Modülü
YZ	:Yapay Zeka
µC	:Mikrodenetleyici

ŞEKİL LİSTESİ

SAYFA NO

Şekil III.1 Biyolojik sinir sisteminin blok gösterimi.....	15
Şekil III.2 Miyelinli bir nöronun yapısı.....	16
Şekil III.3 Aksiyon potansiyeli.....	17
Şekil III.4 Biyolojik nöron.....	17
Şekil III.5 Yapay nöronun temel gösterimi.....	21
Şekil III.6 Doğrusal ve basamak fonksiyonları.....	23
Şekil III.7 Sigmoid fonksiyon.....	24
Şekil III.8 ÇKA yapısı.....	25
Şekil III.9 Kohonen Ağı.....	27
Şekil III.10 ART ağının genel yapısı.....	28
Şekil III.11 ART ağında çıktı oluşturma süreci(aşağıdan yukarı).....	29
Şekil III.12 ART ağında çıktı oluşturma süreci(yukarıdan aşağı).....	30
Şekil III.13 ART ağında yeni bir sınıf oluşturma.....	30
Şekil III.14 Hopfield ağı.....	31
Şekil III.15 Bir ağda geriyayılımlı bir öğrenme algoritmasının uygulanması....	35
Şekil III.16 Danışmanlı öğrenme yapısı.....	35
Şekil III.17 Öğrenmenin hata uzayında gösterimi.....	36
Şekil III.18 Çok boyutlu hata uzayı.....	37
Şekil III.19 Öğrenen bir ÇKA ağının öğrenme eğrisi.....	39
Şekil III.20 Danışmansız öğrenme yapısı.....	40
Şekil III.21 Takviyeli öğrenme yapısı.....	40
Şekil III.22 Takviyeli öğrenme blok diyagramı.....	41
Şekil III.23 Referans vektör örneği.....	42

Şekil III.24 LVQ öğrenme prosedürünün geometrik gösterimi.....	43
Şekil III.25 LVQ öğrenmede ağırlıkların değişim yönü.....	44
Şekil III.26 Basit elektronik YSA modeli.....	45
Şekil III.27 NMOS Çıkış Karakteristiği.....	46
Şekil III.28 Voltaj Kontrollü Direnç.....	47
Şekil III.29 Analog karşılaştırıcı ve çıkış karakteristiği.....	48
Şekil III.30 OPAMP kullanarak elde edilen lineer aktivasyon fonksiyonu	48
Şekil III.31 Sigmoid yaklaşımlı aktivasyon fonksiyonu.....	49
Şekil III.32 OPAMP kullanılarak yapılan nöron modelleri.....	49
Şekil III.33 Gilbert çarpma devresi.....	50
Şekil III.34 Yarım Gilbert çarpma devresi.....	51
Şekil III.35 Fet ve kondansatör ile gerçekleştirilmiş sinaptik ağırlıklar.....	52
Şekil III.36 Çoklayıcı DAC ile ağırlıkların tutulması.....	53
Şekil III.37 Sayısal YSA yonga yapısı.....	54
Şekil IV.1 Temel PIC blok diyagramı.....	59
Şekil IV.2 PIC 16F877 blok diyagramı.....	60
Şekil IV.3 İki mikrodnetleyicili I2C konfigürasyonu.....	64
Şekil IV.4 I2C Bus üzerinde bit transferi.....	65
Şekil IV.5 Start ve Stop şartları.....	65
Şekil IV.6 I2C bus hattı üzerindeki veri transferi.....	66
Şekil IV.7 I2C bus hattı üzerindeki Acknowledge sinyali.....	67
Şekil IV.8 Tek mikrodnetleyicili YSA modeli.....	69
Şekil IV.9 Tek mikrodnetleyicili YSA Donanım modülü.....	71
Şekil IV.10 Tek mikrodnetleyicili YSA donanımında kullanılan programın akış diyagramı.....	72
Şekil IV.11 Paralel veri işleyebilen mikrodnetleyicili YSA modeli.....	74
Şekil IV.12 Çoklu mikrodnetleyicili YSA donanımı.....	76
Şekil IV.13 Çoklu mikrodnetleyicili YSA donanımı akış diyagramı.....	78
Şekil IV.14 IEEE 754 kayan nokta formatı.....	80
Şekil IV.15 RAM değişimine bağlı olarak nöron sayısı değişimi.....	87
Şekil IV.16 RAM ve EEPROM değişimine bağlı nöron değişimi.....	87
Şekil IV.17 Program hafıza değişimine bağlı nöron değişimi.....	88
Şekil IV.18 Program hafıza değişimine bağlı nöron değişimi (araktman çıkartılmış)..	88

Şekil V.1 Doğrusallaştırılması yapılacak olan $y(x)$, $r(x)$, $t(x)$ fonksiyonları ve istenen çıkış eğrisi.....	91
Şekil V.2 Doğrusallaştırma işleminde kullanılan YSA modeli.....	91
Şekil V.3 MATLAB yazılımıyla yapılan ağ eğitimi sonucu.....	92
Şekil V.4 Mikrodenetleyici ve bilgisayar ortamında alınan YSA sonuçlarının karşılaştırılması	92
Şekil V.5 DC motor kontrolünde kullanılan YSA modeli.....	94
Şekil V.6 DC motor kontrolünde kullanılan YSA'nın eğitim modeli.....	95
Şekil V.7 DC motor kontrol uygulaması blok diyagramı.....	95
Şekil V.8 Nöronların PIC'lere dağılımı.....	96
Şekil V.9 DC motor kontrolünde kullanılan YSA'nın simulasyon grafiği.....	96
Şekil V.10 400 devirde nöral kontrolör çalışması.....	97
Şekil V.11 700 devirde nöral kontrolör çalışması.....	97
Şekil V.12 900 devirde nöral kontrolör çalışması.....	98
Şekil V.13 1100 devirde nöral kontrolör çalışması.....	98
Şekil V.14 400 devirde nöral kontrolör çalışması(yüklü).....	99
Şekil V.15 700 devirde nöral kontrolör çalışması(yüklü).....	100
Şekil V.16 900 devirde nöral kontrolör çalışması(yüklü).....	100
Şekil V.17 1100 devirde nöral kontrolör çalışması(yüklü).....	101

TABLO LİSTESİ

	<u>SAYFA NO</u>
Tablo III.1 Beyin ile ilgili bazı özellikleri.....	18
Tablo III.2 Sayılarla sinir hücreleri.....	18
Tablo III.3 Bilgisayar ve beyinin karşılaştırılması.....	19
Tablo III.4 Biyolojik sinir sistemi ile YSA'nın benzeşimleri	20
Tablo III.5 İstatiksel yöntemlerle yapay sinir ağların terminolojik benzeşimleri.....	20
Tablo IV.1 PIC 16F877 entegresinin özellikleri.....	60
Tablo IV.2 I2C veri haberleşmesinde kullanılan terimler ve açıklamaları.....	62
Tablo IV.3 IEEE 754 formatında tek hassasiyetli ve çift hassasiyetli sayıların bitisel dağılımı.....	81
Tablo IV.4 PICC derleyicisinde kullanılan kayan nokta formatı.....	83
Tablo IV.5 PLAN yaklaşımı hesap tablosu.....	85
Tablo V.1 Mikrodenetleyici ile bilgisayar ortamından alınan YSA çıkış değerleri.....	93
Tablo V.2 Hesaplanan DAC çıkış voltajı ile ölçülen DAC çıkış voltajları.....	93

BÖLÜM I

GİRİŞ VE AMAÇ

Teknolojinin gelişmesi izlendiğinde önceleri sadece veri transferi yapmak ve karmaşık hesaplamaları gerçekleştirmek üzere geliştirilen elektronik donanımların zaman içerisinde büyük miktardaki verileri filtreleyerek özetleyebilen ve mevcut bilgileri kullanarak olaylar hakkında yorumlar yapabilen nitelikler kazandığı görülmektedir. Günümüzde ise elektronik donanımlar hem olaylar hakkında karar verebilmekte hem de olaylar arasındaki ilişkileri öğrenebilmektedirler. Özellikle modellenemeyen ve uzman deneyimi gerektiren problemler sezgisel yöntemler yolu ile bilgisayarlar yada elektronik donanımlar tarafından çözülebilmektedir. Bilgisayarları yada elektronik donanımları bu özelliklerle donatan ve bu yeteneklerin gelişmesini sağlayan çalışmalar “yapay zeka” çalışmaları olarak adlandırılmıştır. Günümüz teknolojisi, yapay zeka teknikleri kullanarak birçok alanda önemli ilerlemeler kaydetmiştir.

Yapay zeka teknikleri başarılı bir biçimde uygulandığında karmaşık, adaptif ve esnek kararlar ile çözümlenmesi gereken problemler için önemli bir araç haline gelir.

Yapay sinir ağları yapay zeka teknikleri içerisinde önemli bir yer tutmaktadır. YSA’nın öğrenme yeteneği probleme ait çözümün üretilmesindeki temel faydayı sağlar. Öncelikle problemi tanımlayan veriler bir eğitim seti şeklinde belirlenerek,

ağın bu veriler ile istenen sonuçlara yakınsaması sağlanır. İzleyen aşamada ağ yapısına öğretilmeyen veriler ile de doğru kararlar üretip üretmediği test edilmektedir. Bu aşamada elde edilen başarı, eğitim setinin problemi tanımlayıcılık niteliği ile doğrudan ilişkilidir. Sözü edilen işlemler bir YSA sisteminin oluşturulmasının ana basamaklarını tanımlar ve çoğunlukla bilgisayar ortamında simülasyonlar şeklinde uygulanır. Problem çözümü için oluşturulan YSA sisteminin, gerçek zamanda işleyen bir donanıma dönüştürülmesi ise uygulama aşamasında büyük önem kazanmaktadır. Bu yaklaşım YSA tekniğinin hedeflenen ana kavramı içinde yer almaktadır. Alternatif yaklaşım ise sistem oluşturmada bilgisayar ve getirdiği standartları zorunlu bir araç olarak kullanmaktır.

BÖLÜM II

YAPAY ZEKA KAVRAMLARI VE TEKNİKLERİ

Bu bölümde yapay zeka ve teknikleri hakkında genel bir bilgi verilecektir.

II.1 AKIL VE ZEKA

Akıl ve Zeka kelimeleri birbirine karıştırılmamalıdır. Akıl; düşünme, anlama, kavrama, idrak etme, karar verme ve önlem alma yetenekleridir. Zeka ise gerçekleri algılama, yargılama, ve sonuç çıkarma yeteneği olarak ifade edilmekte bazen de, bir şeyi anlama, kavrama veya önemini anlama olarak da tanımlanmaktadır. Zeka, bireysel bilgi birikimi ve deneyimlerle ilişkilidir.

Akıl, genetik yoldan intikal eden sevgi, korku, kıskançlık, doğal savunma güdülerinin yanı sıra bulunduğumuz çevreden aldığımız etkileşimlerden ve toplumun şartlandırılmalarından etkilenecek gelişmektedir. Dolayısıyla akıl sabit değil, aksine insanın hayatının sonuna kadar artabilen ve gelişebilen bir yetenektir. Akıl, makine, bilgisayar, yazılım veya başka bir yolla taklit edilemez.

Her insan doğuştan belirli bir zekaya sahiptir. Zeka, belirli bir konuda çalışarak, öğretilerek, eğitilerek, edinilen bilgi ve birikimlerle, deneyimlere dayalı becerilerle geliştirilebilir. İlk kez karşılaşılan yada ani olarak gelişen bir olaya uyum sağlayabilme, anlama, öğrenme, analiz yeteneği, beş duyunun, dikkatin ve düşüncenin yoğunlaştırılması zeka ile gerçekleştirilebilmek ve yazılım veya tümleşik yongalarla taklit edilebilmektedir. Bu durumda zeka ‘Yapay Zeka’ olarak adlandırılmaktadır[1].

II.2 YAPAY ZEKA KAVRAMI

Yapay zeka (YZ); zeka ve düşünme gerektiren işlemlerin bilgisayar veya elektronik donanımlar tarafından yapılmasını sağlayacak araştırmaların yapılması ve yeni yöntemlerin geliştirilmesi üzerinde çalışan bilim dalıdır. Daha geniş anlamda YZ:

- “Bilgisayarların bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılması bilimi”

veya

- “İnsanların düşünüş sürecini, çeşitli makineler kullanarak taklit etme”

veya

- “Programlanmış bir bilgisayarın düşünme girişimi” olarakda tarif edilmektedir.

Diğer bir ifadeyle matematiksel olarak formülasyonu kurulamayan veya çözülmesi mümkün olmayan problemler, sezgisel yöntemler yoluyla bilgisayarlar tarafından çözülebilmektedir. Bilgisayarları bu özelliklerle donatan ve bu yeteneklerin gelişmesini sağlayan çalışmalar “yapay zeka “ çalışmaları olarak bilinmektedir[2].

II.3 YAPAY ZEKA TARİHÇESİ

İnsan beyni hakkında ki çalışmalar binlerce yıl öncesine dayanır. Modern elektroniğin gelişmesi ile birlikte, düşünce işlemini kullanmaya çalışmak doğal bir hale gelmiştir. YZ konusunda ilk çalışma sinir hekimi Warren McCulloch ile matematikçi Walter Pitts tarafından gerçekleştirilmiştir. McCulloch ve Pitts, insan beyninin hesaplama yeteneğinden esinlenerek, elektrik devreleriyle bir sinir ağını modellemişlerdir.

1948 yılında Wiener “Cybernetics” isimli kitabında, sinirlerin çalışması ve davranış özelliklerine değinirken 1949 yılında ise Hebb “Organization of Behaviour” isimli kitabında, öğrenme ile ilgili temel teoriyi ele almıştır.

Hebb kitabında öğrenebilen ve uyum sağlayabilen sinir ağı modeli için temel oluşturacak Hebb Kuralı’nı ortaya koymuştur. Hebb Kuralı sinir ağının bağlantı sayısı değiştirilebilirse, sinir ağının öğrenebileceğini öngörmekteydi. 1950’li yıllardan sonra birçok araştırmacı Hebb kuralından esinlenerek yapay sinir ağlarının hesaplama gücünü artırıcı yönde çalışmalar yapmıştır. IBM araştırma laboratuvarların da yapılan bir sinir ağı benzetimi çalışması başarısızlıkla sonuçlanmasına karşın sonraki girişimlerde başarı sağlanmıştır. 1951 yılında Minsky, sinapsların ayarlanmasıyla belirli bir işi başarabilecek 40 nöronlu bir öğrenme sistemi sunmuştur. İlk yapay sinir ağı sistemlerinden olan SNARC, Minsky ve Edmonds tarafından 1951’de MIT’de geliştirilmiştir. 1957 yılında Frank Rosenblatt’ın Perceptron’u gerçekleştirmesinden sonra yapay sinir ağı alanındaki gelişmeler hızlanmıştır. Perceptron, beyin işlevlerini modelleyebilmek amacıyla yapılan çalışmalar neticesinde ortaya çıkan tek katmanlı eğitilebilen ve tek çıkışa sahip olan yapay sinir ağıdır.

1959 yılında, Bernard Widrow ve Marcian Hoff ADALINE ve MADALINE diye adlandırdıkları ağ modellerini geliştirdiler. MADALINE, telefon hatlarında oluşan yankıları yok eden bir uyarlanabilir süzgeç olarak kullanılmış, gerçek dünya sorunlarına uygulanmış olan ilk sinir ağıdır ve hala kullanılmaktadır. Bu dönemde en dikkat çekici çalışma Widrow’un 1963 yılında yaptığı ‘ters sarkaç’ denetleyicisidir.

1969 yılında Minsky ve Papert Perceptron'un yetersizliğini görmüşler ve XOR problemini çözemediğini ispatlamışlardır. Bunun için iki katmanlı ileri beslemeli ağların kullanılabileceğini ileri sürmüşler ve tek katmanlı ağlardaki birçok sınırlamaları ortadan kaldırdığını göstermişlerdir. Fakat gizli katmanların ağırlıklarının nasıl değiştirileceği konusunda bir yöntem önerememişlerdir.

Araştırmaların çoğunda duraksamalar olmuşsa da bazı bilim adamları çalışmalarına devam etmiştir. 1971 yılında Kohonen, ilişkili hafızalar olan **Learning Vector Quantisation** LVQ'yu geliştirmiş daha sonrada yeni versiyonları olan LVQ1, LVQ2, DLVQ gibi yaklaşımlar sunmuştur.

Minsky ve Papert'in savları McClelland ve Rumelhart'ın 1986 yılında yayınladıkları PDP kitabında çürütülmüş ve bu kitapta verilen uygulamalar, yapılacak olan birçok çalışmaya ışık tutmuştur. Bu yıllar da artmaya başlayan YSA araştırmaları, öğrenme algoritmaları, yapılar, nöro-bilgisayarlar, analog ve sayısal paralel donanımlara yönlenmiştir. 1987 yılında yapılan ilk uluslararası YSA konferansında sunulan yüzlerce çalışma ve uygulamayla artık bu alandaki çalışmalar yaygınlaşmaya başlamıştır. Amerika'da 17 temmuz 1990'da "Beyin Yılları 1990-1999" ilan edilmesiyle bu konuya verilen önem daha da artmıştır.

YZ çalışmalarına farklı bilim disiplinlerinden birçok bilim adamı katkıda bulunmuştur. Nörologlar, biyologlar, fizyologlar, mühendisler, matematikçiler, psikologlar ve fizikçiler bazen tek fakat çoğu zaman birlikte işbirliği içinde çalışarak YZ'nin bugünkü seviyeye gelmesine büyük katkıda bulunmuşlardır. Bundan sonrada bu çalışmalar takım veya grup çalışmalarının artmasıyla daha da ileri boyutlara taşınabilecektir.

II.4 YAPAY ZEKA TEKNİKLERİ

Yapay zeka araştırmacılarının baştan beri ulaşmak istediği ideal, insan gibi düşünen ve davranan sistemler oluşturabilmektir. Fakat bunun güçlüğü anlaşılınca çalışmanın yönü rasyonel düşünen ve davranan sistemlerin tasarlanmasına çevrilmiştir.

Genel olarak YZ teknikleri incelendiğinde, insanın sadece zekası veya beynin çalışması yanında vücudundaki diğer sistemlerinin çalışmalarını da taklit ederek

veya farklı bir formata dönüştürülerek problemlerin çözümüne uygulamaya çalışılmıştır. Bu yaklaşımlar, sadece insan vücudundaki sistemlerle de sınırlı kalmamış diğer canlılardan da örnekler alınmıştır.

YZ teknikleri, yapay sinir ağları, bulanık mantık, sezgisel algoritmalar(genetik algoritmalar, tabu arama, karınca algoritması, ısıt işlemler, bağışıklık sistemi gibi) ve uzman sistemler olmak üzere gruplandırılırlar.

II.4.1 Genetik Algoritmalar

Evrimsel programlamanın (evalutinary computing) bir parçası olan Genetik Algoritmalar (GA) evrim teorisinden esinlenerek ortaya çıkmıştır. Evrimsel programlama, 1960'lı yıllarda I. Rechenberg'in "Evrimsel Stratejileri" adlı çalışmasıyla gündeme gelmiştir. Genetik algoritmalar bugünkü biçimi ilk olarak Michigan Üniversitesi'nden, psikolog ve bilgisayar bilimleri uzmanı J.H. Holland tarafından 1975 yılında ortaya konulmuştur. Genetik Algoritmalar, en iyinin korunumu ve doğal seçim ilkesinin benzetim yoluyla bilgisayarlara uygulanmasıyla elde edilen bir arama yöntemidir. Standart bir GA'da, aday sonuçlar eşit boyutlu vektörler olarak ifade edilir. Başlangıçta, bu vektörlerden bir grup, rastlantısal olarak seçilerek belirli bir büyüklükte bir popülasyon (toplum) oluşturulur. Kromozom adı verilen bu vektörler, yeni nesiller oluşturarak değişikliklere uğrar. Bir kromozomun üzerindeki genler, n boyutlu vektörlerin bir boyutuna karşılık gelmektedir.

Her yeni nesilde kromozomların iyiliği ölçülür, yani her vektör(kromozom), amaç fonksiyonuna yerleştirilerek vermiş olduğu sonuç hesaplanır. Bir sonraki nesil oluşturulurken, bazı kromozomlar yeniden üretilir, çaprazlanır ve mutasyona uğratılır. Bu işlemler için bazı özel tip operatörler geliştirilmiştir.

Tekrarlama, çaprazlama ve mutasyon gibi operatörlerden geçecek olan kromozomlar, amaç fonksiyonu değerlerine göre rastlantısal olarak seçilir. Örneğin amaç fonksiyonu değeri daha iyi olan kromozomun çaprazlanma yada tekrar üretilme olasılığı, değeri kötü olana göre daha fazladır. Benzer kromozomlarla çeşitliliği azalan popülasyonlarda yine rastlantısal olarak seçilen kromozomlar mutasyona uğratılırlar. Bunun yanı sıra, popülasyon içinde iyiliği düşük olan kromozomların bir sonraki nesilde yer almamaları mümkündür, yani yok olurlar.

Standart bir “Genetik Algoritma Yordamı” aşağıdaki gibi verilebilir:

1. Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur(çözüm grubu, biyolojide ki benzerliği nedeniyle, toplum(populasyon), çözümlerin kodları da kromozom olarak adlandırılır). Toplumda bulunacak birey sayısı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Toplum bu işlemden sonra rasgele oluşturulur. Birey sayısı belirlendikten sonra probleme bağlı olarak kromozomların kodlanması gerekmektedir. Değişik kodlama biçimleri vardır.
2. Toplumda her kromozomun ne kadar iyi olduğu bulunur. Bu amaçla kullanılan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyon yardımıyla kromozomların uygunluklarının bulunmasına ise evrimleşme adı verilir. Uygunluk fonksiyonu genetik algoritmanın beynini oluşturmaktadır. Genetik algoritmada probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek onların bir bakıma şifresini çözmekte ve sonra bu parametrelere göre hesaplamayı yaparak kromozomların uygunluğunu bulmaktadır. Çoğu zaman genetik algoritmanın başarısı bu fonksiyonun verimli ve hassas olmasına bağlıdır.
3. Seçilen kromozomları eşleyerek yeniden kopyalama ve değiştirme operatörleri uygulanır. Sonuçta yeni bir toplum oluşturulur. Bu eşleme kromozomların uygunluk değerlerine göre yapılır. Bu seçimi yapmak için rulet tekerleği seçimi, turnuva seçimi gibi seçme yöntemleri vardır. Yeniden kopyalama genlerdeki genetik bilginin birinden diğerine geçmesi işlemine benzediği için çaprazlama olarak adlandırılır. Bu işlem, toplumda çeşitliliği sağlar. Değiştirmenin etkisi, yalnızca bir çözüm üzerinde olmaktadır.
4. Yeni kromozomlara yer açmak için eski kromozomlar çıkartılarak sabit büyüklükte bir toplum sağlanır.
5. Tüm kromozomların uygunlukları tekrar hesaplanır ve yeni toplumun başarısı bulunur.
6. İşlemler tekrarlanarak verilmiş zaman içerisinde daha iyi olan yeni nesillerin oluşturulması gerçekleştirilir.

7. Sonuçta toplumların hesaplanması sırasında en iyi bireyler bulunduğu çözüm elde edilmiş olur.

Burada karşılaşılan ilk sorun; kromozomların oluşturulması ve şifreleme türünün seçimi ile ilgilidir. Bunun için, şifreleme işlemi ile birlikte çaprazlama ve mutasyon gibi GA'nın temel operatörlerinin açıklanması gerekecektir.

Diğer soru ise, çaprazlama için ebeveynlerin nasıl seçileceğidir. Bu birçok şekilde yapılabilir ama ana düşünce, daha iyi olan ebeveynlerin seçilmesidir. Böylece daha iyi döl üretilir. Yeni dölle oluşturulan popülasyonda, son popülasyondaki en iyi kromozomun kaybolacağı da düşünülebilir. Bunun önlenmesi için seçicilik kullanılır. Bu durumda, en azından iyi bir sonuç hiçbir değişiklik olmadan yeni popülasyona kopyalanır, böylece bulunan en iyi sonuç korunmuş olur.

II.4.2 Bulanık Mantık

Günümüzdeki bir çok olay belirsiz koşullarda gerçekleşmektedir. Beklenmedik olaylar ortaya çıkmakta ve karar vermeyi etkilemektedir. Seyahate gitmek isteyen bir kişinin yağmur yağması ile planlarını değiştirmesi gibi belirsiz olaylar insanların kararlarını etkilemektedir. Kesin olarak bilinemeyen olaylar hakkında karar vermek için uzmanlar normal, yüksek, düşük, yaklaşık gibi kavramları kullanmaktadırlar.

Hava sıcaklığının belirsiz olması durumunda sıcaklık 20 derece olunca şu işi yap demek yerine sıcaklık normal olunca şu işi yap denilmektedir. Bulanık mantık bu gibi göreceli durumlar için bilgisayarın karar verebilmesini sağlamak amacıyla geliştirilmiş yöntemdir. “Normal” gibi kavramların bilgisayar tarafından yorumlanmasını sağlar[3].

II.4.3 Uzman Sistemler

Uzman sistemler (US), belirli bir alanda sadece o alan ile ilgili bilgilerle donatılmış ve problemlere o alanda uzman bir kişinin getirdiği şekilde çözümler getirebilen sistemlerdir. İyi tasarlanmış sistemler belirli problemlerin çözümünde

uzman insanların düşünme metodolojilerini taklit ederler. Sistemin bir veya daha fazla uzman bilgilerini barındırmasından dolayı uzman sistem adını almıştır.

Uzman sistemler alanında DENDRAL ile başlayan çalışmalardan sonra, 1976 yılında Stanford Üniversitesinde Fiegenbaum ve arkadaşları tarafından geliştirilen ilk uzman sistem olan MYCIN, bakteriyolojik ve menenjitik hastalıkların tedavisine yönelik kullanılan bir sistemdir. Bu sistemde; hastanın geçmiş bilgileri, laboratuvar sonuçları, semptomlarının sorgulanması ve bu bilgilerin derlenmesi ile teşhis konulabilmiş, reçete yazımı ve tedavi süreçlerinin belirlenmesi sağlanmıştır. MYCIN'e belirtilen verilerin girilmesiyle, niçin ve nasıl soruları sistem tarafından cevaplandırılmakta, sistem kullanıcı ile etkileşimli olarak çalışmaktadır.

Son zamanlarda uzman sistemlerin geliştirilmesinde US kabuklarından yararlanılmaktadır. Kabuk sistemler genellikle bir karar mekanizması, boş bir bilgi tabanı, kullanıcı ile sistem arasındaki iletişimi sağlayan bir arayüz ve bilgi mühendisinin kullanabileceği bir US geliştirme biriminden oluşur. Bilgi mühendisi yüksek seviyeli etkileşimli bir programlama dili kullanarak konunun uzmanından edindiği bilgileri bu birim aracılığıyla bilgi tabanına aktarır. LEONARDO ve KAPPA gibi US kabukları bunlara örnek olarak verilebilir. Ayrıca kullanıcının kendisinin özel çıkarım mekanizması geliştirmesine imkan veren daha gelişmiş sistemler de mevcuttur.

US'ların kullanılmasıyla, maliyet ve işleyiş hataları azaltılabilir, verimlilik artışı sağlanabilir, kalite artışı veya iyileştirmesi gerçekleştirilebilir, esneklik ve daha ucuz sistem kurulumu gerçekleştirilebilir ve tehlikeli çevrelerde iş veya işlemler yapılabilir. Bu sayede güvenilirlik arttırılır, cevap verme süresi azaltılır, tam ve kesin olmayan bilgi ile çalışılabilir, sınırlı da olsa bazı problemlerin çözümü kolaylaştırılabilir.

Bu üstünlüklerin yanında, US'ların gerçekleştirilmesinde bazı güçlükler vardır.

- İnsanlardan sisteme uygulanabilecek bilgi almak zordur ve alınan bilgiler her zaman okunabilir uygunlukta olmayabilir.
- Uzman sistemlerin çalışma alanları sınırlı olabilir.
- Sistemi oluşturacak ve kuracak uzmanların azlığı ve yüksek ücret talepleri sistemin maliyetini yükseltebilir.
- Farklı uzman bilgilerinde çelişkilerle karşılaşılabilir.

US'lar, modern bilgi sistemleri olmasına rağmen, ancak karar verme kurallarının çok açık ve bilginin güvenilir olduğu problemlerde başarıyla uygulanabilmektedir. Son yıllarda problemlerin çözümünde bulanık kümeler ve yapay sinir ağları gibi yapay zeka teknikleriyle beraber kullanılmaktadırlar. US'lar, bilgisayar destekli tasarımdan, hastalık teşhisine, arıza ve üretim hatalarının tespitinden, endüstriyel robotlara, iş ve süreç planlamadan, ekonomik analizlere, askeri uygulamalardan uzay çalışmalarına kadar bir çok alanda başarıyla kullanılmaktadır.

II.4.4 Karınca Algoritmaları

Günümüzde, biyolojik yapılardan esinlenerek oluşturulan sistemler önem kazanmakta ve yapay zeka araştırmacılarının ilgisini çekmektedir. Doğadaki bazı canlı toplulukları, sınırlı yetenekli basit bireyler tarafından oluşturulmalarına rağmen kolektif zeka davranışları sergilerler. Problemlere karşı üretilen zeki çözümler, bu bireylerin kendi içerisindeki organizasyonları ve dolaylı iletişimlerinden ortaya çıkar. Bu sistemler, günlük yapay zeka sistemlerinin geliştirilmesinde önemli tekniklerin kaynağı olmaktadır.

Karıncaların kolektif davranışının taklidi olan Karınca Algoritmaları, son zamanlarda geliştirilen popülasyon tabanlı yaklaşımlardır. İlk olarak Marco Dorigo tarafından önerilen Karınca Kolonisi Optimizasyonu (KKO), algoritmalarının temelini gerçek karınca kolonilerinin yiyecek bulmakta kullandıkları yöntem ve davranışlar oluşturmaktadır.

Karınca kolonileri, ilginç davranışlar sergileyen canlı oluşumlarıdır. Karıncalar tek başlarına basit yeteneklere sahip olmalarına rağmen, koloninin bütünü daha yüksek bir yapıdadır; kendilerinden çok büyük cisimleri taşımak, köprüler oluşturmak veya yuva ile yiyecek arasındaki en kısa yolu bulmak gibi karmaşık işlemlere karşı zeki çözümler üretirler. Tek bir karıncanın, yapılacak iş hakkında hiçbir bilgisi yoktur. Karıncanın hareketi, yerel kararlarda temel alınır ve genellikle daha önceden tahmin edilemez. Zeki davranış doğal olarak, karıncalar arasındaki organizasyon ve dolaylı iletişim sonucu ortaya çıkar. Bu sonuç genellikle Gelişen Zeka veya Gelişen Davranış olarak adlandırılır.

Karınca Kolonisi algoritmasının genel mantığını anlamamız için ilk önce karıncaların kendilerine özgü basit kararları kullanarak, yiyecek ile yuva arasındaki en kısa yolu nasıl bulduklarına değinelim. Karıncaların bu yolu nasıl buldukları ilk bakışta anlaşılması zor bir problemdir. Bu sorunun cevabı geçtikleri yolda bıraktıkları feromen izlerinde gizlidir. Her karınca yiyecek ararken geçtiği yerlere değişik miktar ve yoğunluklarda, karın bölgesinde yer alan Dufour bezlerinden salgıladığı feromen adlı özel bir sıvı bırakır. Bu koku karıncaya, yuvasına dönmesini sağladığı gibi, diğer karıncalara da yiyeceğe giden yolu göstermeye yarayan bir izdir. İlginç olaylardan biride, karıncaların yolu bulmada “yanlışıklıkla” daha büyük olanı seçtiklerinde, çok hızlı bir şekilde rotalarını yeniden yapılandırma yeteneklerinin olmasıdır. Sonuçta yol üstündeki feromen miktarı artacaktır. Karınca ancak feromen kokularını takip ederek doğru yolu bulabilmektedir. Ancak hiçbir karınca tek başına yiyecek ile yuva arasındaki en kısa yolu tek başına bulabilme özelliğine sahip olmamaktadır. Bu süreci başarabilecek kavramları veya değişik yolları değerlendirebilecek hafızaları yoktur. Ancak koloni davranışı, kolektif zeka sonucunda bu optimum çözümü düzenleyebilmektedir[4].

BÖLÜM III

YAPAY SİNİR AĞLARI

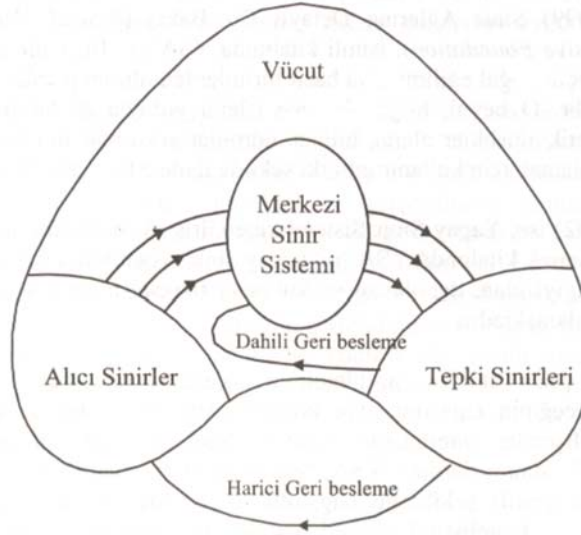
Yapay zeka problemlerinden konuşulduğunda elde edilmiş sonuçlar beynin işleyişi ile doğal olarak karşılaştırılmaktadır. Bu karşılaştırma sonucunda, yapay zeka yöntemlerinin beyin karşısında zayıf kaldığı görülmektedir. Doğal olan bu farklılık, problemlerin kendisinden kaynaklanmaktadır. Öyle ki yapay zekanın çözmesi için sunulan problemler sıradan hesaplama problemleri olmayıp; bilgisayar için zor, fakat beynin işlevine özgü olan problemlerdir. Bilgisayarlar, büyük ölçüde seçim (tarama) gerektiren hesaplama problemlerinde insandan daha hızlı ve güvenli olarak çalışmaktadır. Bilgisayarlar, doğal dili anlayıp bir sonuç üretiyorsa veya satranç oynuyorsa, bu, insan davranışlarının bilgisayara aktarılması değil, bilgisayarların güçlü “hesaplama yeteneğine” dayanmaktadır. Oldukça zor matematiksel hesaplama gerektiren uygulamalarda, bilgisayarın insandan her zaman üstün ve hata olasılığı çok düşük olmasına karşın, gerçek zamanlı doğal dil ve görüntü işleme gibi uygulamalarda, insanın bilgisayara karşı ezici bir üstünlüğü vardır. Bilgisayarların yapı taşı olan transistörlerin nanosaniyeler mertebesinde anahtarlamaına karşın doğal beyin nöronu milisaniyeler mertebesinde ateşlenir. O halde insanı bilgisayarlara karşı üstün kılan nedir? İnsan beyninin yapıtaşı olan nöronlar paralel çalışırlar ve sayıları yüz milyar civarındadır. Beynin nasıl çalıştığını henüz tam olarak çözülememiştir. Fakat burada beyni oluşturan milyarlarca sinir hücresinin

(nöronun) önemli rol oynadığı bilinmektedir. Son yıllarda, yeniden araştırmacıların ilgisini beynin işlevinin modellenmesi, gerçek bir taklidinin tasarlanması çekmektedir. Burada “yeniden” kelimesinin kullanılması bu konuda çalışmaların 1943 senesinde Mc Culloch-Pitts modeli ile başladıktan sonra 80’lerde teknolojinin gelişmesi ile yeniden hız kazanmasıdır. Yapay sinir ağları olarak bilinen bu modeller, canlı organizmalarda bulunan biyolojik sinir yapısından esinlenerek yapılmıştır. Fakat gerçek sinir ağlarıyla arasındaki benzerlik oldukça zayıftır. Doğal sinir ağları ile yapay sinir ağları arasında yapı ve kapasite bakımından büyük farklılıklar bulunmaktadır. Bu farklılıklara rağmen yapay sinir ağları, biyolojik öğrenmeyi temel alan sinir sistemine benzer bir yapıdan örnek alınarak oluşturulmuştur.

İnsanı diğer canlılarda ayıran en önemli özelliği; sahip olduğu öğrenme, düşünme ve yorumlama yeteneklerinin farklılığıdır. İnsan, bu yeteneklerini, temel yapı elemanı olan sinir hücrelerine sahip beynine borçludur. İnsanın akıllı makine yapma isteğinin temelinde, işte bu sinir hücrelerini modelleyebilme ve kısmen de olsa insan beyninin bazı özelliklerini makineye kazandırma yatmaktadır.

III.1 BİYOLOJİK NÖRON

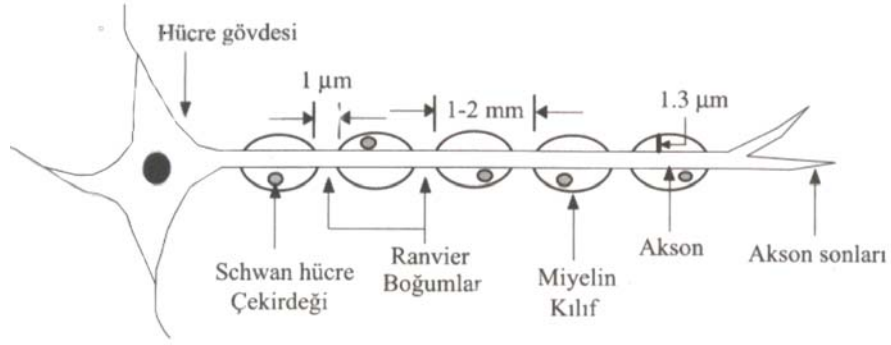
Biyolojik sinir sistemi, bilgiyi alan, yorumlayan ve uygun bir karar üreten bir merkez ve bu merkezin kontrolünde bulunan alıcı ve tepki sinirlerden oluşur. Alıcı sinirler (reseptör), iç veya dış ortamlardan algıladıkları uyarıları, elektriksel sinyallere dönüştürerek beyine iletirler. Tepki sinirleri (efektör) ise beynin ürettiği elektriksel sinyalleri çıktı olarak uygun tepkilere dönüştürür. Şekil-III-1’de biyolojik sinir sisteminin blok şeması görülmektedir. Nöron denilen sinir hücreleri, sinir sisteminin temel birimidir ve gövde(cell body), gövdeye giren sinyal alıcılar(dendrit), gövdeden çıkan sinyal ileticiler (akson) olmak üzere başlıca üç kısımdan oluşur.



Şekil-III.1 Biyolojik sinir sisteminin blok gösterimi

Dendrit, bir nöronda birden fazla bulunabilir ve iç yapıları sinir gövdesi ile aynıdır. Nörona gelen bilgiyi (başka bir nörondan veya reseptörden gelen) alan kısa uzantılardır. Akson, dendritlerden aldığı bilgiyi diğer hücrelere aktaran uzantılardır. Uzunluğu birkaç mikrondan 1m'ye kadar değişebilir. Her nöronda bir akson bulunur. Aksonlar, akson kesecikleri denilen ve içlerinde çok miktarda vezikül bulunan birçok yapı ile sonlanırlar. Bu keseciklerde(vezikül) sinaptik geçişte aktif rol alan kimyasal taşıyıcılar (nörotransmitterlar) bulunur. Aksonlar sahip oldukları örtüye göre miyelinli veya miyelinsiz olmak üzere iki kısma ayrılırlar.

Şekil III-2'da miyelinli bir nöronun yapısı verilmiştir. Miyelinli aksonlarda akson üzerini miyelin denilen lipid yapıda kalınca bir kılıf çevreler. Miyelin kılıf akson boyunca 1-2 mm'de bir düzenli aralıklarla kesintiye uğrayarak boğumlar oluşturur. Bunlara "ranvier boğumları" adı verilir. Uzunlukları çok kısa ve 1µm kadardır. Miyelinli aksonların çapları 1-25µm arasında değişir. Miyelinsiz aksonların çapları 1µm'den küçüktür. Akson üzerini örten miyelin kılıfın yalıtım ve darbe hızını arttırma gibi iki önemli görevi vardır. Miyelin kılıf, aksonu sararak çevre nöronların uyarılarından etkilenmeyi önler veya yalıtımı sağlar. Uyarının yani akson potansiyelinin akson boyunca hızlı bir şekilde iletilmesi darbe hızının arttırılması ile sağlanır.



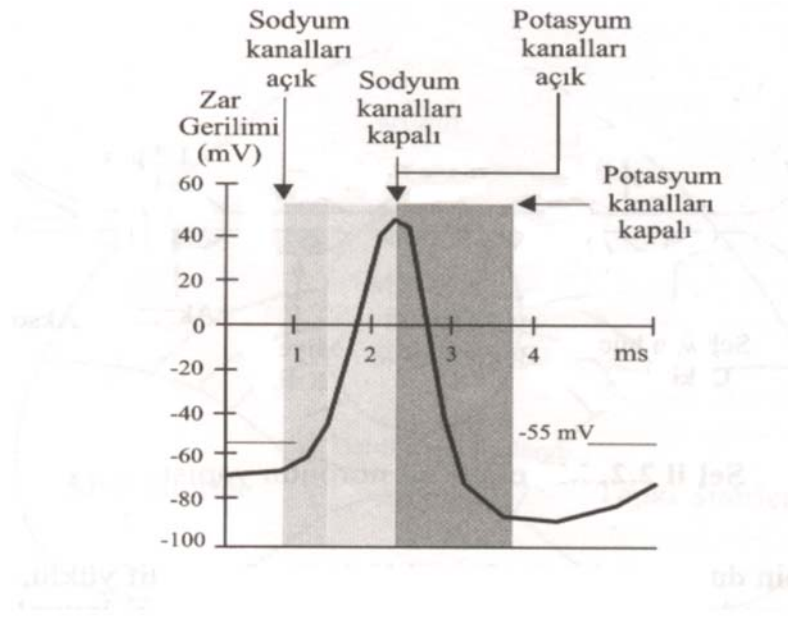
Şekil-III.2 Miyelinli bir nöronun yapısı

Sinir hücrelerinin durumu, dış tarafı pozitif, iç tarafı negatif yüklü, ortasında çift lipid tabakasından kaynaklanan iyi bir yalıtkan bulunan, kutuplanmış bir kondansatörü andırır. Uygun bir uyarıcı ile bu kondansatör boşalabilir, hatta çok kısa bir süre içerisinde ters kutuplanabilir(depolarizasyon). Zarın belirli bir bölgesinde 1 ms kadar bu potansiyel değişikliği zar boyunca yayılmasına devan eder. Akson boyunca iletilen bilginin temel birimi olan bu potansiyel değişikliğine aksiyon potansiyeli denir. Aksiyon potansiyeli yayılma hızını belirleyen faktörler kısaca:

- Birim yüzeydeki sodyum kanalları sayısı çoğaldıkça (sodyum iletkenliği) iletim hızı artmaktadır.
- Birim yüzey başına zar kapasitesi küçüldükçe hız artmaktadır. Miyelin tabakası kapasiteyi azalttığı için iletim hızını arttırıcı rol oynar.
- İletim hızı sıcaklıkla artmaktadır.
- Akson yarıçapı arttıkça iletim hızıda artmaktadır.

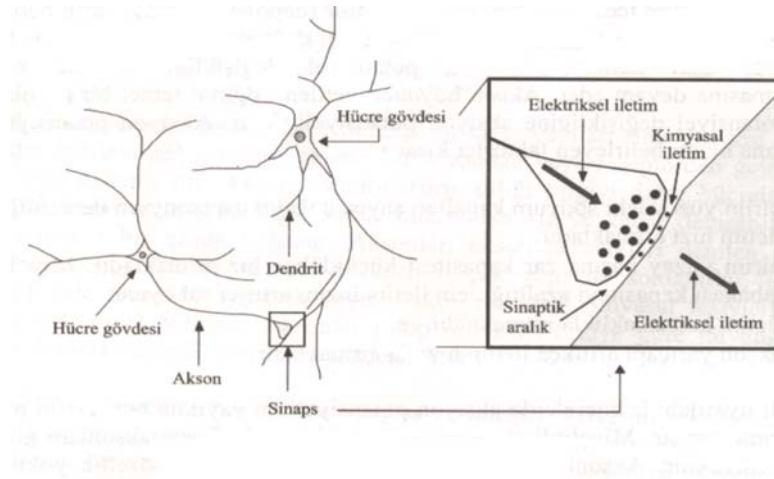
Farklı uyarılabilir hücrelerde aksiyon potansiyelinin yayılma hızı 1-100m/s arasında değişir. Miyelinli aksonlarda iletim hızı miyelinsiz aksonlara göre daha yüksektir. Aksonlarda iletimin yönünü belirleyen bir özellik yoktur. Ortasında uyarılan bir aksonda, aksiyon potansiyelleri her iki yönde de yayılabilir.

Mikroelektrotlarla yapılan ölçümler ve gözlemlerle zar potansiyelinin hücreden hücreye -60 ile -95mV arasında, aksiyon potansiyeli tepe değerinin +20mV ile +50 mV arasında, aksiyon potansiyeli süresinin ise 0.5ms ile 0.5 sn arasında Şekil III.3’de verildiği gibi değişmektedir.



Şekil-III.3 Aksiyon potansiyeli

Şekil-III.4'de bir nöron hücresinin yapısı görülmektedir. Bir nöron, akson, sinaps, dendrit ve hücre gövdesi olmak üzere 4 kısımdan oluşur. Bir nöronun (Şekil III.4) yüzlerce bazen de binlerce dendrit çıkabilir. Bunların uzunluğu genellikle bir milimetreden daha kısadır. Bazıları ise



Şekil-III.4 Biyolojik nöron

birkaç milimetre uzunluğa ulaşabilir. Dendritler çevre hücrelerden gelen sinyalleri (impulse) alıp, gövdeye ulaştırırlar. Her nöron, dendritler vasıtasıyla diğer bir çok nöronun gelen işaretleri alan ve birleştiren basit bir mikroişleme birimidir.

Sinir hücreleri arasında iletişimin gerçekleştiği, yapısal ve fonksiyonel olarak özelleşmiş bölgelere sinaps adı verilir. Mesajı gönderen ve presnaptik hücre olarak adlandırılan hücre ile mesajı alan ve postsnaptik olarak adlandırılan hücre, bu kavşak bölgelerinde birbirlerine oldukça yaklaşırlar. Uyarılabilir hücreler boyunca ve komşu hücreler arasında bilgi aktarımı sırasında, ilk mekanizmalar ne olursa olsun, sonunda ortaya çıkan elektriksel işaretler Şekil-III.3’de gösterilen işaret gibi olacaktır. Sinir hücresinin aksonu (çıkış yolu) sinaps olarak adlandırılan bir kavşak vasıtasıyla diğer nöronların dendritlerine bağlanmıştır. Tablo-III.2’de sinir hücreleriyle ilgili bazı önemli sayılar verilmiştir.

Tablo-III.1 Beyin ile ilgili bazı özellikleri

Beyin Özellikleri	Özellikler
Beyindeki ortalama sinir hücresi sayısı	100 milyar
Beyin kabuğunun tabaka sayısı	6
Beyin kabuğunun kalınlığı	1,5-4,5mm
Sinir hücresi gelişim hızı	Dk’da 250000 sinir hücresi
Dendritlerin üzerindeki sinapsların toplam yüzey alanı	222000 μm^2

Tablo-III.2 Sayılarla sinir hücreleri

Sinir Hücreleri	Özellikler
Sinir hücresinin çapı	3-18 μm
Tipik bir sinir hücresinin ortalama sinaps sayısı	1000-10000
Aksiyon potansiyelinin iletim hızı	0,6-120 m/s
Tek bir sodyum-potasyum pompasının azami taşıma hızı	200 sodyum iyonu/sn 130 potasyum iyonu/sn
Sinir hücresi zarının kalınlığı	5nm
Tipik sinaptik yarık mesafesi	10-20nm
Bir sinaptik kesecikteki sinir ileti madde molekülünün sayısı	10000-100000

Sinapslarda, temel olarak elektriksel ve kimyasal olmak üzere iki farklı türden iletim mevcuttur. Hücreden hücreye iletişimin tamamen elektriksel olarak gerçekleştiği elektriksel sinapslar, kimyasal maddeler aracılığıyla gerçekleştiği sinapslara kimyasal sinapslar denir. Morfolojik yapıları nedeniyle elektriksel sinapslara “bağlantılı sinapslar” denir. Elektriksel sinapslarda iletim iki yönlü de sürebilirken kimyasal sinapslarda iletim ancak tek yönlüdür. Yapı işlevleri daha karmaşık olan kimyasal sinapslarda presinaptik ve postsinaptik zarlar arasındaki uzaklık 20-30 nm kadardır. Presnaptik nöronun son ucunda, iletimde rol alan ve

kimyasal taşıyıcılar adı verilen küçük kesecikler bulunur. Aksiyon potansiyeli sinir ucuna ulaşınca, kimyasal taşıyıcı molekülleri bu keseciklerden sinaps aralığına bırakılırlar. Sinaps aralığını dolduran hücreler arası sıvı içinde yayılma ile ilerleyen bu moleküllerin bir kısmı postsinaptik zarda bazı iyon kanalları açabilmekte veya bazıları kapanabilmektedir. İletkenlik ve geçirgenlik değişimleri sonucu, postsinaptik zarda potansiyel değişimleri olur. Bu potansiyel değişikliği kritik değere ulaşırsa, postsinaptik zarda yayılan bir aksiyon potansiyeli de oluşabilir.

YSA'lar, insan beyninin çalışma prensibi örnek alınarak geliştirilmeye çalışılmıştır ve aralarında yapısal olarak bazı benzerlikler vardır ve bu benzerlikler Tablo-III.3'te verilmiştir.

Tablo-III.3 Bilgisayar ve beynin karşılaştırılması

İşlemler	Beyin	Bilgisayar
İşlem Sırası	Paralel ve Dağıtılmış	Seri ve Sıralı
Anahtarlama hızı	1000 pals/sn	Yaklaşık 10^{-10} sn'den az İşlemci hızına göre
Hesaplama	Düşük	Çok yüksek(işlemci hızıyla artmaktadır.)
Tanımlama	Vektör tanımada	Karmaşık hesaplama gerekir
Veri işleme	Karışık ve hatalı veriyi işler ve dağıtır.	Tam ve doğru kabul eder.
Öğrenme	Var	Programlanabilir

Biyolojik sinir sisteminde tüm aktiviteler merkezi sinir sistemi olarak adlandırılan beyin tarafından yürütülür ve yönetilir. Bir biyolojik sinir sistemi üç katmanlı bir sistemdir. İnsan beyni 10^{12} bit veriyi saklama kapasitesine sahiptir. Biyolojik sistemlerde veriler beyinde dağınık yapıda saklanır. Biyolojik sinir sistemlerinde güçlü önişlemciler söz konusudur ve bu veri derleme sırasında önemli üstünlükler sağlar. Beynin deneyim veya örneklerden öğrenebilme yeteneği vardır. Her bir beyin hücresi için veri işlemede işlem süresi mikrosaniyeler seviyesindedir. Bilgisayarların veri saklama kapasitesi teknoloji ile ilgilidir. Verilere doğrudan adresleme ile ulaşılır ve veriler yerel noktalarda saklanır. Veri işleme veya elde etmede önişlemciler kullanılabilir. Biyolojik sistemlerde işlemler paralel veri işleme yolu ile çok kısa sürede başarılabılır.

III.2 YAPAY NÖRON

YSA'lar, birbirine bağlı doğrusal ve/veya doğrusal olmayan bir çok elemandan oluşurlar. Biyolojik sinir sistemi ile yapay sinir sistemi arasındaki benzerlikler Tablo III.4'de ve istatistiksel prensiplerle YSA arasındaki terminolojik ilişkiler Tablo-III.5'de verilmiştir.

Tablo-III.4 Biyolojik sinir sistemi ile YSA'nın benzeşimleri

Biyolojik Sinir Sistemi	YSA
Nöron	İşlemci eleman
Dendrit	Toplama fonksiyonu
Hücre gövdesi	Transfer fonksiyonu
Aksonlar	Yapay nöron çıkışı
Sinapslar	Ağırlıklar

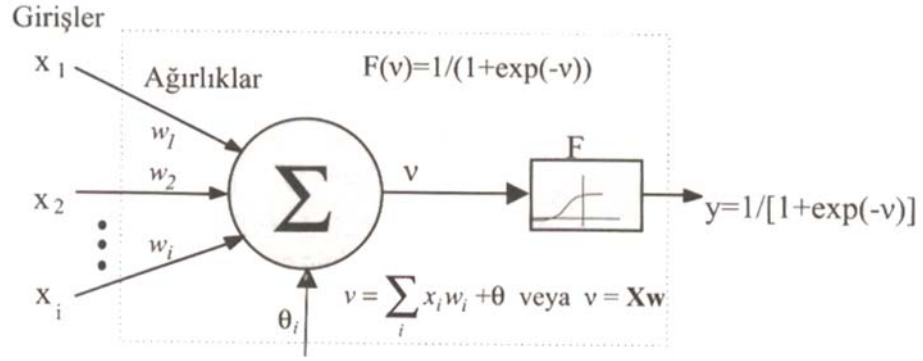
Tablo-III.5 İstatistiksel yöntemlerle yapay sinir ağların terminolojik benzeşimleri

İstatistiksel	YSA
Model	Ağ
Tahmin	Öğrenme
Regresyon	Danışmalı öğrenme
İnterpolasyon	Genelleştirme
Gözlem	Öğrenme algoritması
Parametre	Ağ parametreleri(ağırlıklar, katsayılar)
Bağımsız değişken	Giriş verileri
Bağımlı değişken	Çıkış verileri
Sınır regresyonu	Ağırlık budama işlemi

Bir yapay nöron temel olarak girişler, ağırlıklar, toplama fonksiyonu transfer fonksiyonu ve çıkış olmak üzere beş kısımdan oluşmaktadır. p girişli n çıkışlı bir yapay nöronun farklı gösterimi Şekil III-5'de verilmiştir. Bu şekilde x girişleri, y çıkışı, F fonksiyonu, p nöron giriş sayısını, n çıkış sayısını, w ise ağırlıkları ifade etmektedir.

Dendrit gösteriminden, presinaptik aktiviteler, giriş işaretlerinin p elemanlı sütun vektörü olarak gösterilir.

$$x = [x_1, \dots, x_p]^T \quad (III.1)$$



Şekil-III.5 Yapay nöronun temel gösterimi

Diğer bir ifade ile giriş vektörlerinin uzayı p-boyutludur. Sinapslar, ağırlıklar olarak adlandırılan ayarlanabilir parametreler ile karakterize edilirler. Ağırlıklar, p- elemanlı satır vektörü olarak düzenlenir.

$$\mathbf{w} = [w_1, \dots, w_p] \quad (\text{III.2})$$

İşaret-akış gösteriminde, p tane ağırlığı olan bir nöron, giriş noktalarının bir katmanı şeklinde düzenlenir. Dendrit, bir toplama noktası ile yer değiştirir. Ağırlıklar, giriş ile toplama noktaları arasındaki bağlantılara karşılık gelir.

Sinapslardan ve dendritten geçen giriş işaretleri, toplam post-sinaptik aktiviteyi tanımlayan aktivasyon potansiyeli olarak toplanır.

$$v = \sum_{i=1}^p w_i x_i = \mathbf{xw} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \begin{bmatrix} w_1 & w_2 & \dots & w_p \end{bmatrix} = \mathbf{WX}^T \quad (\text{III.3})$$

Aktivasyon potansiyeli giriş işaretlerinin ve ağırlıkların lineer toplamı olarak şekillenmiştir. Yani, ağırlıklar ile giriş vektörlerinin çarpımıdır. Bu çarpım; III.3 nolu eşitlikle ifade edilebilir. Daha sonra, aktivasyon potansiyeli, çıkış işaretini üreten bir aktivasyon fonksiyonundan $F(\cdot)$ geçirilir ve çıkış değeri fonksiyon III.4'ten elde edilir.

$$y = F(v) \quad (\text{III.4})$$

Aktivasyon fonksiyonu, aktivasyon potansiyelini çıkış işaretinin standart değerleri için normalize eden bir fonksiyondur.

YSA'da girilen giriş değerlerine önce toplama fonksiyonları uygulanır ve her bir işlem elemanının çıkış değeri y 'nin hesaplanması için ilk önce;

$$v = \sum_{i=1}^n x_i w_{ij} - \theta_j \quad (\text{III.5})$$

bulunur. Burada x_i i'inci girişi, w_{ij} j'inci elemandan i'inci elemana bağlantı ağırlığını ve θ_j eşik değerini göstermektedir. Daha sonra bu çıkış değerleri sigmoid bir aktivasyon fonksiyonuna uygulanırsa, sonuçta çıkış değeri:

$$y = F(v) = \frac{1}{1 + e^{-v}} \quad (\text{III.6})$$

ile bulunur. İşlemci eleman çıkışında kullanılan sigmoid fonksiyona göre çıkış değerinin hesaplanması gösterilmiştir. Bu işlemci elemanın çıkış değeri diğer işlemci elemanlarına giriş veya çıkış değeri olabilir.

YSA'larda en çok kullanılan aktivasyon fonksiyonları aşağıda açıklanmıştır.

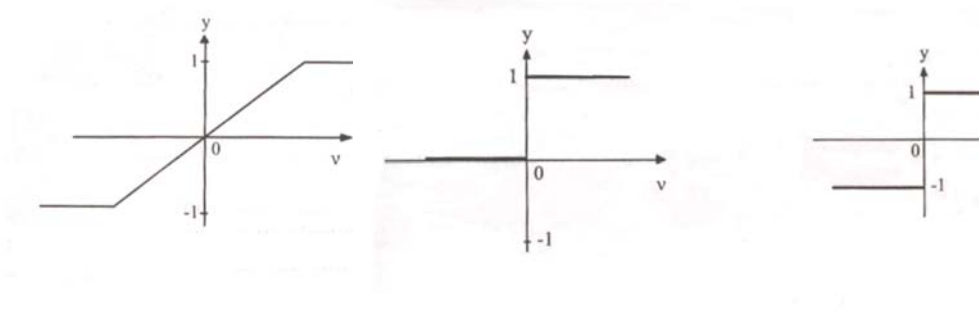
III.2.1 Aktivasyon Fonksiyonları

Transfer veya aktivasyon fonksiyonları öğrenme eğrisi olarak da adlandırılmaktadır. Aktivasyon fonksiyonları bir YSA'da nöronun çıkış genliğini, istenilen değerler arasında sınırlar. Bu değerler çoğunlukla $[0,1]$ veya $[-1,1]$ arasındadır.

YSA'larla kullanılacak olan aktivasyon fonksiyonlarının türevi alınabilir ve süreklilik göstermesi gerekmektedir. YSA'nın kullanım amacına uygun olarak tek veya çift yönlü aktivasyon fonksiyonları da kullanılabilir.

Transfer fonksiyonları olarak çoğunlukla, Şekil-III.6 veya Şekil-III.7'de verilen fonksiyonlardan birisi kullanılabilir. Uygulamalarda, en çok tanjant, hiperbolik veya sigmoid fonksiyonu kullanılmaktadır.

Lineer veya doğrusal olmayan transfer fonksiyonlarının YSA’larda kullanılması karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır. En çok kullanılan aktivasyon fonksiyonları aşağıda açıklanmıştır.



a) Doğrusal Fonksiyon

b) Basamak Fonksiyonları

Şekil-III.6 Doğrusal ve Basamak Fonksiyonları

YSA’ların çıkış katmanında kullanılan doğrusal fonksiyon Şekil-III.6.a’da verilmiştir ve işlemci elemanın girişini doğrudan işlemci elemanın çıkışı olarak verir. Eşitlik III.7’de fonksiyonun matematiksel ifadesi verilmiştir.

$$y=Av \quad (III.7)$$

burada, A sabit bir katsayıdır.

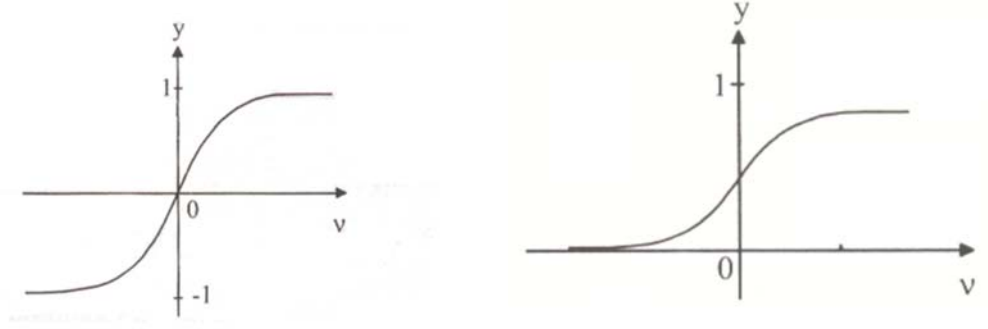
Basamak fonksiyonu tek veya çift kutuplu olabilir. Bu fonksiyonların davranışları Şekil-III.6.b’de verilmiştir. Bu fonksiyonların matematiksel ifadeleride eşitlik III.8 veya III.9’da verilmiştir.

$$y = F(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (III.8)$$

$$y = F(v) = \begin{cases} +1 & v \geq 0 \\ -1 & v < 0 \end{cases} \quad (III.9)$$

YSA uygulamalarında en çok kullanılan aktivasyon fonksiyonlarından bir diğeri de sigmoid fonksiyon ve tanjant hiperbolik fonksiyonudur. Fonksiyon

davranışı Şekil-III.7’de verilmiştir. Formülleri ise eşitlik III.10 ve III.11’de verilmiştir



a) Tanjant Hiperbolik Fonksiyon

b) Sigmoid Fonksiyon

Şekil-III.7 Sigmoid fonksiyonlar

$$y = \frac{1}{1 + e^{-v}} = \frac{1}{2}(\tanh(v/2) + 1) \quad (\text{III.10})$$

$$y = \frac{1 - e^{-2v}}{1 + e^{-2v}} = \tanh(\beta v) \quad (\text{III.11})$$

III.3. YSA MODELLERİ

III.3.1 ADALINE

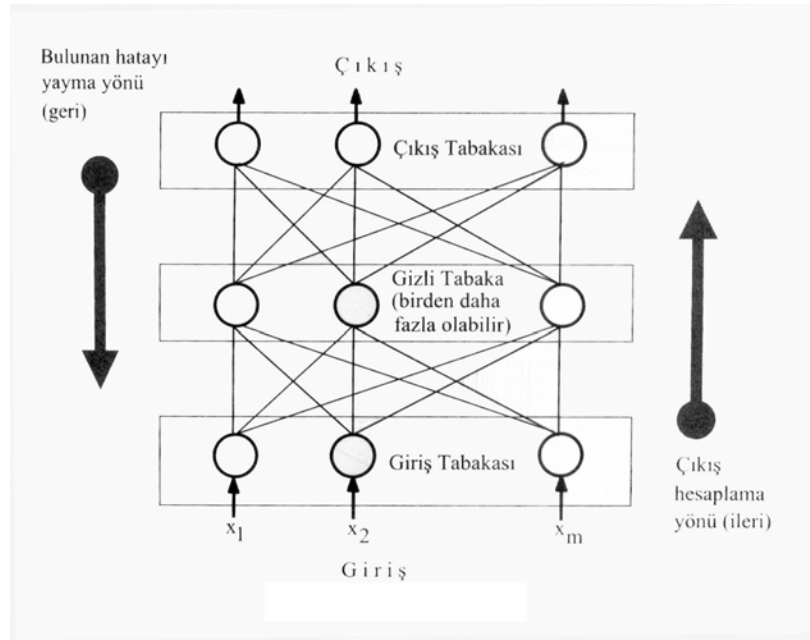
Widrow tarafından önerilen adaptif lineer eleman(adaline), her iterasyonda ortalama karesel hatayı (MSE) azaltarak ağırlıkları ayarlayan ve sınıflandırmayı sağlayan basit bir “perceptron”dur. Adaline yapısı tüm sinir ağlarının en basitidir. Giriş vektörlerini iki sınıfa ayırarak sıralama yeteneği olan basit bir işlemci elemandır. Öğrenme için danışmanlı öğrenmeyi kullanır. Adaline bir çok uygulama için oldukça iyi çalışmasına rağmen lineer problem uzayı ile sınırlıdır. Lineer transfer fonksiyonu kullanırlar. Adaline eğitim kümesindeki giriş vektörleri, lineer ayrıştırılabilir olmalıdır. Aksi takdirde hata yüzeyi parabolü en düşük noktasına ulaştığında, adaline eğitim vektörlerini asla doğru olarak sınıflandıramayacaktır. Bununla birlikte eğitme işlemi ile karışan hata yüzeyi boyunca, engel yoksa adaline kesin olarak minimum noktaya ulaşabilir. Giriş ve istenilen çıkış vektörlerinin tekrar tekrar ağı uygulanmasıyla eğitim gerçekleştirilir. Vektörlerin doğru sınıflara

ayrıştırılmasıyla, hatalar minimize edilerek öğrenme gerçekleştirilir. Eğitimden sonra, adaline girişleri kazandığı deneyime göre sınıflandırabilir.

Bu ağ yapısı ses sinyalleri üzerinde bulunan eko gürültülerinin giderilmesi için kullanılmıştır. Benzer şekilde görüntü sinyalleri üzerinde bulunan bazı gürültülerin giderilmesi içinde kullanıma uygundur.

III.3.2 Çok Katlı Ağlar (ÇKA)

Çok katmanlı bir “perseptron” sinir ağı (ÇKA) modeli Şekil-III.8’de gösterilmiştir. Bir ÇKA modeli, bir giriş, bir veya daha fazla ara veya saklı ve bir de çıkış katmanından oluşur ve her katmanda bir veya daha fazla işlemci eleman veya nöron bulunabilir. ÇKA’daki herhangi bir nöron çıkışı eşitlik III.12 ile hesaplanır.



Şekil-III.8 ÇKA yapısı

$$y_k = f\left(\sum_k w_k x\right) \quad (\text{III.12})$$

Bir katmandaki bütün işlem elemanları bir üst katmandaki bütün işlem elemanlarına bağlıdır. Bilgi akışı ileri doğru olup geri besleme yoktur. Bunun için ileri beslemeli sinir ağı modeli olarak adlandırılır. Giriş katmanında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı tamamen uygulanan problemlerin giriş sayısına bağlıdır. Ara katman sayısı ve ara katmanlardaki işlem

elemanı sayısı ise, deneme yanılma yolu ile bulunur. Çıkış katmanındaki eleman sayısı ise yine uygulanan probleme dayanarak belirlenir.

Bu ağ modeli özellikle mühendislik problemlerinin çözümlerinde ve uygulamalarda en çok kullanılan sinir ağ modeli olmuştur. Bir çok öğretim algoritmasının bu ağ eğitmede kullanılabilir olması, bu modelin yaygın kullanılmasının sebebidir.

III.3.2.1 ÇKA Ağının İşletim Aşamaları

ÇKA ağlarının çalışması şu adımları içermektedir

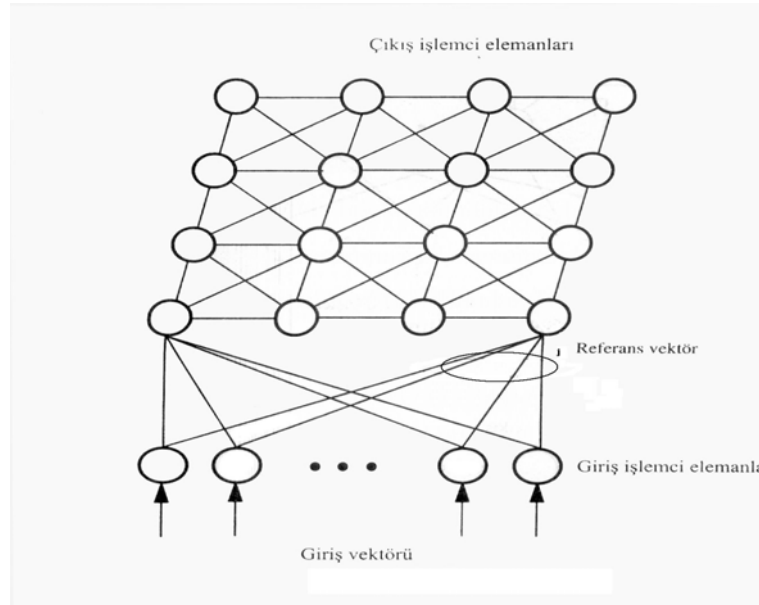
- Örneklerin toplanması: ağın çözmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağın eğitilmesi için örnekler toplandığı gibi(eğitim seti) ağın test edilmesi içinde örneklerin (test seti) toplanması gerekmektedir. Ağın eğitilmesi sırasında test seti ağa hiç gösterilmez. Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanır. Ağ olayı öğrendikten sonra test setindeki örnekler gösterilerek ağın performansı ölçülür. Hiç görmediği örnekler karşısında başarısı ağın iyi öğrenip öğrenmediğini ortaya koymaktadır.
- Ağın topolojik yapısının belirlenmesi: Öğrenilmesi istenen olay için oluşturulacak olan ağın topolojik yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her arakatmanda kaç tane işlemci elemanı ve kaç tane çıktı elemanı olması gerektiği bu adımda belirlenmektedir.
- Öğrenme parametrelerinin belirlenmesi : Ağın öğrenme katsayısı, işlemci elemanlarının toplamı ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.
- Ağırlık başlangıç değerlerinin atanması: İşlemci elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin ataması yapılır. Başlangıçta genellikle rasgele değerler atanır. Daha sonra ağ uygun değerleri öğrenme sırasında kendisini belirler.
- Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi: Ağın öğrenmeye başlaması ve yukarıda anlatılan öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler (Girdi/Çıktı) belirli bir düzeneğe göre gösterilir.
- Öğrenme sırasında ileri hesaplama yapılması: Yukarıda anlatıldığı şekilde sunulan girdi için ağın çıktı değerleri hesaplanır.

- Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması: Ağıın ürettiğı hata değeri 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.

III.3.3 Kohonen Ağı

Kohonen ağı, bir giriş tabakası ve bir de çıkış tabakası olmak üzere iki tabakadan oluşur. Bu ağı Şekil-III.9’da görülmektedir. Çıkış tabakasındaki işlemci elemanlar genellikle düzenli iki boyutlu aralıklar olarak düzenlenir. Çıkıştaki her işlemci eleman(İE), bütün giriş işlemci elemanlarına bağlıdır. Bağlantıların ağırlıkları verilen çıkış işlemci eleman ile ilgili olan referans vektörünün elemanlarını oluşturur.



Şekil-III.9 Kohonen ağı

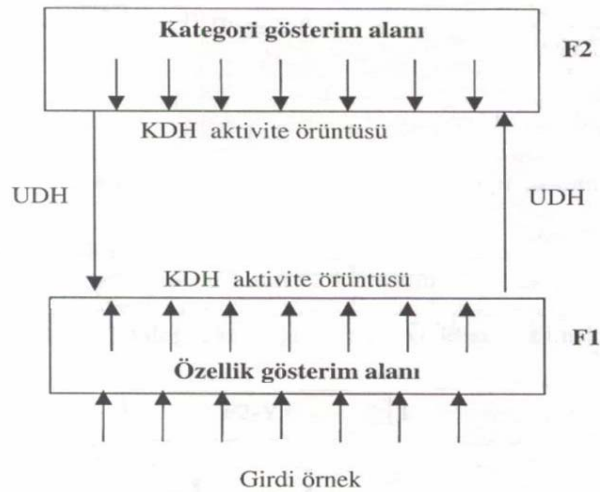
Kohonen ağıının öğrenme adımları aşağıdaki gibidir.

- i) Çıkış İE'lerin bütün referans vektörlerini küçük rasgele değerlere çek,
- ii) Bir giriş vektörünü al
- iii) Kazanan çıkış işlemci elemanını belirle
- iv) Kazanan işlemci elemanın ve onun komşularının referans vektörünü güncelleştir. Bu referans vektörleri giriş vektörüne yaklaştırılır. Bu yaklaştırma kazanan işlemci eleman için en fazla ve bu işlemci elemandan uzaklaştıkça daha azdır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda sadece kazanan işlemci elemanın referans vektörü ayarlanmaktadır.

İyi öğrenmiş bir kohonen ağıda birbirine yakın çıkış işlemci elemanlarının referans vektörleri vardır. Öğrenmeden sonra bir etiketleme işlemine başlanır. Bilinen sınıfların giriş vektörleri tarafından aktif hale gelen çıkış işlemci elemanlarına bu sınıf etiketleri verilir. LVQ ağıında ise bir çıkış işlemci elemanı eğer yarışmayı diğer çıkış işlemci elemanlarına karşı kazanırsa bir giriş vektörü tarafından aktif hale getirildiği hatırlatılır.

III.3.4 ART Ağlarının Yapısı

Adaptif Rezonans Teorisi (ART) ağları genellikle iki katmandan oluşmaktadır. Bu katmanlar F1 ve F2 olarak isimlendirilmiştir. F1 katmanı girdinin özelliklerini gösterirken F2 katmanı kategorileri (ayrıştırılmış sınıfları) göstermektedir. Bu iki katman birbirlerine uzun dönemli hafıza(UDH) ile bağlanmaktadır. Girdi bilgileri F1 katmanından alınır ve sınıflandırma ise F2 katmanında yapılır. Modelin genel yapısı Şekil-III.10'da verilmektedir.



Şekil-III.10 ART ağının genel yapısı

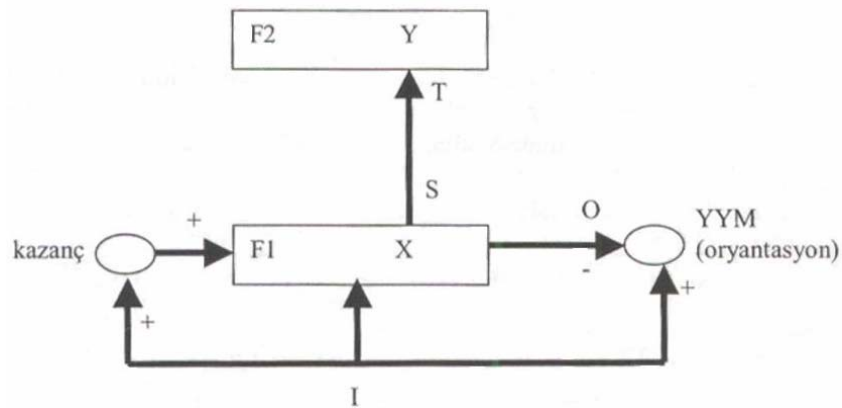
ART ağlarında girdiler direkt olarak sınıflandırılmazlar. Öncelikle girdilerin özellikleri incelenerek F1 katmanının aktivasyonu belirlenir. UDH'da ki bağlantı değerleri ile gelen bilgiler kategorilere ayrılarak F2 katmanına gönderilir. F2 katmanındaki sınıflandırma ile F1 katmanından gelen sınıflandırma birbirleri ile eşleştirilerek, eğer örnek belirlenmiş bir sınıfa uyuyorsa o kategoride gösterilir. Aksi takdirde, ya yeni bir sınıf oluşturulur veya girdinin sınıflandırılması yapılamaz.

III.3.4.1 ART Ağlarının İşletim Aşamaları

Daha önce belirtildiği gibi ART ağları F1 katmanından gelen bilgileri F2 katmanındaki kategorilere eşleştirmektedir. Bu eşleşme sağlanmaz ise yeni bir kategori oluşturulmaktadır. ART çalışması iki yönlü olmaktadır:

- Aşağıdan yukarı(F1 den F2'ye) bilgi işleme
- Yukarıdan aşağıya(F2 den F1'e) bilgi işleme

ART ağlarında aşağıdan yukarı bilgi işleme prensibi Şekil-III.11'de gösterilmiştir. Şekilden de görüldüğü gibi, bir girdi örüntüsü (I) ağa gösterilir. Bu örüntü hem F1 katmanında kısa dönemli hafıza(KDH) da X aktivite örüntüsünü oluşturur hem de oryantasyon sistemini veya diğer bir deyişle yeniden yerleştirme modülünü (YYM) aktif etmek üzere bir işaret gönderir. Benzer şekilde oluşturulan X örüntüsü hem YYM'ne bir engelleyici (inhibitory) işaret (0) göndermekte hemde F1 katmanından bir çıktı örüntüsü (S) oluşturulmaktadır. S sinyali F2 katmanına giden bir girdi örüntüsüne (T) dönüştürülür. Bu girdi örüntüsü ise F2 katmanının çıktısı olan örüntüyü (Y) oluşturur. Bu aynı zamanda ağın çıktısıdır. Bu şekilde aşağıdan yukarı bilgi işleme tamamlanmış olur.

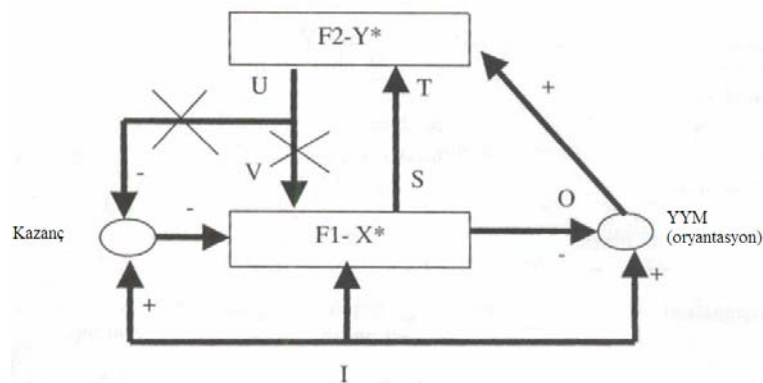


Şekil-III.11 ART ağında çıktı oluşturma süreci (aşağıdan yukarı)

The diagram illustrates a control system for a ship's heading. It consists of the following components and connections:

- Reference Input:** Y (Heading Rate) is the reference input to the outer feedback loop.
- Outer Feedback Loop:** The reference Y is compared with the heading error YYM (Oryantasyon) at a summing junction. The resulting error signal is multiplied by the gain $F2$ and then by the ship's dynamics block X^* to produce the heading rate YYM' .
- Inner Feedback Loop:** The heading rate YYM' is compared with the heading error YYM at another summing junction. The resulting error signal is multiplied by the gain $F1$ and then by the ship's dynamics block X^* to produce the heading error YYM .
- Ship Dynamics:** The ship's dynamics are represented by the block X^* , which takes the heading rate YYM' as input and produces the heading error YYM as output.
- Disturbance:** A disturbance I (Heading Error) is applied to the heading error YYM at a summing junction.

Oluşturulan X^* sinyali oryantasyon sisteminde O işaretin engelleyici etkisini azaltarak YYM'nün(oryantasyon modülünün) F2 katmanına bir sinyal göndermesini sağlar. Bu işaret F2 katmanında Y^* örüntüsünü oluşturur. Böylece, Şekil-III.13'te gösterildiği gibi girilen 1 örüntüsü için doğru sınıfı gösteren Y^* çıktısı üretilmektedir.

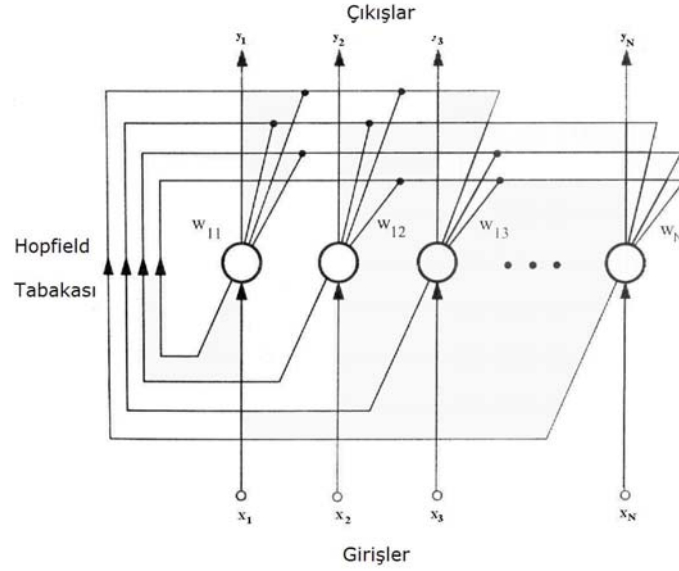


30

Eğer üretilen V şablon örüntüsü ile girdi örüntüsü eşleşir ise o zaman sadece yukarıdan aşağı o girdinin sınıfını gösteren ağırlıklar değiştirilir. Bu değiştirme öğrenme kuralına göre gerçekleştirilir. Her ART modelinin öğrenme kuralı ayırdır.

III.3.5 Hopfield Ağı

Şekil III.14’de Hopfield ağıının bir şekli gösterilmiştir. Bu ağ genellikle ikili(0 veya 1) ve bipolar(+1 veya -1) girişler kabul eder. Tek tabaka işlemci elemanları vardır ve her işlemci eleman bir diğerine bağlanmıştır. Bu da daha önceden belirtilen geri beslemeli yapıdır. Hopfield ağıının eğitilmesi sadece bir adım alır ve ω_{ij} ağırlıkları aşağıdaki gibi hesaplanır.



Şekil-III.14 Hopfield ağı

$$\omega_{ij} = \begin{cases} \frac{1}{N} \sum_{c=1}^P x_i^c x_j^c, & i \neq j \\ 0, & i = j \end{cases} \quad (\text{III.13})$$

Burada w_{ij} , i işlemci elemanından j işlemci elemanına olan bağlantının ağırlığını, x_i^c , c sınıfı için eğitime giriş vektörünün i’inci elemanını, P sınıf sayısını ve N de işlemci eleman sayısını gösterir.

Eşitlik-III.13’de $w_{ij} = w_{ji}$ ve $w_{ii}=0$ durumları ağıın kararlılığını garantiler. Bilinmeyen bir bilgi ağı girildiğinde, ağıın çıkışları bilinmeyen bir vektör elemanına eşitlenir, yani;

$$y_i(0) = x_i \quad 1 \leq i \leq N \quad (\text{III.14})$$

bu başlangıç değerleri ile başlayarak Hopfield ağı bir sonraki denklemini kullanarak minimum enerji durumuna geçmek için döngüye girer.

$$y_i(k+1) = f\left[\sum_{j=1}^N w_{ij}y_j(k)\right] \quad 1 < i \leq N \quad (\text{III.15})$$

Burada f aşağıdaki gibi tanımlanan keskin sınırlama fonksiyonudur.

$$f(x) = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \end{cases} \quad (\text{III.16})$$

III.4 YSA ÖĞRENME KURALLARI

Bir çok araştırmacı ilgi ve çalışma alanlarına göre farklı öğrenme kuralları geliştirmiştir.

III.4.1 Temel Öğrenme Kuralları

Literatürde kullanılan bir çok öğrenme algoritması mevcuttur. Bu öğrenme algoritmalarının çoğunluğu matematik tabanlı olup ağırlıkların güncelleştirilmesi amacıyla geliştirilmişlerdir. Bu algoritmaların bir çoğu Hebb kuralından türetilmiştir.

Öğrenme işlemi çok parametrelili, karmaşık ve matematiksel olarak ifade edilmesi zor bir işlemdir. Bugün kullanılan öğrenme kuralları bu işlemin basitleştirilmiş veya farklı formatlarda ifade edilmiş biçimleridir. Literatürde mevcut olan öğrenme algoritmalarının bir çoğu aslında Hebb, delta, Kohonen ve Hopfield olmak üzere dört farklı öğrenme kuralından esinlenilerek geliştirilmiştir. Bu temel kurallar aşağıda açıklanmıştır.

III.4.1.1 Hebb Kuralı

Hebb'in 1949 yılında yayınlamış olduğu Davranış Organizasyonu isimli kitabında bu öğrenme kuralı açıklanmıştır. Bu kuralın temelinde “bir nöron diğer bir nörondan giriş alıyorsa ve her iki nöronda aktif ise (matematiksel olarak aynı işarete sahip ise), nöronlar arasındaki ağırlıklar kuvvetlendirilir”.

III.4.1.2 Hopfield Kuralı

Bu kural zayıflatma veya kuvvetlendirme büyüklüğü dışında Hebb kuralına benzerdir. Eğer istenilen çıkış ve girişin her ikisi aktif veya her ikisinde aktif değilse, öğrenme oranı tarafından bağlantı ağırlığı artırılır, diğer durumlarda ise azaltılır. Birçok öğrenme algoritmasında, öğrenme katsayısı, oranı veya sabiti vardır. Genellikle bu terim 0 ile 1 arasında değerler almaktadır.

III.4.1.3 Delta Kuralı

Delta kuralı Hebb kuralının değişik bir formudur ve en çok kullanılan öğrenme algoritmalarından biridir. Bu kural, nöronun gerçek çıkışı ile istenilen çıkış değerleri arasındaki farkı azaltan, giriş bağlantılarını güçlendiren ve sürekli olarak değiştiren bir düşünceye dayanmaktadır. Bu kural, ortalama karesel hatayı bağlantı ağırlık değerlerinin değiştirilmesiyle (azaltıp veya arttırma) düşürme prensibine dayanır. Hata, aynı anda bir katmandan bir önceki katmanlara geri yayılarak azaltılır. Ağın hatalarının düşürülmesi işlemi, çıkış katmanından giriş katmanına ulaşınca kadar devam eder. Bu kural, geri yayılım, Widrow-Hoff veya en küçük ortalama karesel öğrenme kuralı olarak da bilinir.

III.4.1.4. Kohonen Öğrenme Kuralı

Bu kural, biyolojik sistemlerdeki öğrenmeden esinlenen Kohonen tarafından geliştirilmiştir. Rekabetçi öğrenme yöntemi olarak danışmansız öğrenme temeline dayanır. Bu kuralda, nöronlar öğrenmek için yarışır, kazanan nöronun ağırlıkları güncellenir. Bu kural “kazanan tamamını alır” olarakta bilinir. En büyük çıkışa sahip işlemci nöron kazanır, bu nöron komşularını uyarma ve yasaklama kapasitesine sahiptir. Kohonen kuralı hedef çıkışa gerek duymadığı için danışmansız bir öğrenme metodudur.

III.4.2 Öğrenme Kurallarının Sınıflandırılması

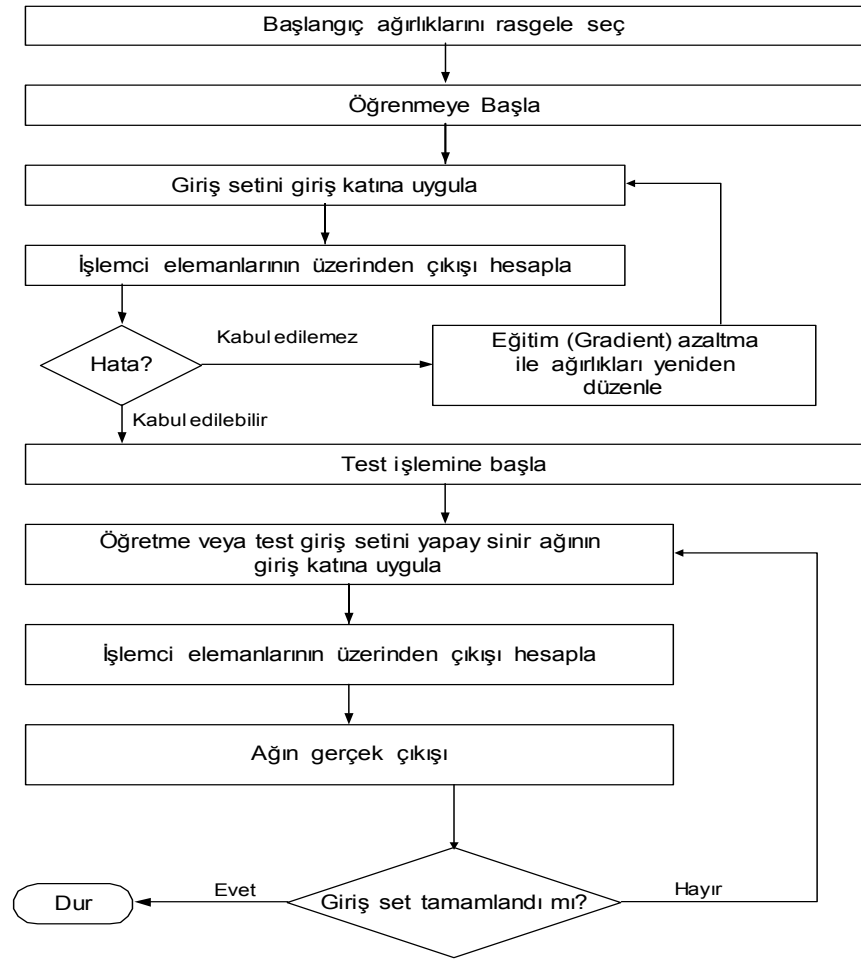
Gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği öğrenme olarak tanımlanmaktadır. Bir takım metot, kural, metodoloji, algoritma, yaklaşım veya gözlem ile bir ağın ağırlıklarının bir probleme göre değiştirilmesiyle öğrenme sağlanır. Öğrenme metodolojisine göre algoritmalar sınıflandırılmıştır. Genel olarak üç öğrenme metodu ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme yaklaşımları danışmanlı, danışmansız ve takviyeli olmak üzere üç farklı gruba ayrılabilir.

III.4.2.1 Danışmanlı Öğrenme

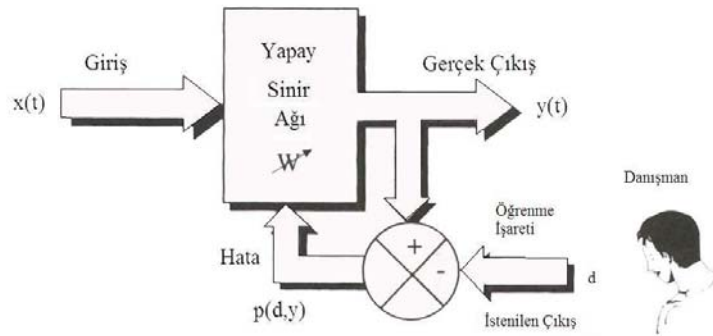
Bu tip öğrenme aşamasında YSA'ya ne öğrenmesi gerektiği örnek bir çıkışla bildirilir. İstenilen veya arzu edilen çıkış ile gerçek çıkış(ağ çıkışı) arasındaki farka göre, nöronlar arası bağlantıların ağırlığı, en uygun çıkışı elde etmek için bir öğrenme algoritmasıyla düzenlenir.

Bu sebeple danışmanlı öğrenme algoritmasının bir öğretmene veya danışmana ihtiyacı vardır. Danışmanlı bir öğrenme algoritmasının daha iyi anlaşılması için, ağın öğrenme aşamasının akış şeması Şekil-III.15'de ve danışmanlı öğrenme yapısı Şekil-III.16'da görülmektedir. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geriyayılım algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.

Bu tip öğrenme yaklaşımında mutlaka bir danışmana veya YSA'ya ne öğrenmesi gerektiğini aktaracak bir yaklaşıma ihtiyaç vardır. Çok Katmanlı Ağlar danışmanlı öğrenme kullanan ağlara örnek gösterilebilir.



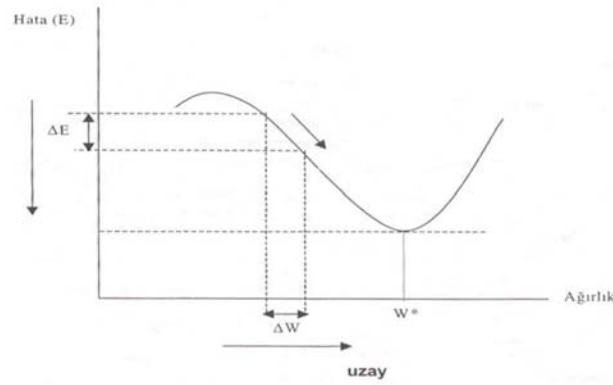
Şekil-III.15 Bir ağda geri yayımlı bir öğrenme algoritmasının uygulanması



Şekil-III.16 Danışmanlı öğrenme yapısı

ÇKA ağlarının eğitilmesi felsefesi diğer ağlarınkinden farklı değildir. Ağın kendisine gösterilen girdi örneği için beklenen çıktıyı üretmesini sağlayacak ağırlık değerleri bulunmaktadır. Başlangıçta bu değerler rasgele atanmakta ve ağa örnekleri gösterdikçe ağın ağırlıkları değiştirilerek zaman içerisinde istenen değerlere ulaşması

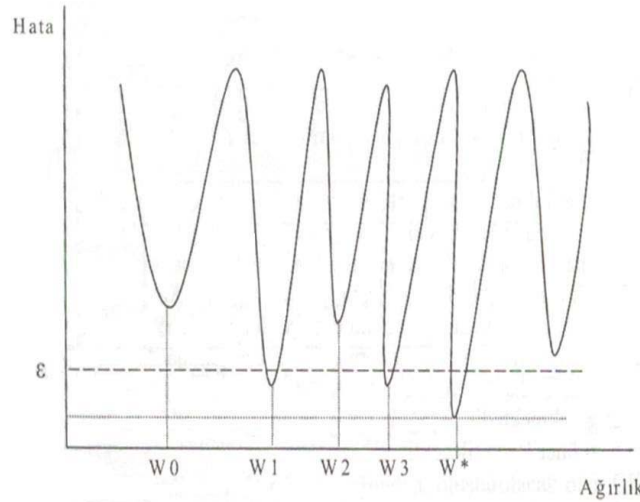
sağlanmaktadır. İstenen ağırlık değerlerinin ne olduğu bilinmemektedir. Bu nedenle yapay sinir ağlarının davranışlarını yorumlamak ve açıklamak mümkün olmamaktadır. Zaten bu ağların diğer yapay zeka tekniklerinden mesala uzman sistemlerden, ayıran en önemli farkı da davranışlarını açıklayamamasıdır. Bunun nedeni bilginin ağ üzerine dağıtılmış olması ve ağırlık değerlerinin kendi başlarına herhangi bir anlam göstermemeleridir. Ağ ile ilgili bilinen konu problem uzayında en az hata verebilecek ağırlık değerlerinin bulunmasıdır. Hatanın en az değeri kavramını anlatabilmek için basit bir problem düşünülürse, ağın öğrenmesi istenen olayın(problem uzayının) Şekil-III.17’de gösterildiği gibi bir hata uzayının olduğu varsayılın. Şekildeki W^* en az hatanın olduğu ağırlık vektörünü göstermektedir.



Şekil-III.17 Öğrenmenin hata uzayında gösterimi

Ağın W^* değerine ulaşması istenmektedir. Bu ağırlık değeri problem için hatanın en az olduğu noktadır. O nedenle her iterasyonda ΔW kadar değişim yaparak hata düzeyinde ΔE kadar bir hatanın düşmesi sağlanmaktadır. Burada bir noktaya dikkatleri çekmek gerekir. Problemin hata düzeyi her zaman böyle basit ve iki boyutlu olmayacaktır. Şekil-III.18 daha karmaşık bir hata düzeyini göstermektedir. Görüldüğü gibi problemin çözümü için en az hatayı veren ağırlık vektörü W^* olmasına rağmen pratikte bu hata değerini yakalamak çoğu zaman mümkün olmayabilmektedir. Bu çözüm ağın sahip olabileceği en iyi çözümdür. Fakat bu çözüm nasıl ulaşacağı konusunda elimizde bir bilgi yoktur. Ağ, eğitim sırasında kendisi bu çözümü yakalamaya çalışmaktadır. Bazen farklı bir çözüme takılabilmekte ve performansı daha iyileştirmek mümkün olamamaktadır. O nedenle kullanıcılar ağların performanslarında ϵ kadar hatayı kabul etmektedirler. Tolerans değeri altındaki her hangi bir noktada olay öğrenilmiş kabul edilmektedir. Şekildeki

W_0 ve W_1 çözümlerinin hataları kabul edilebilir hata düzeyinin üzerinde olduğundan bu W_3 çözümleri en iyi çözüm olmamalarına rağmen kabul edilebilir hata düzeyinin altında bir hataya sahiptirler. Bunlarda yerel çözümler olmalarına rağmen kabul edilebilir çözümlerdir. Görüldüğü gibi bir problem için birden fazla çözüm üretilebilmektedir. Bu nedenle yapay sinir ağlarının her zaman en iyi çözümü ürettikleri söylenemez. Kabul edilebilir bir çözüm olsa bile bunun bilinmesi zordur. Çoğu durumda bilinmesi mümkün değildir.



Şekil-III.18 Çok boyutlu hata uzayı

Şekil-III.18 aynı zamanda başka bir gerçeği daha göstermektedir. Bunun çeşitli nedenleri olabilir. Örneğin,

- Problem eğitilirken bulunan örnekler problem uzayını %100 temsil etmeyebilir.
- Oluşturulan ÇKA ağı için doğru parametreler seçilmemiş olabilir.
- Ağın ağırlıkları başlangıçta tam istenildiği şekilde belirlenmemiş olabilir.
- Ağın topolojisi yetersiz seçilmiş olabilir.

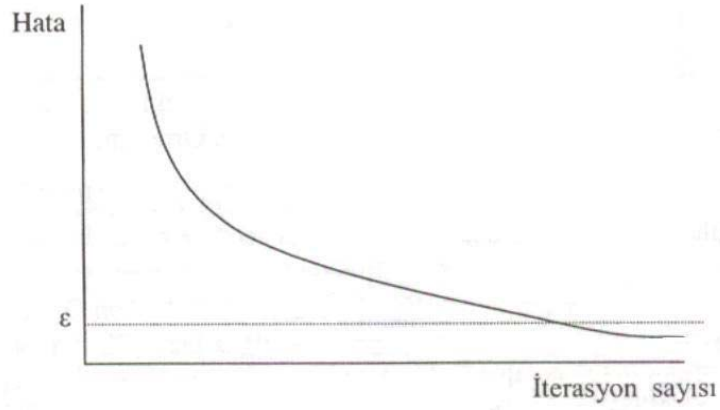
Bu ve benzeri nedenlerden dolayı ağ, eğitim sırasında, hatayı belirli bir değerin altına düşürmeyebilir. Mesala W_1 ağırlıklarını bulur hatayı daha aşağıya düşürmez. Bu aslında yerel bir çözümdür. En iyi çözüm değildir. Hata kabul edilebilir düzeye indiğinden yerel en iyi bir çözüm olarak görülebilir. Global çözümün bulunması da mümkün olabilir. Bu tamamen ağın tasarımına, örneklerin niteliğine ve eğitim sürecine bağlıdır.

Bazı durumlarda ağın takıldığı lokal sonuç kabul edilebilir hata düzeyinin üstünde kalabilir(mesala Şekil-III.18'deki W0 ağırlıklarının bulunması ve hatanın daha fazla azaltılmasının mümkün olmaması). Bu durumda ağın olayı öğrenmesi için bazı değişiklikler yapılarak yeniden eğitilmesi gerekir. Bu değişiklikler arasında şunlar sayılabilir:

- Başka başlangıç değerleri kullanılabilir.
- Topolojide değişiklikler yapılabilir(arakatman sayısını artırmak, işlemci elemanı sayısını artırmak veya azaltmak gibi)
- Parametrelerde değişiklikler yapılabilir(fonksiyonların başka seçilmesi, öğrenme ve momentum katsayılarının değiştirilmesi gibi)
- Problemin gösterimi ve örneklerin formülasyonu değiştirilerek yeni örnek seti oluşturulabilir.
- Öğrenme setindeki örneklerin sayısı arttırılabilir veya azaltılabilir.
- Öğrenme sürecinde örneklerin ağa gösterilmesi

ÇKA ağlarının yerel sonuçlara takılıp kalmaması için momentum katsayısı geliştirilmiştir. Bu katsayısının iyi kullanılması yerel çözümleri kabul edilebilir hata düzeyinin altına çekebilmektedir.

ÇKA ağlarının eğitilmesinde diğer önemli bir sorun ise öğrenme süresinin çok uzun olmasıdır. Ağırlık değerleri başlangıçta büyük değerler olması durumunda ağın lokal sonuçlara düşmesi ve bir lokal sonuçtan diğerine sıçramasına neden olmaktadır. Eğer ağırlıklar küçük aralıkta seçilirse o zaman da ağırlıkların doğru değerleri bulması uzun zamanlar almaktadır. Bazı problemlerin çözümü sadece 200 iterasyon sürerken bazıları 5-10 milyon iterasyon gerektirmektedir. Bu konuda da elimizde bilimsel bir yaklaşım yoktur. Tamamen deneme yanılma yolu ile en uygun başlama koşullarının belirlenmesi gerekmektedir. Aşağıda ÇKA ağlarını daha etkin kullanabilmek için bu kapsamda bazı öneriler verilecektir.

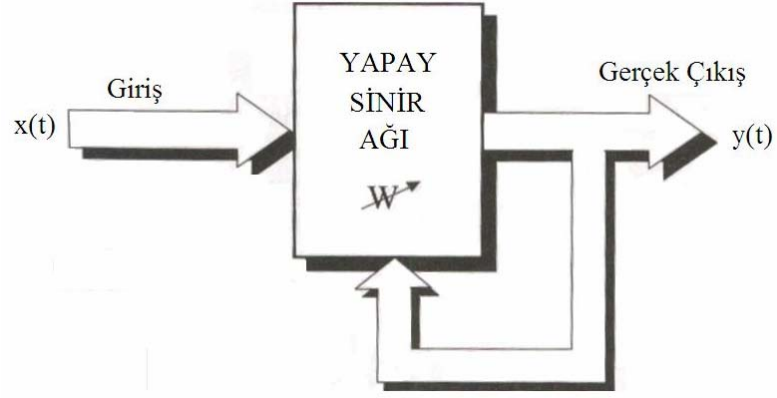


Şekil-III.19 Öğrenen bir ÇKA ağının öğrenme eğrisi

Ağın öğrenmesinin gösterilmesinin en güzel yolu hata grafiğini çizmektir. Öğrenen bir ağ için her iterasyonda oluşan hatanın grafiği çizilirse Şekil-III.19’da gösterildiği şekle benzer bir hata grafiği oluşur. Burada hatanın zaman içerisinde düştüğü görülür. Belirli bir iterasyondan sonra hatanın daha fazla azalmadığı görülür. Bu ağın öğrenmesini durdurduğu ve daha iyi bir sonuç bulunamayacağı anlamına gelir. Eğer elde edilen çözüm kabul edilemez ise o zaman ağ yerel bir çözüme takılmış demektir. Bu durumda ağın yukarıda önerilen değişiklikleri yaparak yeniden eğitilmesi gerekir.

III.4.2.2 Danışmansız Öğrenme

Giriş verilen örnekten elde edilen çıkış bilgisine göre, ağ sınıflandırma kurallarını kendi kendine geliştirmektedir. Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek yoktur. Öğrenme süresince sadece giriş bilgileri ağa uygulanır. Uygulanan girişe göre, bu giriş verileri arasındaki matematiksel ilişkilere göre bağlantı ağırlıkları ayarlanır. Aynı özellikleri gösteren vektörlerde aynı çıkışlar, farklı çıkışlarda ise yeni çıkışlar oluşturulur. Şekil-III.20’te danışmansız öğrenme yapısı gösterilmiştir. Grossberg tarafından geliştirilen ART(adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir.

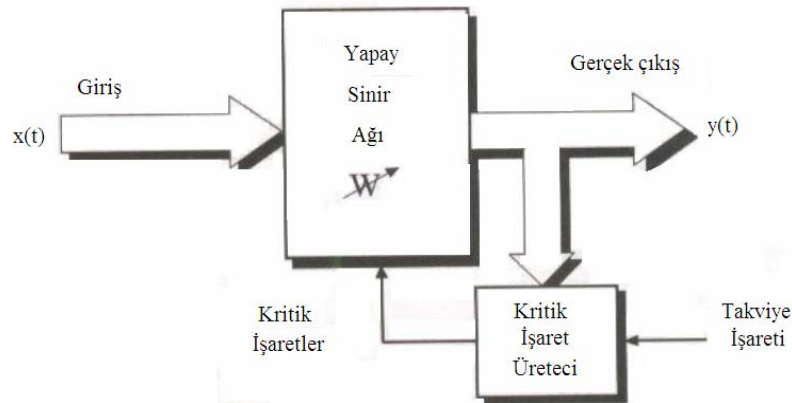


Şekil-III.20 Danışmansız öğrenme yapısı

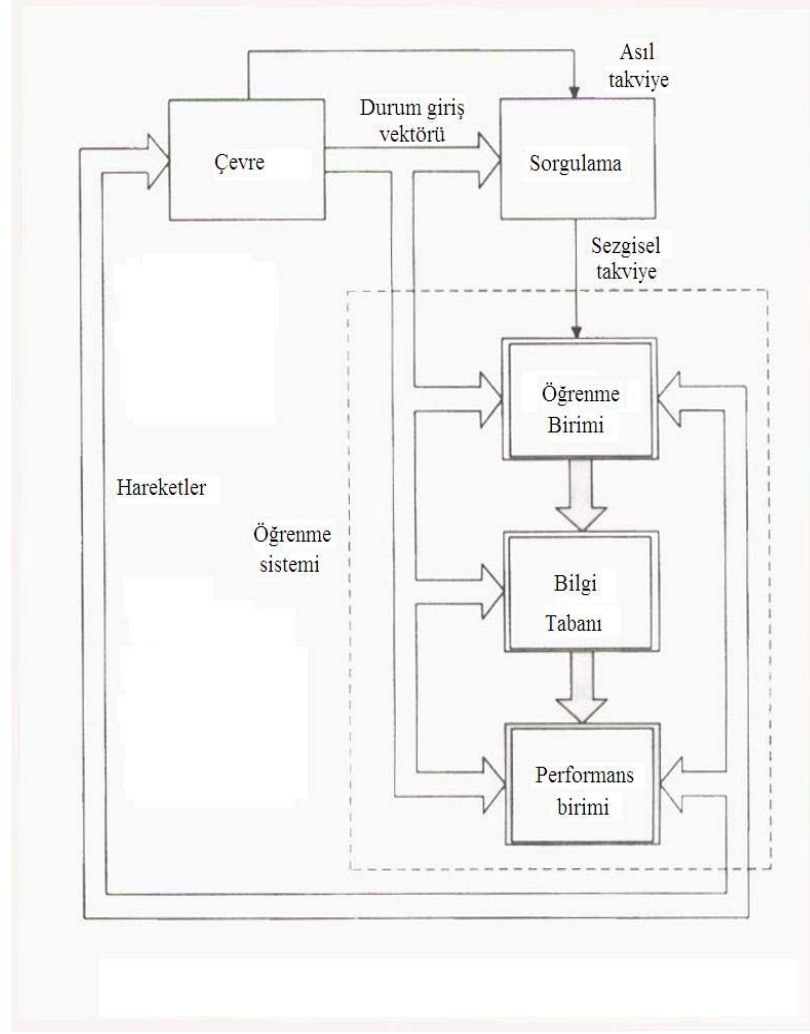
III.4.2.3 Takviyeli Öğrenme

Bu öğrenme kuralı danışmanlı öğrenme algoritmasının özel bir formudur. Şekil-III.21’de takviyeli öğrenme yapısı verilmiştir. Bu algorithmada, giriş değerlerine karşı çıkış değerlerinin bilinmesine gerek yoktur. YSA’ya bir hedef verilmemekte fakat elde edilen çıkışın verilen girişe karşılık uygunluğunu değerlendiren bir kriter kullanılmaktadır. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski’nin geliştirdiği Boltzman kuralı veya Genetik algoritma(GA) takviyeli öğrenmeye örnek olarak verilebilir.

Şekil-III.22’de takviyeli öğrenme algoritmasının blok diyagramı detaylı olarak gösterilmiştir. Bu öğrenme şekli temel olarak çevre, sorgulama ve öğrenme sistemi olan üç modülden oluşmaktadır. Öğrenme sisteminde, öğrenme birimi, bilgi veritabanı ve performans birimi olmak üzere üç modül bulunmaktadır. Takviyeli öğrenmeye örnek olarak LVQ (Linear Vektor Quantization) ağılarından bahsedilecektir.



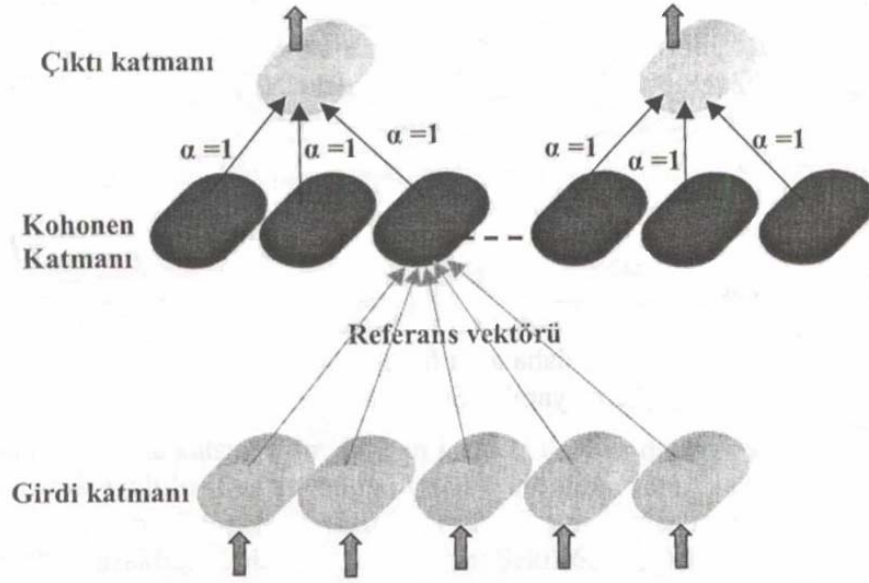
Şekil-III.21 Takviyeli öğrenme yapısı



Şekil-III.22 Takviyeli öğrenme blok diyagramı

LVQ ağınnın eğitilmesinde amaç her iterasyonda girdi vektörüne en yakın referans vektörü bulmaktır. Referans vektörleri kohonen katmanındaki işlemci elemanlarını girdi katmanındaki işlemci elemanlarına bağlayan ağırlık değerleridir. Şekil-III.23 kohonen katmanındaki 3. işlemci elemanını girdi katmanına bağlayan referans vektörü görülmektedir.

Öğrenme esnasında sadece referans vektörlerinin ağırlık değerleri değiştirilir. Değişimin ne şekilde olacağı yukarıda öğrenme kuralı anlatılırken belirtilmiştir. Şekil-III.24 aynı zamanda girdi katmanı ile çıktı katmanı arasındaki ağırlıkların(α) sabit ve 1 değerine sahip olduklarını da göstermektedir. Bu ağırlıklar eğitim sırasında da değiştirilmezler.



Şekil-III.23 Referans vektör örneği

Öğrenme esnasında girdi katmanından gösterilen bir örnek sonucu referans vektörlerinin ağırlık değerleri kullanılarak girdi vektörü ile arasındaki mesafeleri hesaplanır. Bu mesafelerden en küçük değeri hangi işlemci elemanına ait referans vektörü üretiyor ise o işlemci elemanının çıktısı 1 diğerlerinininki 0 olur. Yani kohonen katmanındaki her işlemci elemanın çıktısı ζ^k ise,

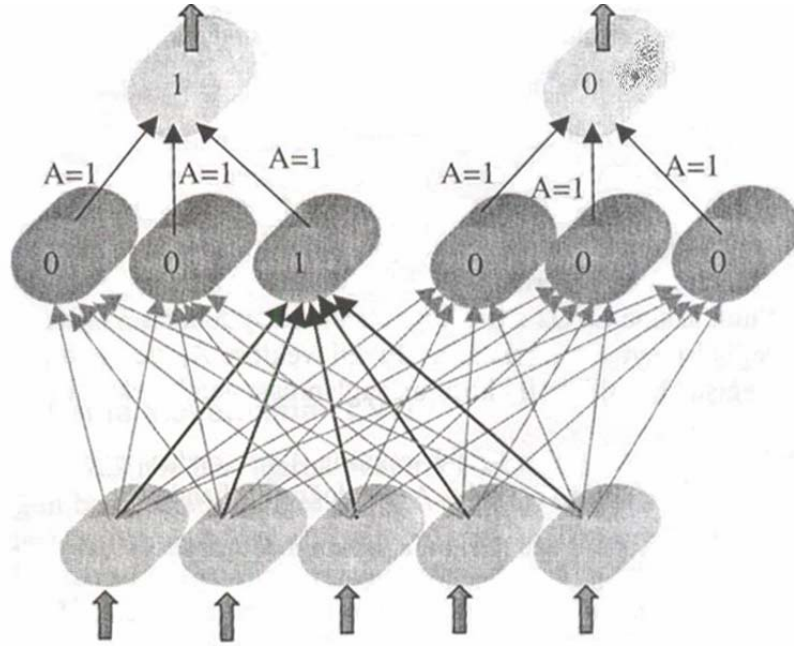
$$\zeta_i^k = \begin{cases} 1 & \text{Eğer } i. \text{ işlemci elemanı yarışı kazanırsa} \\ 0 & \text{Aksi halde} \end{cases}$$

Kohonen katmanındaki işlemci elemanlarının çıktıları bu işlemci elemanları çıktı katmanına bağlayan ağırlık değerleriyle çarpılarak ağın çıktısı hesaplanır. Yani,

$$\zeta_i = \sum_j \zeta_j^k \alpha_{ki} \quad (\text{III.17})$$

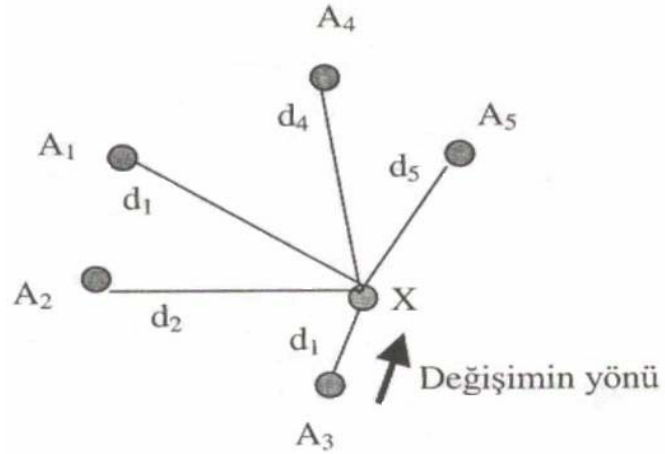
olacaktır. Bu kohonen katmanında yarışmayı kazanan işlemci elemanına bağlı olan çıktı elemanının değerinin 1, diğerlerinin değerinin 0 olması anlamına gelmektedir. Burada kohonen katmanındaki işlemci elemanlarının sadece bir tane çıktı elemanına bağlı olduğu unutulmamalıdır. Şekil-III.24'te bu durum bir örnek uzayı ile gösterilmektedir. Örnekte girdi katmanından gelen bilgileri kohonen katmanındaki 3. işlemci elemanının kazandığı varsayılmaktadır. Kazanan referans vektörü A_3 vektörü ile gösterilmektedir. Ağ 5 girdi 6 kohonen katmanı elemanı ve 2 çıktı elemanından oluşmaktadır. Kohonen katmanındaki 3. eleman çıktı katmanındaki

1. elemana bağlı olduğundan çıktı katmanında 1. eleman 1, diğerleri ise 0 değerini almaktadır. Ağın çıktıları belirlendikten sonra yapılacak iş çıktının doğru sınıflandırılıp sınıflandırılmadığını sorgulamaktır. Bu sorunun cevabına göre kohonen katındaki 3. elemanı girdi katmanına bağlayan ağırlıklar değiştirilir.



Şekil-III.24 LVQ öğrenme prosedürünün geometrik gösterimi

Bulunan sonuç eğer doğru ise o zaman Şekil-III.25’de gösterildiği gibi referans vektörü (A_3) ok yönünde girdi vektörüne (X) yaklaştırılır. Öğrenme setindeki bütün örnekler doğru sınıflandırılincaya kadar bu işlemler tekrarlanır. Hepsi doğru sınıflandırılınca öğrenme gerçekleştirilmiş olur. Bazı durumlarda girdi setinden bir iki (çok az sayıda) adet öğrenilmeyebilir. Bu örneklerin yapısından kaynaklanabilir. Bu durumda da ağ öğrenemedi denilmez. Önemli olan tasarımcının eğitim sonucunda öğrenilen sınıflandırmadan memnun olması ve test seti sınıflandırmasının kabul edilebilir olmasıdır.



Şekil-III.25 LVQ öğrenmede ağırlıkların değişim yönü

III.4.3 Uygulamaya Göre Öğrenme Yöntemlerinin Sınıflandırılması

Uygulamaya göre, öğrenme metotları on-line ve off-line olmak üzere iki gruba ayrılabilir. Uygulama problemine göre bu yaklaşımlardan birisi seçilir. Sistem bilgilerini öğrenmek ve ağırlıklarını değiştirmek için giriş verilerini kullandığında buna eğitim veya öğrenme modu denir. Eğer YSA'ya uygulanan girişlere göre ,ağın hemen cevap vermesi durumundaki moda, çalışma veya çoğu kez hatırlama(racall) modu denir.

III.4.3.1 Off-line Öğrenme

Ağların çoğu off-line çalışma modunda kullanılır. Bu yaklaşımda ağ farklı bir ortamda eğitilir. Eğitim sonucu elde edilen ağırlıklar ile ağın yapısı, daha sonra on-line uygulamalarda kullanılabilir. Bu aşamada hiçbir öğrenme algoritması kullanılmaz sadece mevcut ağ parametreleri kullanılarak ileri yönde bir hesaplama yapılarak çıkış hesaplanır.

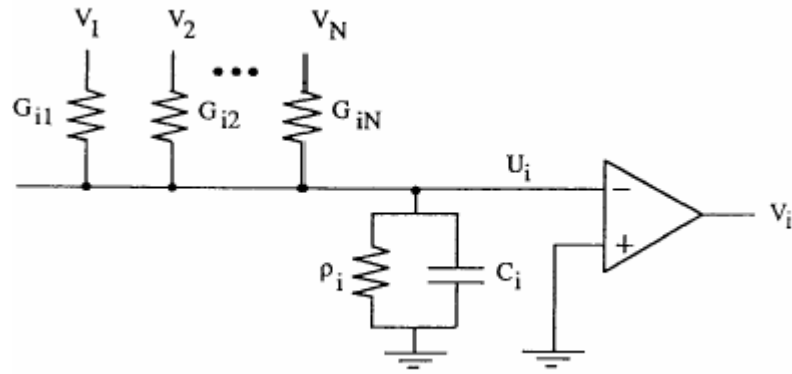
III.4.3.2 On-line Öğrenme

On-line veya gerçek zamanlı öğrenmede, sistem üzerinde bir öğrenme algoritması ve bu yapıyı destekleyen donanım ve yazılımlar mevcuttur. Bu yaklaşım daha çok farklı uygulamaların sıkça kullanıldığı ve öğretilecek sistemin devamlı olarak farklı davranışlar sergilediği uygulamalarda kullanılır. Bu ağ çalışma modunda öğrenme sürekli. Bu tip öğrenme daha karmaşık yapıların tasarımı için kullanılır ve maliyetleri yüksektir.

III.5 YAPAY SİNİR AĞI DONANIMLARI

III.5.1 Analog Olarak Oluşturulan YSA Modelleri

Basit olarak oluşturulan elektronik YSA modeli Şekil-III.26'da gösterilmektedir. Şekilde kullanılan yükselteç hücre gövdesinin işlevini yaparken dirençler ise sinapsların görevini yerine getirmektedir. Temel elektronik nöronda iki eleman vardır; sinaptik bağlantılar(dirençler) ve sinyal işleme elemanı(İE)(yükselteç). Diğer nöronlardan alınan sinyal (gerilim) dirençler vasıtasıyla toplanır. Her bir sinaptik ağırlıklar modelde iletkenlik(G) olarak temsil edilir. Devre girişlerine gelen nöral voltajlara bağlı olarak İE transfer fonksiyonuna göre bir V çıkış sinyali üretir ve üretilen V çıkış sinyali diğer nöronlara iletilir. [5]



U_i : i.ninci yükseltecin giriş voltajı

V_i : i'ninci yükseltecin çıkış voltajı

C_i : i'ninci yükseltecin giriş kapasitansı

ρ_i : i'ninci yükseltecin giriş direnci

G_{ij} : i'ninci ve j'ninci yükselteçler arasında bulunan iletkenlik

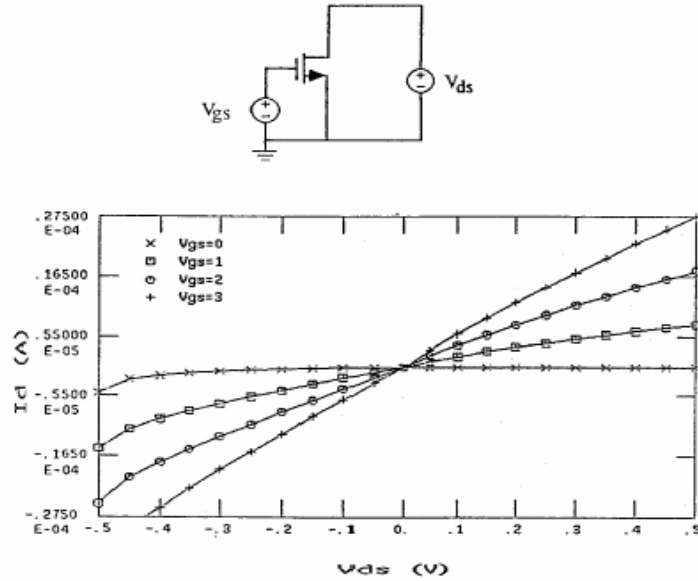
Şekil-III.26 Basit elektronik YSA modeli

YSA'ların en önemli yanlarından biri öğrenme kapasiteleridir. Öğrenme algoritmaları hassas ağırlıklarla çalışmaktadırlar bu nedenle hassas, kesin sonuç veren analog devrelere ihtiyaçları vardır. Analog devrelerde bu hassasiyeti sağlamak için daha fazla silikon alan kullanılmak zorundadır. Dolayısıyla analog devreler yüksek çözünürlük gerektirmeyen hata toleransı fazla uygulamalarda kullanılması elverişlidir.

III.5.1.1 Ağırlıkların Yarı İletken Devre Elemanlarıyla Gerçekleştirilmesi

Daha öncede bahsedildiği gibi sinaptik ağırlıklar dirençlerle gerçekleştirilebilir. Programlanamayan YSA'ları sabit dirençlere yapmak ucuz, kolay ve fabrikasyon olarak üretilmeye elverişlidirler fakat uygulama alanları çok sınırlıdır. Buna karşılık sayısal uygulamalar çok fazla silikon alan kullandığı için VLSI tekniğine uygun değildirler ayrıca DAC ve ADC devrelerine ihtiyaçları vardır.

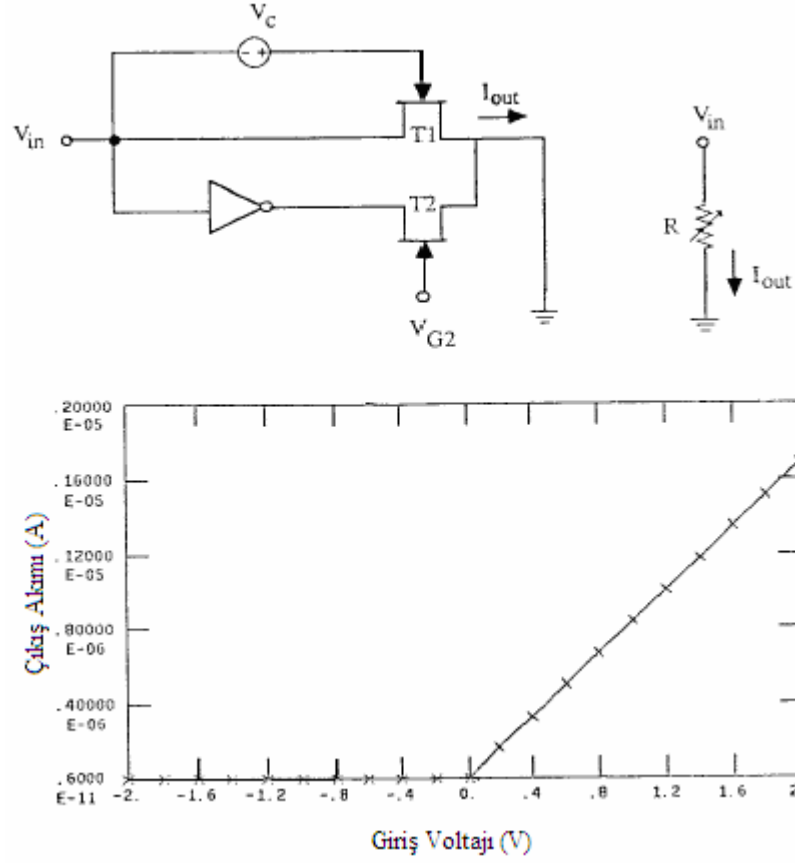
Değişken direnç olarak kullanılabilen en basit devre elemanları JFET(junction field-effect transistors) ve MOSFET(metal-oxide-semiconductor field-effect transistor)'lerdir. Gerek JFET gerekse MOSFET devre elemanları aktif bölgede voltaj kontrollü doğrusal direnç özelliği gösterdikleri için ağ eğitiminde uygun devre elemanlarıdır. Şekil-III.27'de N kanal MOSFET ve karakteristiği görülmektedir. Mosfetin çıkış direnci kapı(gate) ve kaynak(source) girişleri arasında uygulanan gerilimle kontrol edilebilmektedir. Şekil-III.27 'de bir MOSFET'in -0.5V ile 0.5V arasında lineer direnç değişimi görülmektedir.



Şekil-III.27 NMOS çıkış karakteristiği

Analog devrelerle daha fazla silikon alan kullanmak suretiyle daha hassas çözünürlüğe sahip gerilimle beraber lineer direnç değişimi sağlayacak yapılar yapmak mümkündür. Şekil-III.28'de verilen devrede 0-2V gerilimleri arasında lineer bir direnç değişimi görülmektedir. Direnç değişimi T2 transistörünün kapı(gate) girişine bağlı V_{G2} gerilimi ile yapılmaktadır. Çıkış akımı $I_o = \beta * V_{in} (V_c - V_{G2})$

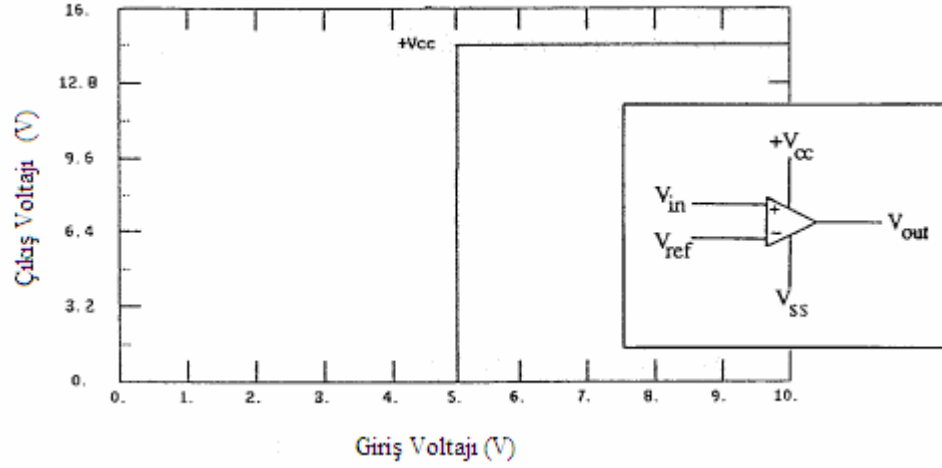
bağıntısıyla hesaplanabilir. β ifadesi transistorün üretimine bağlı iletkenlik parametresidir [6].



Şekil-III.28 Voltaj Kontrollü Direnç

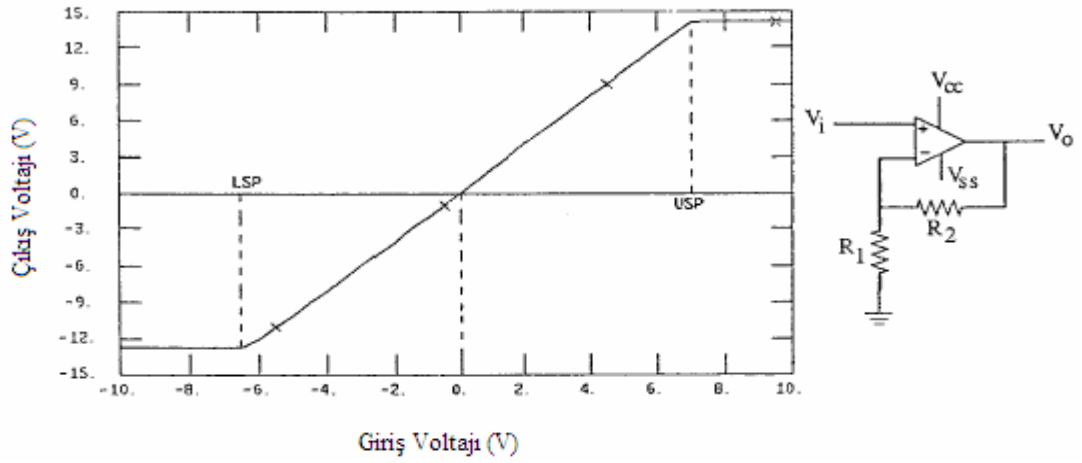
III.5.1.2 Aktivasyon Fonksiyon Üretici

İşlemci eleman YSA'ların en önemli kısmıdır. İşlemci eleman gelen sinyalleri toplar ve aktivasyon fonksiyonuna bağlı olarak bir sinyal üretir. Genel olarak YSA'larda üç temel transfer fonksiyonu vardır. Adım fonksiyonu, lineer fonksiyon ve sigmoid fonksiyonu. Adım fonksiyonu basit bir analog karşılaştırıcı ile yapılır. Şekil-III.29'da OPAMP ile yapılmış bir analog karşılaştırıcı gösterilmektedir. Devrede analog karşılaştırıcının eşik değeri V_{ref} gerilimi ile kontrol edilmektedir.



Şekil-III.29 Analog karşılaştırıcı ve çıkış karakteristiği

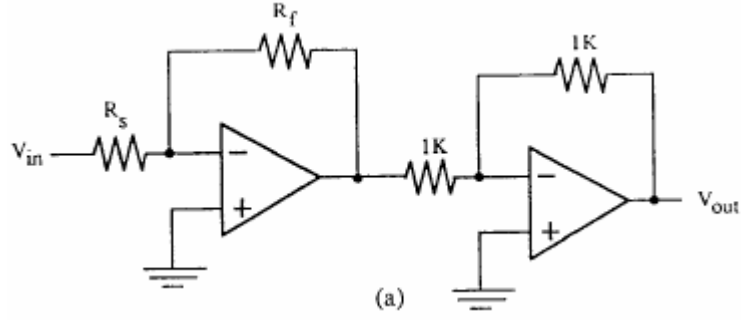
Lineer aktivasyon fonksiyonu ise evirmeyen bir OPAMP vasıtasıyla kolayca gerçekleştirilebilir. Şekil-III.30’da devre ve çıkış karakteristiği gösterilmektedir.



Şekil-III.30 OPAMP kullanarak elde edilen lineer aktivasyon fonksiyonu

Lineer transfer fonksiyonu ayarlanabilir eşik seviyesi ve doğrusallık imkanı verdiği için kullanıcıya daha fazla esneklik sağlar.

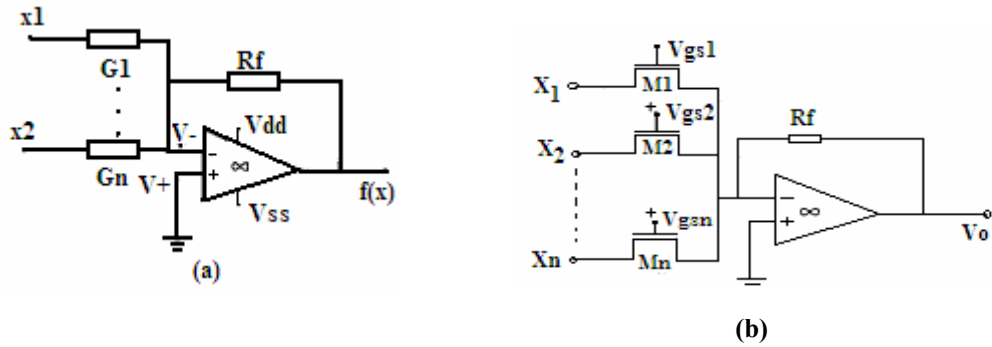
Bir diğer aktivasyon fonksiyonunda sigmoid fonksiyonudur. Şekil-III.31’de sigmoid fonksiyonunu üretecek devre görülmektedir.



Şekil III.31 Sigmoid yaklaşimli aktivasyon fonksiyonu

Sigmoid yaklaşimli aktivasyon fonksiyonu Şekil-III.31’de görüldüğü gibi yüksek kazançlı eviren yükselteç ile birim kazanç sağlayan yükseltecin kaskat bağlanmasıyla oluşturulabilir.

Bir OPAMP ile YSA’daki “node” fonksiyonu kolayca gerçekleştirebilir. Şekil-III.32’de OPAMP ile gerçekleştirilmiş bir nöron görülmektedir [7]



Şekil-III.32 OPAMP kullanılarak yapılan nöron modelleri

Şekil III.32(a)’deki OPAMP’ın çıkış voltajı Eşitlik- III.18 ile hesaplanabilir.

$$f(x) = f\left(\sum_{j=1}^n w_j x_j\right) \quad (\text{III.18})$$

f(net)=net olarak kabul edilirse.

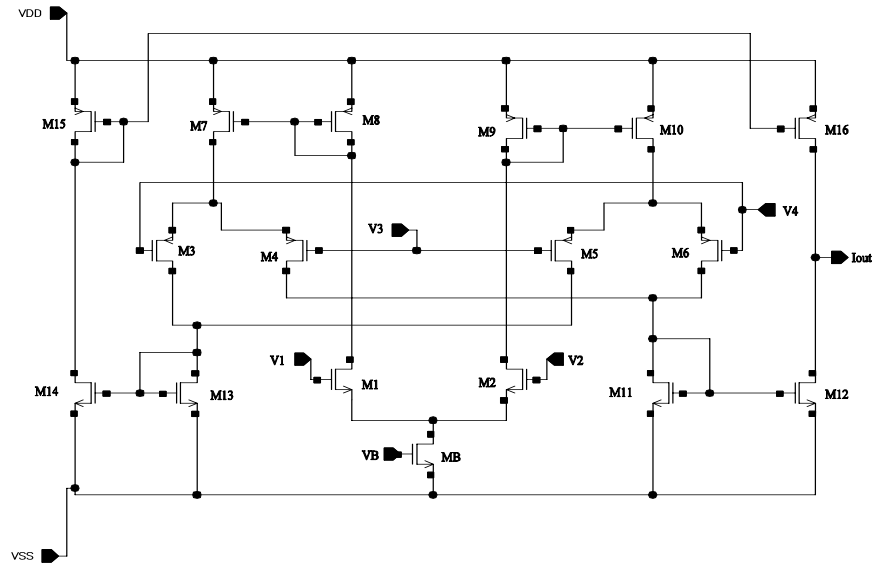
$$f(x) = \sum_{j=1}^n x_j (-R_F G_j) \quad (\text{III.19})$$

$$net = \sum_{j=1}^n x_j (-R_F G_j) \quad (\text{III.20})$$

Eşitlik-III.18 ile Eşitlik-III.19’dan şu sonucu çıkarabiliriz ki OPAMP ile gerçekleştirilen devrede ağırlıkların $w_j = -R_F G_j$ ifadesi ile temsil edildiği görülür.

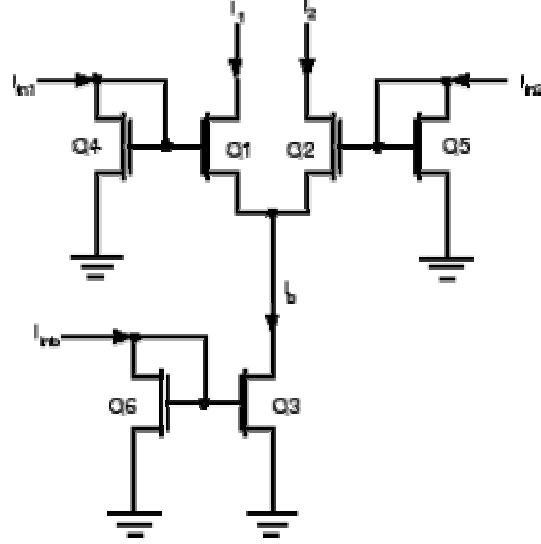
Ağırlık ifadesi w_j , G_j iletkenlik ifadesi ve R_F geri besleme direnci ile orantılıdır. Şekil-III.32’de gösterilen OPAMP devreleri gerek pozitif ağırlıkları gerekse negatif ağırlıkları kolayca temsil edebilmekte ve kısmi doğrusal aktivasyon fonksiyonuna sahiptirler. Bu devreler temel nöron işlevini yerine getirdiği halde kısıtlamaları vardır. Bunlar; transfer fonksiyonunun şekli, yüksek güç tüketimi, fakat en önemli sorunu ağırlıkların dirençlerde saklanıyor olmasıdır. Ağırlıklar dirençlerde saklandığı için kolayca kontrol edilememekte, ayarlanamamaktadırlar dolayısıyla eğitim esnasında kullanılamazlar ve başka bir işlevi yerine getirmek için tekrar eğitilemezler [8].

Hücre fonksiyonunu gerçekleştirmek için sık kullanılan devrelerden bir diğeri de Gilbert çarpma devresidir. Gilbert devresinin avantajı VLSI tekniği ile üretime elverişlidir. Güç tüketimi düşüktür ve iyi bir giriş simetrisine sahiptir. Şekil-III.33’te Gilbert çarpma devresi görülmektedir [9].



Şekil-III.33 Gilbert çarpma devresi

Anlatım kolaylığı olması amacıyla Şekil-III.34’te gösterilen basitleştirilmiş yarım Gilbert çarpma devresi üzerinde açıklama yapılacaktır [10].



Şekil-III.34 Yarım Gilbert çarpma devresi

I_1 ve I_2 akımları Eşitlik-III.21’de ve Eşitlik-III.22’de gösterilmektedir.

$$I_1 = \frac{I_{inb} I_{in1}}{I_{in1} + I_{in2}} \quad (III.21)$$

$$I_2 = \frac{I_{inb} I_{in2}}{I_{in1} + I_{in2}} \quad (III.22)$$

Bu devre, I_{in1} ve I_{in2} akımlarının toplamı 1’e eşit kabul edilirse

$$I_1 = I_{inb} * I_{in1} \quad (III.23)$$

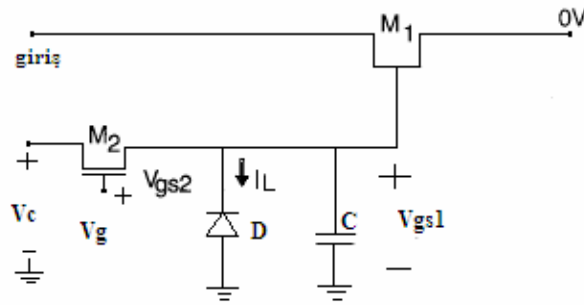
$$I_2 = I_{inb} * I_{in2} \quad (III.24)$$

I_1 akımı $I_{inb} * I_{in1}$ akımlarının çarpımına, I_2 akımı ise $I_{inb} * I_{in2}$ çarpıma eşit olacaktır.

III.5.1.3 Sinaptik Bağlatıların Gerçekleştirilmesi

YSA’lar çok miktarda sinaptik bağlantılar içerirken az sayıda işlemci nörona sahiptirler. YSA’ları analog olarak gerçekleştirmenin en zor tarafı ayarlanabilir ağırlıklara sahip devreler yapabilmektir. Ayarlanabilir ağırlıklara sahip devreler MOS transistorlerle yapılabilirler. Şekil III.32.b’de sinaptik bağlantılara sahip nöron modeli görülmektedir. Bu modelde kullanılan mosfetler sayesinde gerilim kontrollü dirençlere sahip olmaktadır. Ağırlık eğitimi gerilim kontrolü ile yapılabilir fakat ağırlık doğru olarak işlevini yerine getirebilmesi için eğitim sonrasında elde edilen gerilim

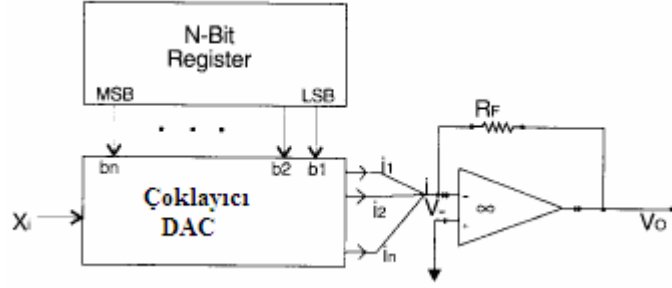
değerlerinin bir yerlerde saklanması gerekecektir. Gerilimi depo etmede kullanılan ortak yöntemlerden biri kondansatörleri kullanmaktır. Kullanılan FET'lerin kapı girişlerine paralel bir kondansatör bağlamak yeterli olacaktır. Kondansatörde uygun gerilimin tutulması Şekil-III.35'de gösterildiği gibi bir örnekle tut devresiyle gerçekleştirilebilir. M_2 transistörü örnekle tut görevini yerine getirirken aynı zamanda ON/OFF işlevini yerine getirmektedir. V_{gs1} gerilimi M_1 transistörünün kanal direncini kontrol eder. Ağırlıkları bu şekilde tutmanın dezavantajı kondansatörlerde depolanan enerjinin zaman içinde azalmasından dolayı kondansatörlerin belli zaman aralıklarında tekrardan şarj edilmeleri gerekir [11].



Şekil-III.35 Fet ve kondansatör ile gerçekleştirilmiş sinaptik ağırlıklar

Ağırlıkları tutmanın bir diğer yolu sayısal kontrollü entegreler kullanmaktır. Bu teknik ağırlıkları kondansatörde tutmaktan daha etkili bir yoldur zira ağırlıklar sayısal olarak programlanabilmekte ve saklanabilmektedir. Ağırlıklar sayısal olarak kaydedicilerde tutulduktan sonra geri kalan işlemlerin hepsi analog ortamda yapılmaktadır. Ağırlıkları sayısal olarak tutmanın bir diğer avantajı ise ağ eğitimleri genellikle bilgisayar yardımıyla yapıldığı için hassas ağırlık değerleri kolayca temsil edilebilmektedir.

Çoklayıcı DAC devreleri sayısal ağırlıkları tutmak için kullanılabilir. Şekil-III.36'da çoklayıcı DAC devresiyle yapılan bir devre görülmektedir. Bu tür bir devre ile ağırlıklar kolayca ayarlanabilir ve elde edilen ağırlıklar kaydedicilerde saklanabilir. Şekil-III.36 daki çevirici devre girişindeki X_i voltajını ağırlık değerlerine bağlı olarak $i_1, i_2 \dots i_n$ akımlarına çevirme işlemi yapmaktadır [12].



Şekil-III.36 Çoklayıcı DAC ile ağırlıkların tutulması

III.5.1.4 Analog Sistemlerin Avantajları ve Dezavantajları

Analog devrelerle YSA donanımı oluşturmanın şu avantajları vardır [13].

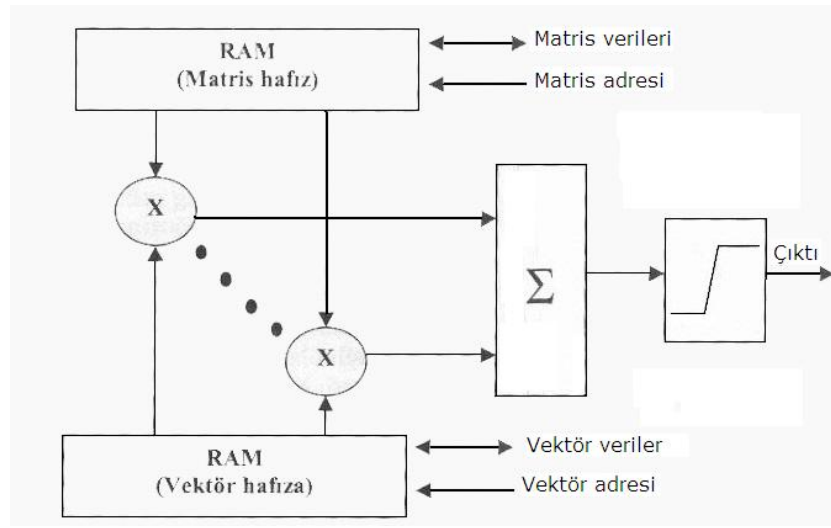
- Basit bir transistör YSA'ların temel fonksiyonlarından olan çarpma işlemini kolayca yerine getirebilmektedir. Aynı işi sayısal olarak yapmak için çok sayıda transistör kullanılmak zorundadır.
- YSA'ların diğer bir özelliği ise toplamadır. Basit bir iletken tel ile birden fazla (transistörden)nörondan gelen akım kolayca toplanabilir.
- Analog YSA sistemlerine giriş değerleri ve çıkış değerleri analog olduğu için sayısalardan analog veriye veya analog veriden sayısal veriye dönüştürmeye gerek kalmaz.
- Gerçek paralel işlem yapan donanımlar oluşturulabilir.
- Sayısal sistemlere göre hızlıdır(aynı işlemi yapmak kaydıyla).

Analog YSA'ların dezavantajları ise

- Analog YSA'lar sıcaklık değişimine karşı duyarlıdır. Sıcaklık değişimi devre çalışmasını etkileyebilir.
- Analog YSA'lar gürültüye karşı duyarlıdır. Gürültüden kolayca etkilenebilirler.
- Sinaptik ağırlıkları uygun hassasiyette üretmek kolay değildir.
- Analog YSA'ları eğitmek zordur.
- Eğitim için ek donanım gerektirirler.

III.5.2 Sayısal Yapay Sinir Ağı Donanımları

Sayısal yapay sinir ağı donanımlarında, ağ üzerindeki bütün hesaplamalarda kullanılan değerler büyüklüğü belirlenmiş ikili vektörlerdir. Sayısal teknolojiye verilerin kısmen 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-III.37’de bir sayısal yapay sinir ağı donanımının yapısı gösterilmektedir.



Şekil-III.37 Sayısal YSA yonga yapısı

Sayısal YSA’lar, programlanabilir elemanları içerebilirler, hem sabit hem değişken ağırlık matrislerini saklayabilirler. Olumsuz yanları ise çarpma ve toplama işlemlerindeki yavaşlık ve genellikle DAC ve ADC gibi yardımcı devrelere ihtiyaç duyarlar.

Sayısal YSA modellerinde işlemciler, programlanabilir devre elemanları veya nöral entegreler kullanılmaktadır. YSA’larda paralel veri işleme gerçekleşmektedir fakat işlemcilerde komut işleme seri olarak gerçekleştiği için işlemci kullanmak YSA’lar ile çelişmektedir. Ayrıca yoğun işlem gerektiren YSA uygulamalarında bilgisayarlar yavaş kalmaktadır. İşlemcilerin veri işleme hızını arttırmak için işlemcileri paralel bağlayıp paralel veri işleme ile ilgili çalışmalar yapılmaktadır. Bu tür çalışmalarda bir efendi işlemci birden fazla köle işlemci ile birlikte çalıştırılmaktadır. Efendi işlemci diğer köle işlemcileri kontrol etmektedir. Sayısal uygulamalarda en fazla öne çıkan donanımlardan biri ise FPGA’lar dır. Son yıllarda

entegre teknolojisindeki gelişmeler sayısal uygulamalarda da FPGA entegrelerine ve uygulamalarına yönelişi arttırmıştır ve FPGA kullanılarak yapılan YSA donanımları ile ilgili makalelerde artış görülmektedir. Son yıllarda mikrodenetleyicilerde ki hız ve kapasite bakımından gelişmelere paralel olarak YSA donanımlarında kullanılmaya başlanmıştır. Fakat hala çok fazla nöron gerektiren uygulamalarda tek başına kullanılan mikrodenetleyiciler yetersiz kalabilmektedir.

Sayısal YSA uygulamalarını gerçekleştirirken karşılaşılan en büyük sıkıntı aktivasyon fonksiyonunu gerçekleştirirken yaşanmaktadır. FPGA vb. programlanabilir entegreler ile yapılan donanımlarda aktivasyon fonksiyonunu gerçekleştirmek çok sayıda alan ve eleman kullanımı gerektirmektedir. Mikroişlemcilerle ve mikrodenetleyicilerle yapılan donanımlarda ise aktivasyon fonksiyonları YSA hızını yavaşlatan en büyük faktör olarak öne çıkmaktadır. Ağırlık değerlerini değiştirmek, bir yerde saklamak gibi işlemler ise RAM vb. hafıza birimleriyle kolayca yapılabilir.

III.5.3 Analog-Sayısal YSA Donanımları

Bu donanım sistemleri hem sayısal hem de analog sistemlerin birleştirilmesi ile elde edilen donanımlardır. Bu sistemlerin amacı her iki yaklaşımında avantajlarını bir araya getirmektir. Genel olarak dış dünya ile ilişki analog sinyaller ile gerçekleşmektedir. Dahili işlemciler ile ya tamamen yada kısmen sayısal devreler ile sağlanmaktadır. Nöral çipler bu gruba girmektedir. Bu nöral çiplere örnek verilecek olursa INTEL firmasının M64 ve ETANN isimli yongaları elektriksel olarak eğitilebilen nöro çiplerdir. ETANN nöro çipinde 64 nöronluk YSA modelleri oluşturulabilir. Benzer şekilde NeuroDynamX firmasının NDX Neural Accelerators, ETANN çipinin bir üst versiyonudur. IC Tech firmasının ürettiği DANN050L isimli çip online öğrenme yeteneğine sahip olan birbirine tamamen bağlantılı 50 nöron girişine izin veren ve saniyede 6 milyon bağlantıya sahip 64 pinli ve 100 mwatt altında enerji sarfıyatı olan bir donanımdır. Diğer bir nöral çip üreticisi olan Bellcore firmasının CLNN-32 ve AT&T ANNA çipleride karma YSA donanımlarındandır.

Bazı genel amaçlı üretilen yongalar YSA donanımı olarakta kullanılabilir. Bu tür yongalardan biri olan Zetex firmasının üretmiş olduğu TRAC (Totally Reconfigurable Analog Circuit) isimli entegre YSA donanımlarında aktivasyon fonksiyonunu gerçekleştirmek amacıyla kullanılabilir [14]. TRAC isimli yongada 20 adet modül bulunur. Bu modüllerden her biri 8 adet farklı fonksiyondan biri olacak şekilde programlanabilir. Modüllerin programlanabileceği fonksiyonlar, logaritma alma, ters logaritma alma, doğrultmaç, evirmeyen geçiş, toplama, ters alma işlemi, türev alma, integral alma olarak belirtilebilir. Bu fonksiyonlar kullanılarak YSA'ların aktivasyon fonksiyonları gerçekleştirilebilir. Yonga içerisindeki modül sayısının az olması yüksek miktarda nöron içeren YSA donanımlarını gerçekleştirmede temel zorluğu oluşturmaktadır.

BÖLÜM IV

MİKRODENETLEYİCİLER

Bir mikro bilgisayar içerisinde bulunması gereken temel bileşenlerden RAM, I/O ünitesinin tek bir chip içerisinde üretilmiş biçimine mikrodnetleyici(μ C) denir. Bilgisayar teknolojisi gerektiren uygulamalarda kullanılmak üzere tasarlanmış olan mikrodnetleyiciler, mikroişlemcili yapılara göre daha basit ve ucuzdur.

Mikrodnetleyicilerin mikroişlemcilere karşı bazı üstünlükleri vardır. Örneğin mikroişlemcili bir sistem yapıldığında mikroişlemcinin yanı sıra hafızalar (RAM, ROM veya EPROM) , I/O giriş çıkış birimi ve buna benzer birçok sistem kullanılmaktadır. Bu karışık sistemin hem tasarlanması ve yapımı zordur hem de maliyeti oldukça yüksektir.

Mikrodnetleyiciler de bir sistemin çalıştırılabilmesi için yalnızca bir mikrodnetleyici ve osilatör devresi yeterli gelmektedir. Sistemde gerekli olan ön bellek ve I/O birimi mikrodnetleyiciler içinde bir yonga halindedir. Dolayısıyla bu yönleriyle pratik uygulamalarda mikroişlemcilere üstünlük sağlarlar.

Tamamen Analog YSA donanımlarını oluşturmak ve eğitmek zaman ve maliyet açısından mikrodnetleyicilere göre daha zor ve zahmetlidirler. Oysa mikrodnetleyiciler ile kısa sürede YSA uygulamalarını pratiğe dökme şansımız

vardır. Bu avantajları nedeniyle mikrodnetleyicilerin çözüm olabileceđi YSA uygulamalarında kullanılmaları yerinde olacaktır.

IV.1 PIC MİKRODENETLEYİCİLERİ

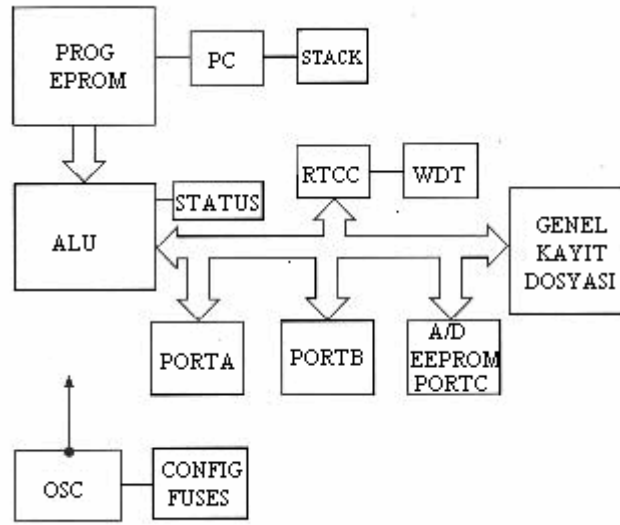
PIC Serisi mikrodnetleyiciler MICROCHIP firması tarafından geliştirilmiş ve üretim amacı çok fonksiyonlu mantıksal uygulamaların hızlı ve ucuz bir mikrodnetleyici ile yazılım yoluyla karşılanmasıdır. PIC'in kelime anlamı Peripheral Interface Controller (Çevresel arabirim denetleyicisi)'dir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme ihtiyaç duyulduđu için geliştirilmiştir. PIC mikrodnetleyicileri RISC tipi mimariye sahip mikrodnetleyicilerdir. PIC serisi tüm mikrodnetleyiciler herhangi bir ek bellek veya giriş / çıkış elemanı gerektirmeden sadece 2 adet kondansatör, 1 adet direnç ve bir kristal ile çalıştırılabilir [15].

IV.1.1 PIC Mikrodnetleyicilerinin İç Yapısı

CPU bölgesinin kalbi ALU'dur (arithmetic logic unit- aritmetik mantık birimi) ve W (working – çalışan) yazmacı adında bir yazmaç içerir. PIC diğer mikrodnetleyicilerden aritmetik ve mantık işlemleri için bir tek ana yazmaca sahip oluşuyla farklılaşır. W yazmacı CPU'daki herhangi bir veriyi transfer etmek için kullanılır.

CPU alanında ayrıca iki katagoriye ayırabileceğimiz veri yazmaç dosyaları (Data Register Files) bulunur. Bunlardan ilki I/O ve kontrol amaçlı olanlar diğeride tamamen kayıt amaçlı olanlardır.

Harvard mimarisi temelli mikrodnetleyicilerde, veri akış miktarını hızlandırmak ve yazılım güvenliğini arttırmak amacıyla ayrı bus'lar kullanılır. Bu mimari veri ve program belleğine eşzamanlı erişimi olanaklı kılar.



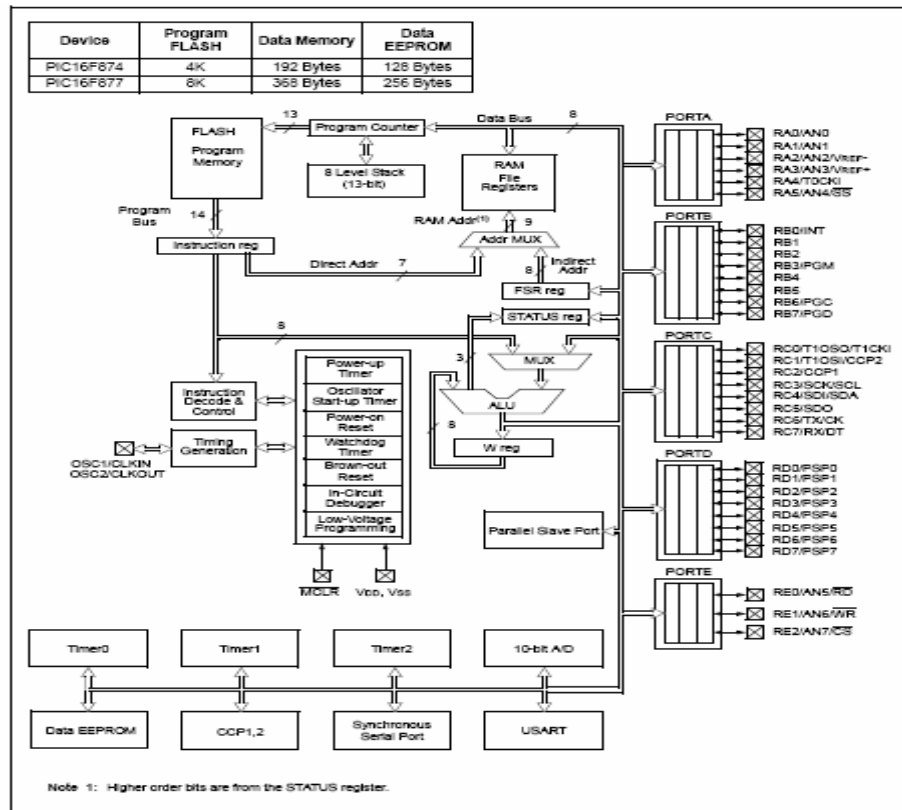
Şekil-IV.1 Temel PIC blok diyagramı

IV.1.2 PIC 16F877 Mikrodenetleyicisi

PIC 16F877, en popüler işlemci olan PIC 16F84'den sonra yeni ve gelişmiş olanaklar sunmaktadır. Program belleği 16f84'de olduğu gibi FLASH ROM'dur. Özellikle 16C6x ve 16C7x ailesinin tüm özelliklerini barındırması, 16F877'yi kod geliştirmede de ideal bir çözüm olarak gündeme getirmektedir. Konfigürasyon bitlerine dikkat etmek şartıyla C6x ve C7x ailesinden herhangi bir işlemci için geliştirilen kodlar hemen hemen hiçbir değişiklik yapmaksızın 16F877'ye yüklenebilmektedir. Bunun yanı sıra 16F877 mikrodnetleyicisi 16C74 ve 16C77 işlemcileriyle de bire bir bacak uyumludur.

Tablo-IV.1 PIC 16F877 entegresinin özellikleri

Özellikleri	PIC 16F877
Çalışma hızı	DC 20 Mhz
Program belleği	8Kx14 word FLASH ROM
EEPROM veri belleği	256 byte
Kullanıcı RAM	368x8
Giriş/Çıkış port sayısı	33
Timer	Timer0, Timer1, Timer2
A/D çevirici	8 kanal 10 bit
Capture	16 bit capture
Compare	16 bit compare
PWM	10 bit PWM çözünürlük
Seri çevresel arayüz	SPI(master) ve I2C(master/slave) modunda SPI portu
Paralel slave port	8 bit, harici RD, WR ve CS kontrollü
USART/SCI	9 bit adresli
Interrupt kaynağı sayısı	14
1 bacadan çekilecek maksimum akım	40mA



Şekil-IV.2 PIC 16F877 Blok Diyagramı

IV.1.3 PIC 16F877'nin Bazı Özel Fonksiyonları

IV.1.3.1 USART

USART, senkron/asenkron alıcı verici 16F877'deki iki seri giriş/çıkış modülünden biridir. Seri iletişim arayüzü (SCI: serial comm.interface) olarak da bilinen USART, monitör veya PC gibi aygıtlara tam çift yönlü asenkron bağlantıda kullanılmak üzere konfigüre edilebilmektedir. A/D veya D/A arayüzlerine, seri EEPROM'lara yarım çift yönlü senkron bağlantıda kullanılmak üzere de konfigüre edilebilir. USART aşağıdaki gibi konfigüre edilebilmektedir.

- Asenkron : tam çift yönlü (full duplex)
- Senkron: Master , yarım çift yönlü (half duplex)
- Senkron: Slave, yarım çift yönlü

RC6 verici, RC7 ise alıcı port olarak kullanılmaktadır. RCSTA (0x18) ve TXSTA (0x98) yazmaçları konfigürasyonda kullanılmaktadır.

IV.1.3.2 Senkronize Seri Port (MSSP)

MSSP modülü, diğer çevre birimleri veya mikroişlemcilerle seri iletişimde kullanılmaktadır. Bu çevre birimleri seri EEPROM, kaydırmalı yazmaçlar (shift register), gösterge sürücüleri, A/D çeviriciler vb. olabilir. MSSP modülü aynı anda aşağıdaki iki moddan birine konfigüre edilebilir:

- Seri Çevresel Arabirim (SPI)

RC5: Seri veri çıkışı (SDO: serial data out)

RC4: Seri veri girişi (SDI: serial data in)

RC3: Seri saat (SCK: serial clock)

- Inter Integrated Circuit (I²C)

RC4: Seri veri (SDA: serial data)

RC3: Seri saat (SCK: serial clock)

Bu modlardan birine göre konfigüre etmek içinse SSPSTAT (senkron seri port durum yazmacı, 0x94), SSPCON (senkron seri port kontrol yazmacı, 0x14) ve SSPCON2 (senkron seri port kontrol yazmacı 2, 0x91) yazmaçları ayarlanmalıdır.

IV.1.3.3 Analog/Sayısal Çevirici Modülü

A/D modülü 16C7x ailesinden farklı olarak 10 bittir. Toplam 8 A/D kanal vardır. 16F877'nin güzel bir özelliği de işlemci "SLEEP" modunda iken bile A/D çeviricinin geri planda çalışmasıdır. A/D kanalları için RA4 hariç diğer RA portları ve RE portları kullanılabilir.

IV.2 I2C SERİ HABERLEŞMESİ VE BUS ÖZELLİKLERİ

Her ne kadar seri bus yapıları paralel bus yapılarının özelliklerine sahip olmasa da seri bus yapıları daha az iletim yolu ve daha az port çıkışı kullanmaktadır. Bir bus sadece cihazları birbirine bağlayan yapılar değil aksine birbirleriyle iletişimi sağlayacak formatları ve standartları olan yapılardır. Seri bus ta oluşabilecek çakışma, veri kaybı ve bilginin bloke olma ihtimaline karşılık her seri bus kendi protokolünü oluşturmak zorundadır. Hızlı cihazlar yavaş cihazlarla haberleşebilmelidir. Sistem bus'a bağlanan cihaza bağlı olmamalıdır aksi taktirde sistemde değişiklik ve geliştirme yapmak imkansızlaşır. Kullanılan prosedür hangi cihazın kontrolde olacağını ve ne zaman kullanılacağını kontrol edebilmelidir. Eğer farklı saat hızlarına sahip cihazlar bus' a bağlanacaksa bus'ın saat hızı belirlenebilir olmalıdır. Bütün bu kriterler I2C –bus yapısında belirlenmiştir [17].

Tablo-IV.2 I2C veri haberleşmesinde kullanılan terimler ve açıklamaları

Terimler	Tanımlar
Verici	Bus 'a veri yollayan cihaz
Alıcı	Bus'tan veri alan cihaz
Efendi	Veri transferini başlatan, saat sinyalini üreten ve sonlandıran cihaz
Köle	Efendi tarafından adresi üretilen cihaz
Çoklu-Efendi	Birden fazla efendinin bus'ın kontrolünü gerçekleştirmesi
Arabulma (Arbitration)	Aynı anda birden fazla efendi bus'ı kontrol etmeye çalıştığında sadece bir efendiye izin verme işlemi.
Senkronizasyon	İki veya daha fazla cihazın saat sinyalini senkronize etme işlemi

IV.2.1 I2C Bus Kavramı

I2C bus yapısı NMOS, CMOS ve bipolar teknolojilerini desteklemektedir. İki hat kullanmaktadır. Seri veri (SDA) hattı ve seri saat (SCL) hattı. Bus'a bağlanacak her cihazın kendine ait bir adresi vardır bu cihazlar mikrodnetleyici, LCD sürücü, hafıza veya tuş takımı arabirimi olabilir. Bu cihazlar fonksiyonlarına göre alıcı veya verici olabilirler. Efendi, veri transferini başlatan saat sinyali üreten ve veri transferini sonlandıran cihaz olarak belirtilirken bu işlemler esnasında efendi ile haberleşen cihazı köle olarak tanımlayabiliriz.

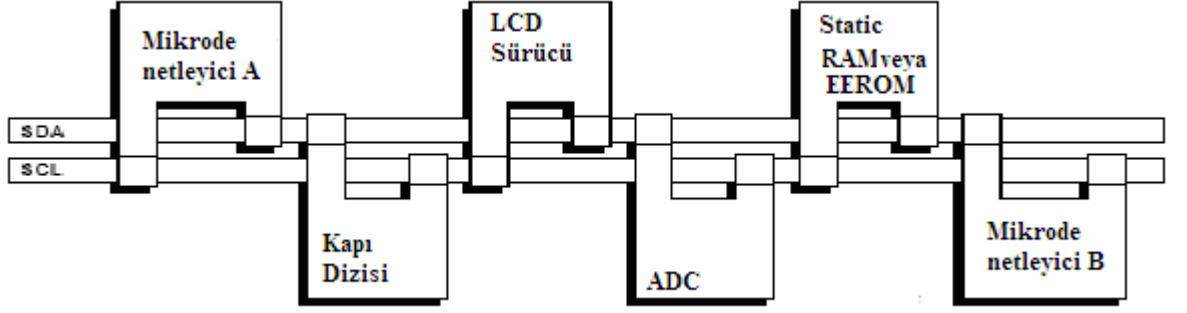
I2C bus'ı çoklu efendiyi destekleyen bir bus yapısına sahiptir. Çoklu-Efendi terimi bus'ın kontrolünün birden fazla efendi tarafından yapılabildiğini belirtmek için kullanılmaktadır ve genellikle efendiler mikrodnetleyiciler olmaktadır. Şekil-IV.3'de yer alan iletişim prensibine göre veri alışverişi şu şekilde gerçekleşir.

1) Mikrodnetleyici A'dan, mikrodnetleyici B'ye veri gönderme işlemi.

- Mikrodnetleyici A (Efendi), mikrodnetleyici B (köle)'nin adresini gönderir.
- Mikrodnetleyici A (Efendi-verici), veriyi mikrodnetleyici B (köle-alıcı)'ye gönderir.
- Mikrodnetleyici A veri transferini sonlandırır.

2) Mikrodnetleyici A'nın, mikrodnetleyici B'den veri okuma işlemi.

- Mikrodnetleyici A (Efendi), mikrodnetleyici B (köle)'nin adresini gönderir.
- Mikrodnetleyici A (Efendi-alıcı), veriyi mikrodnetleyici B (köle-alıcı)'den alır.
- Mikrodnetleyici A veri transferini sonlandırır.



Şekil-IV.3 İki Mikrodenetleyicili I2C Konfigurasyonu

İki mikrodenetleyici aynı anda hattı kontrol etmek istediğinde ilk logic 1'i diğeri lojik 0 üretirken hattın kontrolünü ele alır.

IV.2.2 I2C Çalışma Karakteristiği

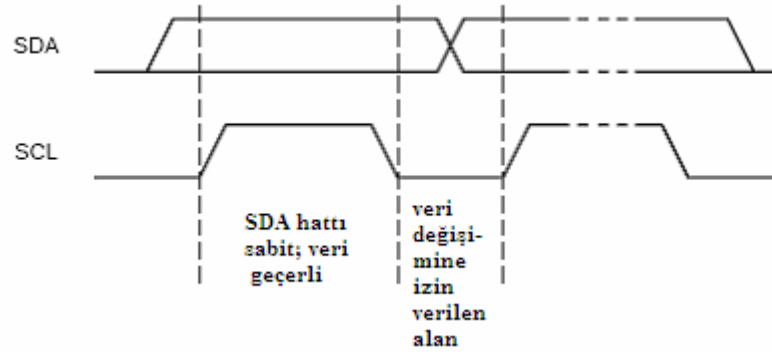
Veri iletiminde kullanılan SDA hattı ve saat sinyalinin iletiminde kullanılan SCL hatlarında iletim iki yönde gerçekleşir. Bu iki hatta pull-up dirençleri ile birlikte voltaj kaynağına bağlanır. İki hatta kullanılmaz iken lojik 1 seviyesindedir. Bus'a bağlanacak olan cihazların açık-drain veya açık-kollektör tipi çıkışa sahip olmaları gerekir. I2C'de veri iletimi standart modda 100kbit/s hızla gerçekleşirken hızlı modda 400kbit/s, en hızlı modda ise 3.4Mbit/s ile gerçekleştirilmektedir. Bus'a bağlanacak toplam cihaz sayısı bus'ın 400pF olan kapasite sınırını aşmamalıdır.

IV.2.2.1 Bit Transferi

I2C Bus hattına bağlanabilecek cihazların teknoloji farklılıkları(NMOS, CMOS, bipolar) nedeniyle lojik 0 ve lojik 1 seviyeleri sabit tutulmamıştır . Lojik seviyeleri kullanılan besleme sinyaline göre değişmektedir. Her bir bit transferi için bir saat sinyali üretilmektedir.

IV.2.2.2 Veri Geçerliliği

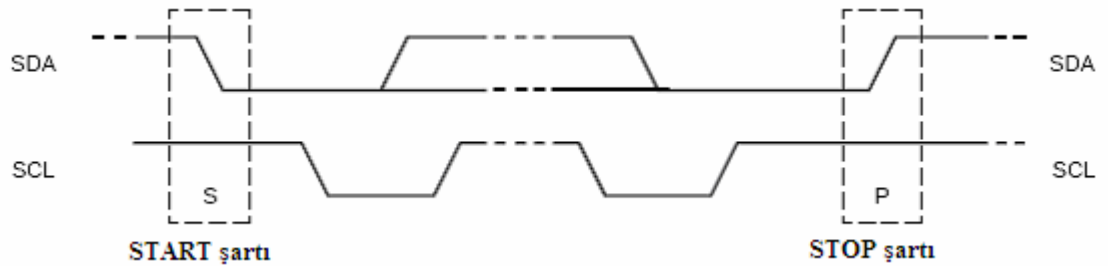
SDA hattındaki veri saat sinyalinin 'yüksek' seviyede olduğu müddet boyunca değiştirilmemelidir. Verinin değişimine sadece saat sinyalinin 'düşük' seviyede olduğu süre içinde izin verilir. Şekil-IV.4 'te bu gösterilmektedir.



Şekil-IV.4 I2C Bus üzerinde bit transferi

IV.2.2.3 START ve STOP Şartları

I2C-bus prosedüründe farklı START(S) ve STOP (P) şartları tanımlanmıştır. (Şekil-IV.5) SCL hattı ‘yüksek’ seviyede iken SDA hattı ‘yüksek’ seviyeden ‘düşük’ seviyeye geçerse START işlemi gerçekleşir. SCL hattı ‘yüksek’ seviyede iken SDA hattı ‘düşük’ seviyeden ‘yüksek’ seviyeye geçerse STOP işlemi gerçekleşir. START ve STOP işlemini sadece efendi cihaz üretebilir. START işleminden sonra bir müddet meşgul kabul edilir ve STOP sinyalinden sonra hat yine serbest olur. STOP sinyali üretilmeden önce tekrar START sinyali üretilirse hat meşgul kalmaya devam eder.

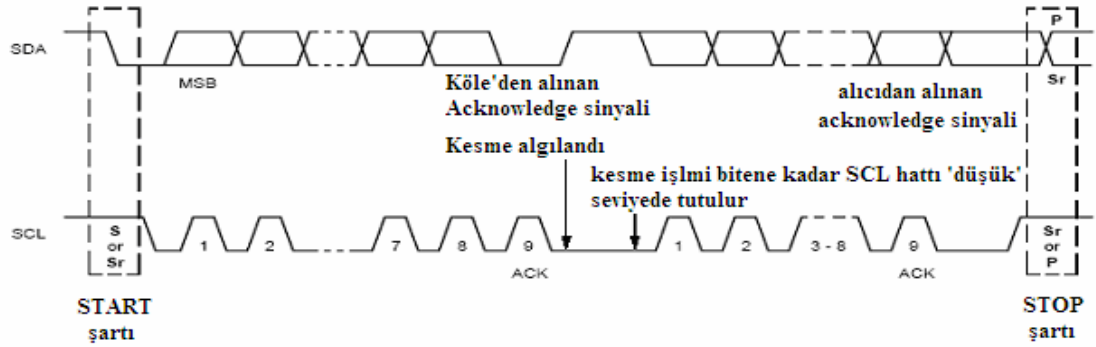


Şekil-IV.5 START ve STOP şartları

IV.2.2.4 BYTE Formatı

SDA hattına aktarılan veri uzunluğu 8 bit olmak zorundadır. Sınırsız sayıda byte iletişim esnasında gönderilebilir. Gönderilen her byte’tan sonra bir acknowledge sinyali üretilir. Byte’lar gönderilirken MSB biti önce gönderilir (Şekil-IV.6). Eğer köle bir byte alma veya gönderme işlemi tamamlanmadan kesme veya veri akışını ertelemesi gereken şartlar ile karşılaşırsa köle SCL hattını ‘düşük’ seviyede tutarak efendi cihazın saat sinyali üretmesini engeller. Kesme işlevi veya veri akışını

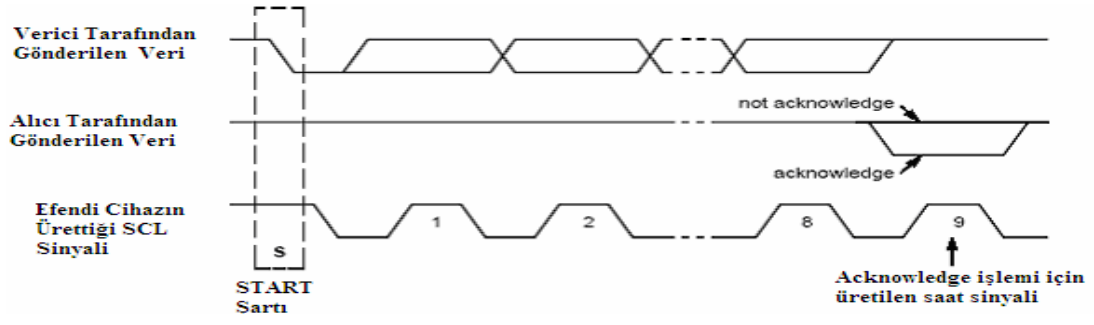
ertelemesi gereken şartlar geçince SCL hattını serbest bırakarak veri akışının kaldığı yerden devam etmesini sağlar.



Şekil-IV.6 I2C Bus hattı üzerindeki veri transferi

IV.2.2.5 Bilgi Alındı Sinyali(Acknowledge)

Veri transferinin bir acknowledge sinyali ile birlikte bitirilmesi bir zorunluluktur. Acknowledge sinyaline bağlı saat sinyali efendi cihaz tarafından üretilir. Acknowledge saat sinyali sırasında verici SDA hattını ‘yüksek’ seviyeye getirir. Alıcı cihaz acknowledge saat sinyali boyunca SDA hattını ‘düşük’ seviyede tutmak zorundadır. Şekil-IV.7’de ayrıntılı biçimde acknowledge sinyali gösterilmektedir. Köle alınan veriye karşılık(acknowledge) vermez ise SDA hattı yüksek seviyede kalır dolayısıyla efendi işlemi iptal edebilir, STOP sinyali üretebilir veya tekrar START sinyali üreterek işlemin tekrar başlamasını sağlayabilir. Köle, efendi tarafından gönderilen adrese cevap verebilir fakat ardından herhangi bir veri alış verişi olmaz ise köle SDA hattını serbest bırakır ve efendi işlemi bitirebilir veya tekrar başlatabilir. Eğer efendi alıcı işlevi görüyorsa , efendi alma işleminin bittiğini köle ye bildirmek için alınan son byte’dan sonra acknowledge sinyali üretmez. Köle –verici cihaz bunun ardından SDA hattını serbest bırakarak efendi cihazın STOP veya tekrar START işlemini başlatmasına izin verir.



Şekil-IV.7 I2C Bus hattı üzerindeki Acknowledge sinyali

IV.2.2.6 I2C ve SPI senkron(eş zamanlı) Veri Haberleşmelerinin Karşılaştırılması

SPI(Serial Peripheral Interface) haberleşme standardında cihaz adreslemesi yapılamamaktadır. Birden fazla cihaz SPI kullanarak haberleşmesi gerektiği takdirde cihazların chip select girişini kullanması gerekmektedir. N adet cihazı birbirine bağlamak için $3+N$ adet pin kullanılmak zorundadır. I2C haberleşme standardında ise 7 bit adresleme ile 128 adet cihazı birbirine bağlamak mümkündür. Ayrıca bu standartta cihaz sayısı ne kadar olursa olsun 2 adet pin veya hat kullanılmaktadır.

- SPI haberleşme standartı I2C haberleşme standardından hızlıdır. SPI'da veri hızları 10Mbit/s'a kadar çıkabilirken bu I2C'de 3.4 Mbit/s olarak gerçekleşmektedir.
- SPI çoklu efendi sistemini dolaylı ve karmaşık yollardan desteklerken I2C standartında ise bu işlem kolayca yapılabilir.
- SPI gürültü bağışıklığı yönünden daha iyidir. I2C standardında adres bilgisinden sonra efendi cihaz bir bit ile okuma veya yazma işlemi yapacağını belirtir. Bu bitin zarar görmesi halinde işlem hatası oluşacaktır. SPI'da ise okuma ve yazma komutları mevcuttur.

Sonuç olarak birden fazla cihaz birbirine bağlanacak, çoklu efendi desteği isteniyor ayrıca ses ve görüntü ile ilgili işlemlerde I2C standardı tercih edilmelidir. Philips firmasının görüntü ve ses ile cihazlardaki üstünlüğü dolayısıyla. Eğer düşük maliyet, yüksek hız ve gürültü bağışıklığının iyi olması isteniyorsa SPI standardı tercih edilmelidir.

IV.3 PIC MİKRODENETLEYİCİ TABANLI YSA DONANIMININ OLUŞTURULMASI

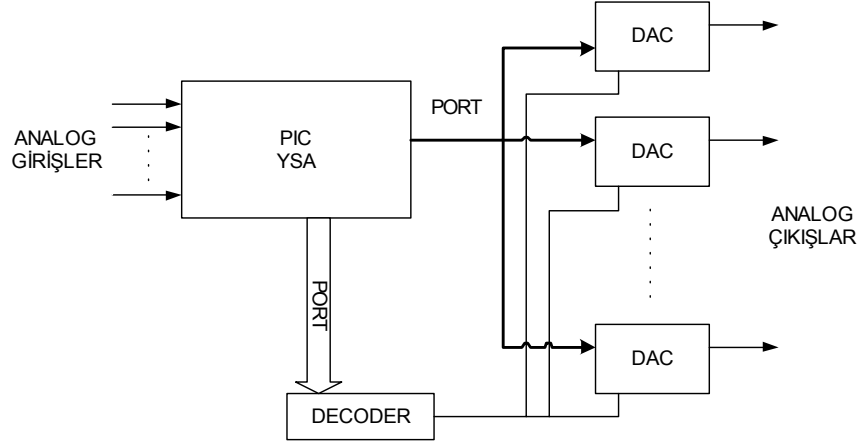
PIC tabanlı YSA donanımı oluşturma çalışmasında, hücre sayısı çok yüksek olmayan ağ yapıları için gerçeklemesi kolay bir donanım oluşturma amaçlamıştır. Uygulamada YSA gerektiren birçok problem onlar seviyesindeki hücre sayısı ile çözümlenebilmektedir. Bu da mikro denetleyici yapıları ile karşılanabilir bir ağ yapısı oluşturulmasına imkân vermektedir.

Sayısal YSA donanımları yanında, analog YSA yapılarına olan ihtiyaç birçok alanda kendini göstermektedir. Bu tür uygulamalarda giriş ve çıkış verileri, akım veya gerilim formatında tanımlıdır ve süreklilik gösteren bir değişim özelliği gösterirler. Tamamen analog elemanlar ile modellenen YSA yapıları ilk bölümde sözü edilen sorunlar nedeniyle sağlıklı ve duyarlı bir çözüm oluşturmakta büyük zorlukları beraberinde getirmektedir. Bu yüzden sayısal kontrollü analog donanımlar daha elverişli bir çözüm olarak yerini korumaktadır.

Bu çalışmada analog giriş çıkış özelliği taşıyan sayısal kontrollü bir YSA donanımı PIC mikro denetleyicisi çevresinde oluşturulmuştur. Tasarlanan donanım, eğitim aşaması tamamlanmış her hangi bir ileri besleme YSA yapısının gerçek zamanda çalışan bir uygulamaya aktarılmasını sağlamaktadır.

IV.3.1 Tek Mikrodenetleyicili YSA Donanımı

Şekil-IV.8'de gösterilen model ihtiyaca göre analog giriş ve çıkış imkanı sunmaktadır. Bir 16F877 PIC mikrodenetleyicisinde bulunan ADC modülü maksimum 8 adet analog girişe izin vermektedir fakat daha fazla analog giriş istenen uygulamalarda dışarıdan eklenecek bir analog çoklayıcı(multiplexer) ile analog giriş sayısı arttırılabilir. Denetleyici çıkışında da analog sinyal elde etmek amacıyla DAC entegreleri kullanılmıştır. Bir kod çözücü(decoder) vasıtasıyla mevcut analog çıkışlar ihtiyaca göre arttırılıp azaltılabilir. Gerektiği taktirde DAC çıkışlarındaki analog sinyalin kalıcı olması için DAC girişlerine uygun hafıza birimleri eklenebilir.



Şekil-IV.8 Tek mikrodnetleyicili YSA modeli

Şekil-IV.9’da gösterilen devredeki YSA donanım modeli 1 analog, 2 sayısal giriş ve 1 analog çıkıştan oluşmaktadır. Devrede, 1 adet mikrodnetleyici 1 adet DAC entegresi 3 adet opamp entegresi, mikrodnetleyici ve diğer entegrelerin çalışması için gerekli olan kondansatörler, dirençler ve kristal kullanılmıştır. Devreye birde devir sayısını ayarlamak ve işlemlerin başlatılıp durdurulması sağlamak için LCD ve butonlar eklenmiştir.

Mikrodnetleyicinin A portu analog veri girişi ve mikrodnetleyici üzerindeki ADC modülü için referans değeri sağlamak üzere ayarlanmıştır. B portu LCD kontrolü için kullanılırken C portu ise devir sayısı ayarında ve motor kontrolünü başlatıp durdurmaya yarayan butonların bağlandığı porttur. Mikrodnetleyici içindeki YSA algoritması tarafından üretilen kontrol sinyali de D portuna bağlı DAC entegresi sayesinde motor sürücü devresine uygulanır.

Devrede standart 2x16 alfanümerik LCD, DAC entegresi olarak 8 bit sayısal girişli tek kutuplu AD558 isimli çevirici entegre kullanılırken U5 ve U3 OPAMP’ları tampon devresi olarak U4 OPAMP’ı ise motordan alınan devir bilgisinin istenen seviyeye yükseltmek amacıyla kullanılmaktadır.

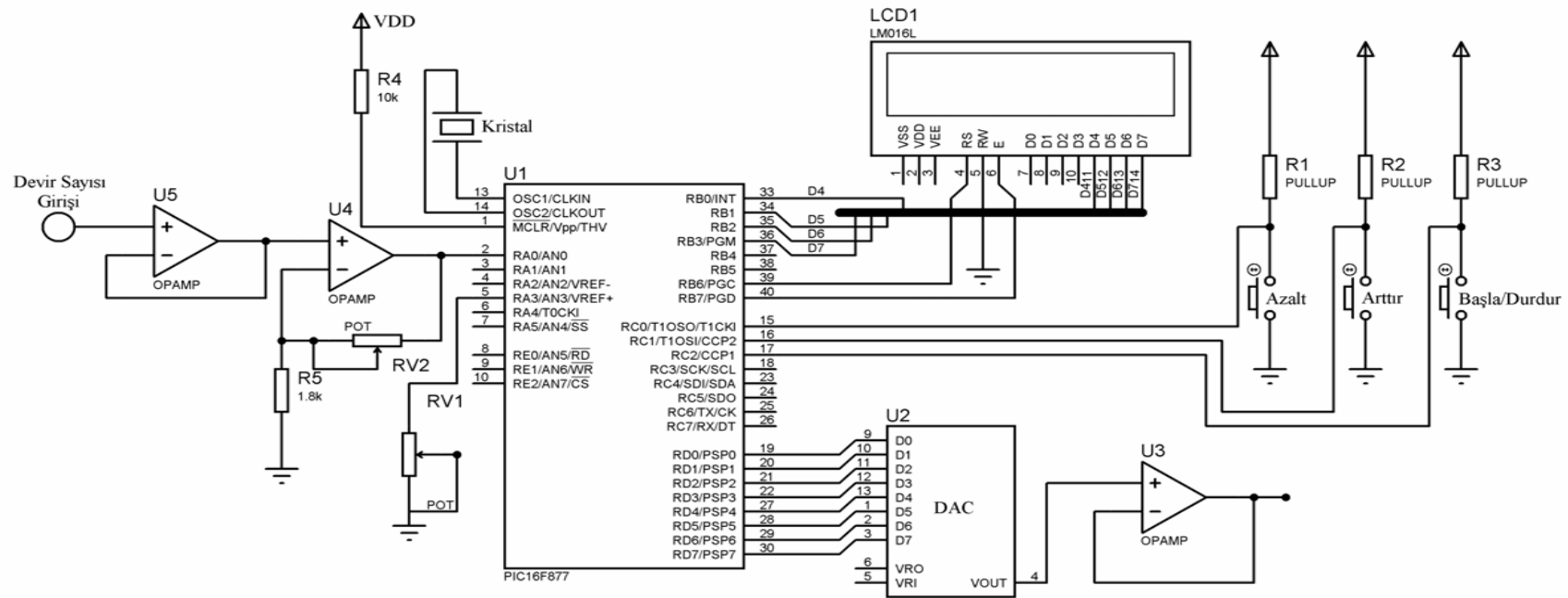
A portunun RA3(Vref+) girişine bağlı potansiyometre ile ADC modülü için gerekli olan referans voltaj değeri ayarlanmaktadır.

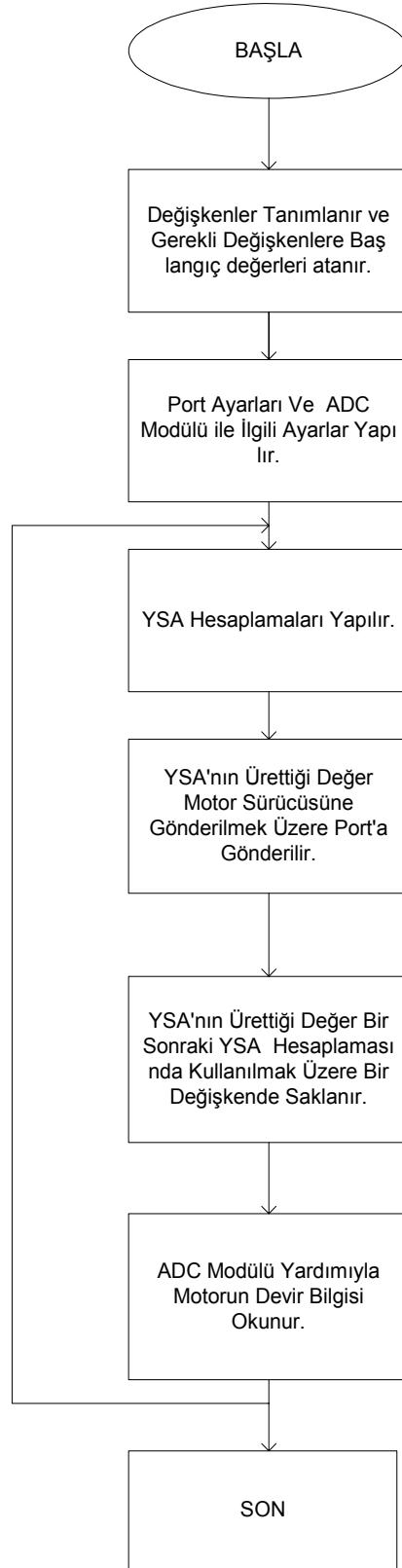
PIC entegresinin analog girişine maksimum 5 V gerilim uygulanabilir. Bu nedenle ADC referans voltajı olarak 4V seçilmiştir. YSA’nın analog girişine uygulanacak gerilim önceden opamplarla maksimum değerinde 4V olacak şekilde

yükseltildikten sonra mikrodnetleyiciye uygulanır. Devrede kullanılan DAC entegreside 0-10V arasında çıkış verebilen 8 bit çözünürlüklü bir entegredir.

IV.3.1.1 Devrenin Çalışması

Devrede ilk olarak RV1 potansiyometresi ile uygun referans gerilimi ve RV2 potansiyometresi ile istenen kazanç değeri ayarlanır. Motordan alınan devir bilgisi U5 opampının evirmeyen girişine uygulanır. U3 OPAMP'ının çıkışı ise motor sürücü devresinin girişine bağlanır. Devreye enerji verildiğinde mikrodnetleyici motor kontrolüne başlaması için Başla/Durdur butonuna basılmasını bekler. Bu butona basmadan önce azalt ve arttır butonlarıyla istenen devir sayısı belirlenebilir. Başlat butonuna basılmasıyla beraber mikrodnetleyici YSA algoritmasını işlemeye başlar ve işlem sonucunda üretilen YSA çıkışı bir katsayı ile çarpılarak DAC entegresine üzerinden motor sürücü devresine uygulanır





Şekil-IV.10 Tek mikrodenetleyicili YSA donanımında kullanılan programın akış diyagramı

IV.3.1.2 Tek Mikrodenetleyici Kullanılan Modelin Program Akış

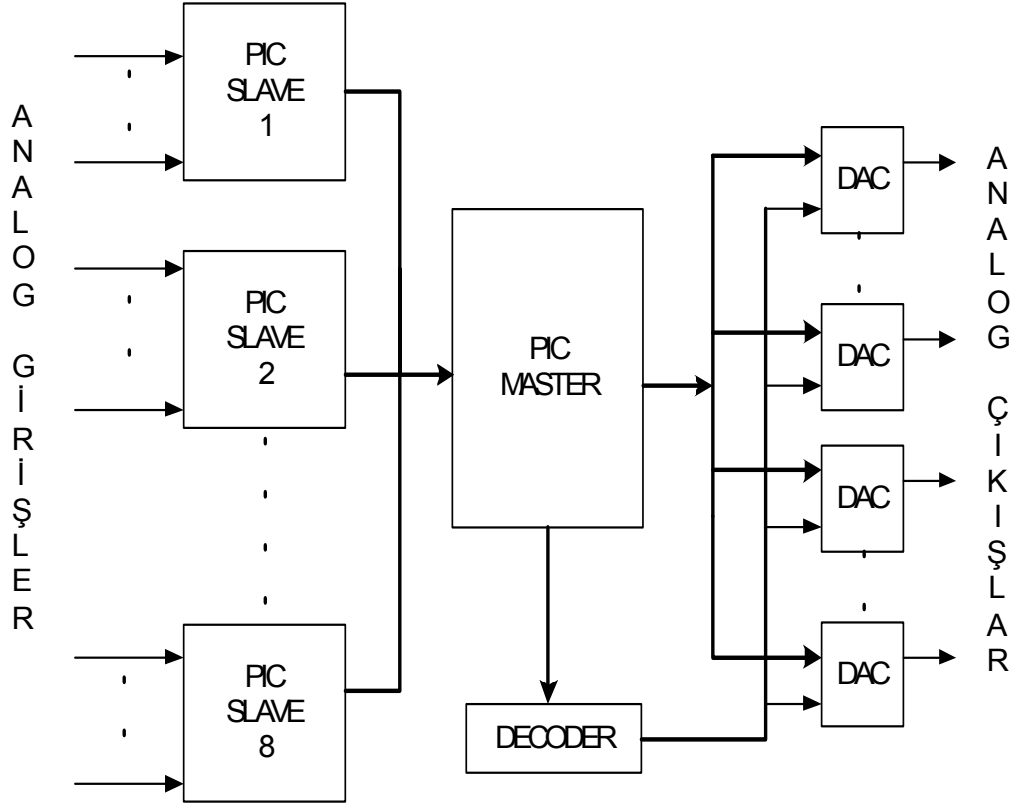
Diyagramı Açıklaması

Şekil-IV.10'da tek mikrodenetleyicili YSA donanımında kullanılan programın akış diyagramı gösterilmektedir. İlk olarak gerekli değişken tanımlamaları yapılmaktadır. Hangi değişkenler float tipine sahip olacak, hangi değişkenler integer tipinde hangilerinin char tipinde olacağı belirlenir. Gerekli olan sabit değerler ve ağırlık değerleri uygun değişkenlere atanır. Mikrodenetleyicinin A portunun RA0 pini analog giriş, RA3 pini ise ADC modülü için referans voltaj girişi olarak ayarlanmıştır. Bu ayarlamalar ADCON1 ve ADCON0 kaydedicileri yardımıyla yapılmaktadır. B portu ve D portundan veri gönderileceği için TRISB, TRISD kaydedicilerinden bu portlar üzerinden çıkış almak üzere ayarlanmıştır. Gerekli olan port ayarları ve başlangıç ayarları yapıldıktan sonra YSA hesaplamaları yapılır. Yapılan hesaplama sonucu, D portu üzerinden DAC'ye gönderilir. Belirli bir süre gecikme uygulandıktan sonra ADC modülü vasıtasıyla motor devir bilgisi okunur ve tekrardan gerekli olan hesaplamaları yapmak üzere YSA hesaplamalarının yapıldığı kısma geri döner.

IV.3.2 Çoklu Mikrodenetleyicili YSA Donanımı

Şekil-IV.12'deki model daha yüksek sayıda hücre elemanlı ağ yapısını oluşturmak için önerilen YSA donanımıdır. Tek μC ile fazla sayıda hücre elemanı gerektiğinde, işletim programı çevrim süresi nöron sayısına paralel olarak artmaktadır. Paralel veri işleme sayesinde tek mikrodenetleyicinin yavaş kaldığı ve boyut olarak yetersiz geldiği alanlarda çoklu mikrodenetleyicili YSA donanımı rahatlıkla kullanılabilir. Nöronlar köle PIC'lere dağıtılarak işlem süresi kısaltılmaktadır. Ayrıca boyut olarak yetersiz gelen noktalarda sistemdeki mikrodenetleyici sayısı artırılarak istenilen sayıda nöron içeren YSA donanımları gerçekleştirilebilir.

Yapılan devrede mikrodenetleyiciler arasındaki iletişim I2C seri haberleşme ile yapılmaktadır. Böylece 10 bit adresleme kullanılırsa teorik olarak 1024 adet mikrodenetleyiciyi birbirine bağlayıp çalıştırmak mümkün olacaktır.



Şekil-IV.11 Çoklu mikrodnetleyicili YSA modeli

Yapılan devrede 1 adet analog giriş, 2 adet sayısal giriş ve bir analog çıkış kullanılmıştır.

1 efendi ve 3 köle mikrodnetleyiciden oluşan devre şeması Şekil-IV.12’de verilmiştir. Mikrodnetleyici olarak PIC 16F877 kullanılmıştır. Mikrodnetleyiciler arasında haberleşme I2C standartında yapılmaktadır [17]. I2C haberleşmesi için PIC’in RC3 pini saat sinyali için RC4 pini ise veri hattı için kullanılmaktadır. PIC’in belirtilen uçları I2C için özelleştirilmiş pinlerdir. Devredeki R1 ve R2 dirençleri I2C haberleşmesi için pull-up dirençleri olarak görev yapmaktadır. Motor devir sayısını ayarlamak için kullanılan butonlar efendi PIC’in RA0, RA1 ve RA2 girişlerine bağlıdır. Başlat butonu kontrol işlemini başlatmaya ve durdurmaya yaramaktadır. Köle PIC’lerin RA0 girişleri analog giriş olarak ayarlanmış ve motordan alınan devir bilgisi buradan PIC’lere uygulanır. ADC için gerekli olan referans gerilimi ise köle PIClerin RA3 girişlerinden uygulanmaktadır. Efendi PIC’in ürettiği kontrol sinyalinin analog bilgiye dönüştürmek amacıyla AD 558JN isimli Analog Device firmasının DAC entegresi kullanılmaktadır. Bu DAC entegresi efendi PIC’in D portuna bağlıdır. Bu DAC entegresinin çıkışına ise tampon olarak görev yapması için

741 OPAMP entegresi bağlanmıştır. LCD olarak standart 2x16 alfanümerik LCD kullanılmıştır. LCD ile haberleşme 4 bit üzerinden yapılmaktadır. Efendi PIC'in RB4,RB5,RB6 ve RB7 pinleri LCD'ye veri göndermek amacıyla kullanılırken RA3 pini LCD'nin RS(Register Select) ucuna RA5 pini ise LCD'nin E(Enable) ucuna bağlanmıştır.

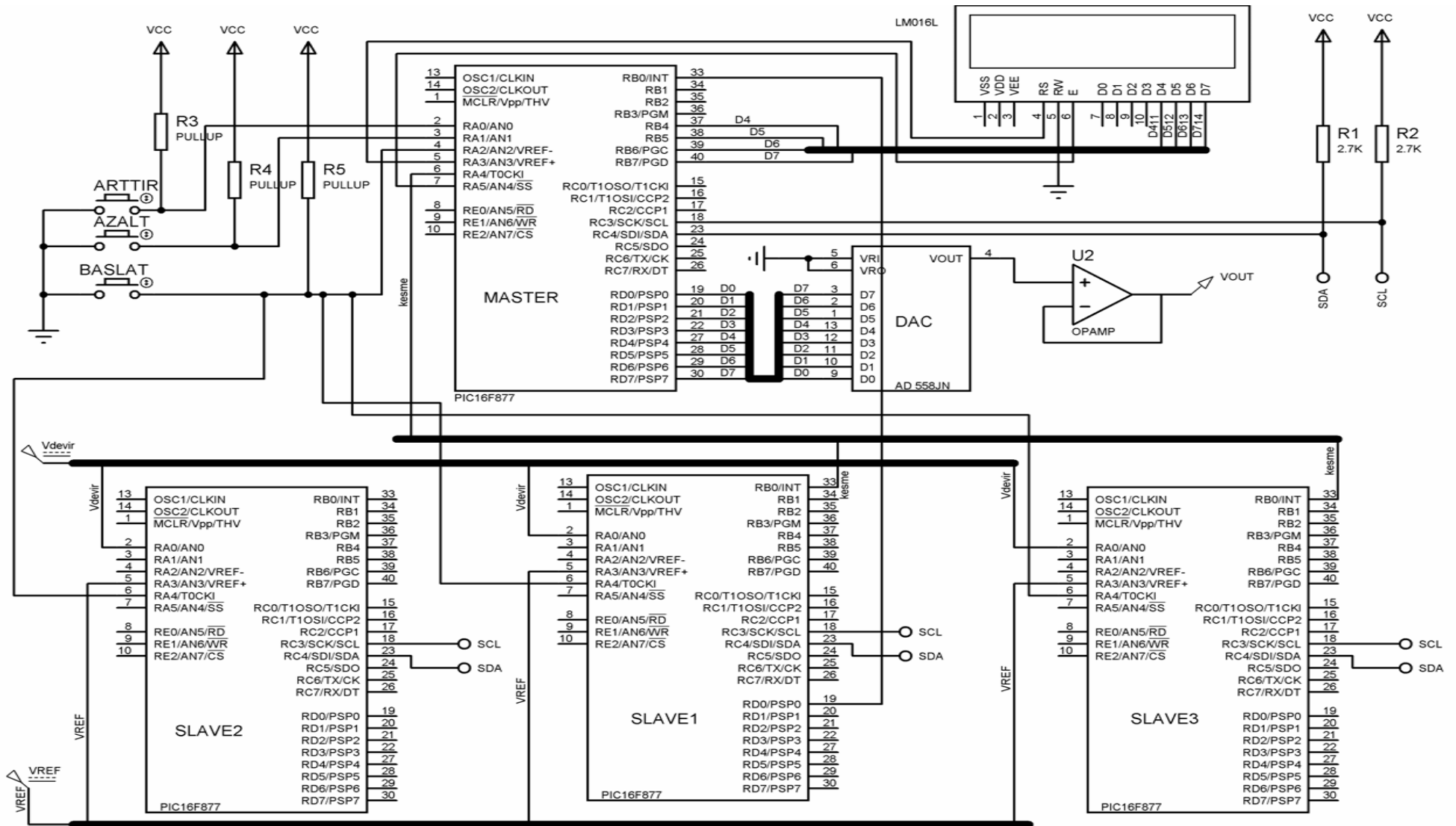
IV.3.2.1 Devre Çalışması

İlk olarak efendi PIC Başlat butonundan başlat sinyalini bekler. Başlat sinyali verilmeden kullanıcı istediği referans değere “arttır” ve “azalt” butonlarıyla ulaşabilir. Efendi başlat sinyalini alınca gerekli hazırlıkları yaparak I2C veri haberleşmesini başlatır. Köle PIC'ler programlandıkları gibi veriyi işlerler ve Efendi PIC'e işlem sonuçlarını bildirirler. Efendi PIC'te kendisinde yazılı olan algoritmaya bağlı olarak istenen sinyali üretir. Bu işlem tekrar başlat butonuna basılarak durdurulabilir ve yeni devir sayısı ayarlanabilir.

IV.3.2.2 Çoklu Mikrodenetleyicili YSA Donanımında Kullanılan Programların Akış Diyagramı Açıklaması

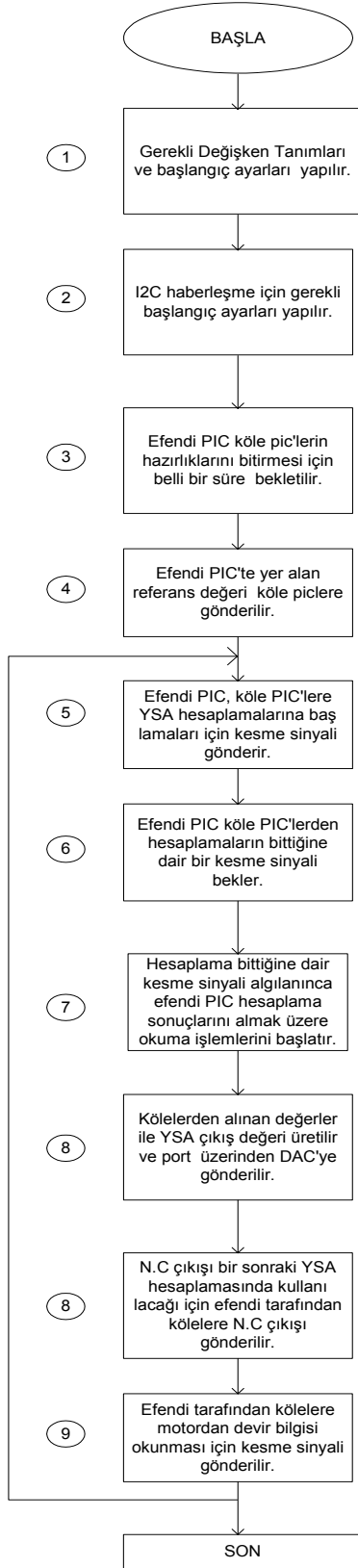
Çoklu mikronetleyicili YSA donanımında kullanılan programların akış diyagramı Şekil-IV.13'te gösterilmektedir. 1 efendi ve 3 köle PIC'ten oluşan modelde temelde iki farklı yazılım yer almaktadır. Birincisi efendi için yazılan program diğeri köle PIC'ler için yazılan programdır. Köle PIC'lerde yer alan programlar birbirleriyle aynı sadece sahip oldukları I2C adresleri farklıdır ayrıca 1.köle PIC'in programına, yapılan işlemlerin bittiğine dair efendi PIC'e kesme göndermesi için ayrı bir kod eklenmiştir. Akış diyagramlarına göre program işleyişi şekil IV.15'de görülebilir.

1. Efendi PIC ve köle PIC'lerin değişken tanımlamaları ve başlangıç atamaları yapılır.
2. Haberleşme esnasında ve veri alışı verişinde kullanılacak port ayarları ve I2C ayarları yapılır.

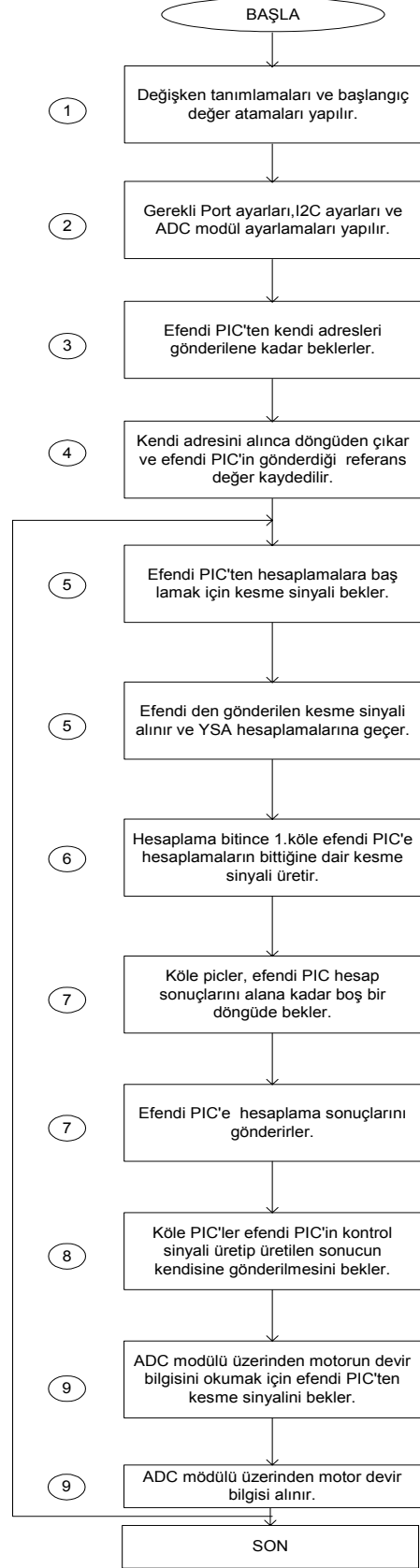


Şekil-IV.12 Çoklu mikrodnetleyicili YSA donanımı

3. Efendi PIC, köle PIC'lerin I2C hazırlıklarının bitmesi için belli bir süre bekler. Köle PIC'ler ise I2C hazırlıklarını bitirip efendi PIC'in kendi adreslerini göndermelerini beklerler.
4. Efendi her bir köle PIC'in adresini gönderir ve ardından referans değerini köle PIC'lere yollar. Köle PIC'ler ise kendi adresleri efendi PIC tarafından gönderilince efendi PIC'in gönderdiği 3 bayt uzunluğundaki referans değeri sırayla bir diziye atılarak saklanır.
5. Efendi PIC köle PIC'lere aynı anda YSA hesaplamalarına başlamaları için bir kesme sinyali gönderir. Köle PIC'ler bu kesme sinyalini alır almaz gerekli YSA hesaplamalarına başlar.
6. Efendi PIC köle PIC'lerin hesaplamalarını bitirmesini boş bir döngü içinde bekler bu esnada köle PIC'ler gerekli hesaplamaları yaparlar ve sadece 1. köle efendi PIC'e hesaplamaların bittiğini bildirmek üzere kesme sinyali gönderir.
7. Efendi PIC 1. köle PIC'ten hesaplamaların bittiğini bildiren kesme sinyalini algılayınca köle PIC'lerin yaptığı hesaplama sonuçlarını almak için okuma prosedürünü başlatır. Bu esnada köle PIC'ler efendi PIC'in sonuçları alması için boş bir döngüde beklerler.
8. Efendi PIC, köle PIC'lerden alınan hesaplama sonuçlarını kullanarak kontrol sinyalini üretir. Ve ardından YSA modelinde bir önceki sonuç bir sonraki kontrol çıkışında kullanılacağı için köle PIC'lere üretilen kontrol sinyali gönderilir. Köle PIC'lerde efendi PIC'in gönderdiği kontrol sinyalini bir sonraki işlemde kullanmak üzere saklar.
9. Efendi PIC köle PIC'lere aynı anda motordan gelen devir bilgisini almaları için kesme sinyali gönderir. Köle PIC'ler ise efendi PIC'ten kesme sinyalini alınca ADC modülü yardımıyla devir bilgisini okur. Ve tekrardan YSA hesaplamaları başlamak üzere 5.adımda belirtilen kısma dönerler.



a) Efendi PIC akış diyagramı



b) Köle PIC akış diyagramı

Şekil-IV.13- Çoklu mikrodeneleyicili YSA donanımı akış diyagramları a) Efendi PIC diyagramı b) Köle PIC akış diyagramı

IV.4 MİKRODENETLEYİCİLER İLE ETKİN YSA DONANIMI OLUŞTURMA

YSA’larda mikrodnetleyicilerin kullanılabilirliğine baktığımızda bazı sorunlarla karşılaşmaktadır. Bu sorunların başında mikrodnetleyicilerin işlem hızı gelmektedir. Mevcut mikrodnetleyicilerin çalışma hızı maksimum 40 mhz’dir. Bu hızdaki bir mikrodnetleyici yoğun hesap ve işlem gerektiren resim işleme gibi uygulamalarda yeterli olmayacaktır. Mikrodnetleyicilerle ilgili diğer bir sorun ise mikrodnetleyicilerin hafıza kısıtlamalarıdır.

Mikrodnetleyicilerde karşılaşılan bu problemlere çözüm paralel veri işleyen mikrodnetleyici ağları ile sağlanabilir.

Yapılan çalışmada mikrodnetleyicilerde işlem hızını ve hafıza doluluk oranını etkileyen belli başlı faktörler şu şekilde belirtilebilir.

- Kullanılan programlama dili(makine dili, c derleyici vb.)
- İşlemlerde kullanılan sayı formatı (tamsayı, kayan nokta(16 bit, 24 bit vb.))
- YSA’da kullanılan aktivasyon fonksiyonu (doğrusal, sigmoid vb.)
- İşlemlerde kullanılan algoritma
- Kullanılan YSA modeli (ileri yönlü vb..)

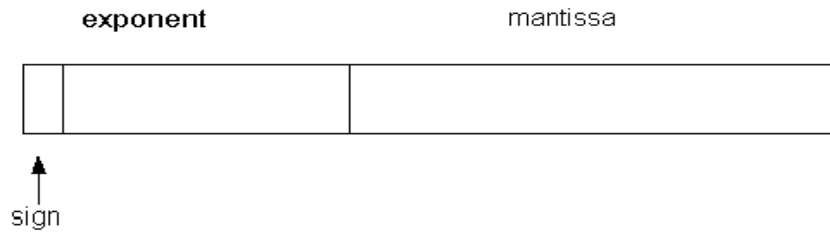
IV.4.1 Kullanılan Programlama Dili

Programda kullanılan programlama dili hafızayı doğrudan etkileyen etmenlerden biridir. Makine dilinde yazılan bir program ile derleyiciler ile yazılan programlar arasında, programlar aynı işlemi gerçekleştiriyor olsa bile hafıza kullanım farklılıkları olmaktadır. Makine dilinde yazılan program derleyicilerde yazılan programlara göre daha az hafıza kullanmaktadır. Fakat programları derleyici ile yazmak makine dilinde yazmaktan çok daha kolay ve kısa zamanda gerçekleşebilmektedir. Ayrıca makine dilinde yazılan uzun ve karmaşık programlarda hata arama işlemi içinden çıkılmaz bir hal alabilmekteyken derleyicilerde yazılan programlarda hata arama işlemi daha kısa sürede

sonuçlanmaktadır. Yapılan çalışmada programlama dili olarak HI-TEC PICC derleyicisi kullanılmıştır.

IV.4.2 Kullanılan Sayı Formatı

Tüm bilgisayar sistemlerinde gerçekleştirilen matematiksel işlemlerde kayan nokta aritmetiği kullanılmaktadır ve kullanılan her programlama dili ilk başlarda birbirlerinden farklı kayan nokta aritmetiği formatına sahiptir. Bu karışıklığı sona erdirmek üzere IEEE 1985 yılında bir format oluşturdu ve oluşturulan bu format diğer üreticilerin büyük çoğunluğu tarafından kabul edilerek IEEE 754 formatı standart olarak kabul edilmiştir. 1987 yılında bu format biraz daha geliştirilerek IEEE 854 standardı belirlenmiştir. IEEE 854 formatı aynı zamanda IEEE 754 formatını desteklemektedir. Bir çok bilgisayar ve derleyicileri float tipi değişkenleri IEEE 754 standardını kullanarak temsil etmektedir.



Şekil-IV.14 IEEE 754 kayan nokta formatı

IEEE 754 standardı tek hassasiyetli 32 bit ve çift hassasiyetli 64 bit olmak üzere iki format belirlemiştir. Tek hassasiyetli formatta bit dağılımı şöyledir. İşaret biti 1(31) bit, üstel kısmı 8(30-23) bit ve 23(22-0) bit mantissa kısmını oluşturur. Çift hassasiyetli formatta ise bit dağılımı şöyledir. İşaret biti 1(63) bit, üstel kısmı 11(62-52) bit ve 52 (51-0) bit mantissa kısmını oluşturur.

Örneğin 0,125 sayısı $1/10 + 2/100 + 5/1000$ şeklinde onluk tabanda ifade edilirken 0,001 sayısı $0/2 + 0/4 + 1/8$ olarak ikilik tabanda ifade edilir ve bu iki sayı birbirine eşittir. Aralarındaki tek fark biri onluk tabana göre yazılmış diğeri ikilik tabanda yazılmıştır. Kayan nokta aritmetiği bilgisayarlarda ikilik tabanda temsil edilir. Standartta üstel kısmın negatif ve pozitif değerler alabilmesi için tek hassasiyetli formatta 127 sayısına çift hassasiyetli formatta ise 1023 sayısına bias edilmiştir.

IEEE 754 formatında sayının onluk düzene çevrilmesi aşağıdaki formatta yapılır.

$$\text{işaret} * 2^{\text{üst}} * \text{kesirli kısım}$$

Tablo-IV.3 IEEE 754 formatında tek hassasiyetli ve çift hassasiyetli sayıların bitsel dağılımı

	İşaret	Üstel	Ondalık	Bias
Tek Hassasiyetli	1 [31]	8 [30-23]	23 [22-00]	127
Çift Hassasiyetli	1 [63]	11 [62-52]	52 [51-00]	1023

Aşağıdaki örnekten daha kolay anlaşılabilir.

```
|S|  Üst  | Kesir  |
|1|10000101|110110101000000000000000|
```

formatında hafızada bulunan bir sayının ondalık karşılığını şu şekilde buluruz. İşaret biti '1', üstel kısmı 133 olarak görülüyor. Bias değeri 127 olduğuna göre $133-127=6$ üst değerini elde ederiz. Kesirli kısmı ise

$1,110110101000000000000000=1,853515125$ değerini elde ederiz.

$-1*2^{(133-127)}*1,853515125= -118,625$ olarak hafızadaki ikilik düzendeki sayıyı ondalık sisteme çevirebiliriz.

12,0820 onluk tabanındaki ondalık sayısının ikilik düzende gösterilmesini inceleyelim. Sayımız pozitif olduğu için işaret biti '0'. İşarete bakmadan sayımızı doğrudan ikilik düzende yazarsak , $1100,00010101 = 1,10000010101*2^3$ sayısını elde ederiz. Virgülden sonraki kısmı 23 biti tamamlamak üzere '0' ile doldurulur. 100000101010000000000000 . Üstel kısmı 3, bu sayı bias değeri ile toplanıp ikilik sisteme çevrildiğinde: $127+3=130=10000010$

```
|S|  Üst  | Kesir  |
|0|10000010|100000101010000000000000|
```

sonuç olarak yukarıdaki sayıyı elde ederiz.

IV.4.2.1 Özel Sayılar

1.a Sıfır Sayılarının Temsili: Standartta sıfır sayısı pozitif sıfır ve negatif sıfır olarak iki farklı şekilde ifade edilmiştir. Sayıdaki bütün bitler '0' ise o zaman sayımız pozitif sıfır olarak kabul edilir eğer sadece işaret biti 1 diğerleri 0 ise sayımız negatif sıfır olarak kabul edilir.

S	Üst	Kesir
0	00000000	000000000000000000000000

 pozitif sıfır

S	Üst	Kesir
1	00000000	000000000000000000000000

 negatif sıfır.

1.b Sonsuz: Sayıların Temsili Eğer sayımız sonsuz ise üstel ifadenin bütün bitleri 1 olarak temsil ediliyor. Eğer sayı $-\infty$ ise işaret biti 1, $+\infty$ ise işaret biti 0 olur.

S	Üst	Kesir
0	11111111	000000000000000000000000

 pozitif sonsuz

S	Üst	Kesir
1	11111111	000000000000000000000000

 negatif sonsuz

1.c Tanımsız ifadelerin: Matematikte bazı ifadeler tanımsızdır. Mesala $0/0$ veya $\infty + (-\infty)$ gibi ifadeler tanımsız ifadelerdir. Bu tür tanımsız ifadeler standartta kesir kısmının hepsi 0'dan farklı olmak üzere aşağıdaki şekilde temsil edilir.

S	Üst	Kesir
X	11111111	XXXXXXXXXXXXXXXXXXXXXXXXXX

 Tanımsız ifade

IV.4.2.2 PICC Derleyicisinde Kullanılan 24 Bit Kayan Nokta

Aritmetiğin İncelenmesi

PICC derleyicisi IEEE 754 standartının değiştirilmiş formatını kullanmaktadır. PICC derleyicisinde 32bit veya 24 bit float tiplerinden birini seçebiliriz. Mikrodenetleyicimizde program hafızası ve veri hafızası sınırlı olduğu ve 24 bit float tipinin hassasiyeti yeterli duyarlılığı sağladığı için derleyicinin 24 bit float tipi seçilmiştir.

arasında simetriktir ve hesaplama yapılırken bir yan kenarın hesaplanması yeterli olmaktadır. Diğer yarısı 1'den çıkarılarak bulunabilir.

Sigmoid fonksiyonu için FPGA ve diğer sayısal ortamlarda kullanmak için çeşitli sigmoid fonksiyonu yaklaşımları önerilmiştir. Bunlardan en önemlilerinden bahsedilmektedir.

IV.4.3.1 Sigmoid Fonksiyon Yöntemleri

Sigmoid fonksiyonunu kısmi doğrusal yaklaşımlarını (Piecwise Linear approximation) aşağıdaki şekillerde inceleyebiliriz. Başarılı bir yaklaşımın kriterleri hız, doğruluk ve gerekli olan alan olarak sıralanabilir.

Kısmi doğrusal metodu, karmaşık işlemleri daha az karmaşık hale getiren bir metottur. Buradaki amaç minimum hata ile birlikte minimum işlem gerektiren yaklaşımlar getirmektedir.

1.a Kural Tabanlı Yaklaşımlar

Myers ve Hutchinson, “**A-law companding**” tekniği kullanarak bir yaklaşım geliştirmişlerdir. Buna göre sigmoid fonksiyon doğrusal segmentlere ayrılmış ve ayrılan segmentler 2 nin katı olacak şekilde ifade edilmiştir. Böylece çarpma gibi matematiksel işlemler kaymalı kaydediciler sayesinde gerçekleştirilebilir [18].

1.b Alippi ve Storti Gajani Yaklaşımı

Bu yaklaşımda kırılma noktaları tamsayı olarak seçilmiştir. Ve y-değerleri ikinin katı olacak şekilde ayarlanmıştır. X ekseninin negatif kısmı için (x) ifadesi x ekseninin tamsayı kısmını tanımlarken ondalık kısmı ise χ olarak gösterilir ve Eşitlik-IV.1 ile ifade edilir [19].

$$\chi = x + |(x)| \quad (IV.1)$$

sigmoid fonksiyonunun yaklaşımı genel olarak Eşitlik (IV.2) ifade edilir.

$$y = \frac{1}{2} + \frac{\chi}{4} \quad (IV.2)$$

Eşitlik-IV.2 sadece toplama ve kaydırma işlemi gerektirdiği için sayısal uygulamalar için uygundur.

1.c PLAN Yaklaşımı

Sigmoid fonksiyonu için düzenlenen bir yaklaşım metodudur [20]. Bakış tabloları ile yapılan yaklaşım basit olmasına karşın çok fazla alan gerektirmektedir. Ve her işlem elemanı için bakış tablosu yapmak pratik değildir. Sigmoid fonksiyonunun doğrudan hesaplanması iki çarpma ve iki toplama işleminden oluşmaktadır ki bu işlem sayısal olarak çok yavaş olmakta ve çok fazla yer kaplamaktadır. Kısmi doğrusal(Piecewise) analizde ise eğri üzerinde belli noktalar belirlenerek bu noktalar hafızada saklanır ve bu noktalar arası doğrusal arbulma işlemi(interpolation) uygulanır.

PLAN (piecewise linear approximation of a nonlinear function) yaklaşımına göre sigmoid fonksiyonunda(Eşitlik-IV.3) yer alan x değerine bağlı olarak y değerinin alacağı değerler Tablo-IV.5'e göre hesaplanır.

$$y = \frac{1}{1 + e^{-x}} \quad (IV.3)$$

Tablodan da anlaşılabilirceği gibi yapılan işlemlerde kullanılan değerler 2'nin üssü biçimde ifade edilmiştir. Bu sayede işlemler toplama ve kaydırma işlemleri ile kolayca yapılabilir bir hale getirilmiştir.

Tablo-IV.5 PLAN yaklaşımı hesap tablosu

İŞLEM	ŞART
Y=1	$ X \geq 5$
$Y=0.03125* X +0.84373$	$2.375 \leq X < 5$
$Y=0.125* X +0.625$	$1 \leq X < 2.375$
$Y=0.25* X +0.5$	$0 \leq X < 1$
Y=1-Y	X<0

Kısmi doğrusal(Piecewise linear) yaklaşımların içinde en iyi çözümü sunan PLAN yaklaşımıdır [21]. Bu nedenle klasik sigmoid hesaplamasıyla PLAN sigmoid yaklaşımı arasındaki farkı görmek için sigmoid fonksiyonunun bir çevriminin ne kadar sürdüğünü ilk önce klasik yöntemlerle hesapladım daha sonrada PLAN yaklaşım yöntemiyle aynı işlevi gerçekleştirdim. Sonuç olarak klasik sigmoid hesaplama için Program ROM'da 1425 word , Program RAM'de ise 48 byte lık yer tutmaktadır. PLAN yaklaşımı ile ise bu değerler Program ROM'da 533 word, Program RAM'de ise 19 byte'a düşmüştür. Fonksiyonun çıkış vermesi için geçen süre klasik yöntemle 10282 saykıl gerektirirken, PLAN yaklaşımı ile ise 2200 saykıl gibi bir süre içinde işlem tamamlanmaktadır.

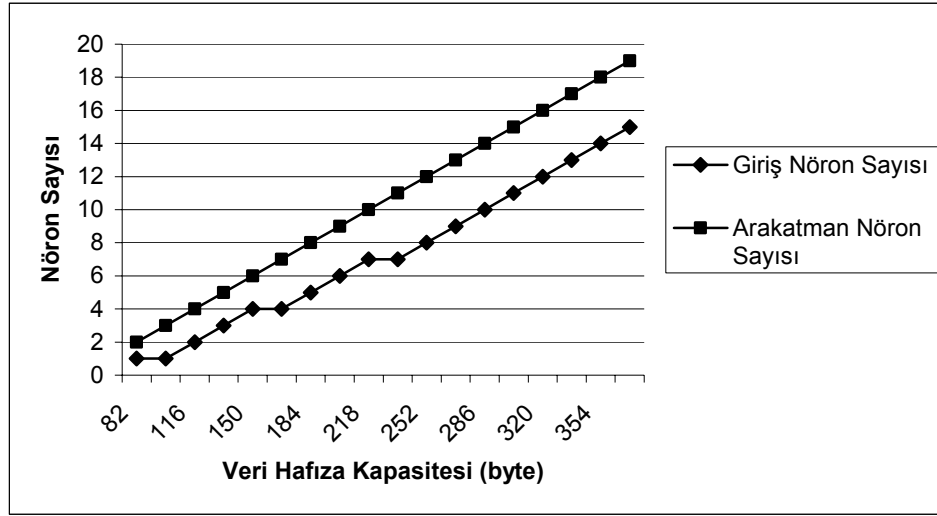
IV.4.4 Kullanılan Algoritma

Mikrodenetleyicide kullanılan program hafızasını ve veri hafızasını etkileyen en önemli faktörlerden bir diğeri ise programda kullanılan algoritmadır. Yapılan çalışmada iki adet birbirinden farklı algoritma kullanılmaktadır. İlki sadece tek PIC ile yapılan program algoritması diğeri ise aralarında I2C haberleşme standardında haberleşen 1 efendi ve 3 adet köleden oluşan program algoritmasıdır.

IV.5 PIC MICRODENETLEYİCİSİNDE NÖRON DEĞİŞİMİNİN ETKİLERİ

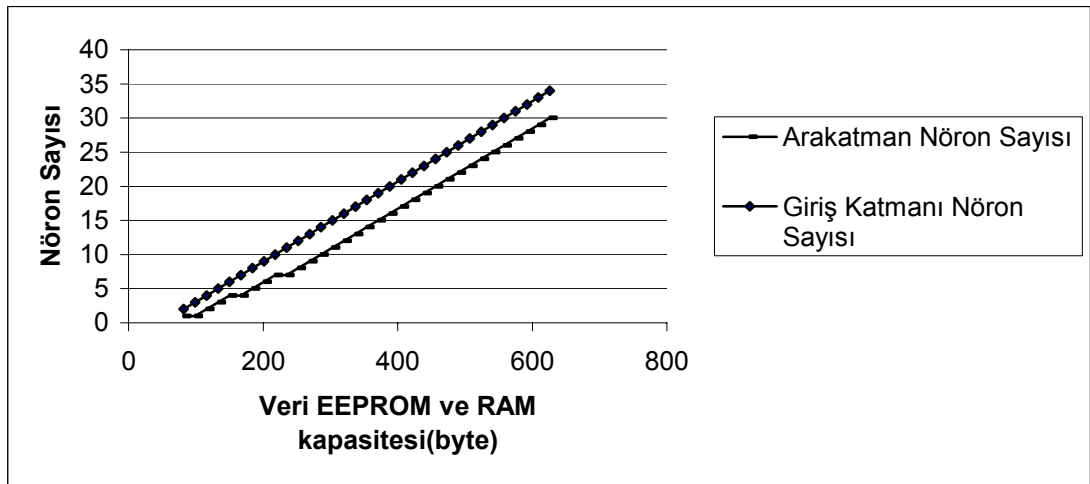
Uygulamada gerçekleştirilen program ile ilgili elde edilen sonuçlar şöyledir. Temel olarak 1 giriş nöronu, 2 arakatman nöronu ve 1 çıkış nöronundan (1-2-1) oluşan YSA modeli program hafızasında 1974 word'lük yer işgal ederken, veri hafızasında 82 byte'lık yer işgal etmektedir. Bu temel YSA modeline göre giriş katmanındaki bir nöron artışı program hafızasını yaklaşık 150 word artırırken veri hafızasında 20 byte'lık artışa neden olmaktadır. Arakatmanda meydana gelen 1 nöronluk artış, program hafızasını 10 word arttırırken veri hafızasında yaklaşık 15 byte'lık artışa neden olmaktadır. Çıkış katmanındaki 1 nöronluk artış ise program hafızasında 200 word'lük artış yaparken, veri hafızasında değişim olmamaktadır. Veri hafızasında değişimin olmamasının nedeni ise çıkış için yeni bir değişken tanımlanması gerekmediği içindir.

Bu belirtilen değerlere göre bir PIC'te kullanılacak nöron sayısı nöronun kullanılacağı katmana göre değişiklik göstermektedir.



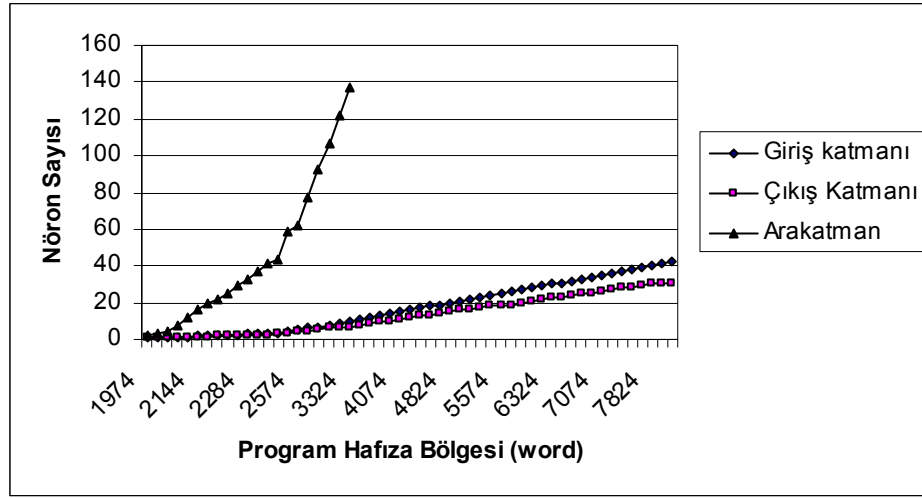
Şekil-IV.15 RAM değişimine bağlı olarak nöron sayısı değişimi

RAM kapasitesine bağlı olarak oluşturulabilecek nöron sayısı Şekil-IV.15'deki grafikte ifade edilebilir. Grafikte çıkış katmanı için bir grafik çizdirilmemiştir. Bunun nedeni çıkış katmanı hesaplanırken yeni değişken tanımlamaya gerek kalmamasıdır. Grafiğe göre maksimum giriş nöron sayısı 16F877 için (16F877'nin RAM kapasitesi 368 byte) 15 nöron olarak görülmekte tabiki bu halde oluşacak model 15 giriş nöronu, 2 arakatman nöronu ve çıkış katmanından oluşacaktır. Oluşturulabilecek arakatman nöron sayısı ise 19 olarak görülmektedir bu durumda oluşacak modelde 1 giriş nöronu, 19 arakatman nöronu olacaktır.

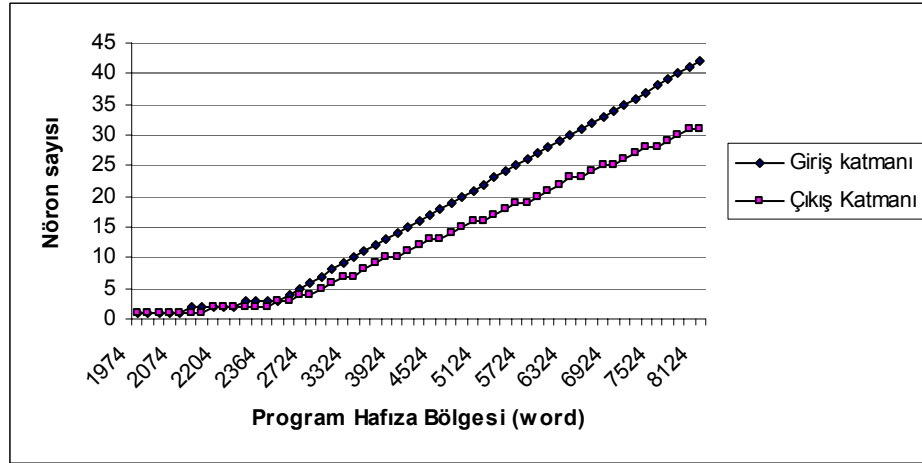


Şekil-IV.16 RAM ve EEPROM değişimine bağlı nöron değişimi

Veri EEPROM ve RAM kapasitesine(256+368=624 byte) bağlı olarak oluşturulabilecek nöron sayı adetleri Şekil-IV.16'daki grafikte gösterilmektedir. Grafiğe göre oluşturulabilecek maksimum giriş nöronu 35'tir. Bu şartlarda oluşturulacak model 35 giriş nöronu,2 arakatman nöronu ve 1 çıkış nöronu bulundurabilir . Arakatman nöron sayısında 30 olarak görülmektedir. Bu şartlarda oluşturulacak YSA modeli ise 1 giriş katmanı 30 arakatman ve 1 çıkış katmanından oluşacaktır.



Şekil-IV.17 Program hafıza değişimine bağlı nöron değişimi



Şekil-IV.18 Program hafıza değişimine bağlı nöron değişimi (arakatman çıkartılmış)

ROM kapasitesine bağlı olarak oluşturulabilecek nöron modeli grafikleri Şekil-IV.17 ve Şekil-IV.18'de verilmiştir. Bu grafiğe göre giriş katmanı nöron sayısı PIC 16F877 için(8K program hafıza) 40 adet giriş nöronu oluşturulabilir. Çıkış

katmanı olarak 30 adet çıkış nöronu olabilir. Arakatman sayısı ise teorik olarak 620'yi bulmaktadır.

Uygulamada kullanılan 3-5-1 modelde kullanılan veri hafızası 185 byte, program hafızasında 2253 word olarak bulunmuştur.

Bu verileri göre kullanılan 16F877 mikrodnetleyicisinde program hafızasına ve veri hafızasına bağılı oluşturulabilecek nöron sayıları yaklaşık olarak birbirlerine eşittir.

Kullanılan programların hız olarak değerdendirdiğimizde 1-2-1 nöron model yapısına sahip yapay sinir ağıının sin15.c'deki(EK-A) gibi bir programla yapılırsa YSA'nın sonuç üretmesi için geçen saykıl sayısı 39423 olarak bulunmuştur. 4MHz kristal kullanılırsa bu süre yaklaşık 39 ms olarak bulunur. 20 MHz'lik kristal kullanılırsa yaklaşık 7.9 ms olmaktadır.

Tek mikrodnetleyicinin kullanıldığı donanımda kullanılan programda YSA'nın kontrol sinyali üretmesi için geçen saykıl sayısı 90311 olarak bulunmuştur. 20 Mhz kullanılırsa bu süre yaklaşık 18 ms olarak hesaplanır. Çoklu mikrodnetleyicili YSA donanımında ise aynı işlevi yapan programda 4 Mhz'lik kristal kullanılırsa sistemin kontrol sinyali üretmesi için geçen süre yaklaşık 50 ms olmaktadır. 20 Mhz'lik kristal kullanılırsa bu süre 13 ms'ye kadar inmektedir. Fakat bu sistemde nöron sayısının artışı sistemin kontrol sinyalini üretmesi için geçen sürede fazla bir artış oluşturmayacaktır.

BÖLÜM V

DENEYSEL ÇALIŞMALAR

Bu bölümde PIC mikrodenetleyicileriyle yapılmış olan YSA donanımlarının farklı uygulamalarla ilgili elde edilen sonuçları incelenecektir. İlk olarak 3 farklı fonksiyonun doğrusallaştırılması ile ilgili uygulamadan, ikinci olarak DC motor kontrolünün YSA donanımı ile gerçekleştirilmesine dair uygulamadan bahsedilecektir [22].

V.1 DOĞRUSALLAŞTIRMA UYGULAMASI

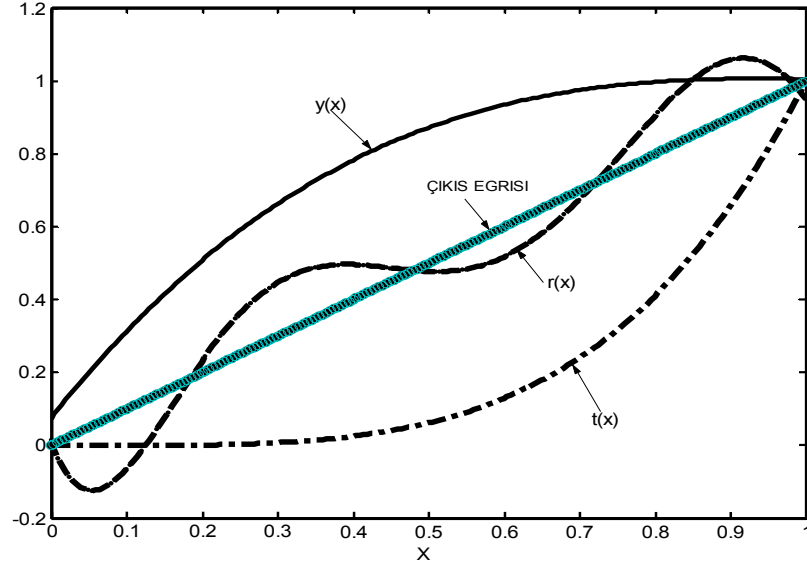
Bu uygulamada üç adet farklı doğrusal olmayan fonksiyon, YSA girişine uygulanmakta ve YSA çıkışında doğrusal bir fonksiyon elde edilmek istenmektedir. Şekil-V.1’de YSA girişine uygulanan 3 adet doğrusal olmayan fonksiyon ve çıkışta istenen doğrusal fonksiyon görülmektedir.

$$y(x) = 0.0738 + (2.618 * x) - (2.399 * x^2) + (0.713 * x^3)$$

$$t(x) = x^4$$

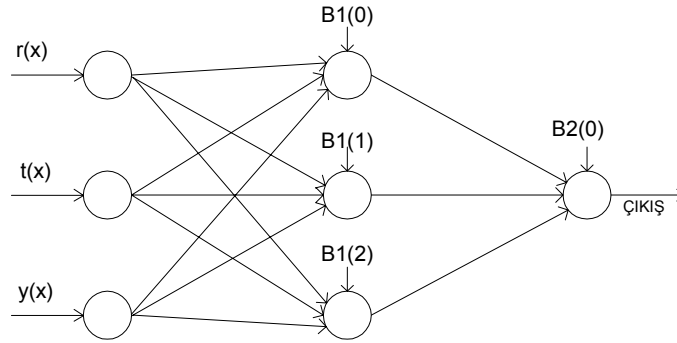
$$r(x) = (0.0169) - (5.689 * x) + (69.62 * x^2) - (245.673 * x^3) + (384.88 * x^4) - (274.26 * x^5) + (72.055 * x^6)$$

$y(x)$, $t(x)$ ve $r(x)$ fonksiyonları YSA girişine uygulanmış ve çıkışta doğrusal bir fonksiyon elde edilmiştir.



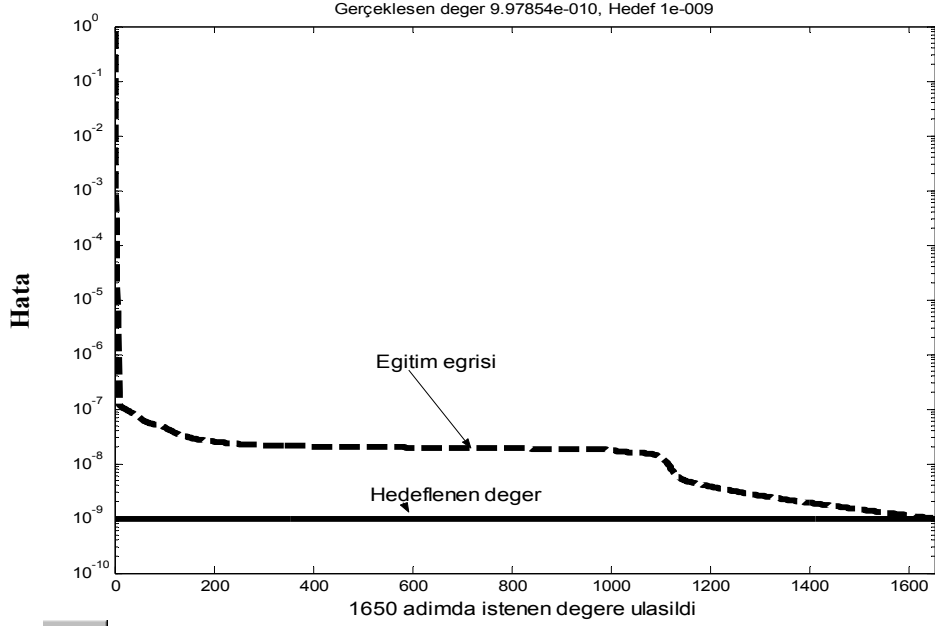
Şekil-V.1 Doğrusallaştırılması yapılacak olan $y(x)$, $r(x)$, $t(x)$ fonksiyonları ve istenen çıkış eğrisi

YSA modeli Şekil-V.2’de gösterilmektedir.



Şekil-V.2 Doğrusallaştırma işleminde kullanılan YSA modeli

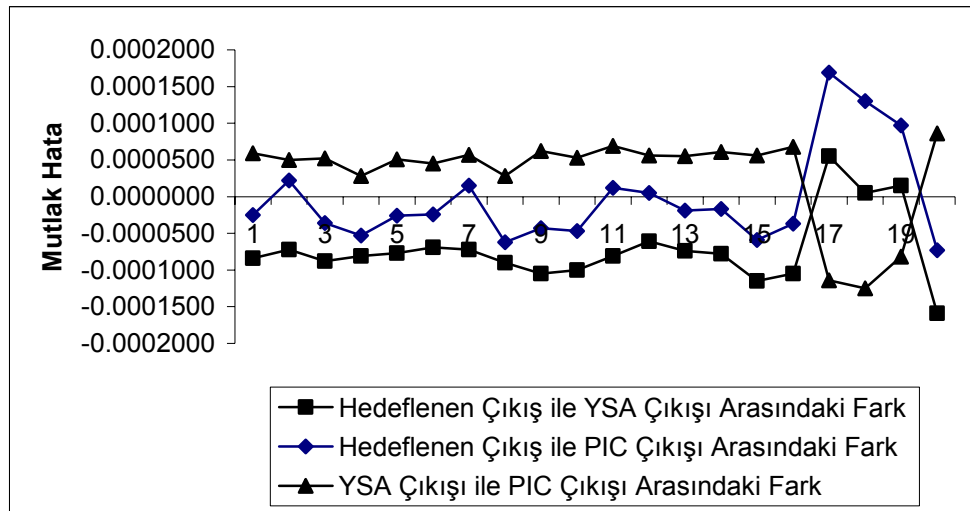
Şekil-V.2’de gösterilen ağ MATLAB yazılımı ile eğitilmiştir [23]. Eğitimde kullanılmak üzere hazırlanan 200 adet veriden 50 tanesi seçilmiştir. Eğitim sonucu Şekil-V.3’de gösterilmektedir.



Şekil-V.3 MATLAB yazılımıyla yapılan ağ eğitimi sonucu

Eğitim aşamasında öğrenme oranı 0.01, momentum sabiti 0.8, hata oranı ise $1e-9$ olarak seçilmiştir. MATLAB programı istenen ağı 1650 adımda eğitmiştir. YSA'da arakatman aktivasyon fonksiyonu olarak tansig fonksiyonu çıkış katında ise lineer aktivasyon fonksiyonları kullanılmıştır. Eğitim fonksiyonu olarak 'trainlm' fonksiyonu kullanılmıştır. Ağ ileri yönlü bir ağ olarak tanımlanmıştır.

MATLAB programı ile elde edilen ağırlık ve eşik değerleri mikrodnetleyicide yazılan programda kullanılmıştır. Mikrodnetleyicide oluşturulan YSA çıkışında elde edilen değerler ile program çıkışında elde edilen değerler Tablo-V.1'de verilmiştir. Çıkışlar arasındaki farklar Şekil-V.4'de gösterilmektedir.



Şekil-V.4 Mikrodnetleyici ve bilgisayar ortamında alınan YSA sonuçlarının karşılaştırılması

Tablo-V.1 Mikrodenetleyici ile bilgisayar ortamından alınan YSA çıkış değerleri

Hedeflenen Çıkış Değeri	YSA Çıkış Değeri	PIC Çıkış Değeri
0.01	0.010084	0.010025
0.15	0.150072	0.149978
0.21	0.210088	0.210036
0.305	0.305081	0.305053
0.325	0.325077	0.325026
0.355	0.355069	0.355024
0.41	0.410072	0.409985
0.49	0.49009	0.490062
0.545	0.545105	0.545043
0.565	0.5651	0.565047
0.615	0.615081	0.614988
0.67	0.670061	0.669995
0.695	0.695074	0.695019
0.725	0.725078	0.725017
0.785	0.785115	0.785059
0.81	0.810105	0.810037
0.865	0.864945	0.864831
0.89	0.889995	0.88987

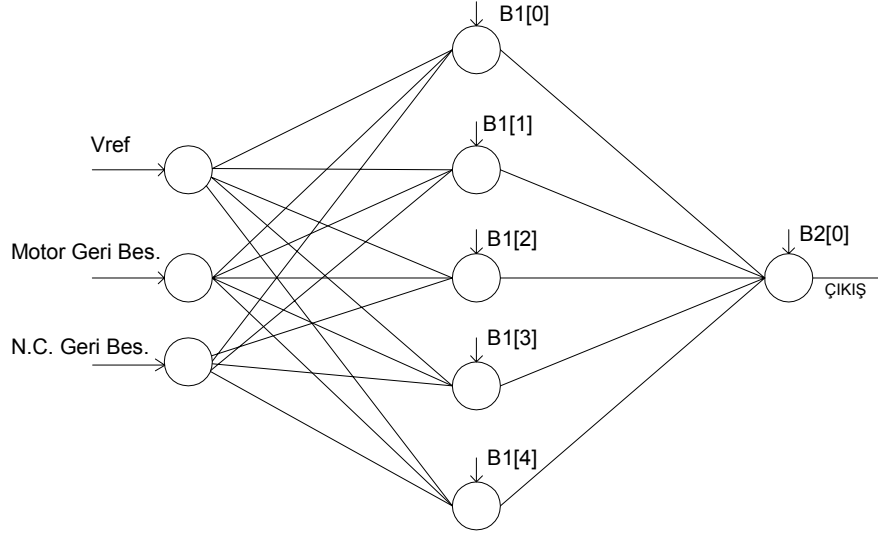
PIC çıkışında değerleri görebilmek için PIC mikrodenetleyicisinin D Portuna 8 bit'lik bir DAC entegresi bağlanmıştır. Bu entegrenin analog çıkışında ölçülen değerler ve olması gereken değerler Tablo-V.2 'de gösterilmektedir. Kullanılan DAC entegresi 0-10V çalışma aralığına sahiptir. 1LSB değişimini $10000\text{mV}/256 = 39.0625\text{mV}$ olarak hesaplanır.

Tablo-V.2 Hesaplanan DAC çıkış voltajı ile ölçülen DAC çıkış voltajları

Hedeflenen çıkış	8 Bite dönüştürülmüş veri	DAC Çıkışı (hesaplanan)(mV)	DAC çıkışı(ölçülen)(mV)	Fark(mV)
0.01	3	99.609375	79	20.60938
0.15	38	1494.140625	1487	7.140625
0.21	54	2091.796875	2073	18.79688
0.305	78	3038.085938	3013	25.08594
0.325	83	3237.304688	3210	27.30469
0.355	91	3536.132813	3522	14.13281
0.41	105	4083.984375	4068	15.98438
0.49	125	4880.859375	4850	30.85938
0.545	139	5428.710938	5397	31.71094
0.565	144	5627.929688	5631	3.07031
0.615	157	6125.976563	6100	25.97656
0.67	171	6673.828125	6647	26.82813
0.695	177	6922.851563	6843	79.85156
0.725	185	7221.679688	7194	27.67969
0.785	200	7819.335938	7819	0.335938
0.81	207	8068.359375	8054	14.35938
0.865	221	8616.210938	8600	16.21094
0.89	227	8865.234375	8834	31.23438

V.2 DC MOTOR KONTROL UYGULAMASI

DC motor kontrolünde kullanılan YSA modeli Şekil-V.5’de verilmiştir. Modelimiz üç nöronlu oluşan giriş katmanı, 5 nöronlu oluşan bir arakatman ve 1 nöronlu oluşan çıkış katmanından oluşmaktadır.



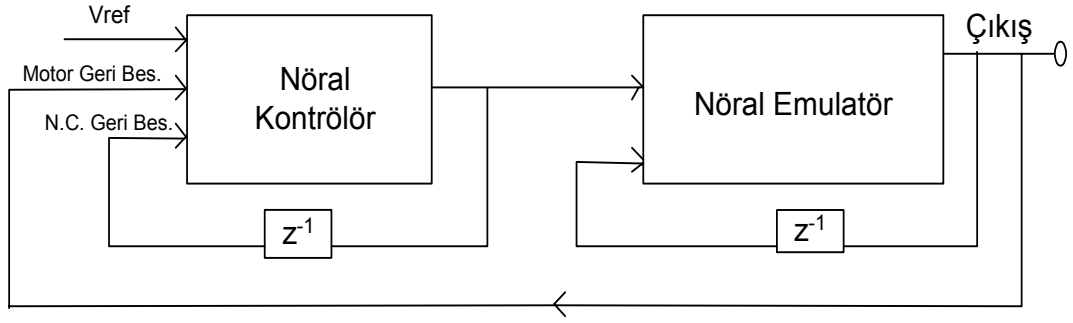
Vref : Motorun referans bilgi girişi

N.C. Geri Bes.: Nöral kontrolörümüzün bir önceki çıkış değeri

Motor Geri Bes: DC motordan alınan devir bilgisi

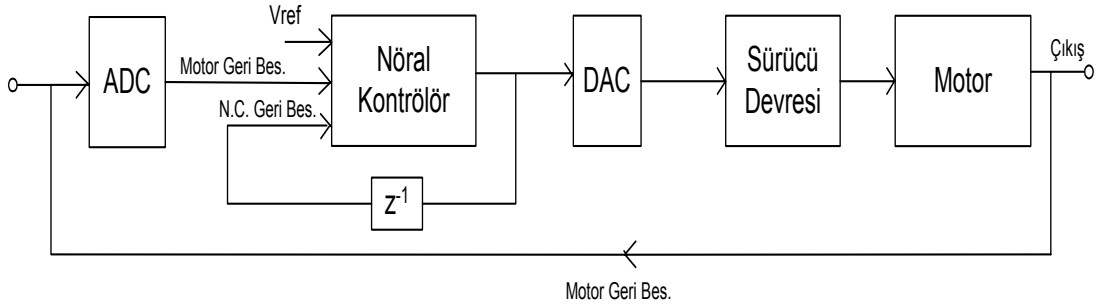
Şekil-V.5 DC motor kontrolünde kullanılan YSA modeli

DC motor kontrolünde kullanılacak ağı eğitimi bilgisayar ortamında gerçekleştirilmiştir [24]. Ağı eğitimi motoru taklit etmesi amacıyla bir nöral emulatör kullanılmıştır. Şekil-V.6’da eğitimde kullanılan model görülmektedir [25]. İlk olarak nöral emulatör deney setinde kullanılan DC motor gibi davranacak şekilde eğitilmiş onun ardından nöral kontrolörün eğitimi gerçekleştirilmiştir.



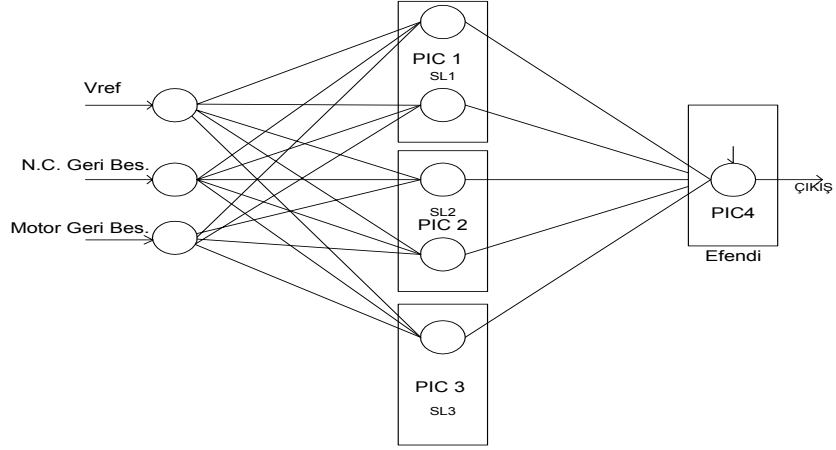
Şekil-V.6 DC motor kontrolünde kullanılan YSA'nın eğitim modeli

Kontrolü yapılan sistemin blok diyagramı Şekil-V.7'de gösterilmektedir. Motor olarak DC motor kullanıldı. Motorun üretmiş olduğu analog devir bilgisi gerilimi ADC yardımıyla sayısal bilgiye çevrildikten sonra nöral kontrolöre uygulanır. Nöral kontrolörde işlenen veri tekrardan analog bilgiye çevrilmek üzere DAC devresine uygulanır. DAC çıkışında 0-10V arasında kontrol sinyaline bağlı olarak bir gerilim elde ederiz. Bu gerilimin motoru sürebilmesi amacıyla 0-110V DC gerilime yükseltmek üzere sürücü devresine uygulanır. Blok diyagramı gösterilen sistemi gerçekleştirmek üzere iki farklı devre kuruldu.



Şekil-V.7 DC motor kontrol uygulaması blok diyagramı

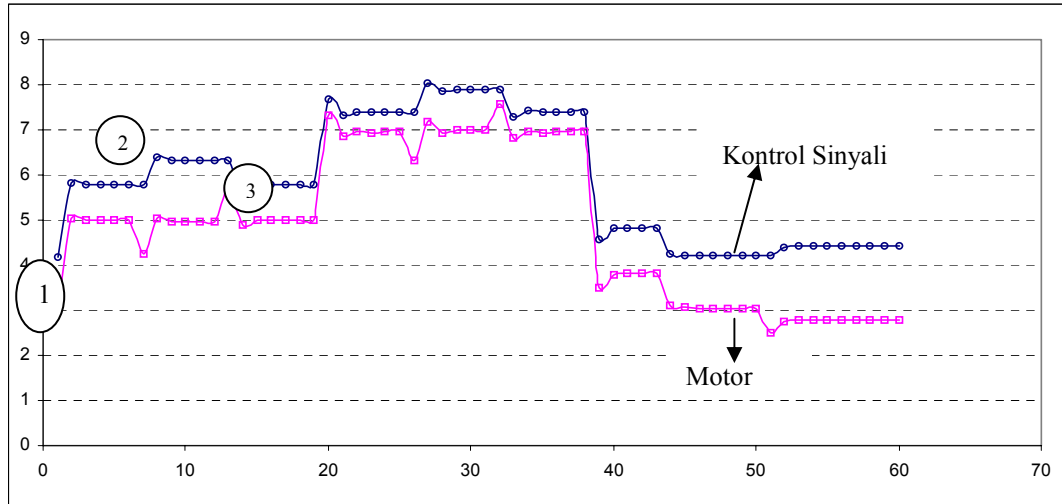
Çoklu mikrodenetleyicili YSA donanımında 4 adet mikrodenetleyici kullanılmıştır. Uygulaması yapılan devrede nöronların PIC'lere dağılımı Şekil V-8'de gösterilmektedir. Buna göre 1. ve 2. PIC'te ikişer nöron 3.PIC'te ise 1 nöron yer almaktadır. 4. PIC ise efendi işlevini görmektedir.



Şekil-V.8 Nöronların PIC'lere dağılımı

Uygulamada kullanılan mikrodnetleyici de oluşturulan model ve kullanılan ağırlıklar ilk olarak bilgisayar ortamında benzetim yoluyla test edilmiştir. Benzetim sonucu YSA modelinin motoru istenilen seviyede kontrol ettiği görülmüştür. Benzetim sonucu alınan veriler Şekil-V.9'da gösterilmektedir.

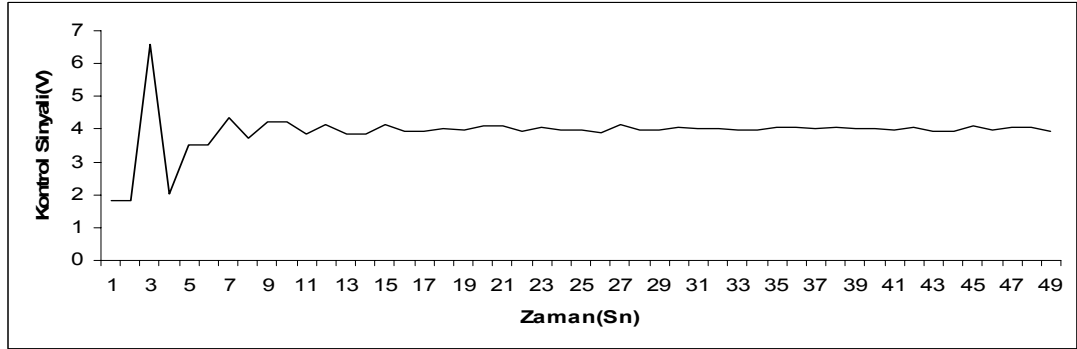
Şekil-V.9'da 1 ile belirtilen yerde motora yük uygulanmıştır. Uygulanan yük bağlı olarak kontrol sinyali de yükselmiştir. 2 ile belirtilen yerde motora uygulanan yük kaldırılmış kontrol sinyali yükün kaldırılmasıyla beraber azalmıştır. 3 ile belirtilen yerde sistemin referans değeri değiştirilmiştir. Şekilde kontrol sinyali ile motor davranışı arasındaki seviye farkı motoru kontrol etmek amacıyla kullandığımız sürücülerden kaynaklanmaktadır.



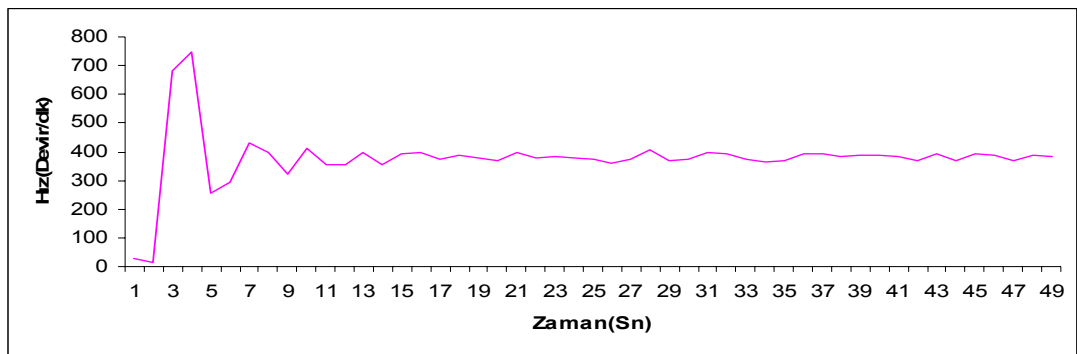
Şekil-V.9 DC motor kontrolünde kullanılan YSA'nın simülasyon grafiği

Bilgisayar ortamında başarıyla gerçekleştirilen uygulama laboratuvar ortamında bölüm IV'te anlatılan donanım ile denenmiş ve aşağıdaki sonuçlar alınmıştır.

Referans değeri 400 devire ayarlanmıştır. Bu değerde motordan alınan devir bilgisi ve motor sürücüsüne uygulanan kontrol sinyalinin bilgileri aşağıda verilmektedir.



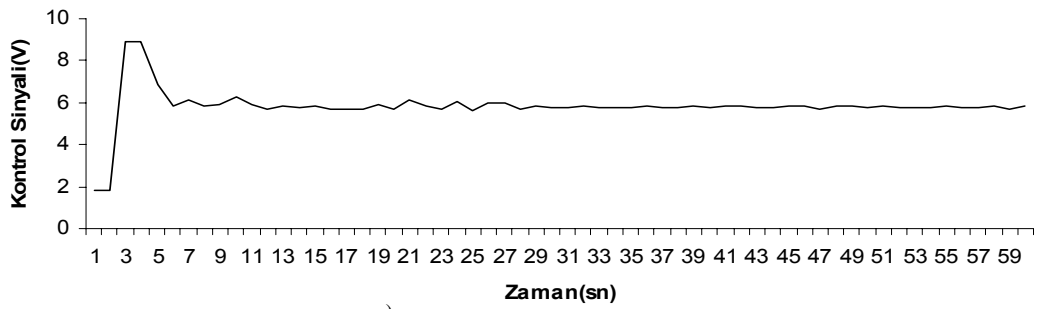
a)



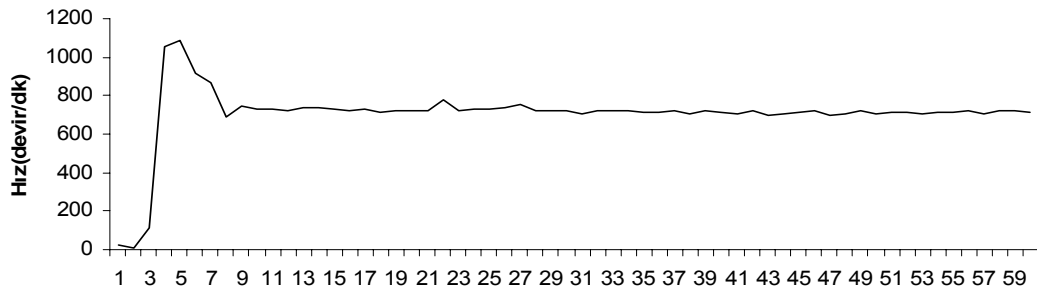
b)

Şekil-V.10 400 devir için a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

Referans değeri 700 devire ayarlanmıştır. Bu değerde motordan alınan devir bilgisi ve motor sürücüsüne uygulanan kontrol sinyalinin bilgileri aşağıda verilmektedir.



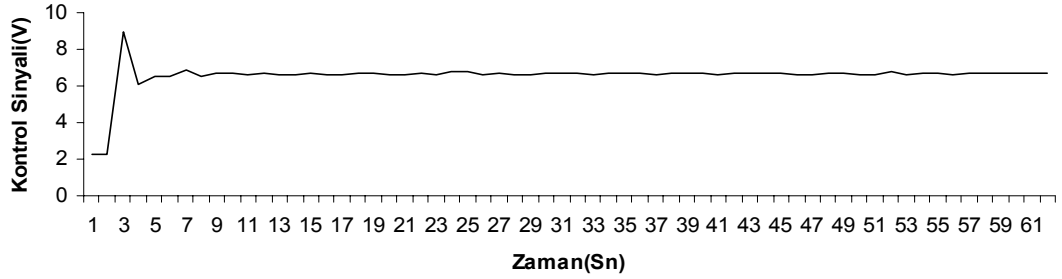
a)



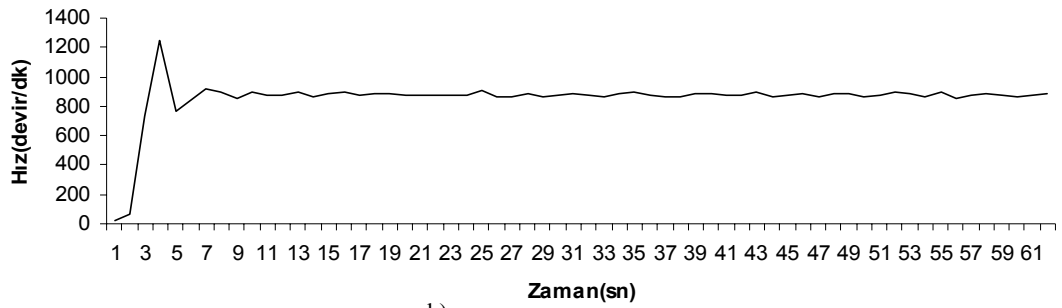
b)

Şekil-V.11 700 devir için a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

Referans değeri 900 devire ayarlanmıştır. Bu değerde motordan alınan devir bilgisi ve motor sürücüsüne uygulanan kontrol sinyalinin bilgileri Şekil-V.12’de verilmektedir.



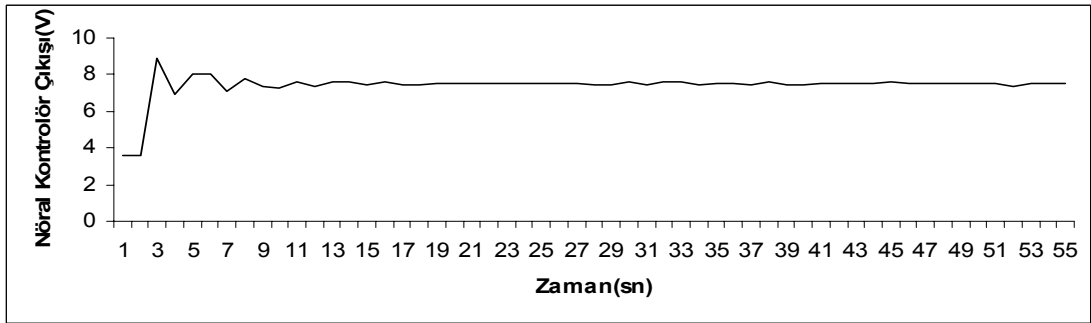
a)



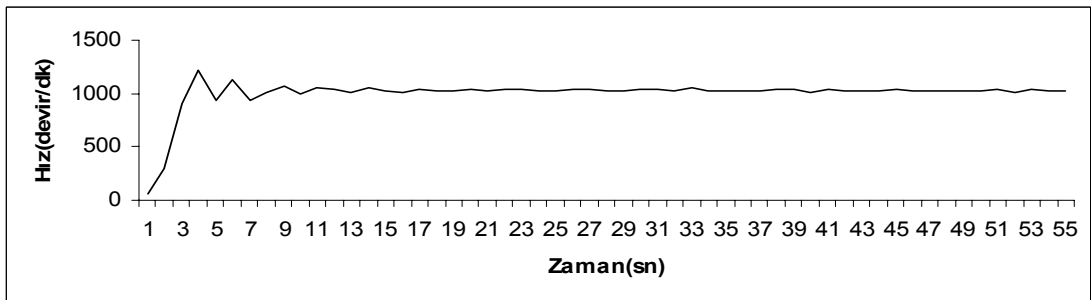
b)

Şekil-V.12 900 devir için a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

Referans değeri 1100 devire ayarlanmıştır. Bu değerde motordan alınan devir bilgisi ve motor sürücüsüne uygulanan kontrol sinyalinin bilgileri aşağıda verilmektedir.



a)



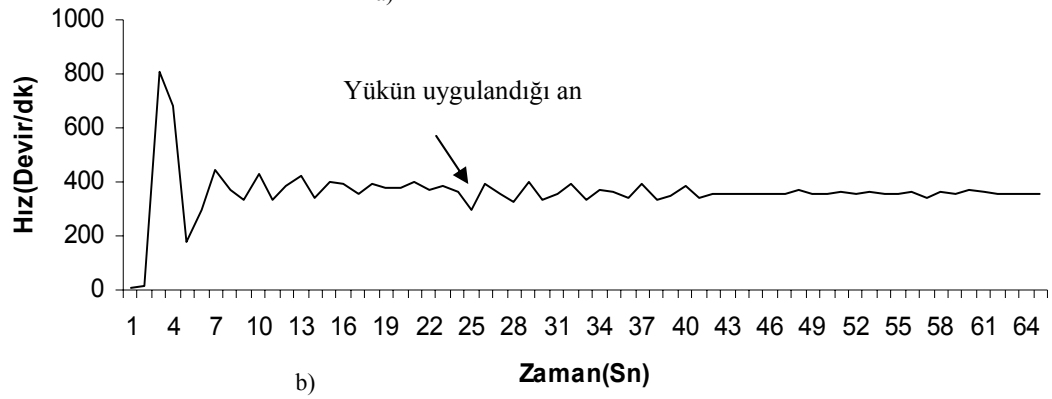
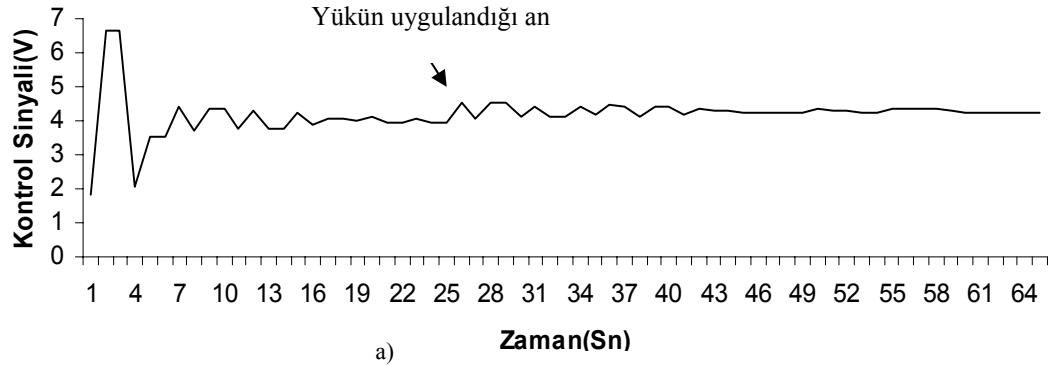
b)

Şekil-V.13 1100 devir için a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

Uygulama sonucu düşük devirlerde örnekleme zamanının etkisiyle sistemin yüksek devirlere göre daha geç sürede kararlı hale geldiği görülmüştür.

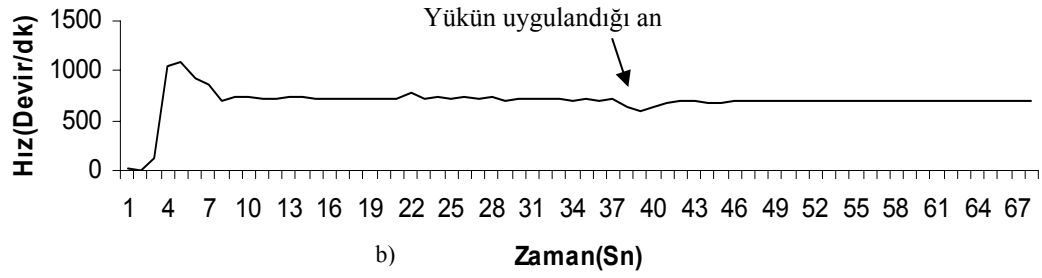
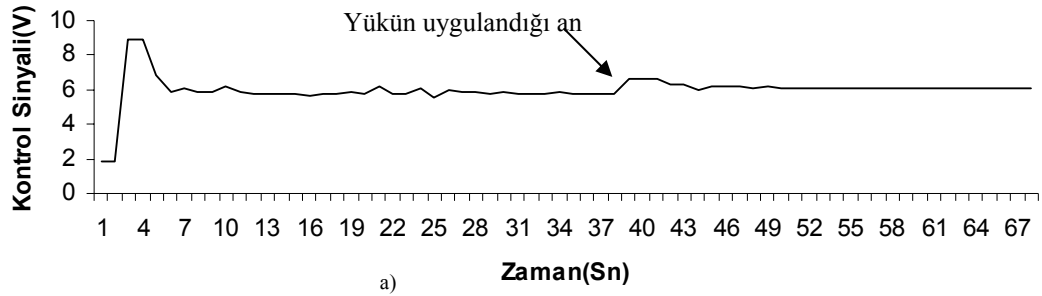
Kontrolörün performansını yük uygulayarak test ettiğimizde aldığımız değerler şöyledir.

400 devirde



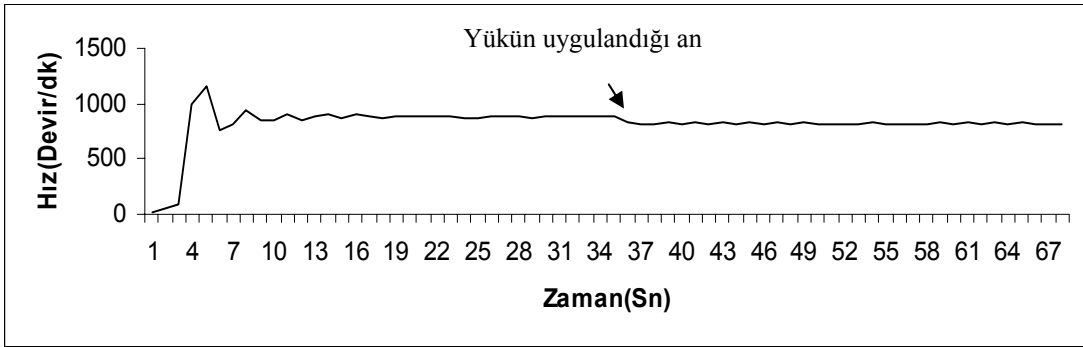
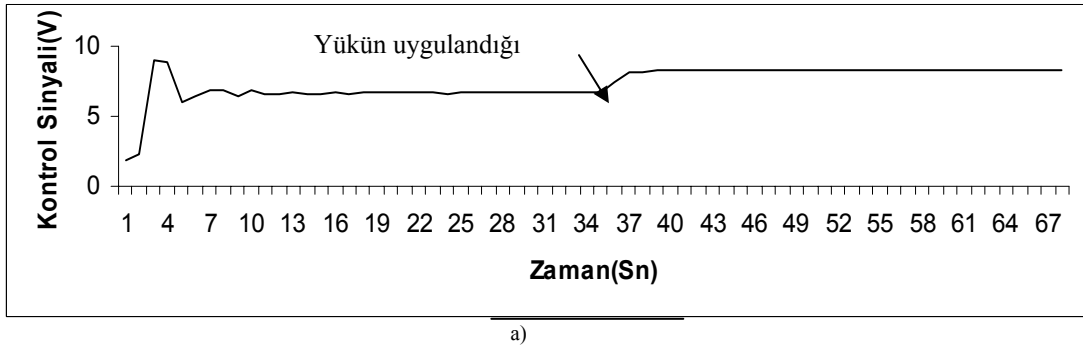
Şekil-V.14 400 devirde yük uygulandığında a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

700 devir



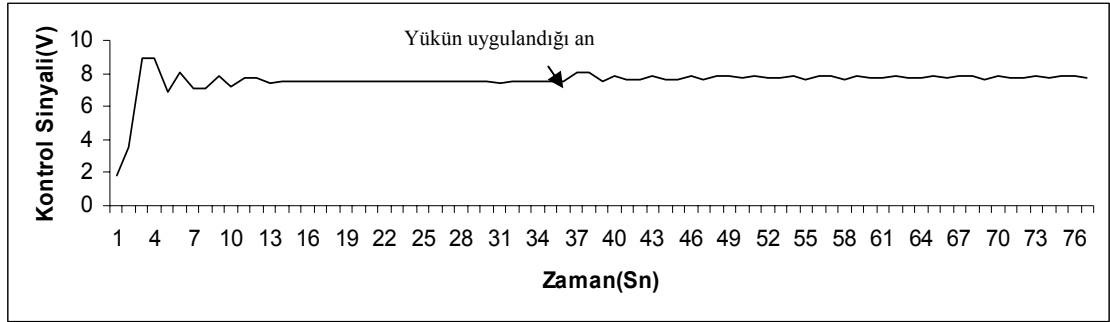
Şekil-V.15 700 devirde yük uygulandığında a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

900 devirde

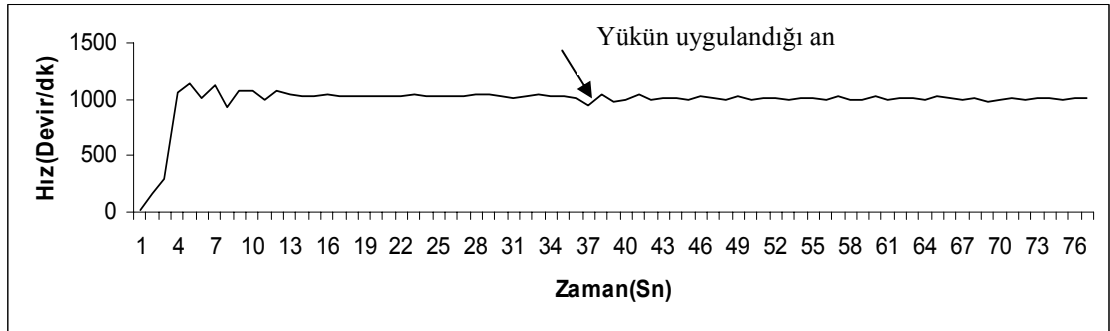


Şekil-V.16 900 devirde yük uygulandığında a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

1100 devirde



a)



b)

Şekil-V.17 1100 devirde yük uygulandığı an a) Kontrol sinyali bilgisi b) Hız (Devir/dk) bilgisi

uygulama sonucunda elde edilen veriler göstermektedir ki mikrodenetleyici ile yapılan donanım kontrol işlemini başarıyla gerçekleştirmektedir.

BÖLÜM -VI

SONUÇ VE DEĞERLENDİRME

Bu çalışmada kısaca yapay sinir ağı donanımlarının genel özelliklerinden bahsedilmiş ve mikrodenetleyiciler ile YSA donanımı yapılmıştır. Çalışmada mikrodenetleyici olarak Microchip firmasının PIC 16F877 isimli denetleyicisi seçilmiştir. Bu denetleyicinin seçilmesinde mikrodenetleyici üzerinde bulunan ADC modülü, USART uyumluluğu ve I2C vb. seri haberleşme standartları için özelleştirilmiş giriş ve çıkışlara sahip bir denetleyici olması nedeniyle bu denetleyici seçilmiştir ayrıca bu denetleyici ile kaynak ve diğer yardımcı yazılım programlarını bulmak kolaydır. Yapılan çalışmada iki adet uygulama yapılmıştır. Bunlardan ilki 2.,4. ve 6. dereceden olan 3 adet farklı fonksiyon girişine bağlı olarak doğrusal bir fonksiyon elde edilmiştir. Bu uygulamada YSA eğitimi MATLAB programı yardımıyla yapılmış ve elde edilen ağırlıklar mikrodenetleyiciye yazılmıştır. Uygulama sonucunda elde edilen çıkış değerleri istenen düzeydedir. 2. uygulama olarak bir DC motorun YSA donanımı ile kontrolü yapılmıştır. Bu uygulamada ağı eğitimi bilgisayar ortamında yapıldıktan sonra mikrodenetleyiciye kaydedilmiştir. DC motorun 300 ile 1100 devir arasında devir kontrolü başarıyla gerçekleştirilmiştir. YSA donanımının performansına baktığımızda düşük devirlerde istenen devir sayısına daha geç oturduğu fakat yük uygulandığında yüke daha iyi cevap verdiği görülmüştür. Yüksek devir değerlerinde ise istenen devire daha kısa sürede ulaştığı fakat yük uygulandığında yüke istenen seviyede cevap veremediği görülmüştür. Motor kontrol uygulamasında özellikle örnekleme zamanının çok etkili olduğu gözlenmiştir.

YSA donanımlarının mikrodnetleyicilerle yapılması YSA'nın gerçek hayatta daha fazla alanda uygulama bulmasına yardımcı olacaktır. Kolay taşınabilir olması, güç tüketiminin düşük olması, farklı sistemlere kolay adapte olması ve diğer çevre elemanlarla kolayca haberleştirilebilir olması mikrodnetleyicili YSA donanımlarının avantajlarından sayılabilir. Mikrodnetleyicili YSA donanımlarının dezavantajı ise hafıza boyutları ve hızları olarak gösterilebilir. Özellikle aktivasyon fonksiyonunu gerçekleştirmek çok fazla işlem zamanı almaktadır. Aktivasyon fonksiyonu diğer alternatif yaklaşım metotları ile yapılırsa hassasiyetten kayıp olacaktır fakat işlem hızının artmasını sağlayacaktır. Bu tür donanımlarda karşılaşılan hız ve kapasite kısıtlamalarına karşın alternatif bir yol olarak paralel veri işleyen YSA donanımları oluşturulabilir. Tez de yapılan diğer yaklaşımda 1 efendi ve 3 adet köle PIC'ten oluşan çoklu mikrodnetleyici yapısı oluşturulmuştur. Bu devrede mikrodnetleyiciler birbirleriyle I2C standardında haberleşmektedirler. Bu standardı kullanarak birbirleriyle haberleşebilen mikrodnetleyici sayısını 128'e çıkarmak mümkündür. Ve her geçen yıl entegreler boyut olarak küçülürken kapasite ve işlem hızı olarak arttığı düşünülürse mikrodnetleyiciler kullanılarak çok daha kapsamlı YSA donanımları kolaylıkla oluşturulabilir hale gelecektir.

Yapılan çalışmada işlem süresi ve hafıza doluluk oranında; kullanılan programlama dili, kullanılan sayı formatı, YSA'da kullanılan aktivasyon fonksiyonu, kullanılan algoritma ve kullanılan YSA modeli etkili olmaktadır.

1 giriş nöronu, 2 arakatman nöronu ve 1 çıkış nöronundan(1-2-1) oluşan bir YSA modelinde mikrodnetleyicinin işlemleri tamamlama süresi 5 Mhz'lik bir işletim hızıyla yaklaşık 8 ms olarak hesaplanmıştır. Uygulamada motor kontrolü için kullanılan 3-5-1'lik modelde tek mikrodnetleyicili sistem 5 Mhz'lik işletim hızıyla 18 ms'de işlemlerini tamamlarken, aynı model için paralel veri işleyen donanımın işlemleri tamamlaması için geçen süre yaklaşık 13 ms olarak hesaplanmıştır. Paralel veri işleyen modelde nöron sayısı artsa bile paralel işleyen mikrodnetleyicilerin sayısını arttırarak işlem süresini istenen seviyede tutmak mümkün olacaktır. Elde edilen bu zaman değerleri göstermektedir ki mikrodnetleyiciler ile yapılacak olan YSA donanımları bir çok kontrol uygulamasında, elektronik donanım olarak kullanılabilir.

KAYNAKLAR

- [1] Elmas,Ç. ,“Yapay Sinir Ağları”, ISBN: 975347612-4, Seçkin Yayıncılık, **Nisan-2003**
- [2] Sağıroğlu,Ş., Beşdok, E., Erler, M., “ Mühendislikte Yapay Zeka Uygulamaları-I”, ISBN: 975-95948-5-4, Ufuk Kitap Kırtasiye-Yayıncılık, **2003**
- [3] Öztemel, E., “Yapay Sinir Ağları”, ISBN: 975-67-97-39-8, Papatya Yayıncılık, **2003**
- [4] Nabiyev, V., “Yapay Zeka”, ISBN: 975 347 687 6, Seçkin Yayıncılık, **Ekim-2003**
- [5] Foo S.Y., Anderson L. R., Takefuji Y. , “Analog Components for the VLSI of Neural Network”, Circuits and Device Magazine,IEEE **Volume: 6, Issue: 4, 1990**
- [6] Czarnul, Z., “Design of Voltage Controlled Linear Transconductance Elements with a Matched Pair of FETs”, IEEE Trans. On Circuits and Systems, **1986**
- [7] Zurada, J.M., ”Analog Implementation of Neural Networks”, Circuits and Device Magazine,IEEE, Vol.8,**1992**
- [8] Zurada, J.M., “Introduction to Artificial Neural Systems”, West Publishing Company, **1992**
- [9] Salmeh, R., Moundy, B.J., “VLSI implementation of An Automatic Q tuning Systems, **2004**
- [10] http://www.eng.yale.edu/pjk/EESrProj_02/luckenbill_html/node5.html
(Erişim Tarihi:28.8.2005)
- [11] Satyanarayana, S., Tsividis,Y., Graf, H.P, “A Reconfigurable Analog VLSI Neural Network Chip” , In Advance In Neural Information Processing Systems, Vol. 2 , California, Morgan Kaufman Publishes, **1990**

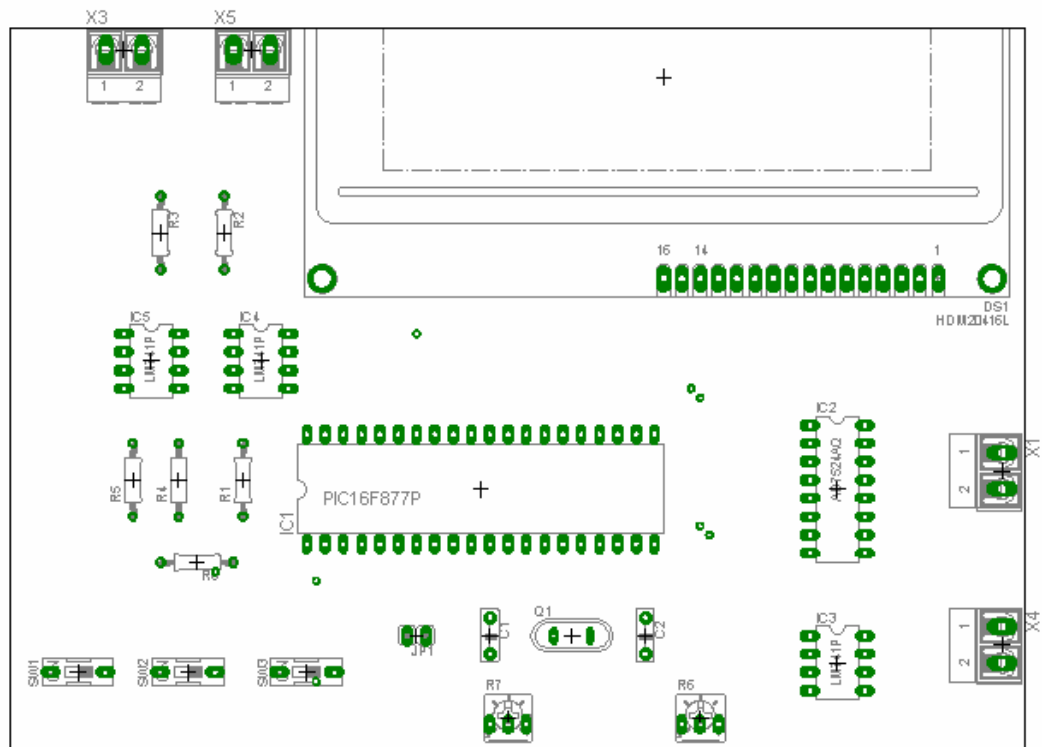
- [12] Zurada, J.M., “Application of Multiplying Digital to Analog Converter to digital control of active filter characteristics”, IEE Proc. Part 6, 91-92, **1981**
- [13] Vittoz, E.A., Arrequit, X.A., “Analog VLSI Hardware : What is missing for Industrial Realization?”, IEEE, Switzerland, 1994
- [14] www.zetex.com (Erişim Tarihi: 7.2.2005)
- [15] www.microchip.com (Erişim Tarihi : 17.5.2005)
- [16] Cadence Design Foundry (UK) Ltd, “ Inter Integrated Circuit (I2C) Technical Datasheet”, **2002**
- [17] Tang, J., Varley, M.R., Peak, M.J, “Hardware implementation of multilayer feed forward neural networks and error backpropagation using 8-bit PIC microcontroller” IEE, Institute of electrical engineers, Londra, **1997**
- [18] Myers, D.J., Hutchinson, R.A., “Efficient Implementation of Piecewise Linear Activation Function for digital VLSI Neural Network”, Electron. Lett. **1989**
- [19] Alippi, C. Storti-Gajani, G., “Simple approximation of sigmoidal functions: realistic design of digital neural networks capable of learning”, Proc. IEEE Int. Symp. on Circuits and Systems, Singapore ,**1991**
- [20] Amin, H., Curtis. K.M., Hayes-Gill, B.R., “Piecewise linear approximation applied to nonlinear function of a N.N”, IEE Proc. Circuits, Devices Sys. **1997**
- [21] Tommiska, M.T. “Efficient digital implementation of the sigmoid function for reprogrammable logic”, IEE proc. Comput. Digit. Tech., Vol 150, **2003**
- [22] Liu, Z., Zhuang, X., Wang, S., “Speed control of a dc motor using BP Neural Networks”, IEEE transaction on industry application, **2003**
- [23] <http://www.mathworks.com/> (Erişim Tarihi: 3.4.2005)
- [24] Çelik, H.H.: “Yapay Sinir Ağı Kullanarak El Yazısı Rakam Tanıma”, Doktora Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1999
- [25] Machbub, C., Setijadi, A., Cahaya, Y.D, “Design and implementation of adaptive neural networks algorithm for DC motor speed control system using simple microcontroller”, IEEE PEDS, Indonesia, **2001**

EK – A

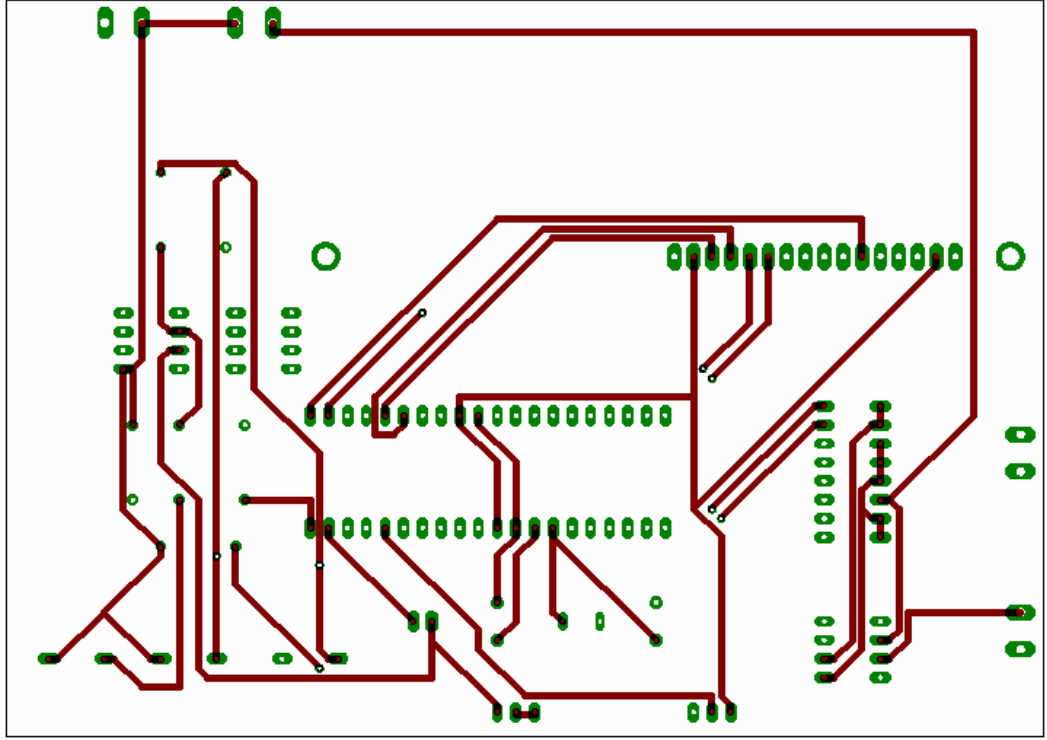
Tezle ilgili yazılımlar CD’de verilmiştir.

EK – B

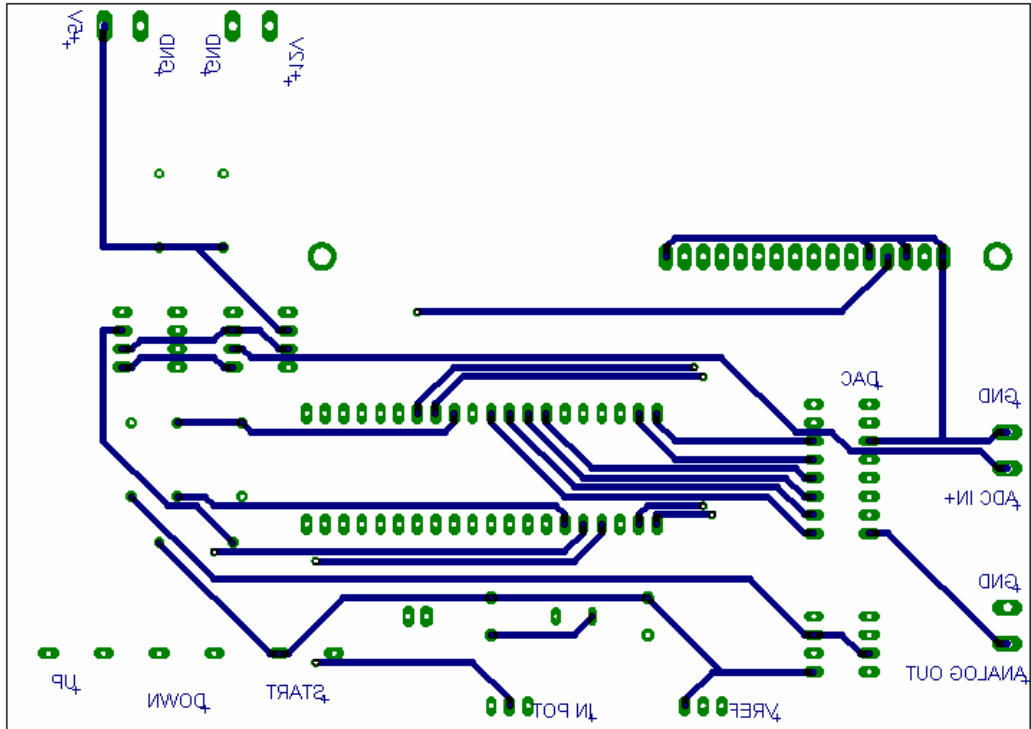
TEZDE KULLANILAN DEVRELERİN ELEMAN YERLEŞİMİ VE BASKI
DEVRE YOLLARI GÖRÜNÜMÜ



Tek mikrodnetleyili modelin devre elemanlarının dizilişı



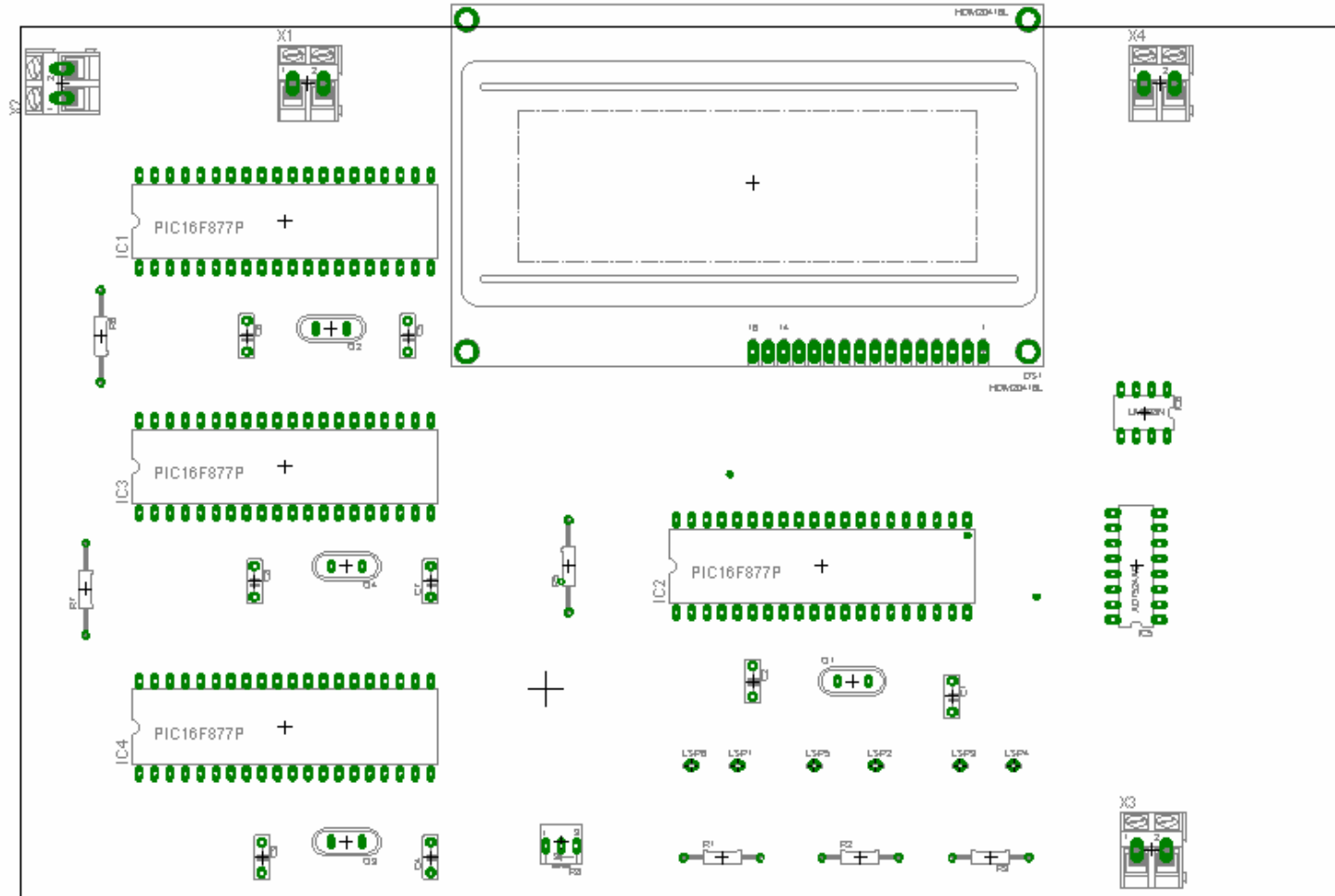
Tek mikrogenetleyicili modelin baskı devresinin üst yollarının görünümü



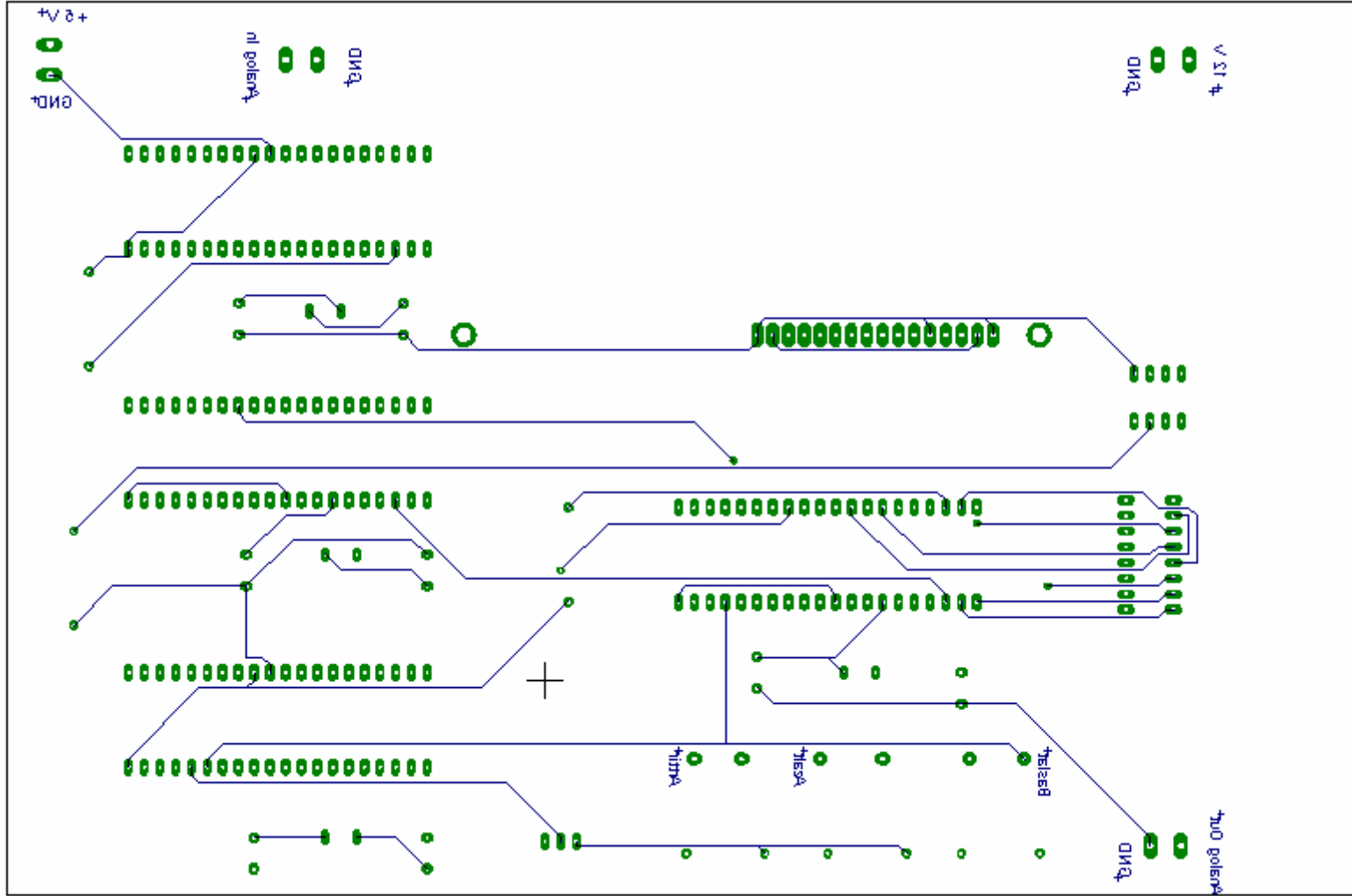
Tek mikrogenetleyicili modelin baskı devresinin alt yollarının görünümü

Tek mikrodnetleyici modelde kullanılan malzemeler

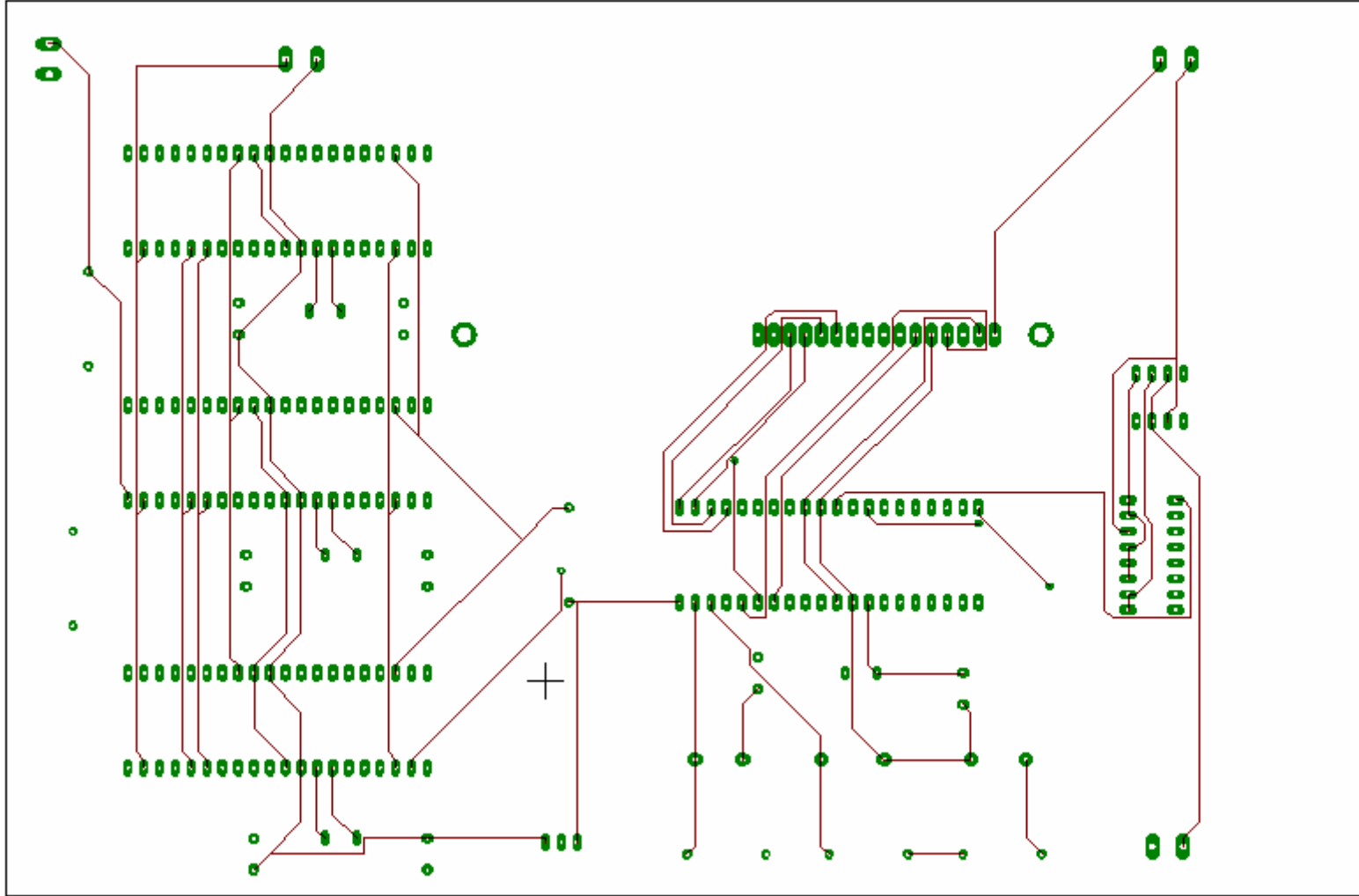
X1,X3,X4 ve X5	Konnektör
IC1	PIC 16F877 Mikrodnetleyici
IC2	AD 558JN DAC entegresi
IC3,IC4 ve IC5	UA741 OPAMP
DS1	2X16 Alfanümerik LCD display
Q1	4 Mhz Kristal
R1,R2,R3,R4,R5,R8	10K Direnç
R6,R7	10K Potansiyometre
C1,C2	22pF Kondansatör
SW1,SW2,SW3	Push buton



Çoklu mikrodnetleyicili modelin devre elemanlarının yerleşimi



Çoklu mikrodnetleyicili modelin alt baskı devre yollarının görünümü



Çoklu mikrodeneleyicili modelin üst baskı devre yollarının görünümü

Paralel veri işleyen 4 mikrodnetleyicili modelde kullanılan malzeme listesi

X1,X2,X3 ve X4	Konnektör
IC1,IC2,IC3 ve IC4	PIC 16F877 Mikrodnetleyici
IC5	AD558JN DAC Entegresi
IC6	LM 308 OPAMP
Q1,Q2,Q3 ve Q4	4Mhz Kristal
DS1	2X16 Alfanümerik LCD display
C1,C2,C3,C4,C5,C6,C7,C8	22pF Kondansatör
R1,R2,R3	10K Direnç
R5,R6	2.7K Direnç
R8	10K Potansiyometre

ÖZGEÇMİŞ

1980 yılında İstanbul’da doğdu. Lise öğrenimini İstanbul Ticaret Odası Anadolu Meslek Lisesi’nde tamamladı. 1998 yılında M.Ü. Teknik Eğitim Fakültesi Elektronik-Haberleşme Öğretmenliğini kazanarak lisans eğitime başladı. 2002 yılında M.Ü. Fen Bilimleri Enstitüsünde yüksek lisans eğitime başladı. 2002 yılından beri M.Ü. Teknik Eğitim Fakültesinde Araştırma Görevlisi olarak görev yapmaktadır.