

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

SCARA TİPİ ROBOTUN YAPAY SİNİR AĞLARI İLE EĞİTİLMESİ

YÜKSEK LİSANS TEZİ

Mak.Müh. Selçuk ÖZTÜRK

Enstitü Anabilim Dalı : MAKİNE MÜHENDİSLİĞİ
Enstitü Bilim Dalı : MAKİNE TASARIM VE İMALAT
Tez Danışmanı : Doç. Dr. Recep KAZAN

Mayıs 2007

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SCARA TİPİ ROBOTUN YAPAY SİNİR AĞLARI İLE EĞİTİLMESİ

YÜKSEK LİSANS TEZİ

Mak.Müh. Selçuk ÖZTÜRK

Enstitü Anabilim Dalı : MAKİNE MÜHENDİSLİĞİ

Enstitü Bilim Dalı : MAKİNE TASARIM VE İMALAT

Bu tez 02 / 05 / 2007 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Doç. Dr.
Recep KAZAN
Jüri Başkanı

Doç. Dr.
Nejat YUMUŞAK
Üye

Yrd. Doç. Dr.
Ömer Kadir MORGÜL
Üye

ÖNSÖZ

Tez çalışması sırasında bana destek olan saygıdeğer hocam Doç. Dr. Recep KAZAN ve bilgileri ile bana yardımcı olan Arş. Gör. Aysun EĞRİSÖĞÜT TİRYAKİ hanıma ve de hayatım boyunca buralara kadar ulaşmam için benden maddi ve manevi desteğini esirgemeyen çok sevgili aileme teşekkür ederim. Ayrıca tez çalışmam esnasında bana her konuda yardımcı olan sevgili abim ve hocam Öğr. Gör. Kadir İSA' ya da teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖNSÖZ.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ.....	xi
ÖZET.....	xii
SUMMARY.....	xiii
BÖLÜM 1.	
GİRİŞ.....	1
BÖLÜM 2.	
ROBOT.....	3
2.1. Giriş.....	3
2.2. Tarihçe.....	5
2.3. Robotların Sınıflandırılması.....	10
2.3.1. Tahrik sistemlerine göre robotlar.....	10
2.3.2. Robot eksenlerine göre sınıflandırma.....	11
2.3.3. Koordinat sistemlerine göre robotlar.....	13
2.3.4. Robot tiplerine göre sınıflandırma.....	17
2.3.5. Kontrol sistemlerine göre robotlar.....	21
2.4. Robotlarda Tutucular.....	23
2.5. Robot Sensörleri.....	24
2.6. Robotların Uygulama Alanları.....	25
2.7. Robotların Kullanım Avantajları.....	28

BÖLÜM 3.

YAPAY SİNİR AĞLARI.....	30
3.1. Giriş.....	30
3.2. Tarihçe.....	33
3.3. Genel Özellikler.....	35
3.4. Yapay Sinir Hücresi.....	39
3.5. Yapay Sinir Hücresinin Çalışma Prensipleri.....	46
3.6. Yapay Sinir Ağlarında Bilgi ve Zeka.....	47
3.7. Yapay Sinir Ağlarında En Çok Kullanılan Modeller.....	47
3.8. Bazı Yapay Sinir Ağları.....	48
3.8.1. İleri beslemeli yapay sinir ağları.....	48
3.8.2. Geri beslemeli yapay sinir ağları.....	49
3.8.3. Bellek hücreli yapay sinir ağ yapıları.....	50
3.8.4. Radyal tabanlı fonksiyon ağları.....	51
3.8.5. Fonksiyonel link ağları.....	52
3.8.6. Çağrışımlı bellek ağları.....	53
3.8.7. Modül yapay sinir ağları.....	53
3.9. Yaygın Olarak Kullanılan Öğrenme Algoritmaları.....	55
3.9.1. Geri yayılım.....	55
3.9.2. Delta bar delta.....	56
3.9.3. Genişletilmiş delta bar delta.....	58
3.10. Yapay Sinir Ağlarında Öğrenme.....	59
3.11. Makine Öğrenmesi ve Öğrenme Türleri.....	60
3.11.1. Öğrenme yöntemleri.....	62
3.11.2. Örneklerden öğrenme.....	63
3.11.3. Öğrenme stratejileri.....	63
3.11.4. Öğrenme kuralları.....	66
3.12. Yapay Sinir Ağ Uygulamalarının Avantajları.....	68
3.13. Yapay Sinir Ağ Uygulamalarının Dezavantajları.....	70
3.14. Yapay Sinir Ağlarının Uygulama Alanları.....	72
3.14.1. Endüstriyel uygulamalar.....	72
3.14.2. Askeri uygulamalar.....	72
3.14.3. Yapay sinir ağlarının mühendislik uygulamaları.....	73

3.14.4. Yapay sinir ađları için uygulama örnekleri.....	73
BÖLÜM 4.	
SCARA TİPİ ROBOTUN GENEL DİNAMİKLERİ VE ANALİZİ.....	76
4.1. Robotun Dinamik Modeli.....	77
4.2. Robotun Yörünge Denklemlerinin Elde Edilmesi.....	81
4.3. Robotun Konum ve Tork Grafiklerinin Elde Edilmesi.....	84
BÖLÜM 5.	
SCARA TİPİ ROBOTUN YAPAY SINIR AĐLARI İLE EĐİTİLMESİ.....	86
5.1. Robotun Yapay Sinir Ađları ile Eđitilmesi Sonucu Performans, Çıkış ve Hata Grafiklerinin Elde Edilişı.....	90
5.2. Robotun Yapay Sinir Ađları ile Test Edilmesi Sonucu Çıkış ve Hata Grafiklerinin Elde Edilişı.....	94
BÖLÜM 6.	
SONUÇLAR VE ÖNERİLER.....	96
KAYNAKLAR.....	99
EKLER.....	101
ÖZGEÇMİŞ.....	114

SİMGELER VE KISALTMALAR LİSTESİ

A	: Ağırlık değeri
G	: Girdi değeri
n	: Bir hücreye gelen toplam girdi
NET	: Proses elemanlarına gelen net girdi değeri
$F(NET)$: Aktivasyon fonksiyonu
\mathcal{C}	: Hücrenin aktivasyon fonksiyonuna göre çıktısı
L	: Lagrange fonksiyonu
K	: Robot kolunun toplam kinetik enerjisi
P	: Robot kolunun toplam potansiyel enerjisi
τ_i	: Sisteme uygulanan genelleştirilmiş moment veya kuvvet
q_i	: Robot kolunun genelleştirilmiş koordinatları
θ	: Açısal konum değişkeni
$M(\theta)$: nxn pozitif tanımlı atalet matrisi
$N(\theta, \dot{\theta})$: nx1 pozitif tanımlı coriolis, merkezkaç, yerçekimi kuvveti
m_1	: 1. eklemin kütlesi
m_2	: 2. eklemin kütlesi
m_3	: 3. eklemin kütlesi
m_4	: 4. eklemin kütlesi
L_1	: 1. eklemin uzunluğu
L_2	: 2. eklemin uzunluğu
d_3	: 3. eklemin stroku
d_4	: 4. eklemin uzunluğu
I_{zz1}	: 1. eklemin kütleli atalet momenti
I_{zz2}	: 2. eklemin kütleli atalet momenti

I_{zz3}	: 3. eklem kütlesel atalet momenti
I_{zz4}	: 4. eklem kütlesel atalet momenti
g	: Yerçekimi ivmesi
τ_1	: 1. eklem tork değeri
τ_2	: 2. eklem tork değeri
τ_3	: 3. eklem tork değeri
τ_4	: 4. eklem tork değeri
θ_1	: 1. uzvun yörüngesi
θ_2	: 2. uzvun yörüngesi
θ_3	: 3. uzvun yörüngesi
θ_4	: 4. uzvun yörüngesi
$\dot{\theta}_1$: 1. uzvun hızı
$\dot{\theta}_2$: 2. uzvun hızı
$\dot{\theta}_3$: 3. uzvun hızı
$\dot{\theta}_4$: 4. uzvun hızı
$\ddot{\theta}_1$: 1. uzvun ivmesi
$\ddot{\theta}_2$: 2. uzvun ivmesi
$\ddot{\theta}_3$: 3. uzvun ivmesi
$\ddot{\theta}_4$: 4. uzvun ivmesi
$u(k)$: giriş
$y(k)$: çıkış
$e(k)$: hata

ABD	: Amerika Birleşik Devletleri
ART	: Adaptif rezonans teorisi
BHYSA	: Bellek hücreli yapay sinir ağları
Bkz	: Bakınız
ÇBA	: Çağrışımlı bellek ağları
DBD	: Delta bar delta öğrenme algoritması

EDBD	: Geniřletilmiş delta bar delta öğrenme algoritması
FLA	: Fonksiyonel link ađları
LVQ	: Vektör kuantizasyon modelleri
MSE	: Hatanın toplamının ortalaması
NASA	: Ulusal Havacılık ve Uzay Dairesi
PNN	: Probabilistik sinir ađları
RBN	: Radyal temelli ađlar
RTFA	: Radyal tabanlı fonksiyon ađları
SCARA	: Seçilenlere uyan montaj robot kolu
SOM	: Kendi kendini organize eden modeller
YGKI	: Yöresel geri, küresel ileri
YSA	: Yapay sinir ađları

ŞEKİLLER LİSTESİ

Şekil 2.1.	İnsan kolu ile mafsallı robotun çalışma alanları arasındaki benzerlik.....	12
Şekil 2.2.	Kartezyen koordinat sistemine ait şematik çizim.....	14
Şekil 2.3.	Silindirik koordinat sistemi.....	14
Şekil 2.4.	Küresel koordinat sistemi.....	15
Şekil 2.5.	Döner koordinat sistemli robotun çalışma alanı.....	17
Şekil 2.6.	Kartezyen robot.....	18
Şekil 2.7.	6 eksenli bir mafsallı robot.....	19
Şekil 2.8.	Robota ait şematik çizim.....	20
Şekil 2.9.	Robotun çalışma alanı.....	21
Şekil 2.10.	Robot tutucu.....	23
Şekil 2.11.	Robot el.....	23
Şekil 2.12.	Malzeme taşıma robotu.....	26
Şekil 2.13.	Montaj robotları.....	26
Şekil 2.14.	1400 °F deki döküm parçası kaldırılırken.....	27
Şekil 2.15.	Döküm parçası buharda temizlenirken.....	27
Şekil 2.16.	Ölçme ve kontrol robotları.....	27
Şekil 2.17.	Boyama robotları.....	28
Şekil 3.1.	Yapay sinir ağına bir örnek.....	32
Şekil 3.2.	Yapay sinir hücresinin yapısı.....	40
Şekil 3.3.	Bazı aktivasyon fonksiyonları.....	45
Şekil 3.4.	Bir yapay sinir ağı hücresinin çalışma örneği.....	46
Şekil 3.5.	İleri beslemeli 3 katmanlı yapay sinir ağı.....	49
Şekil 3.6.	Geri beslemeli iki katmanlı yapay sinir ağı.....	49
Şekil 3.7.	Yöresel geri küresel ileri beslemeli yapay sinir ağı.....	50
Şekil 3.8.	Bellek hücreli yapay sinir ağı ve bellekteki bir hücrenin yapısı....	51

Şekil 3.9.	Radyal tabanlı fonksiyon ağı.....	52
Şekil 3.10.	Fonksiyonel link ağları.....	52
Şekil 3.11.	Çağrışimli bellek ağları.....	53
Şekil 3.12.	Modül yapay sinir ağları.....	54
Şekil 3.13.	Geri yayılım algoritması.....	56
Şekil 3.14.	Öğretmenli öğrenme yapısı.....	64
Şekil 3.15.	Destekleyici öğrenme yapısı.....	65
Şekil 3.16.	Öğretmensiz öğrenme yapısı.....	66
Şekil 3.17.	Otomatik benzin pompası.....	74
Şekil 3.18.	Mekanik yardımcı bir tekerlekli sandalye.....	74
Şekil 3.19.	Tıbbi uygulamalar için geliştirilen mekanik bir görüntüleme sistemi.....	75
Şekil 3.20.	Taşıma robotu.....	75
Şekil 4.1.	Robotun şematik çizimi.....	76
Şekil 4.2.	Her bir eklemin konum ve tork grafikleri.....	85
Şekil 5.1.	Kullanılan yapay sinir ağın modeli.....	86
Şekil 5.2.	Yapay sinir ağlarında modellenen robotun akış diyagramı.....	87
Şekil 5.3.	Robot modelleme için bir yapay sinir ağı çözümü.....	88
Şekil 5.4.	Log-sigmoid transfer fonksiyonu.....	88
Şekil 5.5.	Yapay sinir ağının performans grafiği.....	91
Şekil 5.6.	1., 2., 3. ve 4. eklemin istenen ve eğitim sonucu elde edilen konum değerleri.....	92
Şekil 5.7.	1., 2., 3. ve 4. eklemin istenen ve eğitim sonucu elde edilen konum değerleri arasındaki hata değerleri.....	93
Şekil 5.8.	1., 2., 3. ve 4. eklemin istenen ve test sonucu elde edilen konum değerleri.....	94
Şekil 5.9.	1., 2., 3. ve 4. eklemin istenen ve test sonucu elde edilen konum değerleri arasındaki hata değerleri.....	95

TABLolar LİSTESİ

Tablo 3.1.	Sayısal bilgisayarlar ve yapay sinir ağlarının bazı özellikleri..	31
Tablo 3.2.	Farklı toplama fonksiyonları.....	42
Tablo 3.3.	Aktivasyon fonksiyonları.....	44
Tablo 5.1.	Yapay sinir ağ sisteminde kullanılan fonksiyonlar.....	90
Tablo 6.1.	Hedeflenen çıkış değerleri ile ysa çıkış değerleri arasındaki hata değerleri (eğitim).....	97
Tablo 6.2.	Hedeflenen çıkış değerleri ile ysa çıkış değerleri arasındaki hata değerleri (test).....	98

ÖZET

Anahtar kelimeler: SCARA robot, Robot dinamiđi, Yörünge, Yapay sinir ađları

Verimliliđi arttırma ve kaliteyi iyileştirme ihtiyacından, hassas ve tekrarlanabilir işlerin kolaylıkla yapılabilmesi nedeniyle birçok alanda deđişik robotlar kullanılmaktadır. Robotlar için “bir operatörün çeşitli reflekslerini ve zekasını kullanımının basit bir uygulamasıdır” denilebilir.

Bu çalışmada, dört serbestlik dereceli SCARA tipi robot kolunun yapay sinir ađı ile eğitilmesi konusu ele alınmıştır.

İlk olarak modellenecek olan robotun YSA’ nda kullanılacak yörünge ve Lagrange – Euler dinamik denklemleri çıkarılmıştır. Robot, kübik yörüngelerle ifade edilen üç dönme ve doğrusal yörüngeye sahip bir öteleme hareketi yapmaktadır. Belirlenen zaman aralığı ve sınır şartları kullanılarak kübik ve doğrusal yörüngelerin katsayıları elde edilmiş ve bu sayede her bir eklem için konum denklemlerine ulaşılmıştır.

Tork deđerleri YSA’ na giriş; konum deđerleri ise çıkış olarak verilmiştir. Ađı eğitmek için 4:11:4 yapısında bir YSA mimarisi kullanılmaktadır. Robotun YSA’ da eğitilmesi sonucu; performans grafiđi ile her bir eklem, konumlarının çıkış ve hata deđerlerinin görülebildiđi grafikler elde edilmiştir. YSA’ nın hiç görmediđi test setindeki deđerler ađa gösterilmiştir ve ađın verdiđi cevaplar incelenmiştir.

SCARA ROBOT EDUCATED WITH ARTIFICIAL NEURAL NETWORK

SUMMARY

Key Words: SCARA robot, Robot dynamics, Trajectory, Artificial neural network

Robot manipulators are used in various fields to enhance the efficiency to need to get better quality to make works which are sensitive and repeatable easily for different requirements. We say that “the simple applied for an operator which use different reflex and itelligence” for the robots.

In this study the subject which SCARA manipulator which has four degrees of freedom is educated by using artificial neural network (ANN) is considered.

Firstly Lagrange – Euler dynamic and trajectory equations of the modelled robot are obtained for use in artificial neural network (ANN). The modelled robot has three rotary and one linear motion. The rotary motion is defined by cubic trajectory and linear motion has linear trajectory. The cubic and linear trajectory coefficients are acquired by using the boundary conditions in a particular time period and thus position equations are also carried out.

Torque values are introduced to ANN as input and position values are obtained as output in a particular time period. The ANN architecture which has a structure at 4:11:4 used for educated by network. Performance graphic and graphics which shown position output and error values for each one link are acquired with the educated robot manipulator by working with ANN. The values at the test system which have never been seen by the ANN had been shown to the network and also the replies which were given by network had been analysed.

BÖLÜM 1. GİRİŞ

Robot sistemleri çok parametrelili olmaları ve davranışını belirleyen parametrelerin arasında karmaşık matematiksel ilişkilerin olması dolayısıyla yapay sinir ağlarının tanımlama ve denetim problemlerindeki başarısını üzerinde fikir sahibi olmak için son derece uygun sistemlerdir. Özellikle aşağıdaki maddeler halinde verilen güçlüklerin giderilmesi, işlemsel akıl içeren sistemlerin kullanılmasını teşvik etmektedir [1].

1. Durum vektörü pozisyon bilgisinden ve bu ölçümden nümerik olarak türetilen hız bilgisinden müteşekkil olduğundan, pozisyon kodlayıcılarının çıkışlarına gürültü karıştığı durumlarda tanımlama/denetim sistemi gürbüz davranmalıdır.
2. Robot sistemlerinin dinamik davranışını, kütlesi değişebilen yükleri taşıdığı durumlarda tamamen değişeceğinden dolayı benimsenen tanımlama ya da denetim algoritması bu değişiklikleri başarısını düşürmeden dikkate alabilmeli, uyarlanabilir olmalıdır.
3. Sistemin dinamik davranışında, özellikle mekanik sistemlerde sıkça karşılaşılan sürtünme teriminin hangi modele sahip olacağı genellikle tam olarak bilinemez. Tanımlama ya da denetim algoritmasından ise bu belirsizliği ortadan kaldıracı yapısal özelliklere sahip olması beklenir.
4. Sistemi tanımlayan denklemler nonlineer ve kuple olmuş vaziyette olabilir ve geleneksel tasarımı son derece zaman alıcı hale getirebilir [1].

İnsan beynindeki çalışma sisteminin yapay olarak benzetimi çalışmalarının sonucu ortaya çıkan yapay sinir ağları (YSA), klasik tekniklerle çözümü zor, karmaşık ve doğrusal olmayan problemlerin çözümünde etkin bir yoldur. YSA'nın pratik

kullanımı genellikle, çok farklı yapıda ve formlarda bulunabilen verileri hızlı bir şekilde tanımlama ve algılama üzerinedir [2-3].

SCARA tipi robotlar insan kolunu model alan, özellikle kesme, yükleme, montaj ve üretim hatları gibi sürekliliği getiren ortamlarda kullanılan küçük endüstriyel robotlardır. SCARA sözcüğü dilimizde ‘Seçici Serbest Esnemeli Robot Kolu (Selectively Compliant Articulated Robot Arm)’ olarak çevrilmektedir [3-4].

Bu çalışmada dört eklemlili bir SCARA robotu ele alınmış ve robot dinamiği YSA’ nın kullanımı ile modellenmiştir. Çalışmanın 2. kısmında robotlar ve 3. kısmında robot dinamiği genel anlamıyla tanıtılmıştır. 4. kısmında YSA genel hatlarıyla tanıtılmıştır. 5. kısmında tasarlanan YSA mimarisi, ağırlık eğitimi performansı ve test sonuçları verilmiştir. 6. kısımda ise simülasyon sonuçları yorumlanmıştır.

BÖLÜM 2. ROBOT

2.1. Giriş

Robot kelimesi ilk olarak Çek filozofu ve oyun yazarı Karel Capek' in "Rossum' s Universal Robot (R.U.R.)" isimli oyunu içerisinde 1922 yılında kullanıldı. Çek dilinde robot "işçi" veya "esir" anlamındadır [5-6].

Robotlar için "Bir operatörün çeşitli reflekslerini ve zekasını kullanımının basit bir uygulamasıdır" denilebilir. Robotlarla çevrenin ve nesnelerin sürekli bir etkileşimi vardır. Bu etkileşimin sağlanması, görevin önceden belirlenerek robota tanıtılması sayesinde olur. Günümüzde gerçek anlamda robotlar, gelişen sanayi' ye ayak uydurmakta zorlanan işçi kesiminin süper alternatifleri haline gelmişlerdir. Bunun başlıca üç sebebi vardır [6].

1. Verimliliği artırma ihtiyacı,
2. Kaliteyi iyileştirme ihtiyacı,
3. Hassas ve tekrarlanabilir işlerin kolaylıkla yapılabilmesi.

Robotlar, çeşitli malzemeleri, parçaları, takımları veya özel aletleri, bir dizi ödevin gerçekleştirilmesi için, programlanmış hareketler boyunca taşıyacak şekilde tasarlanmış yeniden programlanabilir manipülatörler olarak da tanımlanabilirler [6].

Webster Sözlüğü (1993) ; "Normal koşullarda insanlara atfedilen işlevleri yapan ya da şekilsel olarak insana benzeyen otomatik bir düzenek" olarak tanımlanmaktadır [6-7].

Günümüzde robot tanımı üzerinde dünyada kesin olarak bir fikir birliđi sađlanamamıştır. ABD ve Avrupa ülkeleri, robot tanımlamalarında, bir robotun üç niteliđe sahip olması gerektiđini belirtmektedirler. Bu nitelikler;

1. Manipülasyon yapma, yani cisimlerin yerini deđiştirme ve üzerlerinde işlem yapma,
2. Programlanabilirlik, yani benzer türdeki, çeşitli işlemleri yapabilme olanađı,
3. Algılama sistemi ile çevre koşullarına göre düşük düzeyde karar verebilme yetkisi.

Amerikan Robot Topluluđu ve İngiliz Robot Birliđi tanımlamalarına göre, algılama sistemi olmayan yapıya robot denemez. İlk iki şartı yerine getiren yapıya programlanabilir manipülatör veya kısaca manipülatör, algılama sistemine de sahip olan manipülatöre de robot denir. Japonlar ise, robot tanımlamalarında ilk iki şartın yeterli olduđunu savunmaktadırlar [6].

Endüstriyel robotlar konusunda ilgili robot kuruluşlarından bazılarının yaptıđı tanımlar aşıđıda verilmiştir:

Alman Endüstriyel Robot Birliđi (M.H.I.-Montage Handhabung Industrieroboter): ISO/TR 8373' deki tanıma göre endüstri robotları; mekanik müdahale olmaksızın hareketleri tamamen serbest programlanabilen tutucu takımlarla donatılmış, en az 3 eksenli iş yapabilme otomatları olan imalat sanayiinin tüm branşlarında kullanılabilen makinelerdir.

İngiliz Robot Birliđi (B.R.A.-British Robot Association): Yeniden programlanabilir, minimum 4 serbestlik dereceli, imalatın performansı için mekanik aksam ve mekanik kısımları veya özel imalat kısımları deđişebilir programlanmış hareketleri yapabilecek şekilde dizayn edilmiş makinelerdir.

Amerikan Robot Enstitüsü (R.I.A.-The Robotics Institute of American): Yeniden programlanabilir, çeşitli deđişik konuları yapabilecek şekilde deđişebilir

programlanmış hareketlere sahip malzeme, parça taşımayı gerçekleştirecek şekilde dizayn edilmiş çok fonksiyonlu mekanik konstrüksiyonlardır.

Japon Endüstriyel Robot Birliği (J.I.R.A.-Japan Industrial Association): Bu birlik robotun tanımını sınıflandırarak yapmaktadır.

Manipülator: İnsan kolunun hareketine sahip bir yerden bir yere uzayda hareket eden bir makinedir.

Playback robot: Bir manipülatörün kaydedilmiş bilgileri okuyup, bu bilgilerin doğrultusunda işlemleri sıra ile yapan robottur.

Akıllı robot: Davranış ve iletişim bağını duyumlarında ve tanımlamalarında belirleyen bir robottur.

Bilim adamı Asimov, 1939-40 yıllarında yazdığı romanında üç temel fikir olarak robotların sahip olması gereken fonksiyon ve sınırları tanımlamıştır.

1. Bir robot, insanlara zarar vermemeli, onlara zarar gelmesine seyirci kalmamalıdır.
2. Birinci kuralla çelişmediği sürece bir robot daima insanlardan aldığı emirlere uymalıdır.
3. Birinci ve ikinci kuralla çelişmediği sürece bir robot kendini, kendisine zarar verecek hareketlerden korumalıdır [6].

2.2. Tarihçe

1800: Jacques de Vaucanson, Pierre & Henri-Louis Jacquet- Droz, Henri Maillerdet otomatik yazı yazan ve müzik enstrümanı çalan makineler geliştirdiler [6-7].

1801: Joseph Jacquard ilk kez delikli kart kullanarak çalıştırılan otomatik dokuma makinesi geliştirdi.

1830: Christopher Spencer mekanik kam denetimli otomatik bir torna tezgahı geliřtirdi.

1892: Seward Babbitt sıcak metal parçaları fırından almak üzere motorlu tutucuya sahip vinç tasarladı.

1920-1921: Çekoslovak Karel Capek'in yazdığı bir tiyatro oyununda ilk kez "Robot" kelimesi kullanıldı. Yazar, bu kelimeyi Çek dilinde "hizmet eden" anlamında kullanılan "robota" dan üretti. Tiyatro oyunu, "insan makine yapar, makine de insanı öldürür" teması üzerine kuruluydu.

1938: De Villbis firması için Willard Pollard ve Harold Roselund programlanabilir püskürtme boyama makinesi geliřtirmişlerdir.

1939: Westinghouse řirketi New York Dünya Fuarı'nda sergilenmek üzere mekanik bir insan ve köpek tasarladı.

1940: Grey Walter ışığa yönelen ilk gezer robotları (machine speculatrix) üretti.

1941: Isaac Asimov "Robot" kelimesinden "Robotik" kelimesini türeterek ilk kez kullandı. Robotik, robot teknolojisi ile ilgili tüm alanları kapsayan bir tanım olarak kabul edilmektedir.

1942: Isaac Asimov, Runaround isimli hikayesinde Robotların üç yasaını yazdı.

1. Bir robot bir insana zarar veremez, veya kayıtsız kalarak bir insanın zarar görmesine neden olamaz.
2. Birinci yasa ile çatışmamak şartı ile bir robot insanlar tarafından verilen emirlere uymak zorundadır.
3. Birinci ve İkinci yasa ile çatışmamak şartı ile bir robot kendi varlığını korumalıdır.

1946: George Devol, genel amaçlı, manyetik kayıt yapabilen ve tekrar çalıştırılabilen bir cihaz geliřtirdi. Whirlwind adlı bir başka bilgisayar, MIT’ de ilk olarak bir bilimsel problemi çözdü.

1948: MIT’ den Norbert Wiener Elektronik, Mekanik ve Biyolojik sistemlerin kontrol ve iletiřimini inceleyen, ‘‘Sibernetik’’ bařlıklı kitabı yayınladı.

1951: Raymond Goertz, ABD Atom Enerjisi komisyonu için uzaktan iřletilen bir kol tasarladı.

1954: George Devol, programlanabilir genel amaçlı robotu tasarladı ve patent bařvurusunda bulundu.

1956: G. Devol ve Joseph F. Engelberger, Unimation Inc. Adlı dünyanın ilk robot firmasını kurdular.

1958: Satıř amaçlı ilk ticari robot üretildi.

1959: Planet firması, ilk genel amaçlı ticari robotu pazarlamaya bařladı.

1960: Harry Johnson ve Veljko Milenkoviç’ in tasarladığı Versatron isimli robot pazarlanmaya bařlandı. Unimation robotlarının adı Unimate Robot sistemleri olarak deęiřtirildi.

1962: General Motors ilk kez bir endüstriyel robotu (Unimate) üretim hattında kullanmaya bařladı. Robot, sıcak parçaları kalıp döküm makinesinden alarak istiflemek amacıyla kullanıldı.

1963: Bilgisayar denetimli, altı eklemlili ilk yapay kol (Rancho arm) geliřtirildi.

1964: Dünyanın önde gelen bazı üniversite ve arařtırma merkezlerinde (MIT, Stanford Arařtırma Enstitüsü, Stanford Üniversitesi, Edinburg Üniversitesi) ilk kez Yapay Zeka arařtırmaları bařladı ve laboratuvarları açıldı.

1966: Nokta kaynağı yapan ilk robotlar üretildi.

1968: Stanford Araştırma Enstitüsünce Shakey isimli ve görme yeteneği olan ilk gezer robot üretildi.

1970: Stanford Üniversitesince bir robot kol geliştirildi ve bu robot kol Stanford kolu adıyla araştırma projelerinde bir standart olarak yerleşti.

1972: Shigeo Hirose, Tokyo Teknoloji Enstitüsünde bir öğrenci, yılan benzeri bir robot yaptı.

1973: Richard Hohn, Cincinnati Milacron Corporation adına ilk mini-bilgisayar denetimli robotu geliştirdi. Geliştirilen robot T3 (The Tomorrow Tool) olarak adlandırıldı.

1976: Viking 1 ve Viking 2 uzay araçlarında robot kollar kullanıldı.

1977: Star Wars sinema filmindeki C3P0 ve R2D2 robot animasyonlarıyla robot kelimesi geniş insan kitlelerine yayıldı.

1977: Asea Brown Boveri Ltd. şirketi mikrobilgisayar kontrollü robotları piyasaya sürdü.

1978: PUMA isimli robot üretildi ve pazarlanmaya başlandı.

1979: Yamanashi Üniversitesi fabrikalardaki montaj hatlarında kullanılmak üzere SCARA kolunu tasarladı.

1979: Stanford Cart isimli gezer robot, üzerine monte edilmiş bir kameradan alınan görüntüleri kullanarak engellerle dolu bir odayı engelleri aşarak boydan boya geçti.

1986: Honda Motor Corporation insansı bir robot geliştirmek amacı ile gizli bir proje başlattı.

1988: Danbury Hastanesi'nde ilk yardımcı robot göreve başladı.

1993-1994: Önceki robotlara göre ucuz maliyetli ERRATIC ve PIONEER1 isimli gezer robotlar üretildi.

1993: MIT' den Rodney A. Brooks bir insan gibi yetiştirilen ve eğitilen robot Cob'u yapmaya başladı.

1994: Dante II, Carnigie Mellon Üniversitesi'nde geliştirilen yürüyen robot Alaska'da aktif bir volkana keşif gezisi yapar ve volkanik gaz örnekleri topladı.

1996: Honda, P2 (prototype 2), yürüyen insansı robot dünyaya tanıtıldı.

1997: İlk yıllık robotlar arası futbol turnuvası "Robocup" Japonya'da düzenlendi.

1997: NASA'nın Pathfinder uzay aracı Mars'a iner ve "Sojourner" robotu Mars yüzeyinde keşif gezisi yaptı.

1998: Robot oyuncak FURBY piyasaya çıktı [6-7].

2000: RoboCup 2000' de üç insansı robot ilk defa karşılaşılır. Batı Avustralya Üniversitesinden Johnny Walker, Japonya Aoyama Gakuin Üniversitesinden Mk-II ve Pino [8].

2003: NASA Mars' a iki robot görevi göndermeyi planlıyor[8].

RoboCup' un asıl hedefi ise 2050 yılında dünya şampiyonu futbol takımına karşı 90 dakika mücadele edecek ve kazanacak tamamen otonom robotlardan oluşan bir futbol takımı kurmak. Merdiven çıkmayı becerebilen insan boyutlarında bir robotun çok yakın zamanda ve ne kadar büyük bir bütçe ile imal edildiği düşünülürse, bu hedefin ne kadar yüksek olduğu daha iyi anlaşılabilir.[9]

Endüstriyel robot, parçaları, maddeleri, aletleri ya da özel işlem aletlerini, çeşitli programların hareketleri yönünde, istenen görev doğrultusunda hareket edebilen bir tasarımdır [6-7].

2.3. Robotların Sınıflandırılması

Robotlar kendi aralarında çeşitli kısıtlara göre sınıflandırılmaktadırlar. Aşağıda en çok yapılan sınıflandırma örnekleri verildi [5-6-10].

2.3.1. Tahrik sistemlerine göre robotlar

Robotun önemli elemanı tahrik sistemidir. Bu tahrik sistemi robotun hareketini sağlar. Robotun kullanacağı alana ya da gerek duyduğu güce göre tahrik sistemleri 3 çeşittir. Genellikle sanayide kullanılan bu sistemler:

1. Hidrolik sistem,
2. Elektrikli sistem,
3. Pnömatik sistem.

Hidrolik sistem:

Hidrolik tahrik sistemi, robota büyük hız ve güç verir. Bu sistem, mafsalların doğrusal ve dairesel hareket etmesini sağlayacak şekilde tasarlanır. Hidrolik sistemin temel dezavantajı robotun fazla yer işgal etmesidir. Ayrıca, sızma problemi vardır. Yüksek hız ve güç sağlandığından bu sistem birçok sanayi robotunda kullanılmaktadır. Sprey boyamadaki gibi elektrikli sistemlerin yangın çıkartma tehlikesi yüksek olan alanlarda hidrolik robotlar kullanılmaktadır.

Elektrikli sistem:

Hidrolik sistemler ile karşılaştırıldığında, elektrikli sistemler, daha az hız ve güç sağlarlar. Bu yüzden elektrikli sistemler daha küçük robotlarda kullanılır. Fakat bu

sistemler daha doğru ve daha iyi tekrarlayabilme kabiliyetinde ve kullanımı daha temizdir. En yaygın olarak sanayide bu tip robotlar kullanılır. Nümerik kontrollü tezgahlarda olduğu gibi bu tip robotlar iki grupta sınıflandırılır: Adım motorlular ve doğru akımlı servo motorlulardır. Adım motorlu robotların çoğu açık döngü tipindedir, fakat geri besleme döngüleri bu robotlarda ortaktır. Servo sistemli robotlar, sistem ile robot arasında sabit olan geri besleme döngülerine sahiptirler.

Pnömatik sistem:

Pnömatik tahrikli sistemler, genellikle daha küçük robotlarda kullanılır. Bu robotlar daha az serbestlik dereceli ve malzemeleri bir yerden alıp başka bir yere nakletme işlemlerinde kullanılır. Bu işlemler genellikle basit ve kısa sürelidir. Pnömatik güç, doğrusal veya dairesel eklemler için kullanılır. Pnömatik robotlar, elektrikli veya hidrolik robotlardan daha ucuzdur. Normal olarak, imalat makineleri ve preslerde kullanılan yaklaşık olarak inç kareye düşen 90 pount' luk sıkıştırılmış hava basıncı, ekstra maliyet gerektirmeksizin robotlarda kullanılır, Fakat çoğunlukla, pnömatik robotlar mekanik olarak her bir eksen için sabit noktalı işlemler yaparlar. Bunlar, sınırlı hareketler yapan sıra robotlardır. Bu robotların büyük avantajı basit modüler yapıda olduğundan standart mevcut parçalar kullanılmasıdır. Bu da, bir firma için maddi açıdan önemli ölçüde kazanç sağlar [5-6].

2.3.2. Robot eksenlerine göre sınıflandırma

Bir robot hareketinin kapasitesi, kontrol edilebilmesi mümkün olan eksenlerdeki hareketlerle belirlenir. Sayısal denetimdeki hareketlere çok benzerdir. Endüstriyel robotlar değişik tip ve boyutlarda yapılmaktadırlar. Çeşitli kol hareketlerini yapabilirler ve farklı hareket sistemlerine sahiptirler [7].

Robot hareketinin eksenleri:

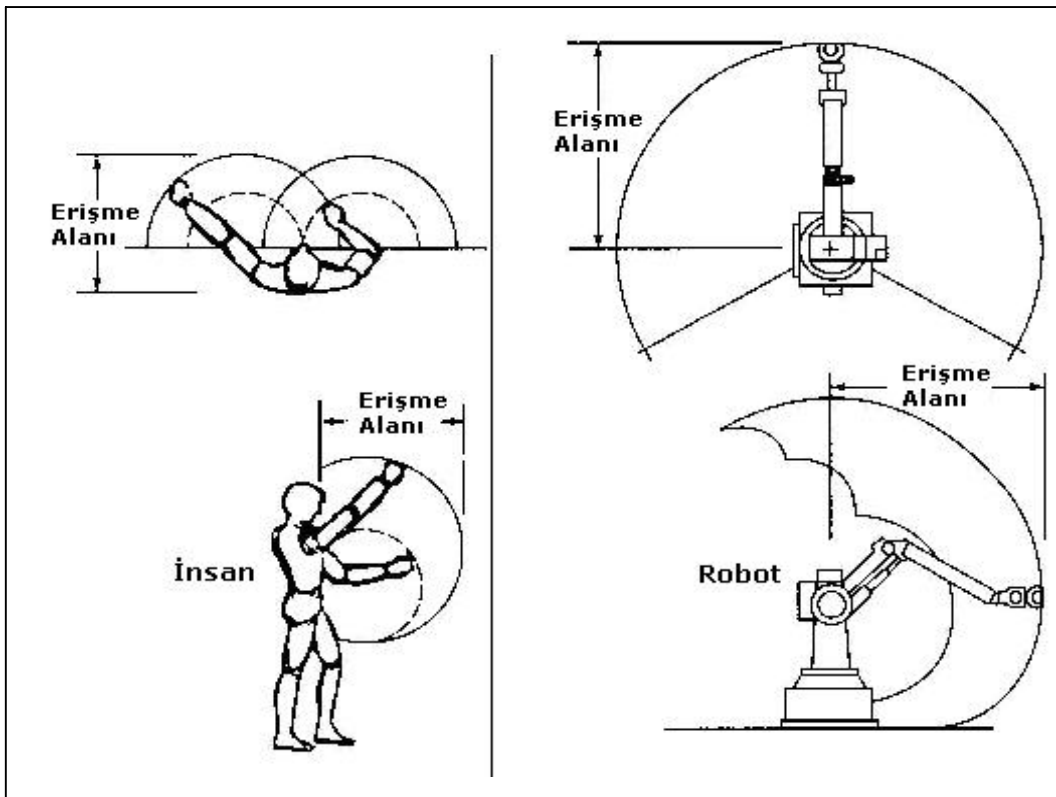
Manipülâtörün serbestlik derecesi diye tanımlanan hareketleri vardır. Eğer bir manipülâtör kendi eksenini etrafında dönüyorsa, bu robota “tek eksenli robot” denir.

Eğer manipülâtör yukarı ve aşağı doğru hareket ediyorsa, bu robota “çift eksenli robot” denir. Kendi ekseni etrafında dönen ve yukarı aşağı hareket eden manipülâtör, yatay ekseninde ileri – geri hareket de edebilir. Bu robota “üç eksenli robot” denir. Endüstriyel robotlar en az üç eksene sahiptirler. Bu hareketler, kendi ekseni etrafında dönmesi, yukarı-aşağı (dikey) ve ileri-geri (yatay) hareket edebilmesidir.

Çalışma alanı:

Robot denildiği zaman aklımıza ilk etapta, insan gibi yürüyen, insan davranışları sergileyen, daha da önemlisi insan gibi düşünen ve karar verebilen makineler geliyor. Bu da demek oluyor ki robotların tasarlanması ve geliştirilmesinde canlıların yaşama uyum sağlamak amacıyla geliştirdikleri karakteristiklerden ilham alınmaktadır.

Robot kolunun yetişebileceği toplam alana, çalışma alanı denir. Şekil 2.1’ de mafsallı robot ile insan kolunun çalışma alanı arasındaki benzerlik görülmektedir [7].



Şekil 2.1. İnsan kolu ile mafsallı robotun çalışma alanları arasındaki benzerlik

2.3.3. Koordinat sistemlerine göre robotlar

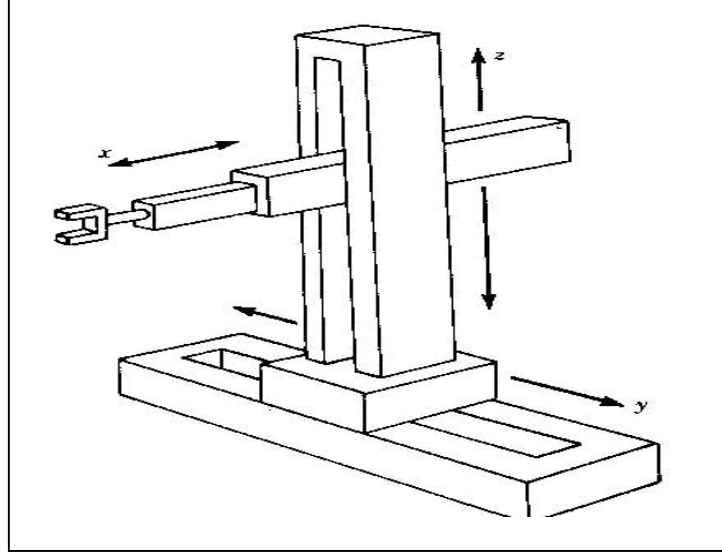
Koordinat sistemlerine göre robotlar dört kısımda incelenir [5-6-7]:

1. Kartezyen koordinat sistemi,
2. Silindirik koordinat sistemi,
3. Küresel koordinat sistemi,
4. Döner koordinat sistemi.

Kartezyen koordinat sistemi:

Bu sistemde bütün robot hareketleri; birbirlerine karşı dik açılı şekilde olur (Bkz. Şekil 2.2). Bu konfigürasyon en kısıtlı hareket serbestine sahip robot tasarım şeklidir. Bazı parçaların montajı için gerekli işlemler kartezyen konfigürasyonlu robotlar tarafından yapılır. Bu robot şekli birbirine dik üç eksenle hareket eden kısımlara sahiptir. Hareketli kısımlar X, Y ve Z kartezyen koordinat sistemi eksenlerine paralel hareket ederler. Robot, üç boyutlu dikdörtgen prizması hacmi içindeki noktalara kolunu hareket ettirebilir.

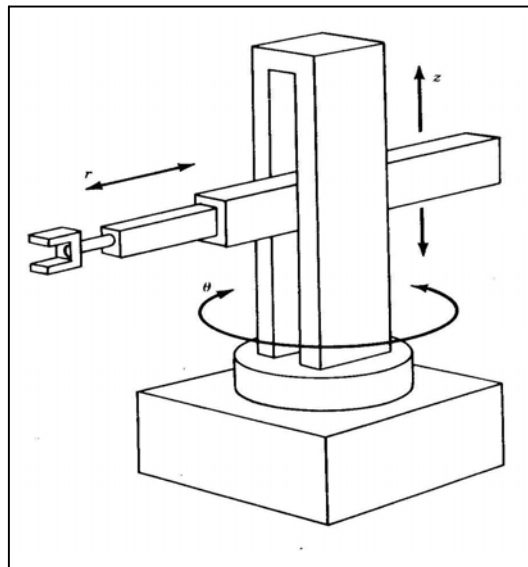
Bir kartezyen koordinat sisteminde, koordinat sistem merkezinin yeri, ilk iki bağlantının birleşme yerinin merkezidir. Merkezine doğru yapılan hareketler dışında, merkez hareket etmez, yani robotun merkezi sabittir. Robotun yerleştirildiği çalışma alanında eğer X yönündeki hattı bir kolona doğru çevrilirse, X hattı daima aynı kolona doğru yönelir robotun programını yaparken döndüğü yönde sorun yoktur. Bunlar verilmiş bir robot donanımı için, yer koordinatları olarak bilinir [6-7].



Şekil 2.2. Kartezyen koordinat sistemine ait şematik çizim

Silindirik koordinat sistemi:

Bu tip robotlar temel bir yatak etrafında dönebilir ve diğer uzuvları taşıyan ana gövdeye sahip özelliktedir (Bkz. Şekil 2.3). Hareket düşeyde ve ana gövde eksen kabul edildiğinde radyal olarak sağlanır. Dolayısıyla çalışma hacmi içerisinde robotun erişemeyeceği, ana gövdenin hacmi kadar bir bölge oluşur. Ayrıca genellikle, mekanik özelliklerden dolayı gövde tam olarak 360° dönemez.



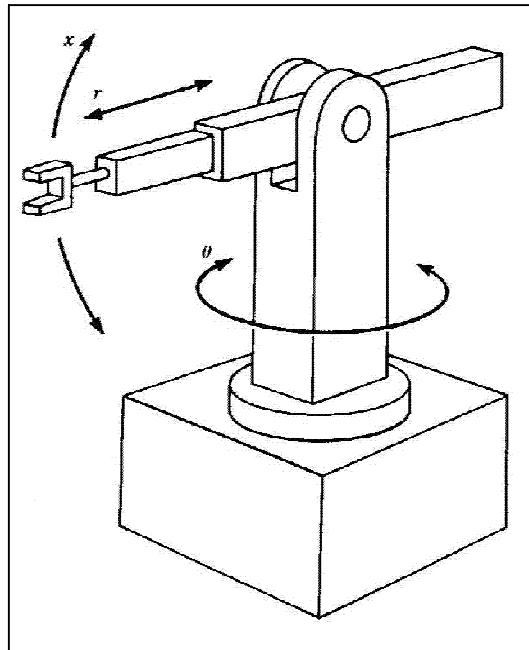
Şekil 2.3. Silindirik koordinat sistemi

Silindirik koordinatlarda tabana dik eksen etrafında dönme ve bu eksen üzerinde ötelenme yapılırken bu eksene dik bir eksende de başka bir öteleme hareketi yapılır. Dönme serbestliğindeki mekanik engellerden dolayı teorik olarak silindirik bir çalışma alanı oluşması beklenirken bazı bölgelerde silindir yapısı tamamlanamaz. Zemine ulaşabilmenin arzu edildiği durumlarda robot kolu zemine açılan bir yuvaya yerleştirilir. Ancak bu durumda da ulaşılacak maksimum yükseklik azalır. Radyal hareketten dolayı, silindirik koordinatlı robotlar montaj, kalıpcılık gibi alanlarda kullanılabilir. Bu tip robotlar da programlama açısından fazla karmaşık değildir. Ancak kartezyen koordinatlı robotlarda olduğu gibi kayar elemanların korozyon ve tozlanmadan korunması gerekir.

Silindirik robotlar genellikle, kendi ekseninde 300° dönmektedir. Geri kalan 60° ise robotun etrafında güvenli bir alan oluşturmak için kullanılır. Bu güvenlik alanına ölü bölge ismi verilmiştir [6-7].

Küresel koordinat sistemi:

Matematiksel olarak küresel koordinat sisteminin iki tane dairesel ve bir de doğrusal eksen olmak üzere üç tane eksen vardır (Bkz. Şekil 2.4).



Şekil 2.4. Küresel koordinat sistemi

Robotikte küresel koordinat sistemi en eski koordinat sistemlerinden biridir. Oldukça çok işlevli, birçok uygulama alanına sahip özelliğinin yanında, yapım ve montaj açısından da oldukça kolaylık sağlamaktadır.

Şekilden de anlaşıldığı gibi temelde iki hareketi mevcuttur. Bunlar yatay ve düşey dönmedir. Üçüncü bir hareket ise doğrusal (uzama kolunun ileri geri hareketi) harekettir. Doğrusal hareket aynen kartezyen koordinatlardan herhangi bir koordinatın hareketi gibi davranış gösterir.

Kutupsal koordinatlarda çalışan bir robotun çalışma hacmi iki kürenin ara hacminden oluşur. Koldaki uzuvlardan biri doğrusal hareket yaparken bunu destekleyen diğer uzuvlardan biri tabana dik eksen etrafında diğeri ise bu eksene dik ve tabana paralel eksen etrafında döner. Ölü bölgeler bu tip robotlarda da vardır. Öteleme hareketi yapan uzvun strokunun yetersizliğinden dolayı zemine ulaşmak mümkün olmaz.

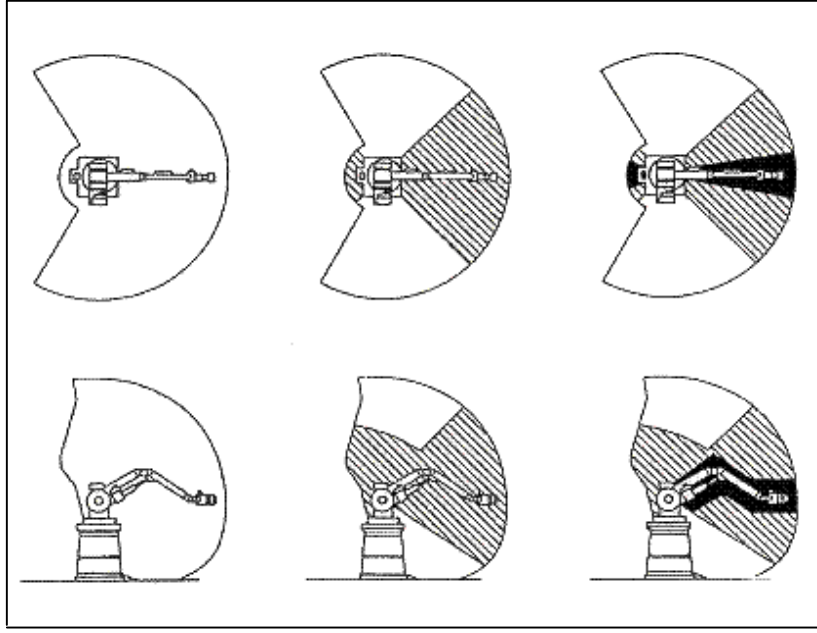
Döner koordinat sistemi:

Eğer bir robot herhangi bir iş yaparken kolu dairesel hareketli bağlamlarla oluşturuyorsa, bu tip robotlara Döner koordinat sistemli robotlar denir.

Robot kolunun bağlantıları gövde üzerine, etrafında dönecek şekilde monte edilmiştir ve dayanak noktaları birbirine benzeyen iki ayrı bölümü taşır. Döner parçalar yatay ve dikey monte edilebilir.

360° dönme sağlanamaz ancak bu kayıplar minimuma indirilebilir. Şekil 2.5' de döner koordinatlarda çalışma hacmi görülmektedir. Bu tip robotlarda robot kolun çalışması zor gözlenir. Çalışma hacmindeki noktalara farklı yörüngelerle ulaşılabilir. Buna göre sistem parametrelerinin en uygun olduğu yol seçilmelidir.

Döner koordinatlı robotlarda kontrol işlemi karmaşıktır, dolayısıyla kontrol donanımının da bu karmaşıklığı karşılayabilecek kapasitede olması gerekir. Ayrıca bu tip robotlarda mafsallarda sızdırmazlık kolayca sağlanabilmektedir [6-7].



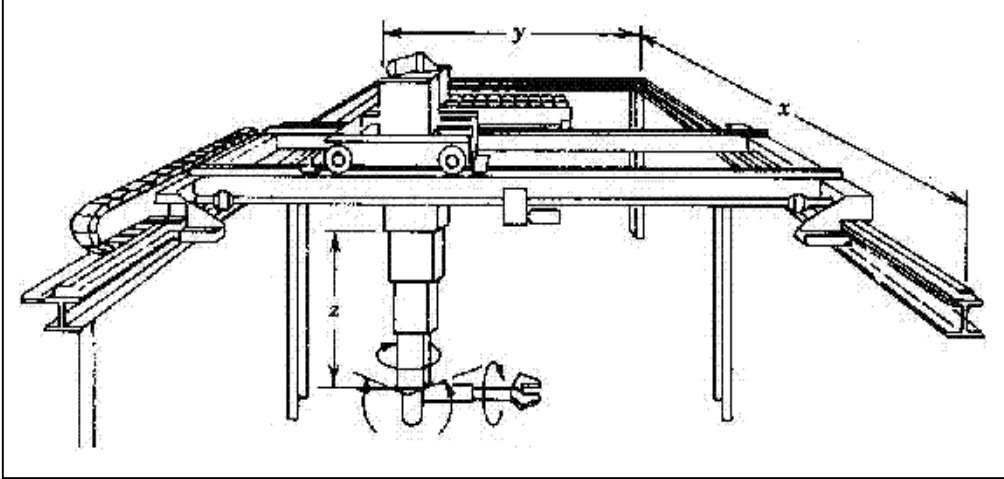
Şekil 2.5. Döner koordinat sistemli robotun çalışma alanı

2.3.4. Robot tiplerine göre sınıflandırma

1. Kartezyen robotlar,
2. Mafsallı robotlar,
3. SCARA tipi robotlar.

Kartezyen robotlar:

Kartezyen koordinat sisteminde bütün robot hareketleri birbirine 90° ' lik açıyla hareket eder. Bu nedenle kartezyen robotlar dikdörtgenimsi bir biçimdedir. Günlük hayatımızda sağa sola, aşağı yukarı vb. hareketlerimiz, kartezyen koordinat hareketlerdir (Bkz. Şekil 2.6).



Şekil 2.6. Kartezyen robot

Bu ürün robotları genellikle özel tatbiklerle sınırlandırılır. Devamlı bir yol alanında, robot, bir köprü ve bir ray sistemi aracılığıyla daha çok işlevlik kazanabilir.

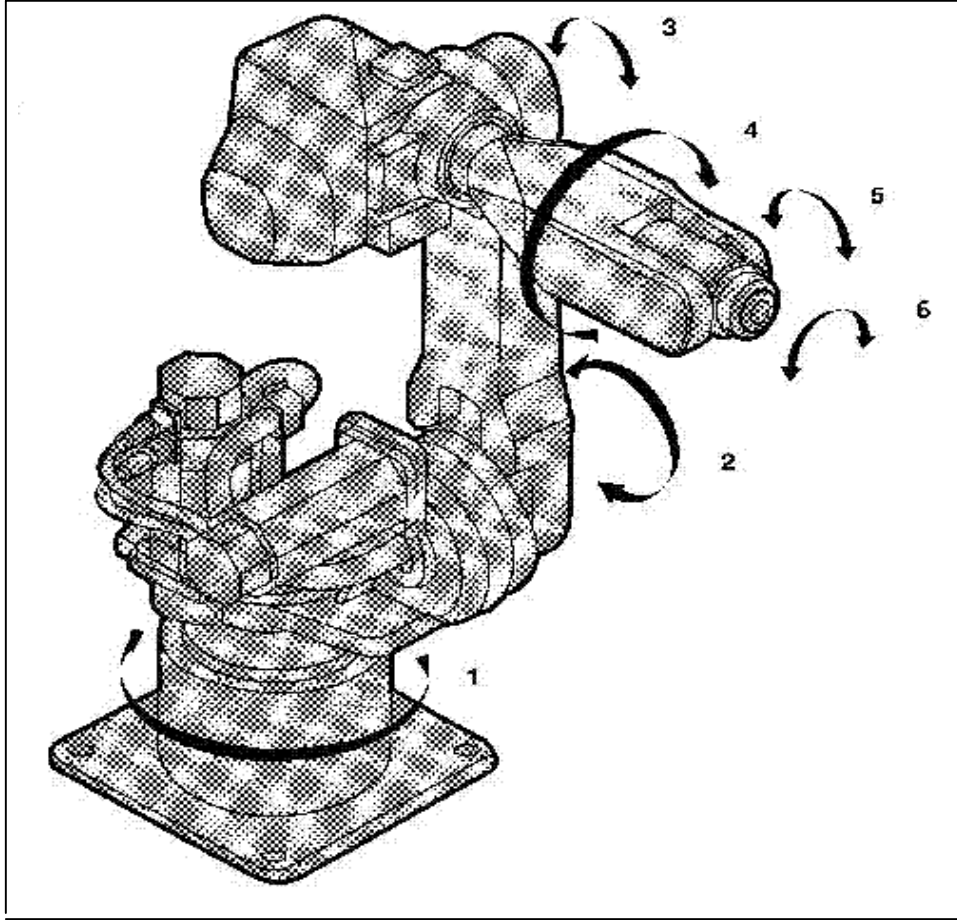
Tavana monte edilerek, birkaç fonksiyonla birçok istasyona hizmet verilebilir. Robotun tavana asılı olmasıyla, zeminde daha fazla boş saha kazanılmış olur.

Kartezyen robotlar, basitlikleri ve konstrüksiyonları sayesinde rijitliği yüksek ve hızlı bir yapıya sahiptir [7-10].

Mafsallı robotlar:

Mafsallı robotların dizaynı insan kolundan esinlenerek yapılmıştır. Kol eklemleri robotlar yeteneklerine göre, insan kolunun yerine getirebileceği görevleri üstlenmek amacıyla yapılmışlardır. Kol eklemleri robotlar insan kollarında olan tüm esnekliğe ve hassasiyete tam olarak sahiptir ve değişik görevlerde insan kolunu taklit eder.

Kol eklemleri robotlar altı eksenle de rahatça hareket ederler. Bu altı eksenle üç tanesi kol hareketi için, diğer üç tanesi ise bilek hareketi içindir (Bkz. Şekil 2.7).



Şekil 2.7. 6 eksenli bir mafsallı robot

İnsan kolunun yapabileceği çok sayıda hareketi yapabilmektedirler. Bu özellikleri kullandıkları koordinat sisteminden (döner koordinat sisteminden) almaktadırlar. Bu koordinat sisteminin gereği olarak omuz, dirsek ve bilek bağlantıları vardır.

Bu bağlantı şeklinin robota kazandırdığı en büyük avantaj, çalışma alanındaki her noktaya rahatça ulaşabilmesidir. Çalışma alanı ise; robot kolunun yatayda dik olarak durması sonucu elde edilir [6-7-10].

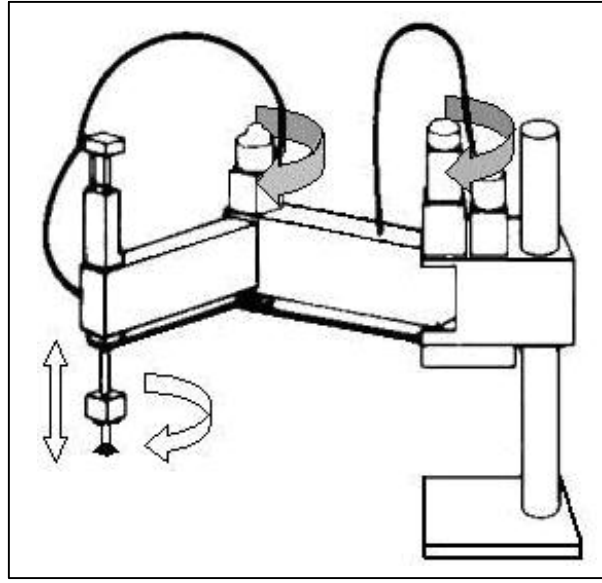
SCARA tipi robotlar:

SCARA, selectively compliant articulated robot arm kelimelerinin baş harflerinden oluşmuştur. Yani seçici serbest esnemeli robot kolu. Bu robot 1970' den sonra Japon Endüstriyel Konsorsiyumu ve bir grup araştırmacı tarafından Japonya' da Yamanashi

Üniversitesinde geliştirilmiştir. SCARA tipi robot, çok yüksek hıza ve en iyi tekrarlama kabiliyetine sahip olan bir robot çeşididir.

Şekil 2.8’ de SCARA tipi bir robota ait şematik çizim verilmiştir. Bu robotta üç genel özellik bulunmaktadır:

1- Doğruluk, 2- Yüksek hız, 3- Kolay montaj.

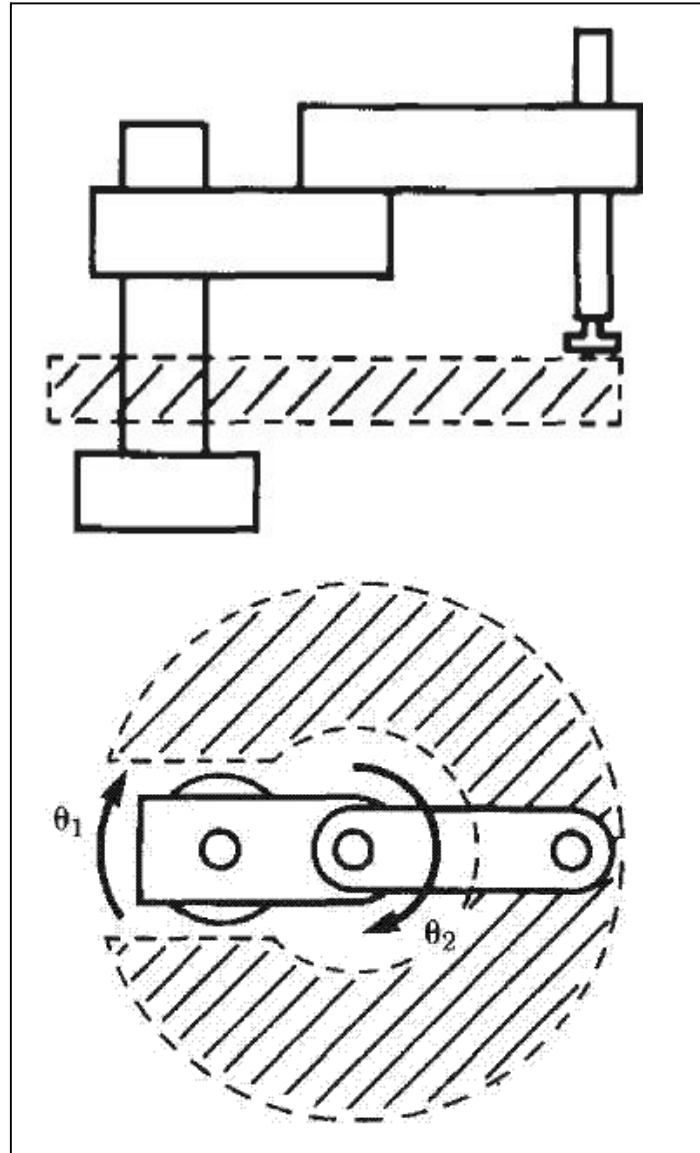


Şekil 2.8. Robota ait şematik çizim

Bu robot genellikle dikey eksen çevresinde dönen 2 veya 3 kol bölümünden meydana gelmiştir. Şekil 2.8’ de görülen 1 numaralı eksen robota ana dönmeyi veren eksenidir. Bu eksen en çok montaj robotlarında kullanılmaktadır.

2 numaralı eksen doğrusal dikey eksenidir. Bu eksende sadece dikey hareket yapılabilmektedir. Bu özellik montaj robotlarında istenildiğinden dolayı, montaj robotlarının büyük bir kısmı aşağıya doğru dikey hareket yapar.

Dikey eksen hareketleri koordinat hareket eksenleri içinde aşağıya doğru yapılan en çabuk ve düzgün hareketlerdir. 3 numaralı eksende robot kolunun erişebileceği uzaklık değiştirilebilir. 4 numaralı eksende ise dönen kol bileği hareket eder. Şekil 2.9’ da robotun çalışma alanına ait çizdiği hacim verilmiştir [6-7-10].



Şekil 2.9. Robotun çalışma alanı

2.3.5. Kontrol sistemlerine göre robotlar

1. Sınırlı hareket robotları,
2. Noktadan noktaya kontrollü playback robotları,
3. Sürekli yol kontrollü playback robotları,
4. Zeki robotlar.

Sınırlı hareket robotları:

Sınırlı hareket robotları, eklemlerinin izafi pozisyonlarını göstermek için servo kontrol kullanmazlar. Bunun yerine, her eklemin hareketi boyunca yapacağı duruşlar switchler ile veya mekanik durdurucularla belirlenir. Pozisyonların ve duruş sınırlarının böyle belirlenmesi anladığımız manada bir robot programlama olmayıp, mekanik bir ayarlama işlemidir. Bu tür kontrol sisteminde eklemler sadece limitleri içerisinde hareket edebilirler [6].

Noktadan noktaya kontrollü playback robotları:

Arzu edilen bir dizi noktada hareket çevrimleri ve benzeri hareketler yapma yeteneğine sahiptirler. Önce her nokta robotun kontrol ünitesine kaydedilir. Playback boyunca robot, bir noktadan ötekine istenilen sırada gidecek şekilde kontrol edilir. Burada robot gidilen yolu takip etmez. Eğer programcı yolda küçük bir değişiklik yapmak isterse, robot programı yeniden değiştirilerek robota yüklenmelidir [5-6].

Sürekli yol kontrollü playback robotları:

Robotun kontrol edildiği yol boyunca hareket çevrimleri yapma yeteneğine sahiptirler. Bu genellikle istenilen yolu tarif eden birbirine yakın noktaların takip edilmesiyle olur. Bu noktalar programcı yerine kontrol ünitesi tarafından sağlanır. Programcı sadece yolun başlangıcını ve bitimini verir. Kontrol ünitesi düz çizgiler oluşturacak şekilde noktalar belirler. Günümüzde bu işlemi yapan kontrol ünitesi olarak bilgisayarlar kullanılmaktadır [5-6].

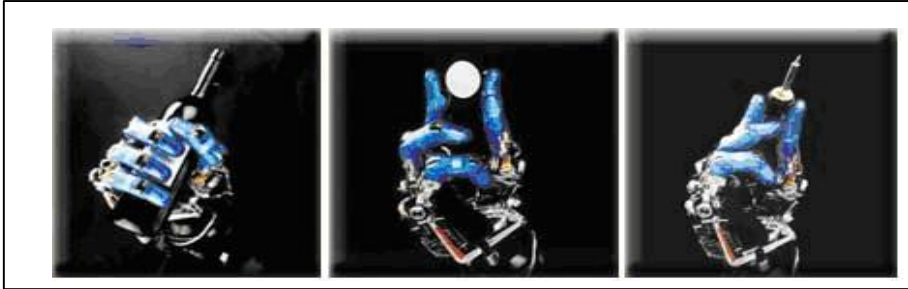
Zeki robotlar:

Sadece programlanmış bir hareketi tekrar etmekle kalmazlar, ayrıca istenildiğinde zeki denebilecek bir şekilde çevresiyle etkileşimde bulunma yeteneğine de sahiptirler. Zeki robotlar, iş yerinde ortaya çıkan koşullara göre programlanmış çevrimini değiştirebilirler. Operasyonda elde edilen verilere göre mantıklı karar verebilirler [6].

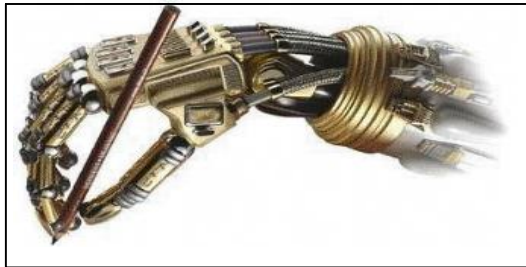
2.4. Robotlarda Tutucular

Robot uygulamalarında uç eleman olan tutucular parçaların taşınmasında montaj işlemlerinde, kaynak işlemlerinde, boyama işlemlerinde çok rahat bir şekilde kullanılmaktadır. Fakat bir montaj hattında aynı tutucunun birden fazla işi yapması veya değişik özellikteki parçaları taşınması düşünüldüğünde bunun işlevsel bakımdan zorlukları görülmektedir. Bu durumda genel maksatlı robot ele ihtiyacı duyulduğu bir gerçektir [5-6-7-10].

Robot ellerde eklemlerin hareketi için gerekli gücü üreten çeşitli teknolojilere dayalı hareketlendiriciler (actuator) vardır. En yaygın olarak kullanılan hareketlendirici teknolojileri elektrik motorları, hidrolik hareketlendiriciler ve pnömatik hareketlendiricilerdir. Bu geleneksel hareketlendiriciler dışında şekil bellekli alaşımları da hareketlendirici teknolojilerine dahil etmek mümkündür (Bkz. Şekil 2.10 ve 2.11).



Şekil 2.10. Robot tutucu



Şekil 2.11. Robot el

Robot tutucuları (iş parçası tutucuları), tutacakları iş parçasının şekline göre çeşitli konstrüksiyonlarda elde edilebilirler.

Endüstriyel robotlarda iş parçasını tutmak, taşımak ve benzeri amaçlar için kullanılan tutucular aşağıda sıralanmıştır [7].

- a. Mekanik tutucular,
- b. Vakumlu tutucular,
- c. Manyetik tutucular,
- d. Yapışkanlı tutucular.

2.5. Robot Sensörleri

Robotun çalışması, eklem ve bileklerindeki belirli hareketlerden meydana gelir. Bu hareketler elde edilirken, robotun çalışmasının belirli ve düzgün olması önemlidir. Tahrik sistemlerinin robot hareketlerinin düzenli olması için uygun araçlarla kontrol edilmesi gerekir. Kontrolleri sırasında, robotların çevrelerindeki karakteristik özelliklerine duyarlı olmalıdırlar. Bu karakteristikler, kontrol sistemlerinin manipülatör hareketlerinin verimli olmasını mümkün kılmak için geri besleme sağlar ve robotlara da daha çok esneklik verir. Görsel sensörler gibi sensörler, daha çok zeki robotlarda kullanışlıdır. Sensörler, faydalarına göre birçok farklı yöntemlerle sınıflandırılabilir. Robotlarda kullanılan bazı sensör çeşitleri şunlardır [5];

- a. Pozisyon sensörleri,
- b. Alan sensörleri,
- c. Hız sensörleri,
- d. Yakınlık sensörleri.

Sensörlerin robotlarda kullanılması:

Endüstriyel robotlarda ve otomatik imalat sistemlerinde kullanılan önemli sensörleri 4 temel kategoriye ayırabiliriz [6].

1. Güvenliđi kontrol edenler,
2. alıřma alanındaki kontrolleri birbirine bađlayanlar,
3. Kalite kontrolü iin paraları denetleyenler,
4. Robot alıřma alanındaki nesnenin pozisyonuna karar vermek ve diđerleri ile iliřkileri hakkında bilgi vermek iin kullanılanlar.

2.6. Robotların Uygulama Alanları

Günümüz alıřma řartları ve rekabet ortamında, yapılan iřin mükemmelliđi ve kalitesi büyük önem kazanmış durumdadır . İřte bu řartlar altında robot kullanımıyla, kalite arttırılmakta, standart üretim sađlanmakta, iřilik ve malzeme giderleri azaltılmaktadır. Böylece robot sistemine sahip řirketlerin rakipleriyle arasındaki rekabet güçleri artmaktadır. Bunların yanında, robotlar insanları monoton ve ağır hacimli iřlerden, kaynakhane ve boyahanenin zehirleyici etkili ortamlarından kurtarırlar. Dar alanlarda bir ok iřlemin yapılması imkanını tanırlar. Pek ok alanda üretime katkıları yadsınamayan robotlar, gelişimleri boyunca hep memnunlukla karşılanmamışlar, zaman zaman toplumsal alkantılara da yol açmışlardır. Buna örnek olarak, otomatik dokuma tezgahlarının son yüzyılda neden olduđu işsizlik gösterilebilir [5].

Ancak, her seferinde teknolojik gelişmenin hemen ardından gelen nesil daha iyi koşullarda alışmış ve alıřma zamanını kısaltmak suretiyle, daha ok serbest zaman elde etmiştir. Son zamanlarda yapılan ve gelişmiş ülkeleri kapsayan bir arařtırmaya göre son 130 yılda kiři başına üretkenlik yaklaşık 25 kat artmıştır.

Herhangi bir alanda robot kullanımının düşünülmesi ařađdaki temel faktörlere bađlıdır;

- a. Üretimde esneklik ve prodüktivitenin yükseltilmesi,
- b. İnsan sađlığını tehdit eden bir tehlikenin bulunması,
- c. İş gücünün zor bulunması ve pahalı olması; ayrıca insanların bu işleri yapmak istememesi,

- d. Üretimde bozuk parça sayısının azaltılması ve malzeme tasarrufu,
- e. Eğitim, hizmet, sağlık, güvenlik alanlarında çeşitli kolaylıkların sağlanması.

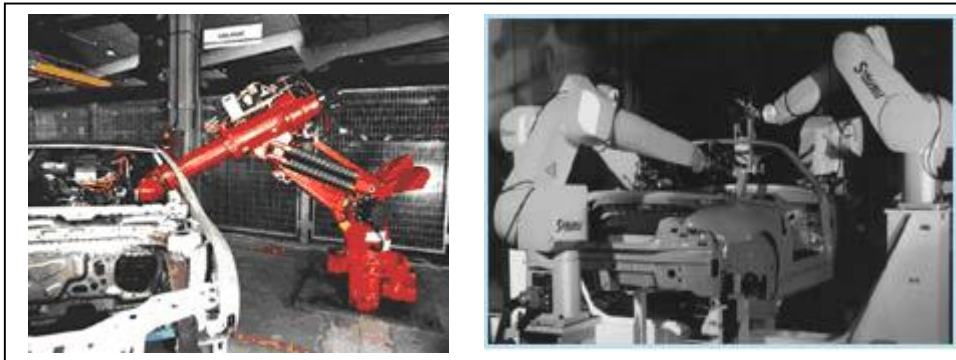
Bununla beraber bugün bazı alanlarda robot kullanılması kaçınılmaz olmuştur. Robotların kullanıldığı alanlar şöyle sıralanabilir:

Parça seçme, sıralama, yerleştirme, tezgaha yöneltme [5-7],



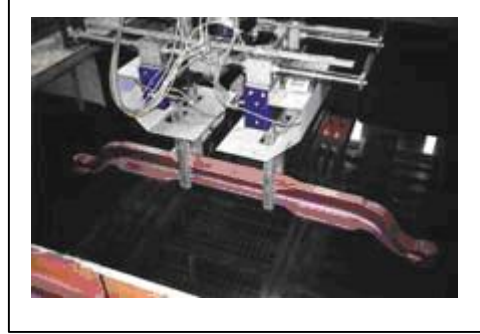
Şekil 2.12. Malzeme taşıma robotu

Parçaların montajı,



Şekil 2.13. Montaj robotları

Sıcak parçaların (dövme döküm gibi) tezgaha yüklenmesi ve boşaltılması (ısıl işlemler),



Şekil 2.14. 1400 °F deki döküm parçası kaldırılırken



Şekil 2.15. Döküm parçası buharda temizlenirken

Bitmiş parçaların ölçü ve kontrolü,



Şekil 2.16. Ölçme ve kontrol robotları

Boya işlemlerinde (özellikle otomotiv sanayisinde),



Şekil 2.17. Boyama robotları

2.7. Robotların Kullanım Avantajları

Günümüz çalışma şartları ve rekabet ortamında, yapılan işin mükemmelliği ve kalitesi büyük önem kazanmış durumdadır.

İşte bu şartlar altında robot kullanımıyla, kalite arttırılmakta, standart üretim sağlanmakta, işçilik ve malzeme giderleri azaltılmaktadır. Böylece robot sistemine sahip şirketlerin rakipleriyle arasındaki rekabet güçleri artmaktadır.

Bunların yanında, robotlar insanları monoton ve ağır hacimli işlerden, kaynakhane ve boyahanenin zehirleyici etkili ortamlarından kurtarırlar. Dar alanlarda bir çok işlemin yapılması imkanını tanırlar.

Son zamanlarda yapılan ve gelişmiş ülkeleri kapsayan bir araştırmaya göre son 130 yılda kişi başına üretkenlik yaklaşık 25 kat artmıştır. Bu üretkenlik artışının yarısı yani 13 kat kadarı fiziki ürün artışı, diğer yarısı da insanların çalışma sürelerinin yaklaşık yarı yarıya düşmesi şeklinde görülmüştür.

Fiziki ürün artışı ancak, otomasyon, anında üretim (just-in-time) ve esnek (flexible) üretim ile gerçekleşebilmektedir. Bugün yarı yarıya çalışıp 13 kat daha yüksek bir refah seviyesinde yaşamak da sadece sanayi devriminin getirdiği makineleşme, otomasyon ve günden güne artan robot kullanımı sayesinde gerçekleşmiştir [5].

Herhangi bir alanda robot kullanımının düşünülmesi aşağıdaki temel faktörlere bağlıdır;

- a. Üretimde esneklik ve üretkenliğin artırılması,
- b. İnsan sağlığını tehdit eden bir tehlikenin bulunması,
- c. İş gücünün zor bulunması ve pahalı olması; ayrıca insanların bu işleri yapmak istememesi,
- d. Üretimde bozuk parça sayısının azaltılması ve malzeme tasarrufu.

BÖLÜM 3. YAPAY SİNİR AĞLARI

3.1. Giriş

Yapay sinir ağı kavramı beynin çalışma ilkelerinin sayısal bilgisayarlar üzerinde taklit edilmesi fikri ile ortaya çıkmış ve ilk çalışmalar beyni oluşturan biyolojik hücrelerin, ya da literatürdeki ismiyle nöronların matematiksel olarak modellenmesi üzerinde yoğunlaşmıştır. Bu çalışmaların ortaya çıkardığı bulgular, her bir nöronun komşu nöronlardan bazı bilgiler aldığı ve bu bilgilerin biyolojik nöron dinamiğinin öngördüğü biçimde bir çıktıya dönüştürüldüğü şeklinde idi. Bugün yapay sinir ağı olarak isimlendirilen alan, birçok nöronun belirli biçimlerde bir araya getirilip bir işlevin gerçekleştirilmesi üzerindeki yapısal olduğu kadar matematiksel ve felsefi sorunlara yanıt arayan bir bilim dalı olmuştur. İnsan beyninin çalışma mekanizmasını taklit etmeye çalışan bu sistemler, her ne kadar günümüz teknolojisinin ürettiği, birim işlem zamanı nanosaniyeler mertebesinde olan silikon lojik kapılar ile gerçekleştirilebilirler de insan beyninin birim işlem zamanı milisaniyeler mertebesindeki nöronlarının toplu biçimde ele alındıklarındaki işlevselliklerinden çok uzakta kalırlar. Yapay sinir ağı, karar hızı açısından insan beyni ile yarışabilecek aşamayı henüz kat edememiş olmalarına rağmen, karmaşık eşleştirmelerin hassas bir biçimde gerçekleştirilebilmesi ve yapısal gürbüzlüğe sahip olmaları nedeniyle gün geçtikçe uygulama alanları genişlemektedir [1].

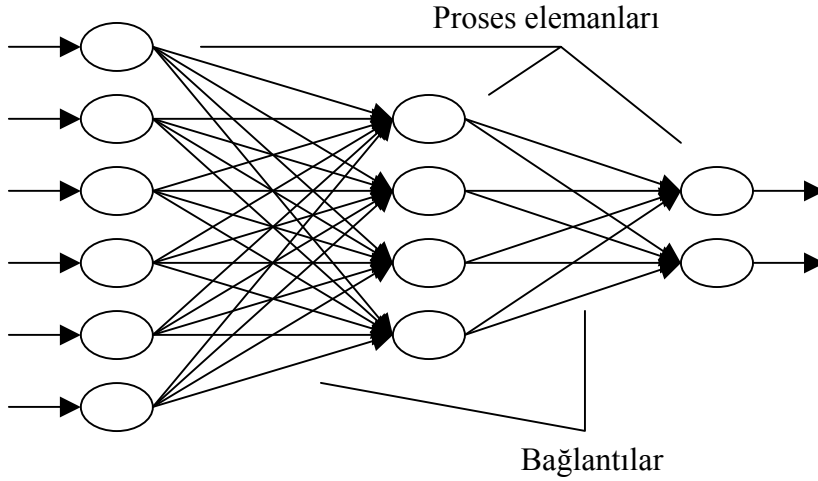
Yapay sinir ağı sayısal bilgisayarlardan çok farklı özellikler gösterirler. Tablo 3.1’ de sayısal bilgisayarlar ve yapay sinir ağlarının bazı özellikleri verilmiştir [11].

Tablo 3.1. Sayısal bilgisayarlar ve yapay sinir ağlarının bazı özellikleri [11]

Sayısal Bilgisayarlar	Yapay Sinir Ağları
Tümdengelimli usavurma: Çıkış üretmek için giriş bilgilerine bilinen kurallar uygulanır.	Tümevarımlı usavurma: Giriş ve çıkış bilgileri (eğitilen örnekler) verilir, kuralları biz koyarız.
Hesaplama merkezi, eş zamanlı ve ardışıldır.	Hesaplama toplu, eşzamansız ve öğrenmeden sonra paraleldir.
Bellek paketlenmiş, hazır bilgi depolanmış ve yer adreslenmiştir.	Bellek ayrılmıştır, dahildir ve içerik adreslenebilir.
Hata toleransı yoktur.	Eğer bilgi, gürültü ve kısmi ise kurallar bilinmiyorsa ya da karışıkta hata toleransı uygulanabilir.
Hızlıdır.	Yavaşdır.
Bilgiler ve algoritmalar kesindir.	Yapay sinir sistemleri deneyimden yararlanır.

Bu anlamda yapay sinir ağları konusu üzerinde çalışırken, bir ağ yapısının çözebileceği problem uzayının, insan beyninin çözebildiği problem uzayının oldukça kısıtlanmış bir alt kümesi olacağı gözden kaçırılmamalıdır. Diğer bir deyişle insan beyninin bilgiyi işlemedeki, kavramların ilişkilendirilmesindeki ve çıkarım mekanizmalarındaki üstünlüğü, bu kitapta tartışılan yaklaşımlar ile çürütülebilecek bir sav değildir. Sinir ağları kavramını çekici kılan aşağıda sıralanmış temel özelliklerin algılanışında bu noktanın gözden kaçırılmamasında yarar vardır [1].

Birinci özellik sistemin paralelliği ve toplumsal işlevin yapısal olarak dağılımlılığıdır. Diğer bir deyişle birçok nöron eşzamanlı olarak çalışır ve karmaşık bir işlev çok sayıda küçük nöron aktivitesinin bir araya gelmesinden oluşur. Bu da, zaman içerisinde herhangi bir nöronun işlev dışı kalması durumunda ağ başarımının dikkate değer ölçüde etkilenmeyeceği anlamına gelir (Bkz. Şekil 3.1) [1-12].



Şekil 3.1. Bir yapay sinir ağı örneği

İkinci özellik ise genelleme yeteneği, diğer bir deyişle ağ yapısının, eğitim esnasında kullanılan nümerik bilgilerden eşleştirmeyi betimleyen kaba özellikleri çıkarsaması ve böylelikle eğitim sırasında kullanılmayan girdiler için de anlamlı yanıtlar üretebilmesidir [1].

Bir başka özellik ise ağ fonksiyonunun nonlineer oluşudur. Yapı üzerinde dağılmış belli tipteki nonlineer alt birimler özellikle, istenen eşleştirmenin denetim ya da tanılama işlemlerinde olduğu gibi nonlineer olması durumunda işlevin doğru biçimde yerine getirilebilmesi için yapısal bir esneklik gerekliliği vurgulanmalıdır. Yani ağ parametreleri, başarıyı arttıracak, ya da maliyeti azaltacak şekilde değiştirilebilmelidir.

Belirtilebilecek son özellik, sayısal ortamda tasarlanan sinir ağı yaklaşımlarının tüm devre gerçekleştirilebilirliklerinin olmasıdır. Bu da yakın gelecekte bu sistemlerin, özellikle robotik uygulamaları ile birlikte düşünüldüğünde, günlük hayatta yaşam kalitesinin artırılmasında nedenli önemli bir rol oynayabileceklerine işaret eder.

Günümüzde birçok alanda yapay sinir ağlarının uygulamalarına rastlamak olasıdır. Özellikle örüntü tanıma, işaret işleme, sistem tanılama ve nonlineer denetim alanlarında yapay sinir ağlarının değişik modelleri ve değişik öğrenme stratejileri başarı ile kullanılmıştır. Burada her bir problemin çözümü için yapay sinir ağları

yaklaşımı ile önerilebilecek çözümler, tasarımcıya bazı seçenekler sunar. Bunlardan bazıları kimi zaman problemin doğası gereği seçenek olmaktan çok zorunluluk haline de gelebilir. Bu nedenle, tasarımcıların bu seçenekleri doğru değerlendirmeleri gibi bir zorunluluk her yaklaşımda olduğu gibi yapay sinir ağları alanında da söz konusudur [1].

3.2. Tarihçe

Yapay sinir ağlarının tarihçesi nörobiyoloji konusuna insanların ilgi duyması ve elde ettikleri bilgileri bilgisayar bilimine uygulamaları ile başlamaktadır. Yapay sinir ağları ile ilgili çalışmaları 1970 öncesi ve sonrası diye ikiye ayırmak gerekmektedir. Çünkü, 1970 yılında bu bilimin tarihinde bir önemli dönüm noktası başlamış ve o zamana kadar olmaz diye düşünülen bir çok sorun çözülmüş ve yeni gelişmeler başlamıştır. Her şey bitti derken yapay sinir ağları yeniden doğmuştur [11-12].

1890- İnsan beyninin yapısı ve fonksiyonları ile ilgili ilk yayının yazılması.

1911- İnsan beyninin bileşenlerinin belirli bir düzenek ile sinir hücrelerinden (nöronlar) oluştuğu fikrinin benimsenmesi.

1943- Yapay sinir hücrelerine dayalı hesaplama teorisinin ortaya atılması ve eşik değerli mantıksal devrelerin geliştirilmesi.

1949- Biyolojik olarak mümkün olabilen öğrenme prosedürünün bilgisayarlar tarafından gerçekleştirilecek biçimde geliştirilmesi.

1956-1962- Adaline ve Widrow öğrenme algoritmasının geliştirilmesi.

1957-1962- Tek katmanlı algılayıcıların geliştirilmesi.

1965- İlk makine öğrenmesi kitabının yayınlanması.

1967-1969- Bazı gelişmiş öğrenme algoritmalarının (Grosberg öğrenme algoritması) geliştirilmesi.

1969- Tek katmanlı algılayıcıların problemi çözme yeteneklerinin olmadığını gösterilmesi.

1969-1972 Doğrusal ilişkilendiricilerin geliştirilmesi.

1972- Korelasyon matris belleğinin geliştirilmesi.

1974- Geriye yayılım modelinin (çok katmanlı algılayıcının) ilk çalışmalarının geliştirilmesi.

Öğretmensiz öğrenmenin geliştirilmesi.

1. 1978- ART modelinin geliştirilmesi.
2. 1982- Kohonen öğrenmesi ve SOM modelinin geliştirilmesi.

1982- Hopfield ağlarının geliştirilmesi.

1982- Çok katmanlı algılayıcının geliştirilmesi.

1984- Boltzman makinesinin geliştirilmesi.

1985- Çok katmanlı algılayıcıların (genelleştirilmiş delta öğrenme kuralı ile) geliştirilmesi.

1988- RBN modelinin geliştirilmesi.

1988- PNN modelinin geliştirilmesi.

1991- GRNN modelinin geliştirilmesi [12].

Bugün, tüm dünyada sinir ağı ile ilgili birçok araştırma yapılmaktadır. Yapay sinir ağlarını eğitmek uzun süreler aldığından başlıca araştırmalar bu konuya odaklanmış görünmektedir. Burada amaç yeni ve daha verimli öğrenme algoritmaları, zamana bağlı olarak değişen modellere karşılık verebilen ağlar ve silikon geliştirmek şeklinde özetlenebilir.

Yapay sinir ağlarını kullanıma sunmak için özel yongalar gerekmektedir. Edinburgh Üniversitesinde bir sinir ağı yongası geliştirilmiştir. Özel bazı şirketler sayısal, analog ve optik olmak üzere 3 tip sinir yongası üzerinde çalışmaktadırlar. Bazı şirketler, özel derlenmiş devreler (ASIC) ile bir sinir ağı uygulamasını oluşturmak için silikon malzemeler üzerinde çalışmaktadırlar. ASIC' lar ve sinire benzer sayısal yongalar yakın gelecekte büyük bir başarıyla kullanılacağı gözükmektedir. Son olarak optik yongalar çok ümit verici gözükmektedir [11].

3.3. Genel Özellikler

Yapay sinir ağları, "biyolojik sinir sisteminde olduğu gibi, gerçek yaşam nesnelileriyle etkileşmeyi amaçlayan basit elemanların ve onların hiyerarşik düzenlemelerinin paralel, içice bağlantılı ağları" olarak tanımlanabilir (Kohonen 1987) [11-12].

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zor veya mümkün değildir.

Yapay sinir ağlarının karakteristik özellikleri uygulanan ağ modeline göre değişmektedir. İlgili modeller anlatılırken her modelin özellikleri ayrıntılı olarak anlatılacaktır. Burada bütün modeller için geçerli olan genel karakteristik özellikler verilmiştir. Bunlar şöyle sıralanabilir [12]:

1. YSA lar makine öğrenmesi gerçekleştirirler: Yapay sinir ağlarının temel işlevi bilgisayarların öğrenmesini sağlamaktır. Olayları öğrenerek benzer olaylar karşısında benzer kararlar vermeye çalışırlar.

2. Programları çalışma sitili bilinen programlama yöntemlerine benzememektedir: Geleneksel programlama ve yapay zeka yöntemlerinin uygulandığı bilgi işleme yöntemlerinden tamamen farklı bir bilgi işleme yöntemi vardır.

3. Bilginin saklanması: YSA da bilgi ağı, bağlantılarının değerleri ile ölçülmekte ve bağlantılarda saklanmaktadır. Diğer programlarda olduğu gibi veriler bir veri tabanında veya programın içinde gömülü değildir. Bilgiler ağın üzerinde saklı olup ortaya çıkartılması ve yorumlanması zordur.

4. YSA örnekleri kullanarak öğrenirler: YSA' nın olayları öğrenebilmesi için o olay ile ilgili örneklerin belirlenmesi gerekmektedir. Örnekleri kullanarak ilgili olay hakkında genelleme yapabilecek yeteneğe kavuştururlar (adaptif öğrenme). Örnek bulunamıyorsa ve yok ise YSA' nın eğitilmesi mümkün değildir. Örnekler ise gerçekleşmiş olan olaylardır. Mesela bir doktor hastasına bazı sorular sorar ve aldığı cevaplara göre teşhis ederek ilaç yazar. Sorulan sorular ve verilen cevaplar ile konulan teşhis bir örnek olarak nitelendirilir. Bir doktorun belirli bir zaman içinde hastaları ile yaptığı görüşmeler ve koyduğu teşhisler not edilerek örnek olarak alınırsa YSA benzer hastalıklara benzer teşhisi koyabilir. Elde edilen örneklerin olayı tamamı ile gösterebilmesi çok önemlidir. Ağa olay bütün yönleriyle gösterilemez ve ilgili örnekler sunulmaz ise başarılı sonuçlar elde edilemez. Bu ağın sorunlu olduğundan değil olayın ağa iyi gösterilemediğindendir. O nedenle örneklerin oluşturulması ve toplanması YSA biliminde özel bir öneme sahiptir.

5. YSA' nın güvenle çalıştırılabilmesi için önce eğitilmeleri ve performanslarının test edilmesi gerekmektedir: YSA' nın eğitilmesi demek, mevcut örneklerin tek tek ağa gösterilmesi ve ağın kendi mekanizmalarını çalıştırarak örnekteki ilişkileri belirlemesidir. Her ağı eğitmek için elde bulunan örnekler iki ayrı sete bölünürler. Birincisi ağı eğitmek için (eğitim seti) diğeri ise ağın performansını sınamak için (test seti) kullanılır. Her ağ önce eğitim seti ile eğitilir. Ağ bütün örneklere doğru

cevaplar vermeye başlayınca eğitim işi tamamlanmış kabul edilir. Daha sonra ağın hiç görmediği test setindeki örnekler ağı gösterilerek ağın verdiği cevaplara bakılır. Eğer ağı hiç görmediği örneklere kabul edilebilir bir doğrulukta cevap veriyor ise o zaman ağın performansı iyi kabul edilir ve ağı kullanıma alınarak gerekirse çevrimiçi (on-line) kullanılır. Eğer ağın performansı yetersiz olursa o zaman yeniden eğitmek veya yeni örnekler ile eğitmek gibi bir çözüme gidilir. Bu işlem ağın performansı kabul edilebilir bir düzeye gelinceye kadar devam eder.

6. Görülmemiş örnekler hakkında bilgi üretebilirler: Ağı kendisine gösterilen örneklerden genellemeler yaparak görmediği örnekler hakkında bilgiler üretebilir.

7. Algılamaya yönelik olaylarda kullanılabilirler: Ağlar daha çok algılamaya yönelik bilgileri işlemede kullanılırlar. Bu konuda başarılı oldukları yapılan uygulamalarda görülmektedir. Bilgiye dayalı çözümlerde uzman sistemler kullanılmaktadır. Bazı durumlarda yapay sinir ağı ve uzman sistemleri birleştirmek daha başarılı sistemler oluşturmayı sağlar.

8. Şekil (örüntü) ilişkilendirme ve sınıflandırma yapabilirler: Genel olarak ağların çoğunun amacı kendisine örnekler halinde verilen örüntülerin kendisi veya diğerleri ile ilişkilendirilmelidir. Diğer bir amaç ise sınıflandırma yapmaktır. Verilen örneklerin kümelenmesi ve belirli sınıflara ayrıştırılarak daha sonra gelen bir örneğin hangi sınıfa gireceğine karar vermesi hedeflenmektedir.

9. Örüntü tamamlama gerçekleştirebilirler: Bazı durumlarda ağı eksik bilgileri içeren bir örüntü veya bir şekil verilir. Ağın bu eksik bilgileri bulması istenir. Örneğin yırtık bir resmin kime ait olduğunu belirlemesi ve tam resmi vermesi gibi bir sorumluluk ağıdan istenebilmektedir. Bu tür olaylarda yapay sinir ağlarının çok etkin çözümler ürettiği bilinmektedir.

10. Kendi kendini organize etme ve öğrenebilme yetenekleri vardır: YSA' nın örnekler ile kendisine gösterilen yeni durumlara adapte olması ve sürekli yeni olayları öğrenebilmesi mümkündür.

11. Eksik bilgi ile çalışabilmektedirler: YSA kendileri eğitildikten sonra eksik bilgiler ile çalışabilir ve gelen yeni örneklerde eksik bilgi olmasına rağmen sonuç üretebilirler. Eksik bilgiler ile de çalışmaya devam ederler. Halbuki geleneksel sistemler bilgi eksik olunca çalışmazlar. Burada bir noktaya dikkatleri çekmekte fayda vardır. YSA' nın eksik bilgiler ile çalışması performanslarının düşeceği anlamına gelmez. Performansın düşmesi eksik olan bilginin önemine bağlıdır. Hangi bilginin önemli olduğunu ağ (network) kendisi eğitim sırasında öğrenmektedir. Kullanıcıların bu konuda bir fikri yoktur. Ağın performansı düşük olunca kayıp olan bilginin önemli olduğu kararına varılır. Eğer ağ performansı düşmez ise eksik olan bilginin önemli olmadığı anlaşılır.

12. Hata toleransına sahiptirler: YSA eksik bilgilerle çalışabilme yetenekleri hatalara karşı toleranslı olmalarını sağlamaktadır. Ağın bazı hücrelerinin bozulması ve çalışamaz duruma düşmesi halinde ağ çalışmaya devam eder. Ağın bozuk olan hücrelerinin sorumluluklarının önemine göre ağın performansında düşmeler görülebilir. Hangi hücrelerin sorumluluklarının önemli olduğuna da yine ağ eğitim esnasında kendisi karar verir. Bunu kullanıcı bilmemektedir. Ağın bilgisinin yorumlanamamasının sebebi de budur.

13. Belirsiz, tam olmayan bilgileri işleyebilmektedirler: YSA belirsiz bilgileri işleyebilme yetenekleri vardır. Olayları öğrendikten sonra belirsizlikler altında ağlar öğrendikleri olaylar ile ilgili ilişkileri kurarak kararlar verebilirler.

14. Dereceli bozulma gösterirler: YSA' nın hatalara karşı toleranslı olmaları bozulmalarının da dereceli olmasına neden olmaktadır. Bir ağ zaman içerisinde yavaş yavaş ve zarif bir şekilde bozulur. Bu eksik olan bilgiden veya hücrelerin bozulmasından kaynaklanır. Ağlar, herhangi bir problem ortaya çıktığı anda hemen anında bozulmazlar.

15. Dağıtık belleğe sahiptirler: YSA' da bilgi ağa yayılmış durumdadır. Hücrelerin birbirleri ile bağlantılarının değerleri ağın bilgisini gösterir. Tek bir bağlantının bir anlamı yoktur. Daha önce belirtildiği gibi ağın bilgilerinin açıklanamamasının sebeplerinden birisi de budur. Bu ağlarda, ağın tamamı öğrendiği olayın bütünü

karakterize etmektedir. O nedenle bilgiler ağı dağıtılmış durumdadır. Bu ise dağıtık bir belleğin doğmasına neden olmaktadır.

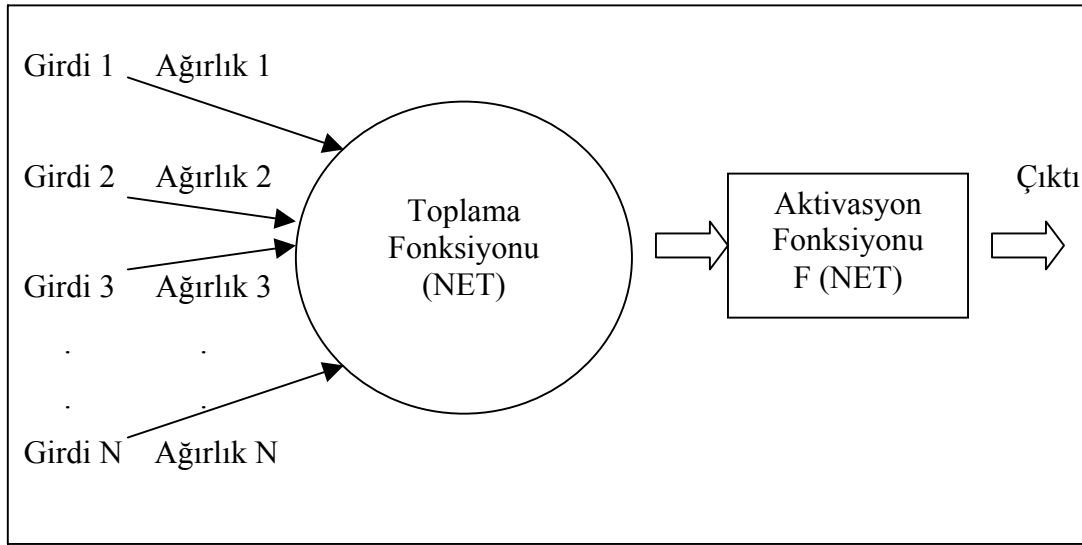
16. Sadece nümerik bilgiler ile çalışabilmektedirler: YSA sadece nümerik bilgiler ile çalışırlar. Sembolik ifadeler ile gösterilen bilgilerin nümerik gösterime çevrilmeleri gerekmektedir. Sembolik bilgilerin nümerik değerler ile ifade edilmesi de bilgilerin yorumlanmasını ve kararların açıklanmasını zorlaştırmaktadır [12].

Yukarıda belirtilen özelliklere ek olarak geliştirilmiş olan her modelin kendisine özgü özellikleri olabilmektedir.

Burada açıklanan özellikler dikkatlice incelenirse aslında yapay sinir ağlarının bilgisayar bilimine oldukça avantajlı katkılarının olduğu görülebilir. Geleneksel bilgisayar yazılım teknolojisi ile çözülemeyen birçok problemin yapay sinir ağları ile çözülebileceği görülebilir. Mesela yapay sinir ağları, eksik, normal olmayan, belirsiz bilgileri işleyebilen en güçlü problem çözme tekniğidir denilirse yanlış olmaz. Belirsiz bilgileri işlemde bulanık önermeler mantığı (fuzzy logic) gibi teknikler olsa bile eksik bilgi ile çalışabilen teknikler bulmak çok zordur [12].

3.4. Yapay Sinir Hücresi

Biyolojik sinir ağlarının sinir hücreleri olduğu gibi YSA da yapay sinir hücreleri vardır. Yapay sinir hücreleri mühendislik biliminde proses elemanları olarak da adlandırılmaktadır. Her proses elemanının 5 temel elemanı vardır (Bkz. Şekil 3.2). Bunlar [12-13]:



Şekil 3.2. Yapay sinir hücresinin yapısı

1. Girdiler: Bir yapay sinir hücresine dış dünyadan bilgiler gelir. Bunlar ağırlık öğrenmesi istenen örnekler tarafından belirlenir. Yapay sinir hücresine dış dünyadan olduğu gibi başka hücrelerden veya kendi kendinden de bilgiler gelebilir.

2. Ağırlıklar: Ağırlıklar bir yapay hücreye gelen bilginin önemini ve hücre üzerindeki etkisini gösterir. Şekildeki Ağırlık 1, Girdi 1'in hücre üzerindeki etkisini göstermektedir. Ağırlıkların büyük yada küçük olması önemli veya önemsiz olduğu anlamına gelmez. Bir ağırlığın değerinin sıfır olması o ağı için en önemli olay olabilir. Eski değerler önemsiz demek değildir. O nedenle artı veya eksi olması etkisinin pozitif veya negatif olduğunu gösterir. Sıfır olması ise herhangi bir etkinin olmadığını gösterir. Ağırlıklar değişken veya sabit değerler olabilirler.

3. Toplama fonksiyonu: Bu fonksiyon, bir hücreye giren net girdiyi hesaplar. Bunun için değişik fonksiyonlar kullanılmaktadır. En yaygın olanı ağırlıklı toplamı bulmaktır. Burada her gelen girdi değeri kendi ağırlığı ile çarpılarak toplanır. Böylece ağa gelen net girdi bulunmuş olur. Net girdi, 3.1 nolu denklem ile formülize edilmektedir.

$$NET = \sum_i^n Gi.Ai \quad (3.1)$$

Burada G girdileri, A ise ağırlıkları, n ise bir hücreye gelen toplam girdi sayısını göstermektedir. Yalnız YSA 'da daima bu formülün kullanılması şart değildir. Uygulanan YSA modellerinden bazıları kullanılacak toplama fonksiyonunu belirleyebilmektedir. Literatürde yapılan araştırmalarda toplama fonksiyonu olarak değişik formüllerin kullanıldığı görülmektedir. Tablo 3.2' de değişik toplama fonksiyonlarına örnekler verilmektedir. Görüldüğü gibi, bazı durumlarda gelen girdilerin değeri dikkate alınırken, bazı durumlarda ise gelen girdilerin sayısı önemli olabilmektedir. Bir problem için en uygun toplama fonksiyonunu belirlemek için bulunmuş bir formül yoktur. Genellikle deneme yanılma yolu ile toplama fonksiyonu belirlenmektedir. Bir YSA bulunan proses elemanlarının tamamının aynı toplama fonksiyonuna sahip olmaları gerekmez. Her proses elemanı bağımsız olarak farklı bir toplama fonksiyonuna sahip olabilecekleri gibi hepsi aynı proses elemanına sahip olabilir. Hatta ağın bazı proses elemanları grup halinde aynı toplama fonksiyonuna sahip olabilir. Diğerleri ise farklı fonksiyonlar kullanabilirler. Bu tamamen tasarımcının kendi öngörüsüne dayanarak verdiği karara bağlıdır.

Tablo 3.2. Farklı toplama fonksiyonları [12]

Net Giriş	Açıklama
Çarpım $NetGirdi = \prod Gi.Ai$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak net girdi hesaplanır.
Maksimum $NetGirdi = Max(Gi.Ai),$ $i = 1....N$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Minimum $NetGirdi = Min(Gi.Ai),$ $i = 1....N$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Çoğunluk $NetGirdi = \sum sgn(Gi.Ai)$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
Kümlatif toplam $NetGirdi = Net(eski) + \sum (Gi.Ai)$	Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi bulunur.

4. Aktivasyon fonksiyonu: Bu fonksiyon, hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonu olarak da çıktıyı hesaplamak içinde değişik formüller kullanılmaktadır. Bazı modeller (mesela çok katmanlı algılayıcı) bu fonksiyonun türevinin alınabilir bir fonksiyon olmasını şart koşmaktadır. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonunda da ağırlık proses elemanlarının hepsinin aynı fonksiyonu kullanması gerekmez. Bazı elemanlar aynı fonksiyonu, diğerleri farklı fonksiyonu kullanması gerekmez. Bazı elemanlar aynı fonksiyonu diğerleri farklı fonksiyonları kullanabilirler. Bir problem için en uygun fonksiyonda yine

tasarımcının denemeleri sonucunda belirleyebileceği bir durumdur. Uygun fonksiyonu gösteren bir formül bulunmuş değildir. Günümüzde en yaygın olarak kullanılan çok katmanlı algılayıcı modelinde genel olarak aktivasyon fonksiyonu olarak Sigmoid fonksiyonu kullanılmaktadır. Bu fonksiyon, 3.2 nolu denklemdeki formül ile gösterilmektedir;

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

NET proses elemanlarına gelen net girdi değerini göstermektedir. Bu değer toplama fonksiyonu kullanılarak belirlenmektedir. Aktivasyon fonksiyonu olarak kullanılacak olan diğer fonksiyonlara örnek ise Tablo 3.3' de verilmiştir [12].

Tablo 3.3. Aktivasyon fonksiyonları [12]

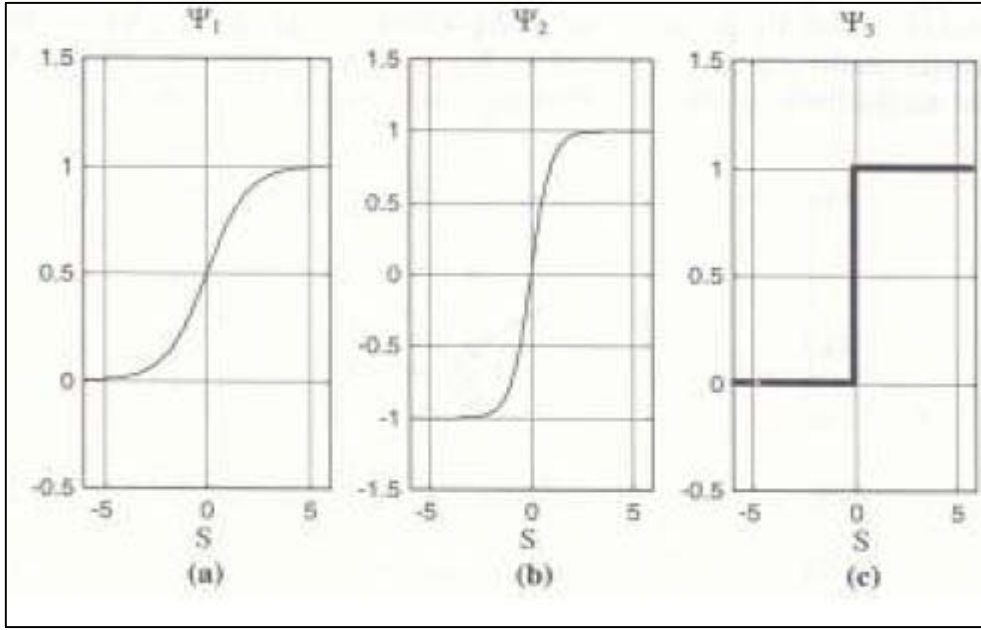
Aktivasyon Fonksiyonları	Açıklama
Lineer fonksiyon $F(NE T) = NE T$	Gelen girdiler olduğu gibi hücrenin çıktısı olarak kabul edilir.
Step fonksiyonu $F(NE T) = \begin{cases} 1 & \text{if } NE T > \text{eşik değ er} \\ 0 & \text{if } NE T \leq \text{eşik değ er} \end{cases}$	Gelen NET girdi değ erinin belirlenen bir eşik değ erinin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değ erlerini alır.
Sinüs fonksiyonu $F(NE T) = \text{Sin}(NE T)$	Öğrenilmesi düşünölen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır.
Eşik değ er fonksiyonu $F(NE T) = \begin{cases} 0 & \text{if } NE T \leq 0 \\ NE T & \text{if } 0 < NE T < 1 \\ 1 & \text{if } NE T \geq 1 \end{cases}$	Gelen bilgilerin 0 veya 1 den büyük veya küçük olmasına göre değ erler alır. 0 ve 1 arasında değ erler alabilir. Bunların dışında değ erler alamaz.
Hiperbolik tanjant fonksiyonu $F(NE T) = (e^{NE T} + e^{-NE T}) / (e^{NE T} - e^{-NE T})$	Gelen NET girdilerin tanjant fonksiyonundan geçirilmesi ile hesaplanır.

Nöron davranışını belirleyen önemli etmenlerden biri nöronun aktivasyon fonksiyonudur. Biyolojik nöronlarda, S ile gösterilen toplam belli bir değ eri aştığında nöronun kısa süreli bir darbe gönderdiği bilinmektedir. Bu davranışa benzer bir davranışı yapay nöronlarla da elde etmek için kullanılan aktivasyon fonksiyonlarından üçü Şekil 3.3.a,b,c' de gösterilmiş, matematiksel ifadeleri ise 3.3, 3.4 ve 3.5 nolu denklemlerdeki şekilde verilmiştir [1].

$$\psi_1(S) = \frac{1}{1+e^{-S}} \quad \text{Sigmoid tipi aktivasyon fonksiyonu} \quad (3.3)$$

$$\psi_2(S) = \frac{e^S - e^{-S}}{e^S + e^{-S}} = \tanh(S) \quad \text{Hiperbolik tanjant tipi aktivasyon fonk.} \quad (3.4)$$

$$\psi_3(S) = \begin{cases} 0 & S \leq 0 \\ 1 & S > 0 \end{cases} \quad \text{Sert geçişli tipte aktivasyon fonksiyonu} \quad (3.5)$$



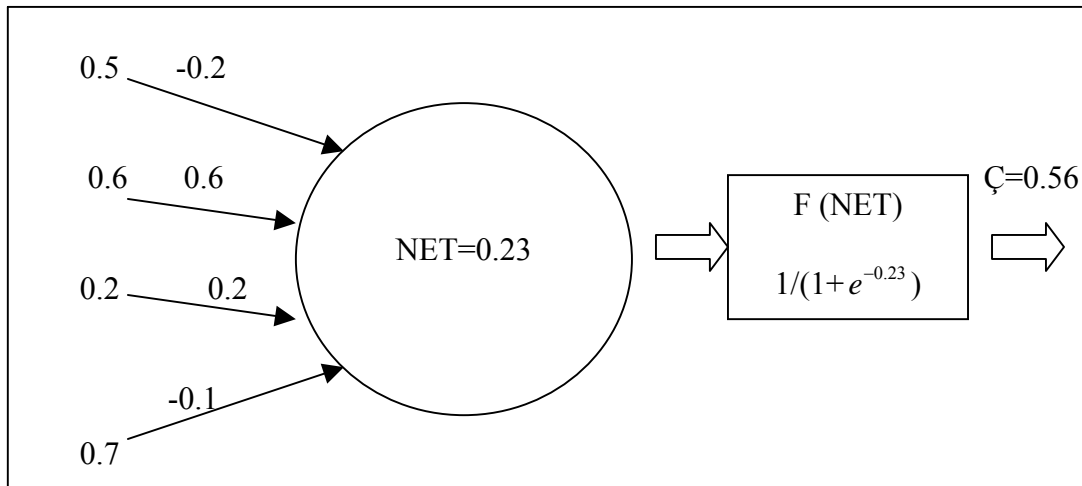
Şekil 3.3. Bazı aktivasyon fonksiyonları

- (a) Sigmoid tipi aktivasyon fonksiyonu
- (b) Hiperbolik tanjant tipi aktivasyon fonksiyonu
- (c) Sert geçişli aktivasyon fonksiyonu [1]

5. Hücrelerin çıktısı: Aktivasyon fonksiyonu tarafından belirlenen çıktı değeridir. Üretilen çıktı dış dünyaya veya başka bir hücreye gönderilir. Hücre kendi çıktısını kendisine girdi olarak da gönderebilir. Bir proses elemanının birden fazla çıktısı olmasına rağmen sadece bir çıktısı olmaktadır. Ağ şeklinde gösterildiğinde bir proses elemanının birden fazla çıktısı varmış gibi görülmektedir. Bu sadece gösterim amacıyla. Aslında bir proses elemanından çıkan tek bir çıktı değeri vardır. Aynı değer birden fazla proses elemanına girdi olarak gitmektedir [12-13].

3.5. Yapay Sinir Hücresinin Çalışma Prensibi

Bir yapay sinir hücresinin nasıl çalıştığını daha kolay anlamak için bir örnek vermek yararlı olur. Bir proses elemanına gelen bilgiler ve ağırlıklar aşağıdaki şekilde gösterildiği gibi varsayalım. Görüldüğü gibi proses elemanının 4 girdisi ve 4 ağırlık değeri vardır (Bkz. Şekil 3.4) [12].



Şekil 3.4. Bir yapay sinir ağı hücresinin çalışma örneği

3.1 nolu denklem kullanılarak, hücreye gelen NET bilgi, ağırlıklı toplam olarak şu şekilde hesaplanır:

$$NET = 0.5 * (-0.2) + 0.6 * 0.6 + 0.2 * 0.2 + 0.7 * (-0.1)$$

$$NET = -0.1 + 0.36 + 0.04 - 0.07$$

$$NET = 0.23$$

3.2 nolu denklem kullanılarak, hücrenin sigmoid fonksiyonuna göre çıktısı (Ç) hesaplanırsa;

$$Ç = 1/(1 + e^{-0.23})$$

$$Ç = 0.56$$

elde edilir.

Bir ağdaki bütün proses elemanlarının çıktılarının bu şekilde hesaplanması sonucu ağın girdilere karşılık çıktıları nasıl ürettiği görülür. Bu konuda ileride örnekler verilmektedir.

3.6. Yapay Sinir Ağlarında Bilgi ve Zeka

YSA'nda bilgi, ağın bağlantılarının sahip olduğu ağırlık değerlerinde saklanır. Diğer bir deyişle, bir yapay sinir ağının zekası ağın bağlantılarının sahip olduğu ağırlık değerlerinde saklıdır. Ağın sahip olduğu ağırlık değerleri ne kadar doğru ise ağın performansı da o kadar yüksek olur. Ağırlık değerleri bütün ağa yayılmış olduğundan ağın belleği de dağıtık bir hafızadır. O nedenle tek bir ağırlık değeri bir anlam ifade etmemektedir. Bilginin dağıtılmış olması bazı ağırlık değerlerinin o veya bu şekilde kaybolması sonucunda dahi ağın çalışmasını sürdürmesine neden olmaktadır. Daha önce belirtildiği gibi bu ise YSA'nın en önemli özelliklerinden ve güçlü yanlarından birisidir.

Yapay sinir ağları, hücrelerin birbirleri ile çeşitli şekillerde bağlanmalarından oluşur. Hücre çıkışları, ağırlıklar üzerinden diğer hücrelere ya da kendisine giriş olarak bağlanabilir ve bağlantılarda gecikme birimi de kullanılabilir. Hücrelerin bağlantı şekillerine, öğrenme kurallarına ve aktivasyon fonksiyonlarına göre çeşitli YSA yapıları geliştirilmiştir. Bu bölümde, çeşitli problemlerin çözümünde kullanılan ve kabul görmüş bazı YSA yapıları ayrıntısına girmeksizin genel özellikleri ile tanıtılacaktır [12].

3.7. Yapay Sinir Ağlarında En Çok Kullanılan Modeller

Bir YSA proses elemanlarının bağlanması sonucu oluşan topoloji, proses elemanlarının sahip oldukları toplama ve aktivasyon fonksiyonları, öğrenme stratejisi ve kullanılan öğrenme kuralı ağın modelini belirlemektedir. Günümüzde çok sayıda model geliştirilmiştir. Bunların en yaygın olarak kullanılanları ve pratik hayatta uygulamaları şunlardır [12]:

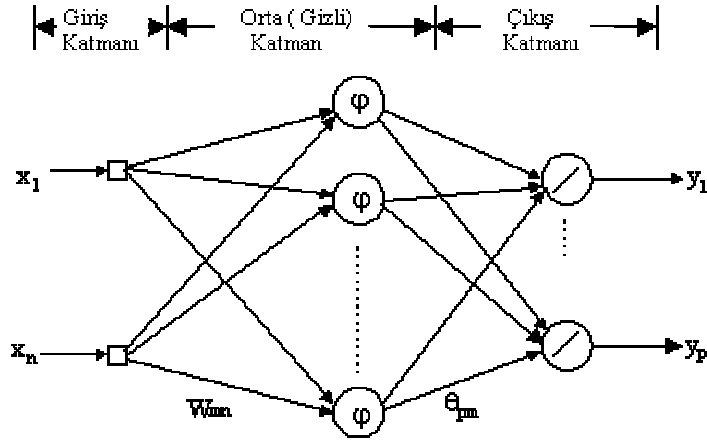
1. Algılayıcılar
2. Çok katmanlı algılayıcılar (hatayı geriye yayma modelleri)
3. Vektör kuantizasyon modelleri (LVQ)
4. Kendi kendini organize eden modeller (SOM)
5. Adaptif rezonans teorisi modelleri (ART)
6. Hopfield ağları
7. Neocognitron ağı
8. Probabilistik ağlar (PNN)
9. Elman ağı
10. Radyal temelli ağlar (RBN)

3.8. Bazı Yapay Sinir Ağları

Bu bölümde, çeşitli problemlerin çözümünde kullanılan ve kabul görmüş bazı YSA yapıları ayrıntısına girmeksizin genel özellikleri ile tanıtılacaktır [11].

3.8.1. İleri beslemeli yapay sinir ağları

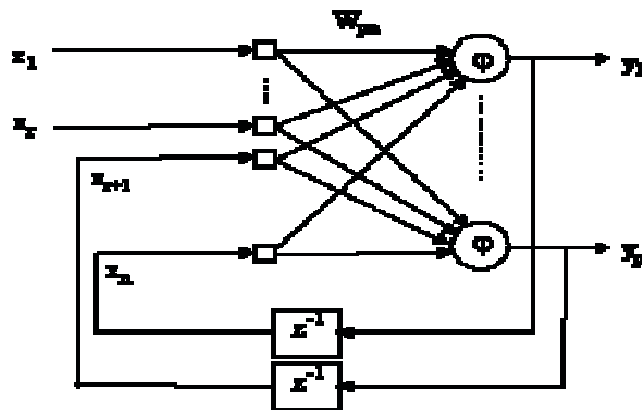
İleri beslemeli YSA' da, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan orta (gizli) katmandaki hücrelere iletir. Bilgi, orta ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı YSA' nın, orta katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen geriye yayılım öğrenme algoritması, bu tip ağların eğitiminde etkin olarak kullanılmakta ve bazen bu ağlara geriye yayılım ağları da denmektedir. Şekil 3.5' de giriş, orta ve çıkış katmanı olmak üzere 3 katmanlı ileri beslemeli YSA yapısı verilmiştir [11-14].



Şekil 3.5. İleri beslemeli 3 katmanlı yapay sinir ağı

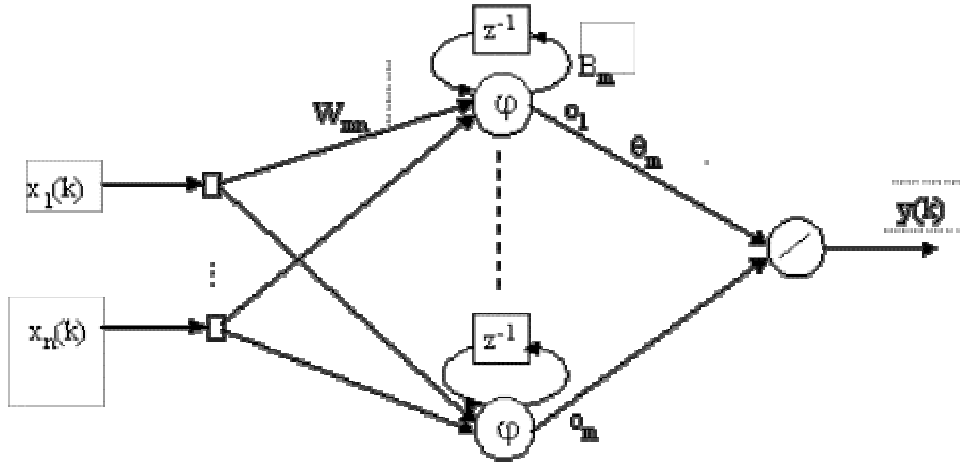
3.8.2. Geri beslemeli yapay sinir ağları

Geri beslemeli YSA' da, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapıyla geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir. Bu nedenle, bu bölümde bazı geri beslemeli YSA yapılarında örnekler verilecektir. Şekil 3.6' da iki katmanlı ve çıkışlarından giriş katmanına geri beslemeli bir YSA yapısı görülmektedir [11].



Şekil 3.6. Geri beslemeli iki katmanlı yapay sinir ağı

Geri beslemeli YSA; hücreler arası yada katmanlar arası geri besleme yapılış şekline göre farklı isimlerle söylenir. Genellikle derecesi bilinmeyen dinamik sistemlerin tanılanmasında kullanılan diğer bir YSA yapısı, gizli katman hücrelerinde öz geri beslemenin kullanıldığı ve yöresel geri-küresel ileri beslemeli (YGKİ) olarak söylenen Şekil 3.7' de verilen YSA' dır.



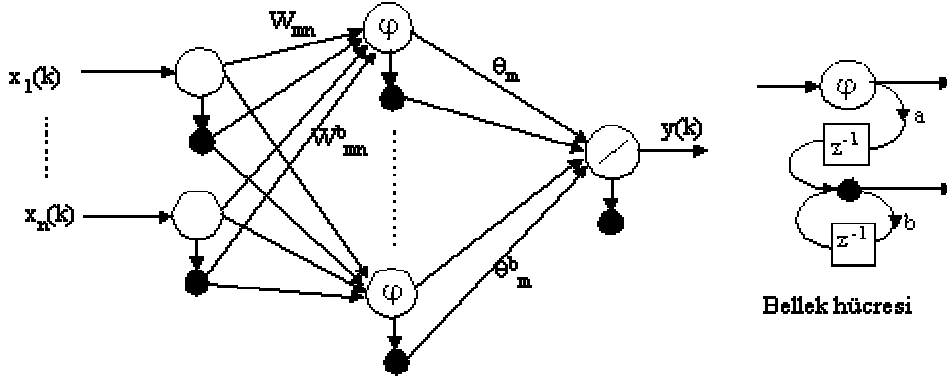
Şekil 3.7. Yöresel geri küresel ileri beslemeli yapay sinir ağı

YGKİ ağlar, ileri beslemeli YSA' nın eğitim algoritmalarında gerçekleştirilen küçük değişikliklerle eğitilebilmeleri nedeniyle ileri ve geri beslemeli YSA' nın ortak özelliklerini taşımaktadır.

3.8.3. Bellek hücreli yapay sinir ağ yapıları

Doğrusal olmayan sistemlerin tanılanması ve denetiminde, katmanlı YSA yapıları etkin olarak kullanılmaktadır. YSA ile sistem tanılamada, doğru model yapısının seçilebilmesi ve model girişlerinin belirlenebilmesi için sistemin giriş ve çıkışının gecikme derecelerinin bilinmesi gerekir. Sistemin derecesinin doğru belirlenememesi, modelde temsil edilemeyen dinamikler nedeniyle kararlı ve değişen dinamik şartlarda doğru bir model elde edilmesini etkiler. Bu nedenle, geri beslemeli YSA yapıları kullanılarak sistemin derecesine ihtiyaç duymayan tanı modelleri geliştirilmiştir. Şekil 3.8' de bellek hücreli yapay sinir ağları (BHYS) olarak

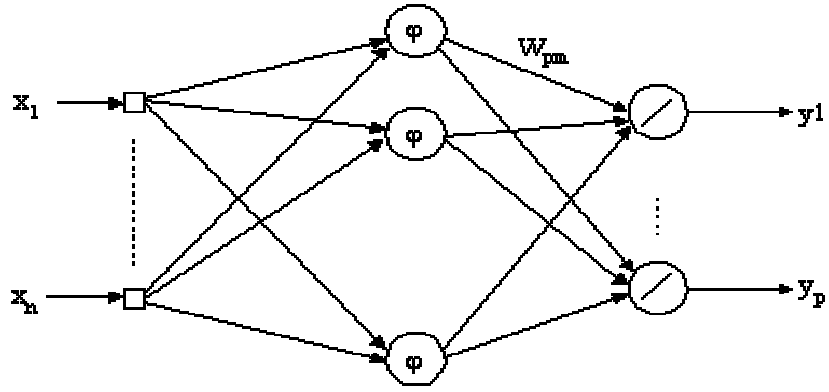
söylenen ve ağdaki her bir hücre için bir bellek hücresinin kullanıldığı katmanlı-geri beslemeli bir ağ yapısı verilmiştir [11].



Şekil 3.8. Bellek hücreli yapay sinir ağı ve bellekteki bir hücrenin yapısı

3.8.4. Radyal tabanlı fonksiyon ağları

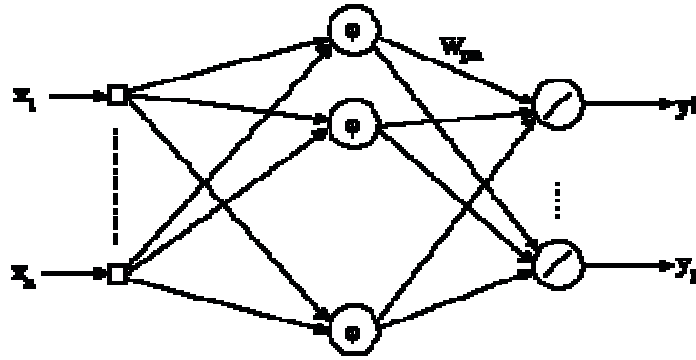
Katmanlı YSA' nın tasarımında eğitici geriye yayılım öğrenme algoritması bir en iyileme uygulamasıdır. Radyal tabanlı fonksiyon ağı tasarımı ise çok boyutlu uzayda eğri uydurma yaklaşımıdır ve bu nedenle RTFA' nın eğitimi, çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemine dönüşür. RTFA' nın genellemesi ise test verilerini interpolate etmek amacıyla, eğitim sırasında bulunan çok boyutlu yüzeyin kullanılmasına eşdeğerdir. Radyal tabanlı fonksiyonlar, sayısal analizde çok değişkenli interpolasyon problemlerinin çözümünde kullanılmış ve YSA' nın gelişmesi ile birlikte bu fonksiyonlardan YSA tasarımında yararlanılmıştır. RTFA, ileri beslemeli YSA yapılarına benzer şekilde giriş, orta ve çıkış katmanından oluşur ancak, giriş katmanından orta katmana dönüşüm, radyal tabanlı aktivasyon fonksiyonları ile doğrusal olmayan sabit bir dönüşümdür. Orta katmandan çıkış katmanına ise uyarlamalı ve doğrusal bir dönüşüm gerçekleştirilir (Bkz. Şekil 3.9) [11].



Şekil 3.9. Radyal tabanlı fonksiyon ağı

3.8.5. Fonksiyonel link ağları

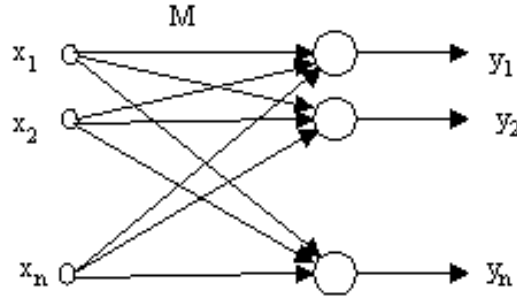
Katmanlı YSA, orta katmandaki doğrusal olmayan aktivasyon fonksiyonları nedeniyle doğrusal olmayan öğrenme algoritmaları ile eğitilmelidir. Bu durum, öğrenme hızını yavaşlatır ve fonksiyon yaklaşımında yöresel en aza ulaşılabilir. Bu sorunlar, ağ girişlerini önce doğrusal olmayan bir dönüşümle genişlettikten sonra doğrusal çıkış katmanlı ağ yapıları ile giderilebilir. Şekil 3.10' da verilen ve fonksiyonel link ağları (FLA) olarak bilinen ağlar iyi bir fonksiyon yaklaşırma performansına sahiptir. Bu yönüyle fonksiyonel link ağları, merkezleri ve genişliği sabit tutulan RTFA' na benzer ancak, FLA' da orta katmanın görevi ve aktivasyon fonksiyonları farklıdır [11].



Şekil 3.10. Fonksiyonel link ağları

3.8.6. Çağrışimli bellek ağları

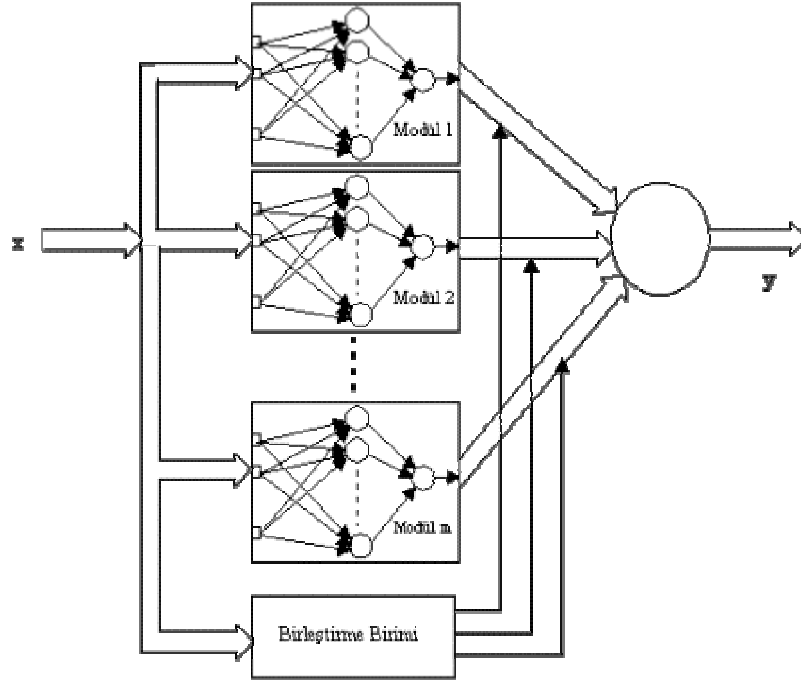
Çağrışimli sistemler, belirli giriş vektörlerini belirli çıkış vektörlerine dönüştüren yada ilişkilendiren sistemler olarak tanımlanır. Dolayısıyla çağrışimli bellek ağları, eğitim sürecinde ağa verilen eğitim örneklerini ağırlıkları aracılığı ile saklar yada ezberler ve hatırlama yada genelleme sürecinde ise saklanmış örneklerin gürültülü versiyonları da ağa verilmiş olsa doğru örnekleri verebilir. Bu yönü ile ÇBA' ları kodlayıcılar ve kod çözücülere benzer işlevleri yerine getirirler ve beynin yapısal karakteristikleri yerine işlevsel özelliklerini benzeştiren ağ yapısı olarak söylenebilir. ÇBA' ları genellikle örüntü tanıma ve eldeki eksik verilerden doğru verileri ortaya çıkarma gibi uygulamalarda yaygın olarak kullanılır (Bkz. Şekil 3.11) [11].



Şekil 3.11. Çağrışimli bellek ağları

3.8.7. Modül yapay sinir ağları

Modül YSA' ları, çok sayıda YSA yapısının birleşiminden oluşur aşağıdaki gibi tanımlanır. Eğer, bir ağın yapması gereken işlemler birbirleriyle de haberleşmeksizin iki yada daha fazla modüle (alt yapıya) ayrılabiliriyorsa bu ağlar modül YSA' ları olarak söylenir (Bkz. Şekil 3.12) [11].



Şekil 3.12. Modül yapay sinir ağı

Modüllerin çıkışları, modüllerden bilgi geri beslenmemek üzere bir birleştirme birimi ile birleştirilir ve birleştirme birimi;

1. Modül YSA çıkışını elde etmek için modül çıkışlarının nasıl birleştirilmesi gerektiğini,
2. Hangi eğitim örneklerini hangi modülün öğreneceğini kararlaştırmalıdır.

Bu nedenle modül YSA' nın, böl ve yönet esasına göre çalıştığı söylenebilir ve böylece karmaşık problemler daha basit alt bileşenlerine ayrılarak çözülebilir ve sonuçta çözümler birleştirilebilir. Örneğin, süreksiz fonksiyonların tek bir YSA ile yaklaştırılması, özellikle süreksizlik noktalarında arzu edilen sonucu vermez. Böyle bir fonksiyonun süreksizlik noktaları arasındaki her bir bölgesi ayrı bir YSA modülü ile yaklaştırılarak sonuç fonksiyon, modül çıkışlarının birleşimi olarak alınabilir. Bu nedenle de, modül YSA' larının eğitiminde genellikle hem eğitici hem de eğitici olmayan öğrenme algoritmalarını birlikte kullanmak gerekir.

3.9. Yaygın Olarak Kullanılan Öğrenme Algoritmaları

1. Geri yayılım
2. Delta bar delta
3. Genişletilmiş delta bar delta [11-13]

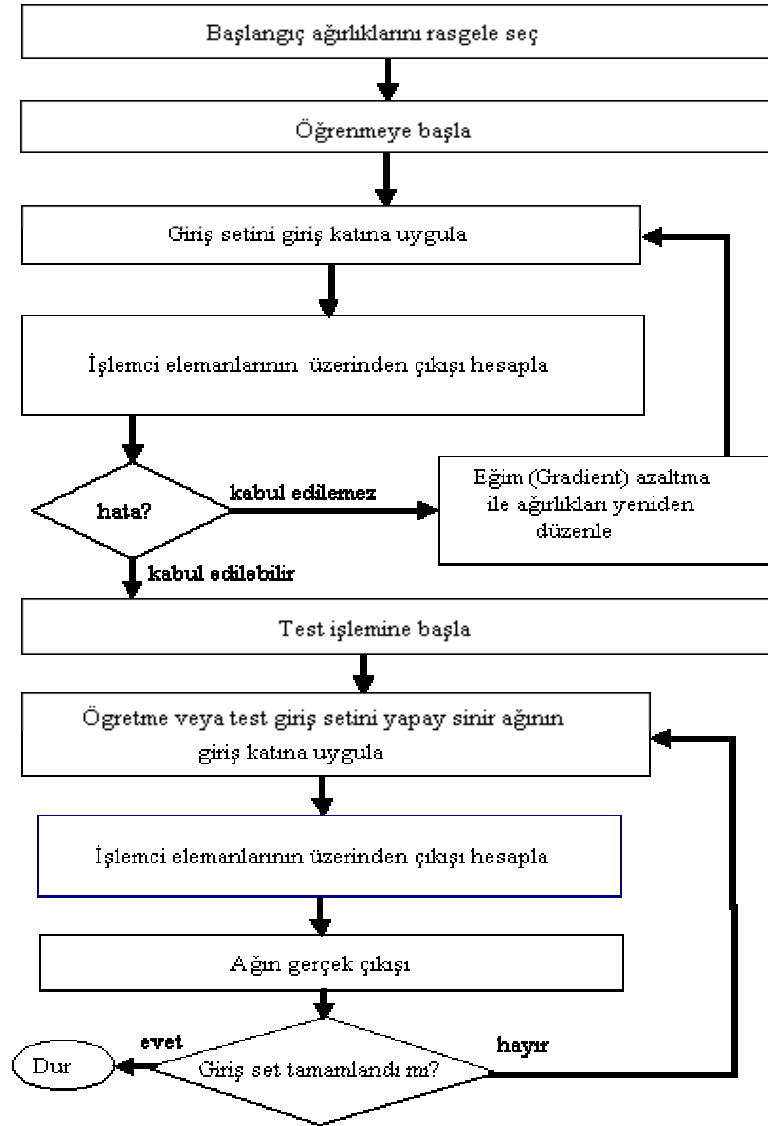
3.9.1. Geri yayılım

Geri yayılım ağı, 1970' lerin başında geliştirilmiş, en popüler, en etkili ve karmaşık, tanımlanamamış problemlere doğrusal olmayan çözümler getirebilen bir ağ çeşididir [11].

Bir çok uygulamalarda kullanılmış en yaygın öğrenme algoritmasıdır. Anlaşılması kolay ve matematiksel olarak ispatlanabilir olmasından dolayı en çok tercih edilen öğretim algoritmasıdır. Bu algoritma, hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır.

Tipik çok katlı geri yayılım ağı, daima bir giriş tabakası, bir çıkış tabakası ve en az bir gizli tabakaya sahiptir. Gizli tabakaların sayısında teorik olarak bir sınırlama yoktur. Fakat genel olarak bir veya iki tane bulunur. Bu algoritmanın akış şeması Şekil 3.13' de verilmiştir.

Geri yayılım ağı için, öğrenme kurallarının bir çok çeşidi vardır. Farklı hata işlevleri, aktarım işlevleri ve hatta aktarım işlevinin türevinin değiştirme yöntemi de kullanılabilir. Öğrenme sürecinde "Momentum" kullanılarak daha hızlı öğrenme sağlanmaya çalışılır. Burada hata işlevi ya da delta bağıl değer eşitliği, bir önceki delta bağıl değerinin bir kısmının mevcut delta bağıl değeri boyunca desteklenmesini sağlayabilmek için değiştirilir. Bu işlem bir nevi alçak geçiren süzgeç işlevi görür [11].



Şekil 3.13. Geri yayılım algoritması [13]

3.9.2. Delta bar delta

Delta bar delta ağı (DBD), bir geri yayılım ağı olarak aynı mimariden yararlanmaktadır. Delta bar delta ağının farklılığı, yalnızca kendine has algoritmik öğrenme metodudur. Delta bar delta, Robert Jacobs tarafından, standart ileri beslemeli, geri yayılım ağlarının öğrenme oranını iyileştirmek amacıyla geliştirilmiştir [11].

Yukarıda ana hatları ile verildiği üzere, geri yayılım süreci, çok dik bir yokuş yaklaşımı üzerine kurulmuştur. Bu yokuş, her bir düğüme bağlantı bağıl değerlerinin değiştirildiği süreç sırasında ağın tahmin hatasını en aza indirir. Standart öğrenme oranları bir katman temeline uygulanır ve momentum terimi genellikle global olarak atanır. Bazı geri yayılım yaklaşımları, öğrenme oranlarının çok büyük miktarlardaki eğitim dizileri ağıdan geçerken, derece derece azalmalarına imkan sağlar. Bu yöntem bir çok uygulamayı çözmekte başarılıdır ancak, sürecin yakınsama oranı bazı pratik problemlerde kullanılmak için daha çok yavaştır.

Delta bar delta, her bir bağıl değer kendine has kendini uyarlayabilen katsayıya sahip olduğu bir öğrenme metodu kullanmaktadır. Ayrıca, geri yayılım mimarisinin momentum faktörünü kullanmaz. İleri beslemeli çağrışım gibi sebeplerin geriye kalan çalışmaları, normal geri yayılım mimarisine benzerdir. Delta bar delta, bir eğitim yapay ağlarına bir sezgisel yaklaşımdır. Bunun anlamı, eski hata değerlerini tahmin etmek için kullanılabileceği demektir. Muhtemel hataları bilmek, sistemin bağıl değerlerini ayarlarken zeka adımları atmasını sağlar. Ancak, bu sürecin bu ampirik kanıt içerisinde karmaşık olması, her bir bağıl değer, hatanın tamamı üzerinde çok farklı etkilere sahip olduğunu ortaya koymaktadır. Jacobs bundan dolayı, geri yayılım öğrenme kurallarının hatanın tamamı üzerindeki etkilerin bu çeşitliliğini göz önüne alması gerektiğini belirten ortak duyu görüşünü ortaya atmıştır. Diğer bir deyişle, bir ağı her bir bağlantı bağıl değerinin kendi öğrenme oranı olmalıdır. Dahası, bu öğrenme oranlarının zaman içinde değişmesini sağlamalıdır. Her bir bağlantıya bir öğrenme oranı atayarak ve bu öğrenme oranının zaman içinde sürekli değişmesine izin vererek, yakınsamaya harcanan zamanı azaltmak için daha fazla özgürlük derecesi girilebilir.

Bu algoritmaya doğrudan uygulanan kurallar ileri doğru düzdür ve uygulaması kolaydır. Her bir bağlantı bağıl değerinin kendi öğrenme oranı vardır. Bu öğrenme oranları, standart geri yayılım ile birlikte bulunan mevcut hataya bağıl olarak değişir. Bağlantı bağıl değeri değiştiğinde, eğer bölgesel hata çeşitli ardıl zaman adımları için aynı sinyale sahipse, o bağlantının öğrenme oranı doğrusal olarak artırılır. Doğrusal olarak artırma, öğrenme oranlarını çok büyük ve çok hızlı hale gelmesini önler. Bölgesel hata sinyalleri sık sık değiştirdiğinde, öğrenme oranı geometrik olarak

azaltılır. Geometrik olarak azaltma, bağlantı öğrenme oranlarının her zaman pozitif olmasını sağlar. Dahası, bu oranlar, hatadaki değişikliğin büyük olduğu bölgelerde daha hızlı bir şekilde azaltılabilir.

Bir ağda, her bir bağlantı bağıl değeri için farklı oranlar olmasına izin vererek, bir çok dik iniş aramasını yapmaya (negatif iniş yönde) gerek kalmaz. Bunun yerine, bağlantı bağıl değerleri, bağıl değerlerin kendisi bakımından hatanın kısmi türevi bazında güncelleştirilir. Bu, ayrıca mevcut nokta bağıl değerinin yakınındaki 'hata yüzeyinin bükülme' tahmine bağlıdır. Ayrıca, bağıl değer değişiklikleri, bölgesellik sınırlamasını karşılar. Yani, bu değişiklikler yalnızca bağlı oldukları işleme elemanlarından gelecek bilgiye ihtiyaç duyarlar [11].

3.9.3. Genişletilmiş delta bar delta

Minai ve Williams, genişletilmiş delta bar delta (extended DBD) algoritmasını Jacobs' un çalışmasının doğal bir uzantısı olarak geliştirmişlerdir. Burada, delta bar delta öğrenme oranı artışına bir üst eksilmesi uygulayıp, momentum unsurunu yeniden buna ekleyip, öğrenme oranı ve momentum katsayısına bir ek koyarak geliştirmişlerdir. Geri yayılım ile ilgili bölümde tartışıldığı üzere, momentum öğrenme oranını düzeltmek için kullanılan bir faktördür. Geçmiş bağıl değer değişikliğine oranlı olan standart bağıl değere eklenen bir terimdir. Bu bağlamda, iyi genel eğilimler güçlendirilmiş ve dalgalanmalar azaltılmıştır [11].

Her bir bağıl değer için öğrenme oranı ve momentum oranı, bunların artış ve azalmalarını kontrol eden ayrı sabitlere sahiptir. Bir kez daha, mevcut hata sinyali, bir artışın mı yoksa bir azalmanın mı uygun olduğunu belirtmek için kullanılır. Azalma ayarlanması, delta bar delta ile aynı şekilde sahiptir. Ancak, öğrenme oranı ve momentum oranı artışları, tartılmış eğim unsurların büyüklüğünün üs rakamlara dayanarak düşen işlevleri olmak üzere değiştirilirler. Böylece, büyük artışlar, büyük eğilmelere değil de, küçük yokuş ya da eğilme alanlarına uygulanır. Bu, delta bar delta zıplama problemine kısmi bir çözüm oluşturur.

Bağıl değerlerdeki büyük zıplamaları ve titreşimleri önlemek amacıyla atılan bir sonraki adım ise, azami sınırların bireysel bağlantı öğrenme oranları ve momentum oranları üzerine yerleştirilir ve düzeltme özelliği olan bir hafıza algoritma içine kurulur. Bunu kullanırken, eğitim verisinin her bir devir sunumundan sonra, biriken hata değerlendirilir. Eğer hata önceki enaz hatadan daha az ise, bağıl değerler hafızaya mevcut en iyiler olarak kaydedilir. Bir tolerans değişkeni düzeltme fazını kontrol eder. Özellikle, eğer mevcut hata önceki enaz hatadan fazla ise, tolerans değişkeni ile değiştirilerek, bütün bağlantı bağıl değerleri hafızadaki kaydedilmiş en iyiler dizisine tahmini olarak dönerler. Dahası, öğrenme oranları ve momentum oranları, düzeltme sürecini başlatmak için azaltılırlar.

3.10. Yapay Sinir Ağlarında Öğrenme

Sinir ağlarının en önemli özelliği, öğrenme yeteneğidir. Bir sinir ağında öğrenmenin anlamı, ağın belirli bir probleme ait doğru çıktıları üretmesini sağlayacak optimum ağırlık değerlerinin bulunmasıdır. Bilgi, ağ boyunca bağlantılarda ağırlıklar şeklinde dağıtıldığı için tek bir bağlantı herhangi bir anlamlı bilgiyi ifade etmez. Daha doğrusu, anlamlı bir bilgi oluşturmak için işlem elemanları olan bir grup bağlantıyı tasarlamak gerekmektedir. Problemin çözümü için ağın, bağlantılarına ait doğru ağırlık değerlerine sahip olması gerekmektedir. Bu öğrenme veya eğitime olarak adlandırılan bir işlem vasıtasıyla gerçekleştirilir. Öğrenme, ağırlık değerlerinin nasıl değiştirilmesi gerektiğini ifade eden bir öğrenme kuralına dayanır. Geliştirilen birçok öğrenme kuralı vardır (Wasserman 1989). Bir öğrenme kuralının temel ilkesi, benimsenen öğrenme stratejisi ile tanımlanır [11-12].

YSA' da proses elemanlarının bağlantılarının ağırlık değerlerinin belirlenmesi işlemine "ağın eğitilmesi" denir. Başlangıçta bu ağırlık değerleri rasgele olarak atanır. YSA kendilerine örnek gösterildikçe bu ağırlık değerlerini değiştirirler. Amaç ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır. Örnekler ağa defalarca gösterilecek en doğru ağırlık değerleri bulunmaya çalışılır. Ağın doğru ağırlık değerlerine ulaşması örneklerin temsil ettiği olay hakkında genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleştirme özelliğine

kavuşması işlemine ağıın öğrenmesi denir. Ağırlıkların değerlerinin değışmesi belirli kurallara göre yürütölmektedir. Bu kurallara öğrenme kuralları denir. Daha önce belirtildiğı gibi, kullanılan öğrenme stratejisine göre değışik öğrenme kuralları geliştirilmiştir [11-12].

YSA' da öğrenme olayının iki aşaması vardır. Birinci aşamada ağı gösterilen örnek için ağıın üreteceğı çıktı belirlenir. Bu çıktı değıerinin doğruluk derecesine göre ikinci aşamada ağıın bağlantılarının sahip olduğı ağırlıklar değıştirilir. Ağıın çıktısının belirlenmesi ve ağırlıkların değıştirilmesi öğrenme kuralına bağılı olarak farklı şekillerde olmaktadır [12].

Ağıın eğıitimi tamamlandıktan sonra öğrenip öğrenmediğini ölçmek için yapılan denemelere ise ağıın test edilmesi denmektedir. Test etmek için ağıın öğrenme sırasında görmediğı örnekler kullanılır. Test etme sırasında ağıın ağırlık değıerleri değıştirilemez. Test örnekleri ağı gösterilir. Ağı eğıitim sırasında belirlenen bağılantı ağırlıklarını kullanarak görmediğı bu örnekleri için çıktılar üretir. Elde edilen çıktılarının doğruluk değıerleri ağıın öğrenmesi hakkında bilgiler verir. Sonuçlar ne kadar iyi olursa eğıitim performansı da o kadar iyi demektir. Eğıitimde kullanılan örnek setine eğıitim seti, test için kullanılan sete ise test seti test seti adı verilmektedir. YSA bu şekilde bilinen örneklerden belirli bilgileri çıkartarak bilinmeyen örnekler hakkında yorumlar yapabilme yeteneğıine adaptif öğrenme denir.

3.11. Makine Öğrenmesi ve Öğrenme Türleri

Makine öğrenmesini anlayabilmek için öncelikle öğrenme kavramının tanımlanması gerekmektedir. Öğrenme kavramı değışik şekillerde tanımlanmakla birlikte genellikle Simon tarafından önerilen tanım etrafında değışiklikler yapılmaktadır. Simon öğrenmeyi, "zaman içinde yeni bilgiler keşfedilmesi yoluyla davranışların iyileştirilmesi süreci" olarak tanımlamaktadır. Makine öğrenmesi ise bu öğrenme işinin bilgisayar tarafından gerçekleştirilmesinin sağlanmasıdır. Burada zaman içerisinde iyileşme kavramına dikkat çekmek gerekmektedir. Bilgisayarın da insan gibi zaman içerisinde tecrübe kazanması istenmektedir. Diđer bir değışle makine

öğrenmesi “bilgisayarın bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek gelecekte oluşacak benzeri olaylar hakkında kararlar verebilmesi ve problemlere çözüm üretebilmesidir.” denilebilir. Bilgisayarın öğrenebilmesi ve tecrübe sahibi olabilmesi, bilgisayarın ilgili olay hakkında bilgiler ile donatılmasına bağlıdır. YSA yolu ile öğrenen bilgisayarların bilgiler ile donatılması, örnekler yolu ile sağlanmaktadır. Öğrenecek olan bilgisayar sistemleri önce bir örnek almakta ve bu örnekten bazı bilgileri öğrenmektedir. Daha sonra ikinci örneğe bakarak biraz daha bilgi edinmektedir. Bu işlemi öğrenilecek olay ile ilgili bütün örnekleri defalarca gözden geçirerek tekrarlamak sonucunda olay ile ilgili genellemeler yapılmaktadır. Bu olaya tecrübelerde öğrenmenin bir yolu olarak bakmak mümkündür. Yapay sinir ağlarının dışında da değişik öğrenme şekilleri vardır. Bu konuda değişik sınıflandırmalar yapılmıştır. Bunların bazılarını şöyle sıralamak mümkündür [12]:

1. Alışkanlık yolu ile öğrenme
2. Görerek öğrenme
3. Talimatlardan öğrenme
4. Örneklerden öğrenme
5. Analoji yolu ile öğrenme
6. Açıklamalardan öğrenme
7. Deney yolu ile öğrenme
8. Keşfetmek yolu ile öğrenme

Geliştirilen herhangi bir makine öğrenme sistemi yukarıdaki öğrenme türlerinden birini veya birkaçını birlikte kullanabilir. Genel olarak bakıldığında makine öğrenmesi çalışmalarının iki amaçla gerçekleştirildiği görülmektedir. Bunlar [12];

- a. Farklı çağrışım: Burada bir olay ister gözlemleyerek olsun, ister talimatlar yolu ile olsun, ister örnekler yolu ile olsun değişik açılarda incelenerek olayın genel yönleri ortaya çıkartılmakta ve daha sonra oluşan benzeri bir durumda bu genel yönler kullanılarak problemler çözülmektedir. Örnekleri kullanmak durumunda, öğrenilecek olay hakkında elde bulunan örnekler kullanılarak genellemeler yapılmakta ve sınıflandırma problemlerinde olduğu gibi örneklerin ortak noktası öğrenilmektedir. Bu tür öğrenmede, öğrenme yapacak sistemin girdisi ve çıktısı

birbirinden farklıdır. Makine öğrenmesi sisteminin görevi, örneklerin çağrıştırdığı çıktıları belirlemektir. Sınıflandırma ve tahmin etme problemlerinde bu tür öğrenme yaygın olarak kullanılmaktadır. Günlük döviz kurlarındaki değişime bakarak bir sonraki günün döviz kurunu tahmin etmek güzel bir örnek olabilir. Bir prosten gözlem yolu ile elde edilen 40-50 adet ölçüm ile prosesin normal davranış sergileyip sergilemediğini belirlemek gibi bir olayda bu tür öğrenmeye örnek olarak verilebilir.

b. Oto çağrışım: Bu durumda ise bir olay öğrenilerek daha sonra eldeki var bilgilerle o olayın kendisi karakterize edilmektedir. Eldeki bilgilerin eksik olması durumunda makine öğrenmesi sayesinde sistem eksik bilgileri tamamlayabilmektedir. Bu durumda, öğrenme yapacak sisteme (mesela yapay sinir ağlarına) bir örnek girdi olarak verilir ve aynı örnek çıktı olarak istenir. Bu, bir insanın resmini öğretmek gibi bir olaydır. Burada amaç, ağa örnekleri göstererek ilgili olay hakkında eksik bilgilerin olması durumunda ilgili olayın bilgilerinin tamamını oluşturmaktır. Bir insanın resmini öğrendikten sonra yırtık bir resmin sahibinin belirlenmesi bu tür öğrenmeye güzel bir örnektir [12].

3.11.1. Öğrenme yöntemleri

Öğrenme üzerine yapılan çalışmaları, makine öğrenmesi haline dönüştürebilmek için çeşitli paradigmlar ve yaklaşımlar geliştirilmiştir. Bunlar arasında şunları saymak mümkündür [12].

1. Sembol işleme yöntemi : Bu sistemler yapay zeka teknolojisine dayanan öğrenme sistemleridir. Belirli formlarda bilgisayara sunulan bilgilere dayanarak muhakeme yolu ile öğrenme gerçekleştirilir.
2. Yapay sinir ağları : Bu sistemler örneklerden genelleme yaparak öğrenirler.
3. İstatistiksel örüntü tanıma : Bu tür öğrenmede bir veri setinin istatistiksel özellikleri ve dağılımı incelenerek veri hakkında genellemeler yapmak söz konusudur.

4. Genel algoritmalar ve evrimsel programlama : Bu paradigma problemlerin çözümü için başlangıç çözümleri rasgele atamak ve bu çözümlerden yeni çözümler üretmek ve daha sonra yeni çözümlerden daha iyi ve yeni çözümler üretmek amaçlanmaktadır. Bu işlemi tekrar ederek sürekli daha iyi çözümler üretilmektedir. Bu iyileştirme en son halini alıncaya kadar devam etmektedir.

5. Vaka tabanlı öğrenme : Bu teknoloji örneklerden de öte vakalara bakarak öğrenme esasına dayanmaktadır. Her vaka başlı başına bir olayı göstermekte ve birden fazla örnekten oluşmaktadır. Bir olay söz konusu olunca benzeri vakalara bakarak o vakalar için verilen kararların benzeri kararlar verilmektedir [12].

3.11.2. Örneklerden öğrenme

YSA gibi öğrenme yöntemleri örneklerden öğrenmeye dayanmaktadır. Örneklerden öğrenmenin temel felsefesi bir olay hakkındaki gerçekleşmiş örnekleri kullanarak olayın girdi ve çıktıları arasındaki ilişkileri öğrenmek ve bu ilişkilere göre daha sonra oluşacak olan yeni örneklerin çıktılarını belirlemektir. Burada bir olay ile ilgili örneklerin girdi ve çıktıları arasındaki ilişkinin olayın genelini temsil edecek bilgiler içerdiği kabul edilmektedir. Değişik örneklerin olayı değişik açılardan temsil ettiği varsayılmaktadır. Farklı örnekler kullanılarak böylece olay değişik açılardan öğrenilmektedir. Burada bilgisayara sadece örnekler gösterilmektedir. Bunlardan başka herhangi bir ön bilgi verilmemektedir. Öğrenmeyi gerçekleştirecek sistem aradaki ilişkiyi kendi paradigmasını kullanarak keşfetmektedir. Sistem örneklerden öğrendiğine göre, örnek kavramının irdelenmesi gerekir. Örnek nasıl oluşturulmaktadır? Örnek nasıl formülize edilmektedir? Bilgisayar örneği nasıl anlayacaktır? Başarılı bir öğrenme sistemi geliştirmek için bu ve benzeri soruların cevaplarının bilinmesi gerekmektedir [12].

3.11.3. Öğrenme stratejileri

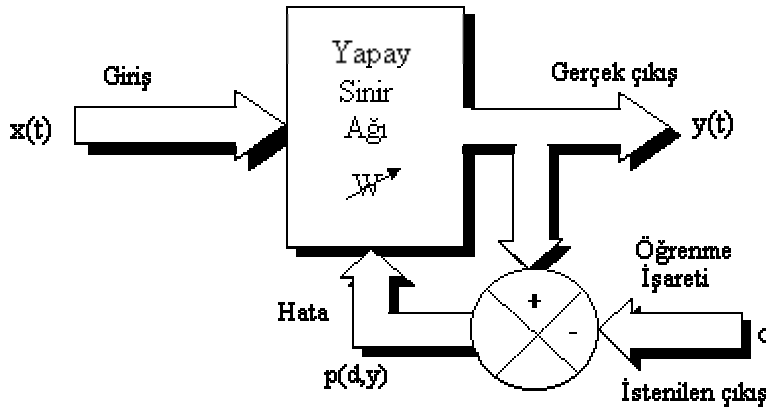
Yapay sinir ağları gibi örneklerden öğrenen sistemlerde değişik öğrenme stratejileri kullanılmaktadır. Öğrenmeyi gerçekleştirecek olan sistem ve kullanılan öğrenme

algoritması bu stratejilere bağılı olarak deęişmektedir. Genel olarak 3 öğrenme stratejisinin uygulandıęı görülmektedir. Bunlar [11-12]:

3.11.3.1. Öğretmenli öğrenme

Bu tür stratejide öğrenen sistemin olayı öğrenebilmesine bir öğretmen yardımcı olmaktadır. Öğretmen sisteme öğrenilmesi istenen olay ile ilgili örnekleri girdi/çıkı seti olarak verir. Yani, her örnek için hem girdiler hem de o girdiler karşılığında oluşturulması gereken çıktılar sisteme gösterilirler. Sistemin görevi girdileri öğretmenin belirledięi çıktıları haritalamaktır. Bu sayede olayın girdileri ile çıktıları arasındaki ilişkiler öğrenilmektedir (Bkz. Şekil 3.14) [12].

Öğretmenli öğrenmede, yapay sinir ağı kullanılmadan önce eğitilmelidir. Eğitime işlemi, sinir ağına giriş ve çıkış bilgileri sunmaktan oluşur. Bu bilgiler genellikle eğitime kümesi olarak tanımlanır. Yani, her bir giriş kümesi için uygun çıkış kümesi ağı sağlanmalıdır [11].

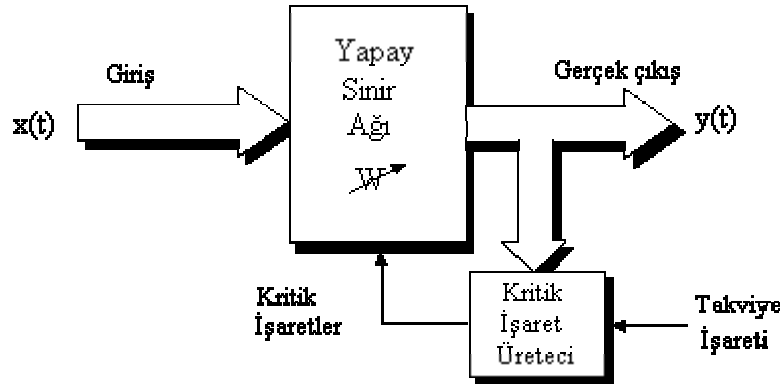


Şekil 3.14. Öğretmenli öğrenme yapısı [10]

3.11.3.2. Destekleyici öğrenme

Bu tür stratejide de öğrenen sisteme bir öğretmen yardımcı olur. Fakat bu öğretmen her girdi seti için olması gereken (üretilmesi gereken) çıktı setini sisteme göstermek yerine sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve

üretileen ıktının dođru veya yanlış olduđunu gösteren bir sinyal üretir. Sistem, öđretmenden gelen bu sinyali dikkate alarak öđrenme sürecini devam ettirir (Bkz. Şekil 3.15) [12].

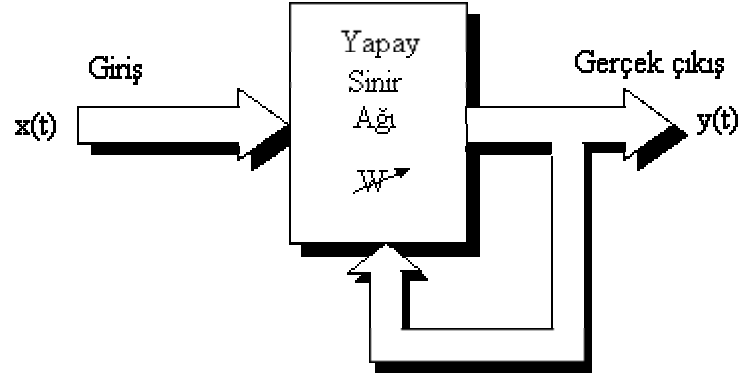


Şekil 3.15. Destekleyici öđrenme yapısı [10]

3.11.3.3. Öđretmensiz öđrenme

Bu tür stratejide sistemin öđrenmesine yardımcı olan herhangi bir öđretmen yoktur. Sisteme sadece girdi deđerleri gösterilir. Örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendisine öđrenmesi beklenir. Bu, daha çok sınıflandırma problemleri için kullanılan bir stratejidir. Yalnız sistemin öđrenmesi bittikten sonra çıktıların ne anlama geldiđini gösteren etiketlendirmenin kullanıcı tarafından yapılması gerekmektedir (Bkz. Şekil 3.16) [12].

Öđretmensiz öđrenmede ađ istenen dıř verilerle deđil girilen bilgilerle alıřır. Bu tür öđrenmede gizli sınırlar dıřarıdan yardım almaksızın kendilerini örgütlemek için bir yol bulmalıdırlar. Bu yaklaşımda, verilen giriş vektörleri için önceden bilinebilen performansını ölçebilecek ađ için hiçbir çıkıř örneđi sađlanmaz, yani ađ yaparak öđrenmektedir [11].



Şekil 3.16. Öğretmensiz öğrenme yapısı [10]

3.11.3.4. Karma stratejiler

Yukarıdaki 3 stratejiden birkaçını birlikte kullanarak öğrenme gerçekleştiren ağlarda vardır. Burada kısmen öğretmenli kısmen ise öğretmensiz olarak öğrenme yapan ağlar kastedilmektedir [12].

3.11.4. Öğrenme kuralları

YSA gibi öğrenen sistemlerde öğrenme, yukarıda anlatılan stratejilerden hangisi uygulanırsa uygulansın bazı kurallara göre gerçekleştirilmektedir. Bu kuralların bazıları çevrimiçi (on-line) bazıları ise çevrimdışı (off-line) çalışmaktadır [12].

3.11.4.1. Çevrimiçi öğrenme kuralları

Bu kurallar gerçek zamanlı çalışabilmektedir. Bu kurallara göre öğrenen sistemler gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte diğer taraftan ise öğrenmeye devam etmektedir. Art ağının öğrenme kuralı ve Kohonen öğrenme kuralı bu sınıfta bulunan öğrenme kurallarına örnek olarak verilebilir [12].

3.11.4.2. Çevrimdışı öğrenme kuralları

Çevrimdışı öğrenme kurallarına dayalı öğrenen sistemler kullanıma alınmadan önce örnekler üzerinde eğitilirler. Bu kuralları kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. Sistemin öğrenmesi gereken yeni bilgiler söz konusu olduğunda sistem kullanımdan çıkarılmakta ve çevrimdışı olarak yeniden eğitilmektedir. Eğitim tamamlanınca sistem yeniden eğitilmektedir. Eğitim tamamlanınca sistem tekrar kullanıma alınmaktadır. YSA' da yaygın olarak kullanılan "Delta Öğrenme Kuralı" bu tür öğrenmeye örnek olarak verilebilir [12].

3.11.4.3. Öğrenme kurallarından bazıları

Öğrenme sistemlerinde kullanılan değişik öğrenme kuralları vardır. Yapay sinir ağlarından bu öğrenme kurallarının çoğu Hebb kuralına dayanmaktadır [12-13].

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

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

c. Delta kuralı: Bu kural hebb kuralının biraz daha geliştirilmiş seklidir. Bu kurala göre beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak için yapay sinir ağının elemanlarının bağlantılarının ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanarak geliştirilmiştir. Ağın ürettiği çıktı ile üretilmesi gereken (beklenen) çıktı arasındaki hatanın karelerinin ortalamasını en aza indirmek hedeflenmektedir.

d. Kohonen kuralı: Bu kurala göre ağın elemanları (hücreleri) ağırlıklarını değiştirmek için birbirleri ile yarışır. En büyük çıktıyı üreten hücre kazanan hücre olmakta ve bağlantı ağırlıkları değiştirilmektedir. Bu, o hücrenin yakınındaki hücrelere karşı daha kuvvetli hale gelmesi demektir. Hem kazanan elemanların hem de komşuları sayılan elemanların (hücrelerin) ağırlıklarını değiştirmesine izin verilmektedir [12].

3.12. Yapay Sinir Ağ Uygulamalarının Avantajları

YSA'nın uygulamaya alınmasının arkasında haklı gerekçeler ve yararlar vardır. Bunları şu şekilde sıralamak mümkündür [12-13].

1. YSA matematik olarak modellenmesi mümkün olmayan veya zor olan karmaşık problemleri çok rahat modelleyerek çözebilmektedir.
2. YSA kullanılarak problemleri başarılı bir şekilde çözebilmek için problemin çok iyi modellenmesi gerekmektedir. Bu modelleme, problemi çözebilmek için sadece söz konusu olay ile ilgili örneklerin belirlenip toplanmasına yardımcı olacaktır. Örneklerin dışında herhangi bir ön bilgiye ihtiyaç yoktur. Örnek bulmak bilgi bulmaktan çok daha kolaydır. Yeter ki modele uygun örnekler olsun. Bunlar da genellikle bir sorun oluşturmaktadır.
3. Gerçek dünyada olaylar ve olayların arkasındaki değişik faktörlerin birbirleri ile ilişkilerini ve birbirinin üzerindeki etkileri gerçek hayatta bilmek zordur. YSA bu ilişkileri otomatik olarak örneklerden öğrenir. Kullanıcıların bu ilişkileri bilmesi

ve ađa söylemesi beklenmemektedir. Geleneksel yöntemlerde bu ilişkileri belirlemek veya yok saymak gerekmektedir. Bu özellik belki de yapay sinir ağlarının en önemli avantajlarından birisidir.

4. Benzer şekilde, gerçek dünya olayları ve bu olayların arkasındaki faktörlerin birbirleri ile ilişkileri doğrusal olmaz ise bu ilişkileri modellemek çok zordur. O nedenle gerçek hayatta problemleri çözmek için bazı varsayımlar yapmak gerekmektedir. Bunda modellenen sistemin gerçek sisteme uygunluđunu azaltmakta ve gerçek sistemin davranışlarını kontrol altına almayı zorlaştırmaktadır. Özellikle insan hayatının söz konusu olduđu yerlerde bu sebepten dolayı geleneksel sistemleri kullanmak sorun olabilmektedir. Yukarıda anlatılanların aksine, yapay sinir ağları için ise ilişkilerin doğrusal olup olmaması önemli değildir. Çünkü ilişkileri gerçekleşen örnekler üzerinde kendiliđinden hesaba katılmaktadır. O nedenle, bu ilişkilerin modellenmesi sorunu geleneksel sistemdeki kadar zor değildir. Örnekler gerçek sistemi temsil etmektedirler. Onun için bu örnekleri kullanarak öğrenen ağlar tarafından verilen kararlar daha gerçekçi olmaktadır.
5. YSA uygulamaları hem pratik hem de maliyet bakımından daha ucuzdurlar. Sadece örneklerin belirlenmesi ve basit bir program problemi çözmek için yeterli olabilmektedir.
6. YSA zaman bakımından da çok verimli çalışırlar. Örneklerin bulunması, ağların oluşturulması, olayı öğrenmesi, gerçek zamanda kullanıma alınması çok kısa bir zaman diliminde mümkün olabilmektedir. Aynı zamanda yapay sinir ağlarının çalışması da geleneksel sistemlerden daha hızlıdır.
7. YSA yeni bilgilerin ortaya çıkması ve ortamda bazı deđişikliklerin olması durumunda yeniden eğitebilirler. Bazı ağların eğitilmesine de gerek yoktur. Kendileri ortama uyumu öğrenerek gerçekleştirebilirler.
8. YSA paralel çalışabilmeleri onların gerçek zamanlı kullanımlarını kolaylaştırmaktadır [12].

3.13. Yapay Sinir Ağ Uygulamalarının Dezavantajları

1. YSA donanım bağımlı çalışmaları önemli bir sorun olarak görülebilir. Ağların temel varoluş nedenlerinden biriside paralel işlemciler üzerinde çalışabilmeleridir. Ağların özellikle, gerçek zamanlı bilgi işleyebilmeleri paralel çalışabilen işlemcilerin varlığına bağlıdır. Günümüzdeki makinelerin çoğu seri şekilde çalışabilmekte e aynı zamanda sadece tek bir bilgiyi işleyebilmektedir. Paralel işlemleri seri makinelerde yapmak ise zaman kaybına yol açmaktadır. Bunun yanı sıra bir ağın nasıl oluşturulması gerektiğini belirleyecek kuralların olmaması da başka bir dezavantajdır. Her problem farklı sayıda işlemci gerektirebilir. Bazı problemleri çözebilmek için gerekli olan paralel işlemcilerin tamamının bir arada çalıştırmak mümkün olmayabilir [12].
2. Probleme uygun ağ yapısının belirlenmesi genellikle deneme yanılma yolu ile yapılmaktadır. Bu ise önemli bir problemdir. Çünkü eğer problem için uygun bir ağ oluşturulmaz ise çözümü olan bir problemin çözülememesi veya performansı düşük çözümlerin elde edilmesi söz konusu olabilir. Bu aynı zamanda bulunan çözümün en iyi çözüm olduğunu da garanti etmez. Yani YSA kabul edilebilir çözümler üretebilir. Optimum çözümü garanti etmez [11-12].
3. Bazı ağlarda ağın parametre değerlerinin, her katmanda olması gereken proses elemanı sayısı, katman sayısı vb. belirlenmesinde de bir kural olmaması diğer bir problemdir. Bu, iyi çözümler bulmayı zor durumda bırakan bir etken olarak görülebilir. Bu parametrelerin belirlenmesi de kullanıcının tecrübesine bağlıdır. Her problem için ayrı faktörleri dikkate almayı gerektirmektedir. Bu parametre değerleri için belirli standartların oluşturulması çok zor olduğundan her problem için ayrı ayrı değerlendirmeler yapılması gerekmektedir. Bu da önemli bir dezavantaj olarak görülebilir.
4. Ağın öğreneceği problemin ağa gösterimi de çok önemli bir problemdir. YSA yukarıda belirtildiği gibi sadece nümerik bilgiler ile çalışmaktadırlar. Problemin nümerik gösterime dönüştürülmesi lazımdır. Bu ise kullanıcının becerisine bağlıdır. Uygun bir gösterim mekanizmasının kurulamamış olması problemin

çözümünü engelleyebilir veya düşük performanslı bir öğrenme elde edilebilir. Problemin nümerik gösterimi mümkün olsa bile bunun ağa gösteriliş şekli problemin başarılı bir şekilde çözülmesini yakından etkiler. Örneğin bir olay hem ayrık hem de sürekli değerler ile gösterilebilir. Bunun hangisinin daha başarılı bir öğrenme gerçekleştireceği ise bilinmemektedir. Bu konuda, kullanıcının tecrübesi de yeterli olmayabilir. Bu günümüzde bir çok olayın YSA ile çözülememesinin en önemli nedenlerinden birisidir.

5. Ağın eğitiminin ne zaman bitirileceğine karar vermek için de geliştirilmiş bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli bir değerin altına indirilmesi eğitimin tamamlanması için yeterli görülmektedir. Fakat neticede en iyi öğrenmenin gerçekleştiği söylenememektedir. Sadece iyi çözümler üretebilen bir ağ oluştu denilmektedir. Optimum neticeleri veren bir mekanizma henüz geliştirilmemiştir. Bu konuda oldukça önemli olup çözülmesi için araştırmalar yapılması gerekmektedir.
6. Bir diğer sorun ise; belki de yukarıdakilerin en önemlisi, daha önce açıklandığı gibi ağın davranışlarının açıklanamamasıdır. Bir probleme çözüm üretildiği zaman bunun nasıl ve neden üretildiği konusunda bir bilgi bulmak mümkün değildir. Bu ise ağın sonucuna olan güveni azaltmaktadır [12].

Bütün bu dezavantajlara rağmen yapay sinir ağları tarafından her problem için değişik şekillerde çözümler üretebilmekte ve başarılı uygulamalar oluşturmak mümkün olabilmektedir. Bu nedenle, bu dezavantajları YSA olan ilgiyi düşürmek için görmemek gerekir. YSA' nın her derde ilaç gibi olmadığını vurgulamak için bunlar burada açıklanmıştır. Ağların bu dezavantajlardan kurtularak problemlere çözüm üretebilmesi için ağların oluşturulmasını titizlik ile gerçekleştirmek gerekmektedir. Hem çözülecek olan problemler hem de YSA konusunda yeterli oranda bilgi sahibi olmak başarılı sonuçlar elde edilmesini sağlayabilir. YSA geliştirecek olanların bu gerçeği göz ardı etmeden problemlere çözüm üretecek bir ağ oluşturmanın mümkün olabileceğini fakat bunun o kadar kolay olmayacağını bilmesi gerekmektedir [12].

3.14. Yapay Sinir Ağlarının Uygulama Alanları

Uygulama alanına göre kullanılan yapay sinir ağları aşağıdaki gibi sınıflandırılabilirler [11-12-13].

3.14.1. Endüstriyel uygulamalar

- a. YSA bir endüstriyel proseste fırınların ürettiği gaz miktarının tahmini
- b. İmalatta, ürün tasarımı, proses ve makinelerin bakımı ve hataların teşhisi, görsel kalite kontrolü.
- c. Kimyasal proseslerin dinamik modellenmesi
- d. Otomobillerde otomatik rehber sisteminin geliştirilmesi
- e. Robotlarda görme sistemleri ve manipülatörlerin kontrol edilmesi
- f. Cep telefonlarında ses ile çalışabilme
- g. Araba pistonlarının üretim şartlarının belirlenmesi
- h. Elektronik yonga hata analizleri
- i. Optimizasyon çalışmaları (üretim planlama ve kontrol çalışmalarında)
- j. Müşteri tatmini ve Pazar verilerinin değerlendirilmesi ve analiz edilmesi
- k. Kömür güç istasyonları için çevrimiçi karbon akımı ölçülmesi [12]

3.14.2. Askeri uygulamalar

- a. Hedef tanıma ve takip sistemleri
- b. Yeni sensörlerin performans analizleri
- c. Radar ve görüntü sinyalleri işleme
- d. Sensör fizyonu
- e. Askeri uçakların uçuş yörüngelerinin belirlenmesi
- f. Mayın dedektörleri [11-12]

3.14.3. Yapay sinir ağlarının mühendislik uygulamaları

- a. Yorumlama
- b. Tahmin
- c. Teşhis
- d. Planlama
- e. Onarım
- f. Kontrol
- g. Sınıflandırma
- h. Yazıyı konuşurmaya dönüştürme
- i. Karakter tanıma
- j. El yazısı tanıma
- k. Bomba detektörü
- l. Robotik
- m. Sinyal işleme
- n. Optimizasyon [12-13]

3.14.4. Yapay sinir ağları için uygulama örnekleri

Otomatik benzin pompaları ile park eden otomobile yakıt yükleme işlemi oldukça kolaylaşmaktadır. Müşterinin kullanacağı bir kart ile kimlik numarasını ve yakıt miktarı bilgisini alan sistem, otomobilin yerini, yakıt tankı kapağının yerini otomatik olarak tespit edebilmekte ve kapağı açıp doğru oktan numarasına sahip yakıt türünden istenen miktarda yakıt araca aktarabilmektir (Bkz. Şekil 3.17) [1].



Şekil 3.17. Otomatik benzin pompası [1]

Tekerlekli bir sandalyeye iliştirilmiş 6 hareket serbestisine sahip bir mekanik kol ile kapı açmak, kahve yapmak veya benzeri basit görevler için hastaya yardımcı olunabilmektedir. Kullanılmadığı esnada yavaşça katlanan bu kola hastanın durumuna göre uygun kumanda cihazları takılmak suretiyle kullanıcıya göre kullanım kolaylığı sağlanabilmektedir (Bkz. Şekil 3.18).



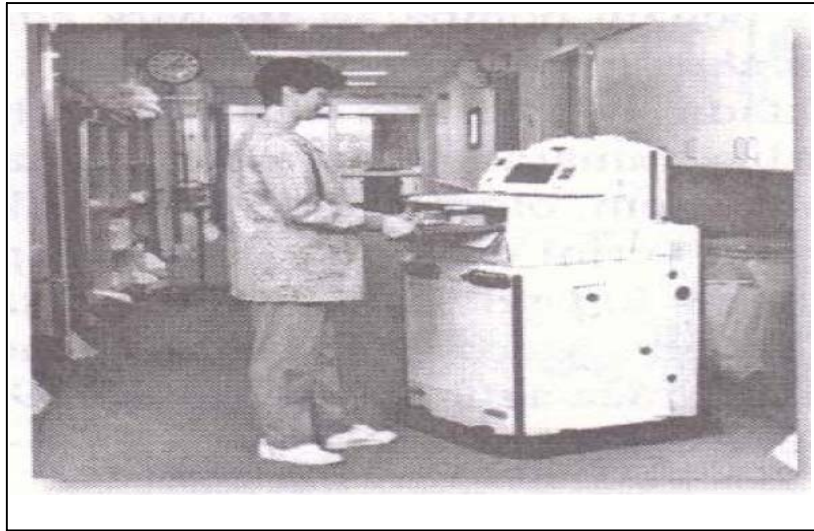
Şekil 3.18. Mekanik yardımcı bir tekerlekli sandalye [1]

Tıbbi uygulamalarda kullanılan bu robot sistemi 6 hareket serbestisine sahiptir. Denetim döngüsünde yer alan bir ana bilgisayarın yanı sıra görselleştirme işlemi için kullanılan bir iş istasyonu ile ameliyatlarda kullanılan kameraları taşımak amacıyla üretilmiştir. Mekanik sistemin hareketi önceden programlanan bir yörünge olabildiği gibi eşzamanlı üretilen veriler de işlenebilmektedir (Bkz. Şekil 3.19).



Şekil 3.19. Tıbbi uygulamalar için geliştirilen mekanik bir görüntüleme sistemi [1]

Akıllı makineler için verilebilecek bir başka örnek ise taşımacılık işlemleri için geliştirilen robotlardır. Özellikle hastaneler için tasarlanan bu sistem, yemeklerin, dokümanların veya hastaların ihtiyaç duyduğu ilaçların taşınması işleminde kullanılmaktadır. Sistem kapı açabilmekte ve asansör kullanabilmektedir (Bkz. Şekil 3.20) [1].

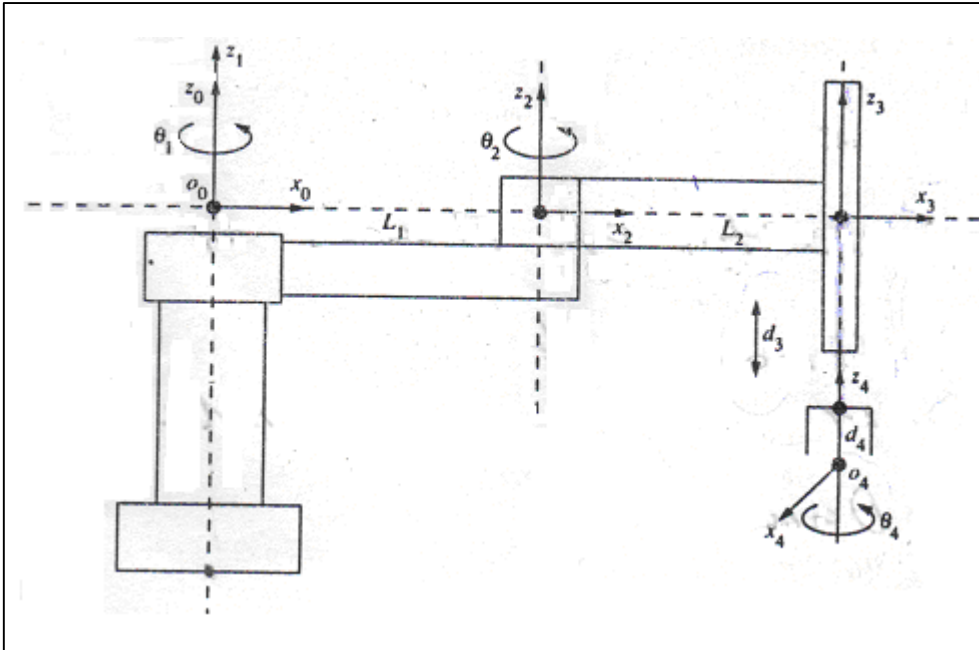


Şekil 3.20. Taşıma robotu [1]

BÖLÜM 4. SCARA TİPİ ROBOTUN GENEL DİNAMİKLERİ VE ANALİZİ

Bu çalışmada kullanılan dört serbestlik dereceli SCARA tipi robot Şekil 4.1' de gösterilmiştir. İlk olarak Lagrange – Euler dinamik denklemleri ile her bir ekleme uygulanacak tork değerleri ve robotun her bir eklemının yörüngelerinin belirlenmesi ile de her bir eklemının konum değerleri belirlenir. Bu elde edilen değerler daha sonra YSA' nda kullanılır [15].

SCARA robotunun modellenmesi ve de analizi için MATLAB 7.0 programı ve de MATLAB 7.0 Neural Network Toolbox' ı kullanılmıştır [16-17].



Şekil 4.1. Robotun şematik çizimi

4.1. Robotun Dinamik Modeli

Bir robot kolun dinamik modeli, robot kolun dinamik davranışlarını belirleyen hareket denklemlerinden oluşur. Bu denklemlerin bilgisayar yardımı ile çözümü robot ve robot kontrol tasarımı açısından büyük kolaylık sağlar. Robot kolunun dinamik analizi, eklemlere tahrik elemanları tarafından uygulanan moment veya kuvvet büyüklükleri ile robot kolunun zamana göre konumu, hızı ve ivmesi arasındaki ilişkilerin incelenmesi demektir. Böyle bir analiz için, robot kolun dinamik davranışlarını tanımlayan lineer olmayan diferansiyel denklem takımlarının elde edilmesi ve çözülmesi gerekmektedir.

Dinamik problem düz dinamik (forward dynamics) ve ters dinamik (inverse dynamics) olmak üzere ikiye ayrılmaktadır. Düz dinamik problem, herhangi bir t anında istenen kuvvet ve momentler verildiğinde robot kolunun alacağı konum ve yönelme için gerekli olan genelleştirilmiş koordinatları, eklem hız ve ivmelerini bulma problemidir. Ters dinamik problem ise, robot kolunun istenen bir konum ve yönelmeye gelmesi için (genelleştirilmiş koordinatlar, bunların birincisi (hız) ve ikinci (ivme) türevleri verildiğinde) gerekli olan kuvvet ve momentlerin hesaplanmasıdır [18].

Robot kolunun dinamik denklemlerini elde etmek için bilinen birçok metot vardır. Bunlar Lagrange-Euler (L-E), Recursive Lagrange (R-L), Newton-Euler (N-E), Genelleştirilmiş D'Alambert (G-D) prensibi gibi yaklaşımlardır. Bu yöntemlerden en çok kullanılanları L-E ve N-E yaklaşımlarıdır.

Bu çalışmada robot kolun dinamik modelinin çıkarılmasında, iyi planlanmış bir yapıya sahip ve çıkarılması N-E yaklaşımından daha basit ve sistematik olan L-E yaklaşımı ile elde edilmiş denklemler kullanılacaktır. Bu yaklaşım sistemin içerdiği toplam iş ve enerji ile ifade edilir. Lagrange-Euler eşitliğinin genel ifadesi 4.1 nolu eşitlikteki gibidir:

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \left[\frac{\partial L}{\partial q} \right] = \tau_i \quad i = 1, 2, 3, \dots, n \quad (4.1)$$

$$L = K - P \quad (4.2)$$

4.2 nolu denklem ile ifade edilir. Burada;

L = Lagrange fonksiyonu

K = Robot kolunun toplam kinetik enerjisi 4.3 nolu denklemdeki gibidir,

$$K = \frac{1}{2} \sum_{i=1}^4 m_i V_i^2 + \frac{1}{2} \sum_{i=1}^4 I_i \dot{\theta}_i^2 \quad (4.3)$$

P = Robot kolunun toplam potansiyel enerjisi 4.4 nolu denklemdeki gibidir,

$$P = m_i g d_i \quad (4.4)$$

q_i = Robot kolunun genelleştirilmiş koordinatları (Döner eklemlerde θ_i ve kayar eklemlerde d_i olarak alınır.)

τ_i = Sisteme uygulanan genelleştirilmiş moment veya kuvvet

Dinamik denklem matris formunda 4.5 nolu eşitlikteki gibidir;

$$\tau = M(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) \quad (4.5)$$

Burada θ açısal konum değişkeni, $M(\theta)$ nxn pozitif tanımlı atalet matrisini, $N(\theta, \dot{\theta})$ coriolis, merkezkaç, yerçekimi ve sürtünme kuvvetini, τ tork değerini simgelemektedir [6-15-19-20-21-22].

$$M(\theta) = [m_{ij}(\theta)]_{n \times n} \quad i, j=1, \dots, n \quad n, \text{ eklem sayısıdır.} \quad (4.6)$$

$$m_{ij}(\theta) = m_{ji}(\theta) \quad (4.7)$$

$$N(\theta, \dot{\theta}) = [n_i(\theta, \dot{\theta})]_{n \times 1} \quad (4.8)$$

Dört serbestlik derecesine sahip olan SCARA robotun genel dinamikleri 4.5, 4.6, 4.7 ve 4.8 nolu denklemler kullanılarak 4.9 nolu denklemdeki gibi elde edilmiştir [15].

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} \quad (4.9)$$

Matrislerin çarpımı ile 4.10, 4.11, 4.12 ve 4.13 nolu denklemler elde edilmiştir.

$$\tau_1 = m_{11}\ddot{\theta}_1 + m_{12}\ddot{\theta}_2 + m_{13}\ddot{\theta}_3 + m_{14}\ddot{\theta}_4 + n_1 \quad (4.10)$$

$$\tau_2 = m_{21}\ddot{\theta}_1 + m_{22}\ddot{\theta}_2 + m_{23}\ddot{\theta}_3 + m_{24}\ddot{\theta}_4 + n_2 \quad (4.11)$$

$$\tau_3 = m_{31}\ddot{\theta}_1 + m_{32}\ddot{\theta}_2 + m_{33}\ddot{\theta}_3 + m_{34}\ddot{\theta}_4 + n_3 \quad (4.12)$$

$$\tau_4 = m_{41}\ddot{\theta}_1 + m_{42}\ddot{\theta}_2 + m_{43}\ddot{\theta}_3 + m_{44}\ddot{\theta}_4 + n_4 \quad (4.13)$$

Dört serbestlik derecesine sahip olan SCARA robotun dinamikleri [15];

$$\begin{aligned} m_{11} = & (I_{ZZ1} + I_{ZZ2} + I_{ZZ3} + I_{ZZ4}) + \left(\frac{m_1}{4} + 2m_3 + m_4\right)L_1^2 + \left(\frac{m_2}{2} + 3m_3 + m_4\right)L_1L_2C_2 + \\ & \left(\frac{m_2}{4} + m_3 + m_4\right)L_2^2 \end{aligned} \quad (4.14)$$

$$m_{12} = (I_{ZZ2} + I_{ZZ3} + I_{ZZ4}) + \left(\frac{m_2}{4} + m_3 + m_4\right)L_2^2 + (2m_3 + m_4)L_1L_2C_2 \quad (4.15)$$

$$m_{13} = 0 \quad (4.16)$$

$$m_{14} = I_{ZZ4} \quad (4.17)$$

$$m_{22} = (I_{ZZ2} + I_{ZZ3} + I_{ZZ4}) + \left(\frac{m_2}{4} + m_3 + m_4\right)L_2^2 \quad (4.18)$$

$$m_{23} = 0 \quad (4.19)$$

$$m_{24} = I_{ZZ4} \quad (4.20)$$

$$m_{33} = m_3 m_4 \quad (4.21)$$

$$m_{34} = 0 \quad (4.22)$$

$$m_{44} = I_{zz4} \quad (4.23)$$

$$n_1 = \left(\frac{m_2}{2} - m_3\right) L_1 L_2 \dot{\theta}_1^2 - (4m_3 + 2m_4) L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \quad (4.24)$$

$$n_2 = \left(\frac{m_2}{2} + m_3 + m_4\right) L_1 L_2 S_2 \dot{\theta}_1^2 \quad (4.25)$$

$$n_3 = (m_3 + m_4) g \quad (4.26)$$

$$n_4 = 0 \quad (4.27)$$

4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26 ve 4.27' de verilen dinamikler, 4.10, 4.11, 4.12 ve 4.13 nolu denklemlerde yerlerine yazıldığında; robotun tork denklemleri, 4.28, 4.29, 4.30 ve 4.31 nolu denklemlerdeki gibi elde edilmektedir;

$$\begin{aligned} \tau_1 = & (I_{zz1} + I_{zz2} + I_{zz3} + I_{zz4}) \ddot{\theta}_1 + \left(\frac{m_1}{4} + 2m_3 + m_4\right) L_1^2 \ddot{\theta}_1 + \left(\frac{m_2}{2} + 3m_3 + m_4\right) L_1 L_2 C_2 \ddot{\theta}_1 + \\ & \left(\frac{m_2}{4} + m_3 + m_4\right) L_2^2 \ddot{\theta}_1 + (I_{zz2} + I_{zz3} I_{zz4}) \ddot{\theta}_2 + \left(\frac{m_2}{4} + m_3 + m_4\right) L_2^2 \ddot{\theta}_2 + (2m_3 + m_4) L_1 L_2 C_2 \ddot{\theta}_2 + \\ & I_{zz4} \ddot{\theta}_4 + \left(\frac{m_2}{2} - m_3\right) L_1 L_2 \dot{\theta}_1^2 - (4m_3 + 2m_4) L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \end{aligned} \quad (4.28)$$

$$\begin{aligned} \tau_2 = & (I_{zz2} + I_{zz3} I_{zz4}) \ddot{\theta}_1 + \left(\frac{m_2}{4} + m_3 + m_4\right) L_2^2 \ddot{\theta}_1 + (2m_3 + m_4) L_1 L_2 C_2 \ddot{\theta}_1 + (I_{zz2} + I_{zz3} I_{zz4}) \ddot{\theta}_2 + \\ & \left(\frac{m_2}{4} + m_3 + m_4\right) L_2 \ddot{\theta}_2 + I_{zz4} \ddot{\theta}_4 + \left(\frac{m_2}{2} + m_3 + m_4\right) L_1 L_2 S_2 \dot{\theta}_1^2 \end{aligned} \quad (4.29)$$

$$\tau_3 = m_3 m_4 \ddot{\theta}_3 + (m_3 + m_4) g \quad (4.30)$$

$$\tau_4 = I_{zz4} (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_4) \quad (4.31)$$

Denklemlerde kullanılan uzuv kütle değerleri ve boyutları [19];

$$\begin{array}{lll} m_1 = 2kg & L_1 = 2m & g = 9.8 \frac{m}{sn^2} \\ m_2 = 2kg & L_2 = 1m & C_2 = \cos(\theta_2) \\ m_3 = 1kg & d_3 = 0.75m & S_2 = \sin(\theta_2) \\ m_4 = 0.5kg & d_4 = 0.25m & \end{array}$$

Uzuvların atalet momentleri ise aşağıdaki gibidir;

$$\begin{array}{l} I_{zz1} = 8kg.m^2 \\ I_{zz2} = 2kg.m^2 \\ I_{zz3} = 0.5625kg.m^2 \\ I_{zz4} = 0.03125kg.m^2 \end{array}$$

Bu verilen değerler ile eklemlere uygulanacak tork değerleri bulunmaktadır.

4.2. Robotun Yörünge Denklemlerinin Elde Edilmesi

Modellenen SCARA robotu üç dönme ve bir öteleme hareketi yapmaktadır. Bu yüzden dönme hareketleri 4.32, 4.33 ve 4.35 nolu eşitliklerde verilen kübik yörüngeler ile, öteleme hareketi ise 4.34 nolu eşitlikte verilen doğrusal yörünge ile ifade edilmiştir [6-15-19-21].

$$\theta_1 = a.t^3 + b.t^2 + c.t + e \quad (\text{Birinci uzuv}) \quad (4.32)$$

$$\theta_2 = f.t^3 + h.t^2 + k.t + d \quad (\text{İkinci uzuv}) \quad (4.33)$$

$$\theta_3 = v.t + y \quad (\text{Üçüncü uzuv}) \quad (4.34)$$

$$\theta_4 = m.t^3 + n.t^2 + p.t + z \quad (\text{Dördüncü uzuv}) \quad (4.35)$$

Birinci uzuv için, 4.32 nolu denklemin birinci ve ikinci türevleri alındığında 4.36 ve 4.37 nolu denklemler elde edilir;

$$\theta_{1T} = 3.a.t^2 + 2.b.t + c \quad (4.36)$$

$$\theta_{1TT} = 6.a.t + 2.b \quad (4.37)$$

elde edilir. Bu denklemlerin,

$$t = 0(sn) \Rightarrow Q1 = 0^\circ \Rightarrow Q_{1T} = 0(rd / s)$$

$$t = 10(sn) \Rightarrow Q1 = 30^\circ \Rightarrow Q_{1T} = 0(rd / s)$$

başlangıç ve bitiş şartlarında çözümü yapıldığında;

$$c = e = 0$$

$$a = -0.001047$$

$$b = 0.0157$$

katsayıları elde edilir.

İkinci uzuv için, 4.33 nolu denklemin birinci ve ikinci türevleri alındığında 4.38 ve 4.39 nolu denklemler elde edilir;

$$\theta_{2T} = 3.f.t^2 + 2.h.t + k \quad (4.38)$$

$$\theta_{2TT} = 6.e.f + 2.h \quad (4.39)$$

elde edilir. Bu denklemlerin,

$$t = 0(sn) \Rightarrow Q2 = 0^\circ \Rightarrow Q_{2T} = 0(rd / s)$$

$$t = 10(sn) \Rightarrow Q2 = 60^\circ \Rightarrow Q_{2T} = 0(rd / s)$$

başlangıç ve bitiş şartlarında çözümü yapıldığında;

$$k = d = 0$$

$$f = -0.002095$$

$$h = 0.031425$$

katsayıları elde edilir.

Üçüncü uzuv için, 4.34 nolu denklemin birinci ve ikinci türevleri alındığında 4.40 ve 4.41 nolu denklemler elde edilir;

$$\theta_{3T} = v \quad (4.40)$$

$$\theta_{3TT} = 0 \quad (4.41)$$

elde edilir. Bu denklemlerin,

$$t = 0(\text{sn}) \Rightarrow Q_3 = 0(\text{m}) \Rightarrow Q_{3T} = 0(\text{m/s})$$

$$t = 10(\text{sn}) \Rightarrow Q_3 = 0.5(\text{m}) \Rightarrow Q_{3T} = 0(\text{m/s})$$

başlangıç ve bitiş şartlarında çözümü yapıldığında;

$$y = 0$$

$$v = 0.05$$

katsayıları elde edilir.

Dördüncü uzuv için, 4.35 nolu denklemin birinci ve ikinci türevleri alındığında 4.42 ve 4.43 nolu denklemler elde edilir;

$$\theta_{4T} = 3.m.t^2 + 2.n.t + p \quad (4.42)$$

$$\theta_{4TT} = 6.m.t + 2.n \quad (4.43)$$

elde edilir. Bu denklemlerin,

$$t = 0(\text{sn}) \Rightarrow Q4 = 0^\circ \Rightarrow Q4T = 0(\text{rd} / \text{s})$$

$$t = 10(\text{sn}) \Rightarrow Q4 = 120^\circ \Rightarrow Q4T = 0(\text{rd} / \text{s})$$

başlangıç ve bitiş şartlarında çözümü yapıldığında;

$$p = z = 0$$

$$m = -0.001397$$

$$n = 0.02095$$

Belirlenen başlangıç ve bitiş şartlarında verilen denklemlerin çözümü yapıldığında 4.44, 4.45, 4.46 ve 4.47 nolu denklemler elde edilmektedir;

$$\theta_1 = (-0.001047)t^3 + (0.0157)t^2 \quad (4.44)$$

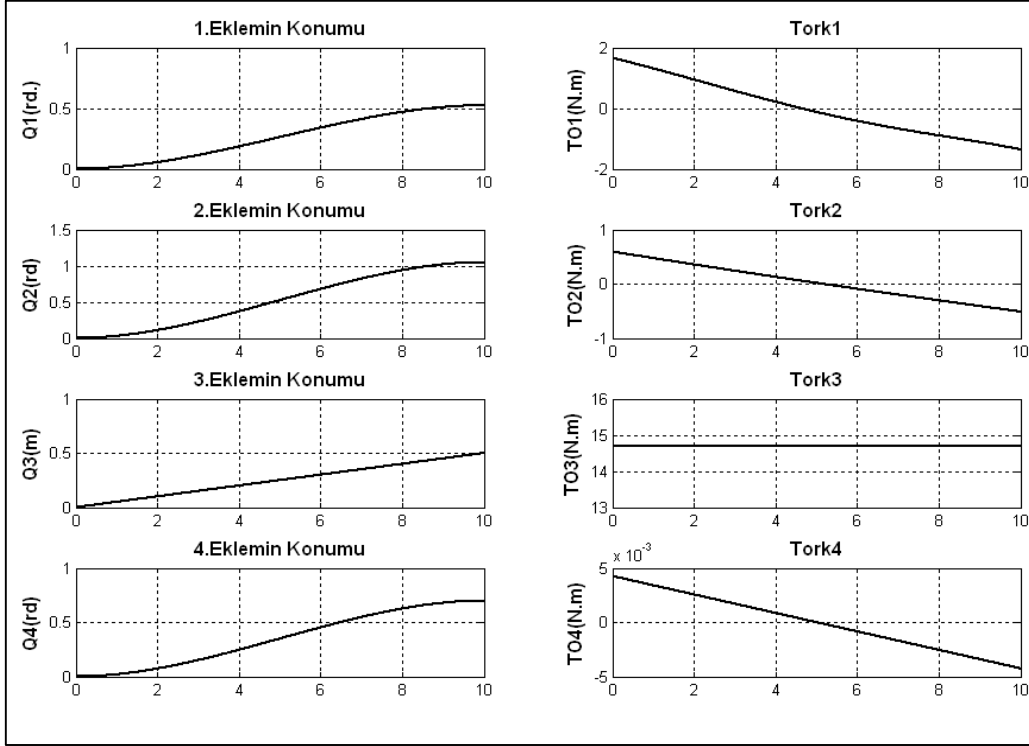
$$\theta_2 = (-0.002095)t^3 + (0.031425)t^2 \quad (4.45)$$

$$\theta_3 = 0.05t \quad (4.46)$$

$$\theta_4 = (-0.001397)t^3 + (0.02095)t^2 \quad (4.47)$$

4.3. Robotun Konum ve Tork Grafiklerinin Elde Edilmesi

Zaman aralığı 0' dan 10 saniye' ye kadar alınmış ve 0.01 artımlar ile konum değerleri elde edilmiştir. Elde edilen konum, hız ve ivme değerleri; Lagrange – Euler dinamik denklemlerinde yerine yazılarak tork değerleri belirlenmiştir. Şekil 4.2' de her bir eklemin konum ve tork grafikleri verilmiştir [16-17].

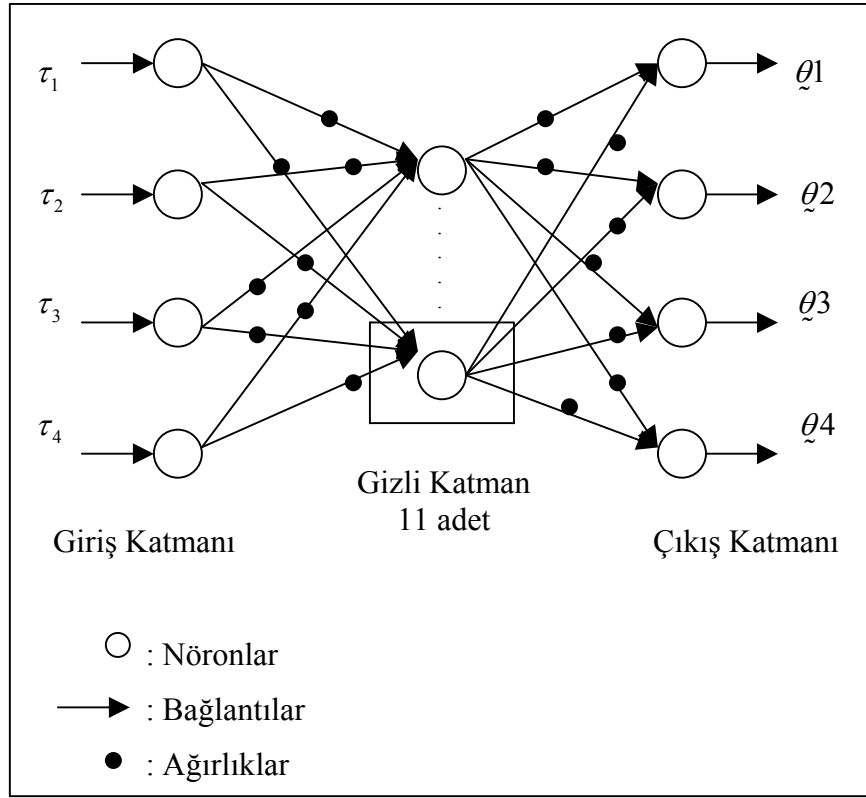


Şekil 4.2. Her bir eklemin konum ve tork grafikleri

Bu elde edilen sonuçlar için Matlab 7.0 komutları ile yazılan m-file dosyası Ek A' da verilmiştir [16-17].

BÖLÜM 5. SCARA TİPİ ROBOTUN YAPAY SİNİR AĞLARI İLE EĞİTİLMESİ

Oluşturulan 4:11:4 mimarisindeki yapay sinir ağının girişleri Lagrange – Euler dinamik denklemleri sonucu bulunan tork değerleri, çıkışları ise her bir eklemlerin hareketi için elde edilen konum değerleri olarak alınmaktadır [3].

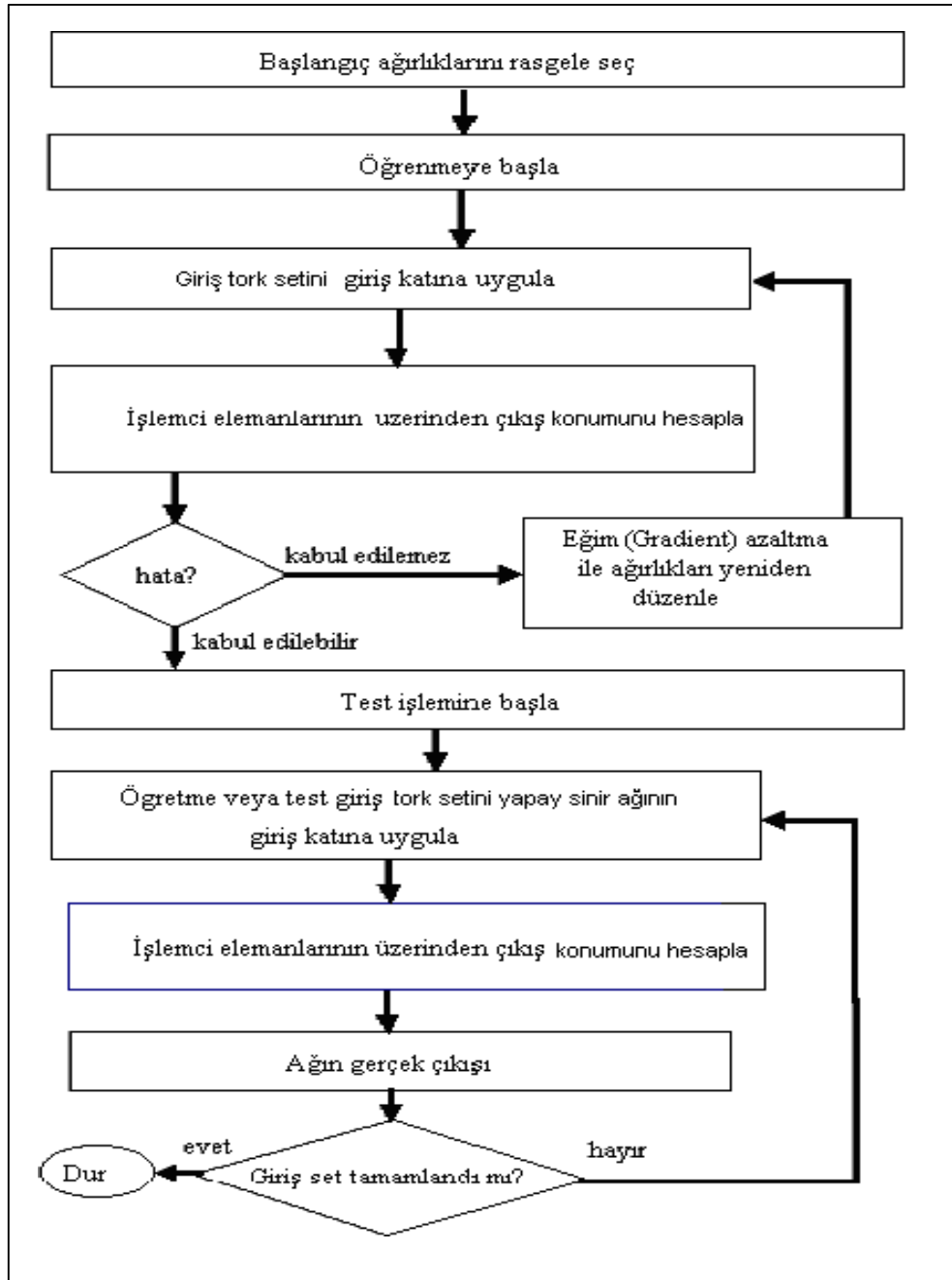


Şekil 5.1. Kullanılan yapay sinir ağın modeli

YSA' nın giriş katmanındaki giriş nöronlarının sayısı sistemin girişlerinin sayısı, çıkış katmanındaki çıkış nöronlarının sayısı da sistemin çıkışlarının sayısı ile belirlenmektedir. Buna göre oluşturulan YSA' nın giriş nöron sayısı 4, çıkış nöron sayısı da 4' tür. Gizli katmanda ise 11 adet nöron kullanılmıştır. Yapılan literatür araştırmalarına göre; gizli katmandaki nöron sayısı, giriş katman sayısı ile çıkış

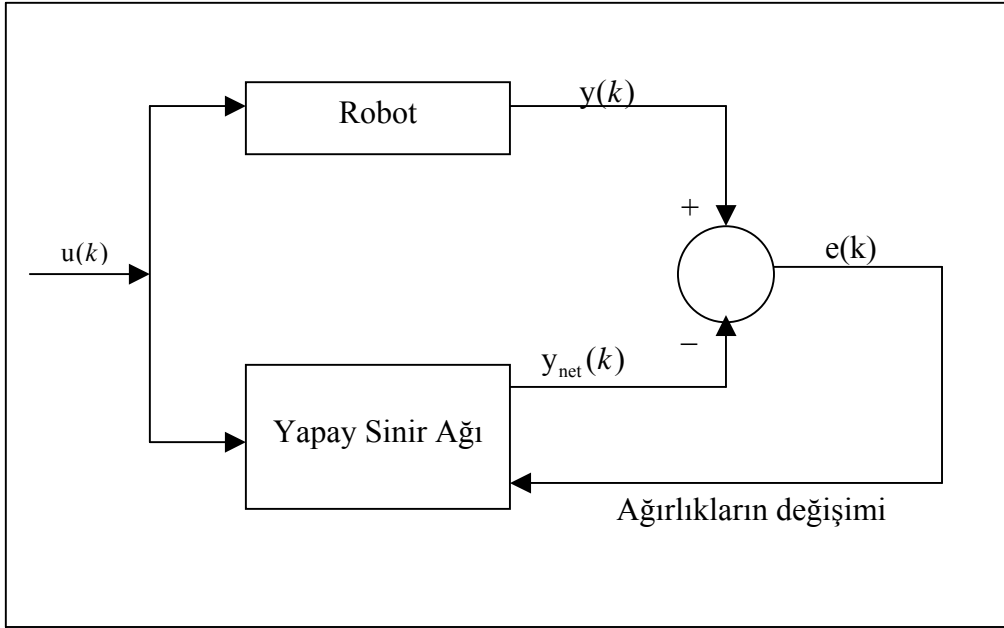
katman sayısının toplamının en az bir fazlası olarak belirlenmektedir. Bu yapılan çalışmada ise; gizli katmanda, deneysel (9,10,11,12....15 adet nöron kullanılarak) olarak yapılan çalışmalara göre en başarılı sonuçlar 11 adet nöron kullanılarak elde edilmiştir. Kullanılan YSA modeli Şekil 5.1' de ki gibidir [3-11-12-23].

Şekil 5.2' de oluşturulan sistemin akış şeması verilmiştir.



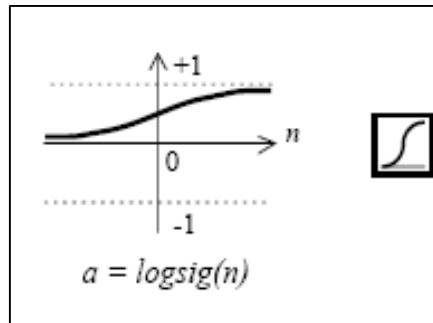
Şekil 5.2. Yapay sinir ağlarında modellenen robotun akış diyagramı

YSA ile robot modellemeye ait diyagram Şekil 5.3' de görülmektedir.



Şekil 5.3. Robot modelleme için bir yapay sinir ağı çözümü [$u(k)$:giriş, $y(k)$:çıkış ve $e(k)$:hata]

Oluşturduğumuz bu sistemde, ileri beslemeli geri yayılım ağı ve öğrenme stratejisi olarak da eğitici öğrenme kullanılmıştır. Gizli katmanda ve çıkış katmanında transfer fonksiyonu olarak Log-sigmoid kullanılmıştır. Bu fonksiyon 0 ile 1 arasındaki değerleri işlemektedir. Log-sigmoid transfer fonksiyonu Şekil 5.4' de ki gibidir.



Şekil 5.4. Log-sigmoid transfer fonksiyonu

Sistemin yapılan matematiksel işlemler sonucu elde edilen 1001 adet giriş-çıkış vektörü bulunmaktadır. Sistemin giriş ve çıkışlarının sayısal değerlerinden oluşan veriler 0.1-0.9 arasında olmak üzere normalize edilmektedir. Çünkü giriş ve çıkış

değerleri, kullanılan Log-sigmoid transfer fonksiyonunun sınırları arasında olmalıdır. Normalizasyon işlemi 5.1 nolu formülle elde edilmektedir [3];

$$\begin{aligned}
 A &= \min(\text{veri}); \\
 B &= \min(A) \\
 C &= \max(\text{veri}); \\
 D &= \max(C) , \text{ olmak üzere} \\
 \text{norveri} &= ((\text{veri}-B)/(D-B))*(0.9-0.1)+0.1; \text{ şeklindedir.} \quad (5.1)
 \end{aligned}$$

Oluşturulan geri yayılım eğitim fonksiyonunun ağırlık ve bias değerlerini güncelleştirmek için Levenberg-Marquardt optimizasyonu kullanılmıştır. Levenberg-Marquardt optimizasyon algoritması maksimum komşuluk fikri üzerine kurulan en küçük kareler tahmin metodudur. Levenberg-Marquardt metodu Gauss-Newton tekniğinin ve Steepest-descent metodunun en iyi özelliklerini birleştirir ve bu algoritmaların sınırlamalarının çoğundan kaçınır. Genellikle hızlı bir yakınsama yapar [3-24-25].

Yapay sinir ağının eğitimi için performans ölçütü olarak hataların kareleri ortalamasını kullanan Levenberg-Marquardt algoritması çok uygundur. Verilen girişlere karşılık YSA' nın bulduğu çıkışlar ile hedef (istenen) çıkış karşılaştırılır. Network çıkışı ve istenen çıkış arasındaki fark hata olarak hesaplanmaktadır. Bu hatanın toplamının ortalaması minimize edilmek istenmektedir. Bu minimize edilmek istenen değer MSE (Mean Squared Error) olarak tanımlanmaktadır ve ağ performansını belirlemektedir [3-24-26].

MSE 5.2 nolu denklemdeki gibi formülize edilmektedir [3-17];

$$MSE = \frac{1}{Q} \sum_{k=1}^Q (e(k))^2 = \frac{1}{Q} \sum_{k=1}^Q (t(k) - y(k))^2 \quad (5.2)$$

$e(k)$ = Hedef ve YSA çıkışları arasındaki hata

$t(k)$ = Hedef çıkış

$y(k)$ = YSA çıkış

Yukarıda kısaca açıklamaya çalışılan ve YSA sisteminde kullanılan fonksiyonlar şunlardır (Bkz. Tablo 5.1);

Tablo 5.1. Yapay sinir ağı sisteminde kullanılan fonksiyonlar

Fonksiyon	Görevleri
newff	İleri beslemeli geri yayılım ağı oluşturma fonksiyonu
logsig	Gizli katmandaki ve çıkış katmanındaki transfer fonksiyonu
trainlm	Levenberg-Marquardt geri yayılım eğitim fonksiyonu
learnqdm	öğrenme fonksiyonu
mse	performans fonksiyonu (hatanın toplamının ortalaması)

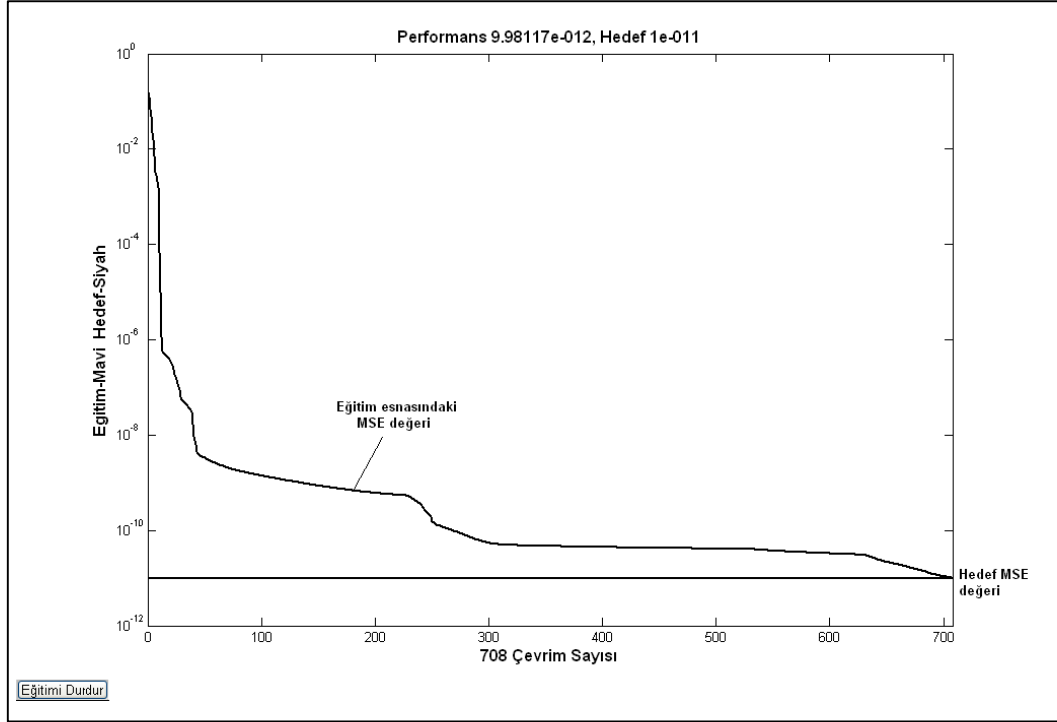
YSA sistemini oluşturmak için, normalize edilen giriş (tork) ve çıkış (konum) verilerinin 801 adedi YSA' nın eğitiminde, 200 adedi de YSA' nın testinde kullanılmaktadır. Burada dikkat edilmesi gereken, verilerin büyük bir kısmı eğitim seti olarak belirlenmelidir [3-17].

Eğitim aşamasında ağı giriş ve çıkış verileri kullanılmış, test aşamasında ise giriş verileri verilmek suretiyle ağ çıkışı elde edilmiştir.

5.1. Robotun Yapay Sinir Ağları ile Eğitilmesi Sonucu Performans, Çıkış ve Hata Grafiklerinin Elde Edilişi

Robotun YSA ile eğitilmesi işlemi Matlab 7.0 programı yardımı ile yapılmıştır. Matlab komutlarıyla yazılan dosya Ek B' de verilmektedir. Şimdi de yazılan programın Matlab 7.0 programı ile çalıştırılması ile elde edilen sonuçlar incelenmektedir [17].

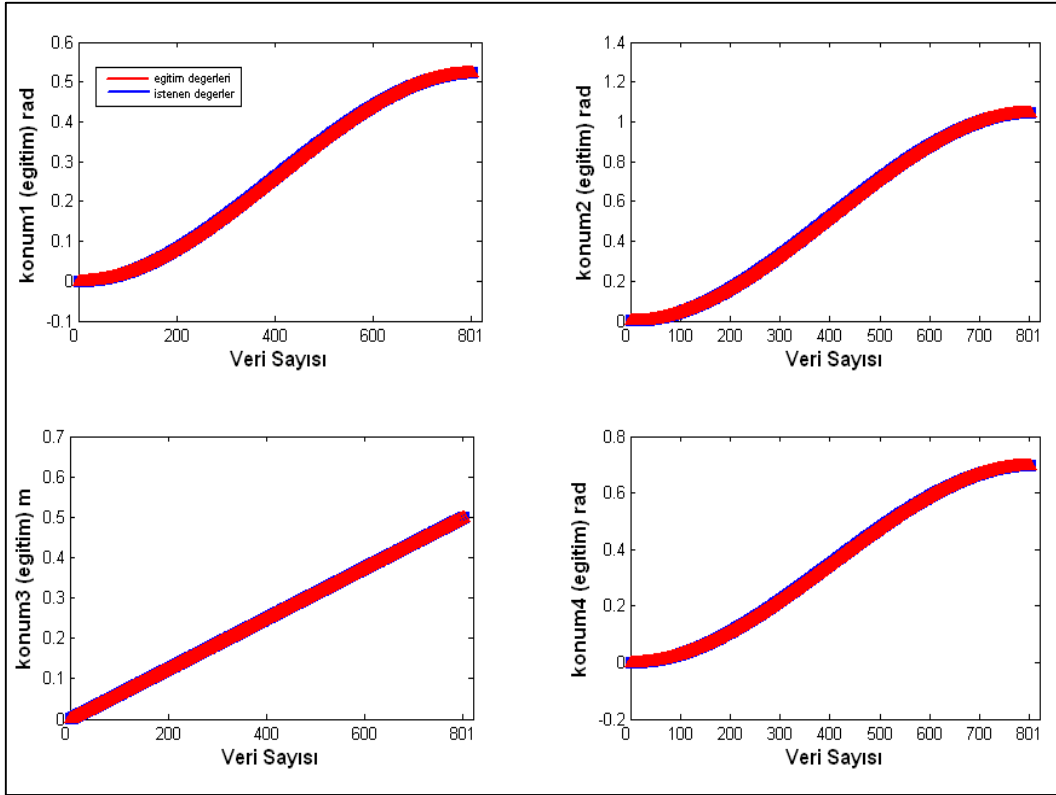
YSA' nın eğitimi esnasında hesaplanan MSE değerlerinin bulunduğu performans grafiği Şekil 5.5' de gösterilmektedir.



Şekil 5.5. Yapay sinir ağının performans grafiği

Eğitim sırasında elde edilen performans grafiğinde de görüldüğü gibi hedeflenen MSE değeri 10^{-11} olarak verilmiş ve eğitim sonucunda bulunan MSE değeri hedefe oldukça yaklaşarak 9.98117×10^{-12} olarak bulunmuştur. Böylece YSA 708 çevrim sonucunda hedeflenen MSE değerini yakalayarak eğitimini tamamlamıştır [3-17].

Modelin YSA' da eğitilmesi sonucu elde edilen konum çıkış değerleri ve istenen konum çıkış değerlerinin birlikte gösterildiği grafikler Şekil 5.6' da görülmektedir. Bu grafiklerde görünen mavi renkli çizim istenen değerler, kırmızı renkli çizim ise eğitim sonucunda elde edilen değerlerdir.

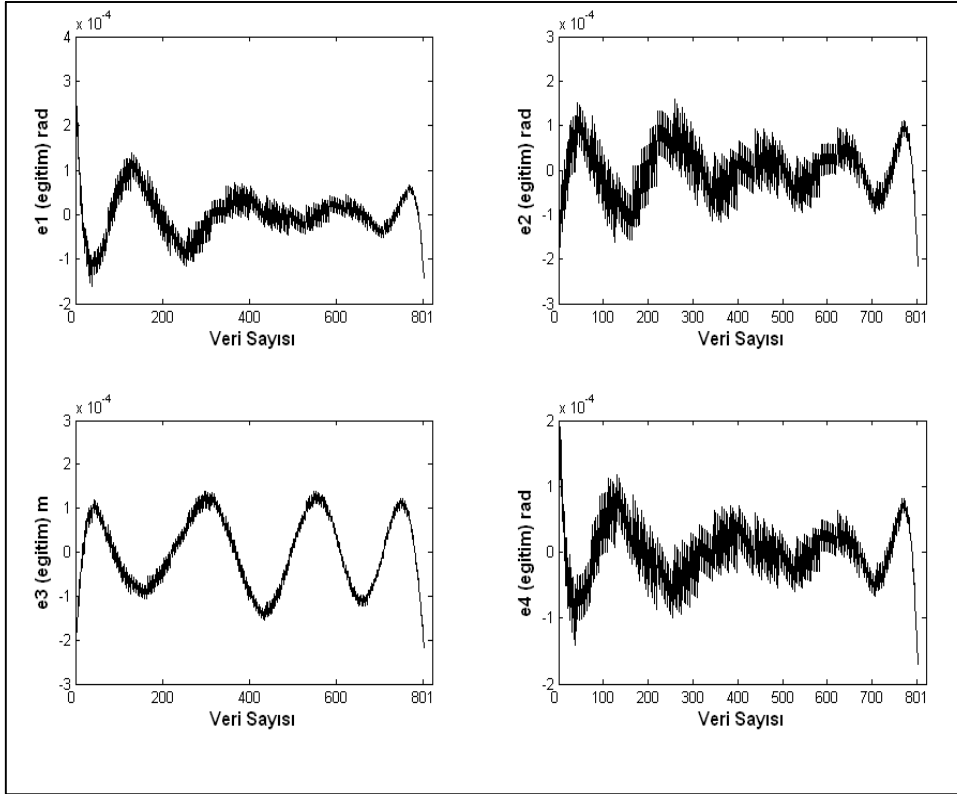


Şekil 5.6. 1., 2., 3. ve 4. eklemin istenen ve eğitim sonucu elde edilen konum değerleri

Modelin YSA' nda eğitilmesi sonucunda elde edilen çıkış değerleri iyi bir performans sonucu istenen çıkış değerlerine oldukça yakın olarak bulunmuştur.

Ayrıca istenen çıkış değerleri ve modelin YSA' nda eğitilmesi sonucunda elde edilen çıkış değerleri arasındaki konumlara ait hata grafikleri sırasıyla Şekil 5.7' de gösterilmiştir. Hata değerleri 5.3 nolu eşitlikteki gibi ifade edilmiştir [3-17];

$$\text{Hata} = \text{İstenen çıkış} - \text{YSA çıkışı} \quad (5.3)$$



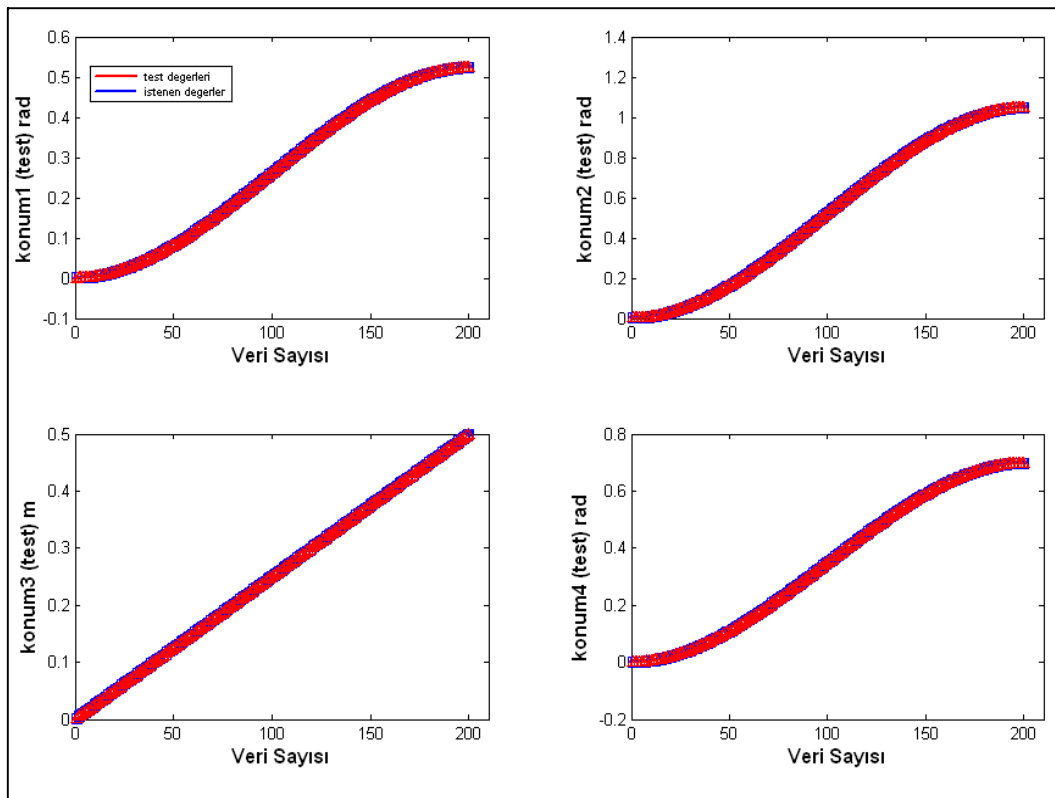
Şekil 5.7. 1., 2., 3. ve 4. eklem için istenen ve eğitim sonucu elde edilen konum değerleri arasındaki hata değerleri

YSA'nın eğitimi sonucunda Şekil 5.7'de de görüldüğü üzere hedeflenen çıkışlar ile YSA çıkışları arasındaki konumların hata değerleri; 1.eklem için maksimum hata değeri 2.432×10^{-4} rad, minimum hata değeri 1.094×10^{-7} rad, 2.eklem için maksimum hata değeri 2.154×10^{-4} rad, minimum hata değeri 3.456×10^{-7} rad, 3.eklem için maksimum hata değeri 2.166×10^{-4} m, minimum hata değeri 1.967×10^{-7} m, 4.eklem için maksimum hata değeri 1.892×10^{-4} rad, minimum hata değeri 3.117×10^{-7} rad olarak hesaplanmıştır.

5.2. Robotun Yapay Sinir Ağları ile Test Edilmesi Sonucu Çıkış ve Hata Grafiklerinin Elde Edilişi

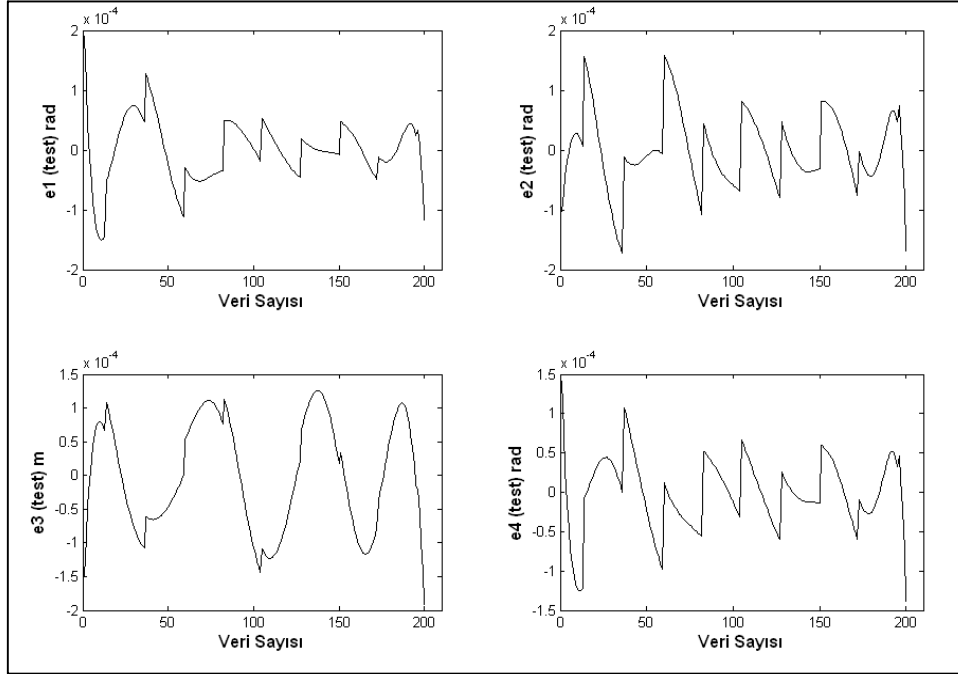
Eğitimi tamamlanan YSA' nın test edilmesi işlemi Matlab 7.0 programı yardımı ile yapılmıştır. Matlab komutlarıyla yazılan dosya Ek B' de verilmektedir. Şimdi de yazılan programın Matlab 7.0 programı ile çalıştırılması ile elde edilen sonuçlar incelenmektedir [17].

Eğitimi tamamlanan YSA' nın testi sonucunda elde edilen çıkış konum değerleri ve istenen çıkış konum değerlerinin birlikte gösterildiği grafikler Şekil 5.8' de görülmektedir. Şekil 5.8' de ki grafiklerde görünen mavi renkli çizim istenen değerler, kırmızı renkli çizim ise test sonucunda elde edilen değerlerdir.



Şekil 5.8. 1., 2., 3. ve 4. eklemin istenen ve test sonucu elde edilen konum değerleri

Modelin YSA' nda testi edilmesi sonucunda elde edilen çıkış değerleri iyi bir performans sonucu istenen çıkış değerlerine oldukça yakın olarak bulunmuştur.



Şekil 5.9. 1., 2., 3. ve 4. eklem için istenen ve testi sonucu elde edilen konum değerleri arasındaki hata değerleri

YSA' nın testi sonucunda Şekil 5.9' da da görüldüğü üzere hedeflenen çıkışlar ile YSA çıkışları arasındaki konumların hata değerleri; 1.eklem için maksimum hata değeri 1.91×10^{-4} rad, minimum hata değeri 1.59×10^{-7} rad, 2.eklem için maksimum hata değeri 1.727×10^{-4} rad, minimum hata değeri 2.248×10^{-7} rad, 3.eklem için maksimum hata değeri 1.906×10^{-4} m, minimum hata değeri 1.897×10^{-7} m, 4.eklem için maksimum hata değeri 1.408×10^{-4} rad, minimum hata değeri 6.17×10^{-7} rad olarak hesaplanmıştır.

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Bu çalışmada dört serbestlik dereceli SCARA tipi robota 4:11:4 yapısında bir YSA mimarisi uygulanmıştır. Elde edilen konum, hız ve ivme değerleri; Lagrange – Euler dinamik denklemlerinde yerine yazılarak tork değerleri belirlenmiştir. YSA' nın girişleri Lagrange – Euler dinamik denklemleri sonucu bulunan tork değerleri, çıkışları ise her bir eklem hareketi için elde edilen konum değerleri olarak alınmaktadır. Zaman aralığı 0' dan 10 saniye' ye kadar alınmış ve 0.01 artımlar ile 1001 adet konum ve tork değeri elde edilmiştir. Normalize edilen değerlerin 801 adedi eğitim seti, 200 adedi de test seti olarak belirlenmiştir. İlk olarak eğitim seti YSA' ya uygulanmıştır. Eğitim işlemi tamamlandıktan sonra test setindeki değerler ağa gösterilmiştir ve sonuçlar incelenmiştir.

Modelin YSA' nda eğitimi sonucunda, istenen değerler ile YSA eğitimi sonucu elde edilen değerler arasında çok düşük miktarda hatalar oluşmuştur. Böylece ağı eğitimi tamamladığı kabul edilmektedir.

YSA' nın eğitimi sonucunda hedeflenen çıkışlar ile YSA çıkışları arasındaki konumların hata değerleri; 1.eklem için maksimum hata değeri 2.432×10^{-4} rad, minimum hata değeri 1.094×10^{-7} rad, 2.eklem için maksimum hata değeri 2.154×10^{-4} rad, minimum hata değeri 3.456×10^{-7} rad, 3.eklem için maksimum hata değeri 2.166×10^{-4} m, minimum hata değeri 1.967×10^{-7} m, 4.eklem için maksimum hata değeri 1.892×10^{-4} rad, minimum hata değeri 3.117×10^{-7} rad olarak hesaplanmıştır.

Yapılan araştırmalara göre; 1 nolu kaynak olan 'Yapay Sinir Ağları ve Uygulamaları' adlı kitapta ve incelenen değişik literatür çalışmalarındaki hata

miktarları (genel olarak 10^{-2} ila 10^{-4} mertebesinde) sonuçların yorumlanması için referans alınmıştır [1-3-14-17].

Tablo 6.1' de hedeflenen çıkışlar ile YSA (eğitim) çıkışları arasındaki konumların hata değerleri verilmiştir. Derece ile radyan arasındaki matematiksel ilişki 6.1 nolu eşitlikte verilmiştir.

$$\text{Derece} = \text{Radyan} * 57.2958 \quad (6.1)$$

Tablo 6.1. Hedeflenen çıkış değerleri ile ysa çıkış değerleri arasındaki hata değerleri (eğitim)

	Minumum Hata Değeri	Maximum Hata Değeri
1. Eklem	0.000006 °	0.014 °
2. Eklem	0.00002 °	0.012 °
3. Eklem	2.166×10^{-4} m.	1.967×10^{-7} m.
4. Eklem	0.00002 °	0.011 °

Daha sonra eğitimi tamamlanan ağın hiç görmediği test setindeki değerler ağa gösterilerek, ağın verdiği cevaplar incelenmiştir. Ağın test setindeki hiç görmediği değerlere verdiği cevaplar ile istenen değerler arasında kabul edilebilir hatalar oluşmuştur. Yani istenen değerler ile YSA testi sonucu elde edilen değerler arasında çok düşük miktarda hatalar oluşmuştur.

Eğitimi tamamlanan YSA' nın testi sonucunda hedeflenen çıkışlar ile YSA çıkışları arasındaki konumların hata değerleri; 1.eklem için maksimum hata değeri 1.91×10^{-4} rad, minimum hata değeri 1.59×10^{-7} rad, 2.eklem için maksimum hata değeri 1.727×10^{-4} rad, minimum hata değeri 2.248×10^{-7} rad, 3.eklem için maksimum hata değeri 1.906×10^{-4} m, minimum hata değeri 1.897×10^{-7} m, 4.eklem için maksimum hata değeri 1.408×10^{-4} rad, minimum hata değeri 6.17×10^{-7} rad olarak hesaplanmıştır.

Yapılan arařtırmalara gre; 1 nolu kaynak olan ‘Yapay Sinir Ađları ve Uygulamaları’ adlı kitapta ve incelenen deđiřik literatr alıřmalarındaki hata miktarları (ortalama olarak 10^{-2} ila 10^{-4} mertebesinde) sonuların yorumlanması iin referans alınmıřtır [1-3-14-17].

Tablo 6.2’ de hedeflenen ıkıřlar ile YSA (test) ıkıřları arasındaki konumların hata deđerleri verilmiřtir (Derece=Radyan*57.2958).

Tablo 6.2. Hedeflenen ıkıř deđerleri ile ysa ıkıř deđerleri arasındaki hata deđerleri (test)

	Minumum Hata Deđerı	Maximum Hata Deđerı
1. Eklem	0.000009 °	0.011 °
2. Eklem	0.00001 °	0.01 °
3. Eklem	1.897×10^{-7} m.	1.906×10^{-4} m.
4. Eklem	0.00004 °	0.008 °

Elde edilen sonulardan da anlařılabileceđi gibi ađ iyi bir performans sergileyerek drt serbestlik dereceli SCARA robotun modellenmesi iřlemine iyi cevap vermiřtir ve tasarlanan ađ kullanılabilir bir durumdadır. Robot kontrol uygulamalarında dinamik modelin kullanılması gerektiđi durumlarda, bu alıřmadan elde edilen model, hataların az olmasından dolayı rahatlıkla kullanılabilir. İstendiđi takdirde bu modele uygun bir kontrol algoritmasının da seilmesi ile SCARA robotun kontrol gerekleřtirilebilir.

KAYNAKLAR

- [1] EFE M.Ö., KAYNAK O., “Yapay Sinir Ağları ve Uygulamaları”, Boğaziçi Üniversitesi, 2004.
- [2] EĞRİSÖĞÜT A., KAZAN R., “Çimento Hammadde Karışım Prosesinin Yapay Sinir Ağları ile Matematiksel Modellenmesi”, Makina Tasarım ve İmalat Teknolojileri Kongresi, 2-3 Kasım 2001, Konya, Türkiye.
- [3] KAZAN R., EĞRİSÖĞÜT A., “Scara Robot Dinamiğinin Yapay Sinir Ağları Kullanarak Modellenmesi”, Mühendis ve Makina Dergisi, 550. Sayı, Kasım 2005.
- [4] DAŞ M.T., DÜLGER L.C., “SCARA Tipi Robotun Programlanabilir Mantık Denetleyicisiyle (PLC) Kısmi Hareket Denetimi”, 11. Ulusal Makina Teorisi Sempozyumu, 4-6 Eylül 2003, Ankara, Türkiye.
- [5] KOCABIÇAK Ü., “Bilgisayar Destekli Tasarım ve İmalat – Ders Notları”, Sakarya Üniversitesi, Mühendislik Fakültesi, 2005.
- [6] KAZAN R., “Robotik – Ders Notları”, Sakarya Üniversitesi, Mühendislik Fakültesi, 2004.
- [7] İNAL E.P., “Endüstriyel Robotlar ve Uygulama Alanları”, Balıkesir Üniversitesi, Makine Mühendisliği Bölümü, 2006.
- [8] Boğaziçi Robot Sitesi, “Robotların Tarihçesi, 2. Kilometre Taşları”. <http://www.teknoturk.org/docking/yazilar/tt000052-yazi.htm>
- [9] Boğaziçi Üniversitesindeki Robot Araştırmaları, “Geleceğe giden yol”. <http://www.teknoturk.org/docking/yazilar/tt000052-yazi.htm>
- [10] ÇENGELCİ B., ÇİMEN H., “Endüstriyel Robotlar, Makine Teknolojileri”, Elektronik Dergisi, Teknik Not, 2005.
- [11] EMAS Ç., “Yapay Sinir Ağları (Kuram, mimari, eğitim, uygulama)”, Seçkin Yayıncılık, 2003.
- [12] ÖZTEMEL E., “Yapay Sinir Ağları”, Papatya Yayıncılık, 2006.

- [13] SAĞIROĞLU Ş., “Yapay Sinir Ağları ve Mühendislik Uygulamaları”.
<http://muhendislik.erciyes.edu.tr/tr/bilgisayar/bilakad/ss/index.htm>
- [14] YILDIRIM Ş., ERKAYA S., UZMAY İ., “Yapay Sinir Ağları Kullanarak Concorde Uçaklarının Kontrolü”, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 20 (1-2), 1-12, 2004.
- [15] LEWIS, F.L., ABDALLAH, C.T., DAWSON, D.M., “Control of Robot Manipulators”, 1993.
- [16] İBRAHİM D., “Öğrenciler İçin Matlab”, Bileşim Yayınevi, 2003.
- [17] ARİFOĞLU U., “MATLAB 7.04 Simulink ve Mühendislik Uygulamaları”, ALFA Basım Yayım Dağıtım Ltd. Şti., 2005.
- [18] ÇALIŞKAN M., “İki Uzunlu Bir Robot Kolunun Hesaplanmış Moment Metoduyla Kontrolü”, Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, 1996.
- [19] SCHILLING R.J., “Fundamentals of Robotics Analysis and Control”, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
- [20] DELİBAŞ A., “Üç Serbestlik Dereceli Anthropomorphic Robot Kolun Modellenmesi ve Ters Dinamik Kontrolü”, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2004.
- [21] DAS M.T., DÜLGER L.C., “Mathematical Modelling, Simulation and Experimental Verification of A Scara Robot”, Gaziantep Üniversitesi, Makine Mühendisliği Bölümü, 2005.
- [22] SEZGİN İ., “Bir Robotik Manipulatörün Dinamik Analizi ve Simülasyonu”, Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, 2004.
- [23] YILDIRIM Ş., “Artificial Neural Network Applications to Control”, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 18 (1-2), 1-11, 2002.
- [24] HAYKİN S., “Neural Networks, Bölüm1”, Macmillan College Printing Company, New Jersey, 1994.
- [25] FLETCHER R., “Unconstrained Optimization, Practical Methods of Optimization, 1”, John Wiley&Sons, ISBN 0-471-27711-8, 1980.
- [26] HAGAN M.T., DEMUTH H.B., BEALE M., “Neural Network Design”, PWS Publishing Company, Boston, 1995.

EKLER

- Ek A : Robotun konum ve tork grafiklerinin elde edildiđi Matlab 7.0 komutları ile yazılan m-file dosyası
- Ek B : Robotun yapay sinir ađları ile modellenmesi için Matlab 7.0 komutları ile yazılan m-file dosyası

Ek A : Robotun konum ve tork grafiklerinin elde edildiği Matlab 7.0 komutları ile yazılan m-file dosyası

```

%%%%%%%%%%
% Kontrol Zaman Sınırı ve Zaman Sekmesi
t=0:0.01:10;
% Yörünge Katsayıları
a=-0.001048;
b=0.015714;
f=-0.002095;
h=0.031429;
m=-0.001397;
n=0.020952;
c=0;
e=0;
k=0;
d=0;
p=0;
z=0;
v=0.05;
y=0;
% Yörünge Denklemleri
Q1=a*t.^3+b*t.^2+c*t+e;
Q1T=3*a*t.^2+2*b*t+c;
Q1TT=6*a*t+2*b;
Q2=f*t.^3+h*t.^2+k*t+d;
Q2T=3*f*t.^2+2*h*t+k;
Q2TT=6*f*t+2*h;
Q3=v*t+y;
s=1:1:1001;
Q3T(s)=v;
Q3TT(s)=0;
Q4=m*t.^3+n*t.^2+p*t+z;

```

```

Q4T=3*m*t.^2+2*n*t+p;
Q4TT=6*m*t+2*n;
% Tork İçin Gerekli Değerler
M1=2;
M2=2;
M3=1;
M4=0.5;
L1=2;
L2=1;
g=9.8;
IZZ1=8;
IZZ2=2;
IZZ3=0.25;
IZZ4=0.005;
C2=cos(Q2);
S2=sin(Q2);
% Tork Denklemleri
T11=IZZ1+2*IZZ2+2*IZZ3+IZZ4*3;
T12=((M1/4)+2*M3+M4)*L1^2;
T13=C2.*((M2/2)+5*M3+2*M4)*L1*L2;
T14=((M2/2)+2*M3+2*M4)*L2^2;
T15=Q1T.^2*((M2/2)-M3)*L1*L2;
T16=Q1T.*Q2T.*(4*M3+2*M4)*L1*L2;
%
T21=2*IZZ2+2*IZZ3+IZZ4*3;
T22=((M2/2)+2*M3+2*M4)*L2^2;
T23=C2.*(2*M3+M4)*L1*L2;
T24=S2.*(Q1T.^2)*((M2/2)+M3+M4)*L1*L2;
%
T31=M3*M4;
T32=M3*g+M4*g;

```

```

%
TO1=Q1TT.*(T11+T12+T14)+Q1TT.*T13+Q2TT.*(T11+T12+T14)+Q2TT.*T13+
Q3TT.*(T11+T12+T14)+Q3TT.*T13+Q4TT.*(T11+T12+T14)+Q4TT.*T13+T15-
T16;
TO2=Q1TT.*(T21+T22)+Q1TT.*T23+Q2TT.*(T21+T22)+Q2TT.*T23+Q3TT.*(T2
1+T22)+Q3TT.*T23+Q4TT.*(T21+T22)+Q4TT.*T23+T24;
TO3=Q1TT.*T31+Q2TT.*T31+Q3TT.*T31+Q4TT.*T31+T32;
TO4=Q1TT.*IZZ4*3+Q2TT.*IZZ4*3+Q3TT.*IZZ4*3+Q4TT.*IZZ4*3;

% Grafiklerin Elde Edilmesi
subplot(4,4,1),plot(t,Q1),title('1.Eklemin Konumu'),ylabel('Q1'),grid;
subplot(4,4,2),plot(t,Q1T),title('1.Eklemin Hizi'),ylabel('Q1T'),grid;
subplot(4,4,3),plot(t,Q1TT),title('1.Eklemin ivmesi'),ylabel('Q1TT'),grid;
subplot(4,4,5),plot(t,Q2),title('2.Eklemin Konumu'),ylabel('Q2'),grid;
subplot(4,4,6),plot(t,Q2T),title('2.Eklemin Hizi'),ylabel('Q2T'),grid;
subplot(4,4,7),plot(t,Q2TT),title('2.Eklemin ivmesi'),ylabel('Q2TT'),grid;
subplot(4,4,9),plot(t,Q3),title('3.Eklemin Konumu'),ylabel('Q3'),grid;
subplot(4,4,10),plot(t,Q3T),title('3.Eklemin Hizi'),ylabel('Q3T'),grid;
subplot(4,4,11),plot(t,Q3TT),title('3.Eklemin ivmesi'),ylabel('Q3TT'),grid;
subplot(4,4,13),plot(t,Q4),title('4.Eklemin Konumu'),ylabel('Q4'),grid;
subplot(4,4,14),plot(t,Q4T),title('4.Eklemin Hizi'),ylabel('Q4T'),grid;
subplot(4,4,15),plot(t,Q4TT),title('4.Eklemin ivmesi'),ylabel('Q4TT'),grid;
subplot(4,4,4),plot(t,TO1),title('Tork1'),ylabel('TO1'),grid;
subplot(4,4,8),plot(t,TO2),title('Tork2'),ylabel('TO2'),grid;
subplot(4,4,12),plot(t,TO3),title('Tork3'),ylabel('TO3'),grid;
subplot(4,4,16),plot(t,TO4),title('Tork4'),ylabel('TO4'),grid;

% Verilerin transpozmesi(satır matrisiyken sütun matrisi haline getirilmesi)
D1=(Q1)';
D2=(Q2)';
D3=(Q3)';
D4=(Q4)';
D1T=(Q1T)';

```

```

D2T=(Q2T)';
D3T=(Q3T)';
D4T=(Q4T)';
D1TT=(Q1TT)';
D2TT=(Q2TT)';
D3TT=(Q3TT)';
D4TT=(Q4TT)';
Tork1=(TO1)';
Tork2=(TO2)';
Tork3=(TO3)';
Tork4=(TO4)';

```

```

% Tüm matrisleri yan yana sütunlar halinde gösterebilmek için
veri1=[Tork1 Tork2 Tork3 Tork4 D1 D2 D3 D4];
save veri1.dat veri1 /ascii

```

```

% Elde Edilen Matrisin Normalizasyon İşlemi
A=min(veri1);
B=min(A)
C=max(veri1);
E=max(C)
%
veri2=((veri1-B)/(E-B))*(0.9-0.1)+0.1;
save norveri2.dat veri2 /ascii

```

```

% "norveri2.dat" dosyasındaki verilerin 801'tanesi "egitim.dat", 200'tanesi
% de "test.dat" olarak YSA'da kullanılmak üzere ayrılarak kaydedildi!
%%%%%%%%%

```

Ek B : Robotun yapay sinir ağı ile modellenmesi için Matlab 7.0 komutları ile yazılan m-file dosyası

```

%%%%%%%%%%
load egitim.dat
load test.dat
load veri1.dat

% Eğitim için giriş ve çıkış matrislerinin oluşturulması
U=egitim;
Q=size(U,1);
time=1:1:Q;
% Eğitim giriş matrisleri
a=egitim(:,1);
b=egitim(:,2);
c=egitim(:,3);
d=egitim(:,4);
% Eğitim Çıkış Matrisleri
h=egitim(:,5);
k=egitim(:,6);
m=egitim(:,7);
n=egitim(:,8);
% Eğitim Giriş-Çıkış matrislerinin Sütun Matrisi haline getirilmesi.
F=[a b c d];
F=(F)';
Y=[h k m n];
Y=(Y)';

% 11: Gizli nöron
% 4: Çıkış nöronu
%'logsig': Gizli Nöronlar ve Çıkış Nöronları için SİGMOİD fonksiyonunu ifade eder
%'trainlm': Geri dönüşüm için kullanılan fonksiyon
%'learnngdm': Öğrenme için kullanılan fonksiyon

```



```

%'mse': Eğitimin yapılması sırasındaki hataları gösterir
nnet=newff(minmax(F),[11 4],{'logsig' 'logsig'},'trainlm','learnngdm','mse');
% n: Grafikteki "epochs" değerinin gösterim aralıkları
%nnet.trainParam.show=n;
nnet.trainParam.show=100;
% Epoks çevrim sayısı(Her değer için m'kez dener)
%nnet.trainParam.epochs=m;
nnet.trainParam.epochs=1000;
% Ulaşılması hedeflenen hata değeri
nnet.trainParam.goal=(0.000000000001);

% YSA'ile giriş-çıkış arasında ilişki kurularak eğitiminin gerçekleştirilmesi
mnet=train(nnet,F,Y);
save ysa.mat mnet
% Eğitim Sonucunda YSA'nın bulduğu değerler.
nnnet=sim(mnet,F);
nnnet=(nnnet)';

% Değerlerin Ters Normalize edilmesi.
E1=min(veri1);
S1=min(E1);
O1=max(veri1);
L1=max(O1);
veri3=((egitim-0.1)/(0.9-0.1))*(L1-S1)+S1;
% Ters Normalize Edilmiş değerlerin yeni bir dosyada saklanması.
save gercekegitim.dat veri3 /ascii;
% Elde edilen değerlerin çıkış sütunlarının alınması.
tk1=veri3(:,5);
rk1=veri3(:,6);
yk1=veri3(:,7);
zk1=veri3(:,8);

```

```

% Eğitim değerlerinin Ters Normalize edilmesi.
veri4=((nnnet-0.1)/(0.9-0.1))*(L1-S1)+L1;
% Ters Normalize Edilmiş değerlerin yeni bir dosyada saklanması.
save egitimcikis.dat veri4 /ascii;
% Elde edilen değerlerin 4 sütun haline getirilmesi.
tk2=veri4(:,1);
rk2=veri4(:,2);
yk2=veri4(:,3);
zk2=veri4(:,4);

% Eğitim İçin Hata Değerleri
% e=Hata
e1=tk1-tk2;
e2=rk1-rk2;
e3=yk1-yk2;
e4=zk1-zk2;

% Eğitim grafikleri
% pause: Giriş sinyali grafiklerini görmek için
% herhangi bir tuşa basarak görmeye yarar.
% clf : Grafik penceresini muhafaza eder ve bir tuşa basıldığında sadece içeriği
% silerek varsayeni grafiklere ortam hazırlar.

pause
clf
subplot(2,2,1);
plot(time,tk1,'bs:',time,tk2,'r^-');
xlabel('Data Number');
ylabel('konum1(egitim)');

pause
subplot(2,2,2);
plot(time,rk1,'bs:',time,rk2,'r^-');

```

```
xlabel('Data Number');  
ylabel('konum2(egitim)');
```

```
pause  
subplot(2,2,3);  
plot(time,yk1,'bs:',time,yk2,'r^-');  
xlabel('Data Number');  
ylabel('konum3(egitim)');
```

```
pause  
subplot(2,2,4);  
plot(time,zk1,'bs:',time,zk2,'r^-');  
xlabel('Data Number');  
ylabel('konum4(egitim)');
```

% Eğitim İçin Hata Grafikleri

```
pause  
subplot(2,2,1);  
plot(time,e1,'r^-');  
xlabel('Data Number');  
ylabel('e1(egitim)');
```

```
pause  
subplot(2,2,2);  
plot(time,e2,'r^-');  
xlabel('Data Number');  
ylabel('e2(egitim)');
```

```
pause  
subplot(2,2,3);  
plot(time,e3,'r^-');  
xlabel('Data Number');  
ylabel('e3(egitim)');
```

```
pause  
subplot(2,2,4);
```

```

plot(time,e4,'r^-');
xlabel('Data Number');
ylabel('e4(egitim)');

%Test için giriş ve çıkış matrislerinin oluşturulması
V=test;
P=size(V,1);
time1=1:1:P;

%Test giriş matrisleri
A=test(:,1);
B=test(:,2);
C=test(:,3);
D=test(:,4);
% Test Çıkış Matrisleri
H=test(:,5);
K=test(:,6);
M=test(:,7);
N=test(:,8);
% Test Giriş-Çıkış matrislerinin Sütun Matrisi haline getirilmesi.
J=[A B C D];
J=(J)';
G=[H K M N];
G=(M)';
%Test Sonucunda YSA'nın bulduğu değerler.
uunet=sim(nnet,J);
uunet=(uunet)';

veri5=((test-0.1)/(0.9-0.1))*(L1-S1)+L1;
%Ters Normalize Edilmiş değerlerin yeni bir dosyada saklanması.
save gercektest.dat veri5 /ascii;

```

```

%Elde edilen deęerlerin ıkıř stnlarının alınması.
fk1=veri5(:,5);
gk1=veri5(:,6);
jk1=veri5(:,7);
lk1=veri5(:,8);

%Test deęerlerinin Ters Normalize edilmesi.
veri6=((uunet-0.1)/(0.9-0.1))*(L1-S1)+L1;
%Ters Normalize Edilmiř deęerlerin yeni bir dosyada saklanması.
save testcikis.dat veri6 /ascii;
%Elde edilen deęerlerin 4 stn haline getirilmesi.
fk2=veri6(:,1);
gk2=veri6(:,2);
jk2=veri6(:,3);
lk2=veri6(:,4);

% Test İin Hata Deęerleri
% e=Hata
e1=fk1-fk2;
e2=gk1-gk2;
e3=jk1-jk2;
e4=lk1-lk2;

% Test Grafikleri
pause
clf

pause
subplot(2,2,1);
plot(time1,fk1,'bs:',time1,fk2,'r^-')
xlabel('Data Number');
ylabel('konum1(test)');

```

```
pause
subplot(2,2,2);
plot(time1,gk1,'bs:',time1,gk2,'r^-')
xlabel('Data Number');
ylabel('konum2(test)');
```

```
pause
subplot(2,2,3);
plot(time1,jk1,'bs:',time1,jk2,'r^-')
xlabel('Data Number');
ylabel('konum3(test)');
```

```
pause
subplot(2,2,4);
plot(time1,lk1,'bs:',time1,lk2,'r^-')
xlabel('Data Number');
ylabel('konum4(test)');
```

```
% Test İçin Hata Grafikleri
```

```
pause
subplot(2,2,1);
plot(time1,e1,'bs:');
xlabel('Data Number');
ylabel('e1(test)');
```

```
pause
subplot(2,2,2);
plot(time1,e2,'bs:');
xlabel('Data Number');
ylabel('e2(test)');
```

```
pause
subplot(2,2,3);
plot(time1,e3,'bs:');
xlabel('Data Number');
```

```
ylabel('e3(test)');  
pause  
subplot(2,2,4);  
plot(time1,e4,'bs:');  
xlabel('Data Number');  
ylabel('e4(test)');
```

```
echo off
```

```
%%%%%%%%%%
```

ÖZGEÇMİŞ

Selçuk Öztürk, 18.10.1979 yılında İstanbul’ da doğdu. Yeniköy İlkokulundan 1990 senesinde, Özdemir Sabancı Emirgan Ortaokulundan 1993 senesinde, Behçet Kemal Çağlar Lisesinin süper lise kısmından da 1997 senesinde mezun oldu. 1997 yılının güz dönemi ile birlikte Erciyes Üniversitesinde lisans eğitimine başladı. 1 sene İngilizce eğitimi alıp, tamamlamaya hak kazandıktan sonra Mühendislik Fakültesi, Makine Mühendisliği bölümünde mesleki eğitimine başladı. 2003 yılında eğitimini tamamlayarak mezun olmaya hak kazandı. Kısa bir süre İstanbul’ da çalıştıktan sonra askerlik görevini tamamlamak için kısa dönem olarak Hatay’ a gitti. 2004 yılında askerlik görevini tamamladı. 2004 yılında Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Makine Mühendisliği anabilim dalı, Makine Tasarım ve İmalat bilim dalında yüksek lisans eğitimine başladı. Halen bu eğitime devam etmektedir.