

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

**ÜÇ FAZLI KAFESLİ ASENKRON MOTORLARDA
TOPLAM HARMONİK DİSTORSİYONUNUN YAPAY SİNİR
AĞLARI YÖNTEMİ KULLANILARAK SAPTANMASI**

Hıdır Selçuk NOĞAY

**DOKTORA TEZİ
ELEKTRİK EĞİTİMİ ANABİLİM DALI
ELEKTRİK EĞİTİMİ PROGRAMI**

**DANIŞMAN
Yrd. Doç. Dr. Yaşar BİRBİR**

İSTANBUL 2008

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÜÇ FAZLI KAFESLİ ASENKRON MOTORLARDA
TOPLAM HARMONİK DİSTORSİYONUNUN YAPAY SİNİR
AĞLARI YÖNTEMİ KULLANILARAK SAPTANMASI

Hıdır Selçuk NOĞAY
(141201220020092)

DOKTORA TEZİ
ELEKTRİK EĞİTİMİ ANABİLİM DALI
ELEKTRİK EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Yaşar BİRBİR

İSTANBUL 2008

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
KABUL VE ONAY BELGESİ

**ÜÇ FAZLI KAFESLİ ASENKRON MOTORLARDA TOPLAM
HARMONİK DİSTORSİYONUNUN YAPAY SİNİR AĞLARI
YÖNTEMİ KULLANILARAK SAPTANMASI**

Hıdır Selçuk NOĞAY' ın "ÜÇ FAZLI KAFESLİ ASENKRON MOTORLARDA TOPLAM HARMONİK DİSTORSİYONUNUN YAPAY SİNİR AĞLARI YÖNTEMİ KULLANILARAK SAPTANMASI" isimli Lisansüstü tez çalışması, M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih vesayılı kararı ile oluşturulan jüri tarafından Elektrik Eğitimi Anabilim Dalı Elektrik Eğitimi Programında DOKTORA Tezi olarak Kabul edilmiştir.

Danışman : Yrd.Doç.Dr. Yaşar BİRBİR Marmara Üniversitesi
Üye : Prof.Dr. Asım KASAPOĞLU Yıldız Teknik Üniversitesi.....
Üye : Doç. Dr. Fevzi BABA Marmara Üniversitesi
Üye : Yrd.Doç.Dr. Hasan ERDAL Marmara Üniversitesi
Üye : Yrd.Doç.Dr. Caner AKÜNER Marmara Üniversitesi

Tezin Savunulduğu Tarih : 08.05.2008

ONAY

M.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararı ile Hıdır Selçuk NOĞAY'ın Elektrik Eğitimi Anabilim Dalı Elektrik Eğitimi Programında Doktora (Dr, PhD.) derecesi alması onanmıştır.

Marmara Üniversitesi
Fen Bilimleri Enstitüsü Müdürü

TEŐEKKÜR

Tezin hazırlanması sırasında bilimsel katkıları, emeđi ve tecrübesi ile bana her konuda yardımcı olan danışman hocam Sayın Yrd. Doç. Dr. Yaşar BİRBİR'e en içten teşekkür ve saygılarımı sunarım. Araştırmalarım süresince, bilgi ve deneyimlerinden yararlandığım Sayın Yrd. Doç. Dr. Vedat TOPUZ' a ve Sayın Yrd. Doç. Dr. Fevzi KENTLİ' ye , bu çalışmanın oluşumu için büyük çaba harcayan Sayın Emekli Öğretim Görevlisi Fahri ERDİNÇ 'e ayrı ayrı teşekkür ederim.

Bu tez çalışması, Marmara Üniversitesi Bilimsel Araştırma Projeleri Komisyonu Başkanlığı tarafından desteklenerek gerçekleştirilmiştir. Proje numarası FEN-DKR-151105-0226' dır. Marmara Üniversitesi Bilimsel Araştırma Projeleri Komisyonu Başkanlığına ve personeline teşekkürlerimi sunarım.

Deneysel çalışmamın oluşumunda tezimi destekleyen, bana tavsiyeleri ve destekleri ile yardımcı olan bütün hocalarıma ve mesai arkadaşlarıma teşekkür ederim. Özellikle de bu yoğun çalışma temposunda gösterdiği sabır ve anlayış için sevgili eşime ve aileme teşekkürlerimi sunarım.

Mart 2008

Hıdır Selçuk NOĞAY

İÇİNDEKİLER

SAYFA NO

TEŞEKKÜR	i
İÇİNDEKİLER	ii
ÖZET	vii
ABSTRACT	viii
YENİLİK BEYANI	ix
SEMBOL LİSTESİ	xi
KISALTIMA LİSTESİ	xiv
ŞEKİLLER LİSTESİ	xv
TABLolar LİSTESİ	xviii
I. GİRİŞ VE AMAÇ	1
I.1. GİRİŞ	1
I.2. AMAÇ	2
II. YAPAY SİNİR AĞLARI	4
II.1. AKIL VE ZEKA	4
II.1.1. Yapay Zeka	4
II.1.2. Yapay Sinir Ağları	5
II.1.3. Öğrenme	7
II.1.3.1. Öğrenmenin Psikolojik Temelleri	7
II.1.3.2. Bilgisayarlı Öğrenme	10
II.2. YAPAY SİNİR AĞLARININ YAPISI VE TEMEL ELEMENLARI	13
II.2.1. Biyolojik Bir Beyin Sinir Hücresinin Yapısı	13

II.2.2.	Yapay Sinir Hücresi (İşlem Elemanı)	16
II.2.2.1.	Girişler	17
II.2.2.2.	Ağırlıklar	18
II.2.2.3.	Toplama Fonksiyonu	18
II.2.2.4.	Aktivasyon Fonksiyonu	18
II.2.2.5.	Ölçekleme ve Sınırlama	22
II.2.2.6.	Hücrenin Çıkışı	22
II.3.	YAPAY SİNİR AĞLARIN SINIFLANDIRILMASI	23
II.3.1.	Mimari Yapılarına Göre Sınıflandırılması	23
II.3.1.1.	İleri Beslemeli Ağlar	24
II.3.1.2.	Geri Beslemeli Ağlar	24
II.3.2.	Öğrenme Yaklaşımlarına Göre Sınıflandırılması	25
II.3.2.1.	Danışmanlı Öğrenme (Supervised Learning)	26
II.3.2.2.	Danışmansız Öğrenme (Unsupervised Learning)	26
II.3.2.3.	Takviyeli öğrenme (Reinforcement learning)	27
II.3.3.	Öğrenme Kurallarına Göre Sınıflandırılması	28
II.3.3.1.	Çevrimiçi (On-line) öğrenme kuralları	28
II.3.3.2.	Çevrimdışı (Off-line) öğrenme kuralları	28
II.4.	ÖĞRENME KURALLARI	28
II.4.1.	Hebb Kuralı	28
II.4.2.	Hopfield Kuralı	29
II.4.3.	Delta Kuralı	29
II.4.4.	Kohenen Kuralı	29
II.5.	ÖĞRENME ALGORİTMALARI	29
II.5.1.	Geri Yayılım Öğrenme Algoritması	30
II.5.1.1.	İleri hesaplama	34
II.5.1.2.	Geriye hesaplama	34
II.5.1.3.	Ağırlıkların ve Dengelemelerin Dağılımı	38
II.5.2.	Levenberg- Marquardt Algoritması (LMBP)	39
II.6.	ÜSTÜNLÜKLERİ VE DEZAVANTAJLARI	45
II.6.1.	Üstünlükleri	45
II.6.2.	Dezavantajları	45

III. ASENKRON MOTORLAR.....	48
III.1. SİNCAP KAFESLİ ASENKRON MOTORLAR.....	48
III.1.1. Stator Sargı Yapıları	50
III.1.1.1. Dört Kutuplu Dengeli El Tipi Sargı.....	51
III.1.1.2. Tam Kalıp Sargı (İki Tabakalı).....	52
III.1.2. Stator Sargılarında Endüklenen Gerilim ve Harmonikler.....	54
III.1.2.1. Periyodik Fonksiyonların Fourier Serileri ile Gösterilişi.....	54
III.1.2.2. Bir Sargı İletkeninde Endüklenen Gerilim	55
III.1.2.3. Bir Çap Bobininde Endüklenen Gerilim	56
III.1.2.4. Krişlenmiş Bir Bobinde Endüklenen Gerilim.....	56
III.1.2.5. Endüviye Yayılmış Bir Sargıda Endüklenen Gerilim.....	57
III.1.2.6. Kirişlenme ile Bazı Harmoniklerin Yok Edilmesi.....	58
III.2. ASENKRON MOTORLARIN KONTROLÜ.....	59
III.2.1.1. Skalar Kontrol Yöntemleri	59
III.2.2. Sinüzoidal Olmayan Beslemenin Asenkron Motora Etkisi.....	60
III.2.2.1. Motor Harmonik Akımları.....	61
III.2.2.2. Harmonik Kayıplar.....	62
III.2.2.3. Moment Salınımları	63
III.2.3. Darbe Genişlik Modülasyonu (PWM)	63
III.2.3.1. Sinüzoidal PWM (SPWM)	63
IV. PROGLAMLANABİLİR LOJİK KONTROLÖR (PLC).....	66
IV.1. YAPISI.....	66
IV.1.1. Giriş Arabirimi	67
IV.1.1.1. Sayısal Giriş Modülü.....	68
IV.1.1.2. Analog Giriş Modülü	69
IV.1.2. Merkezi İşlem Birimi	70
IV.1.2.1. Diyagnostikler.....	71
IV.1.3. Bellek	73
IV.1.4. Programlama Aletleri.....	74
IV.1.5. Çıkış Arabirimi	74
IV.1.5.1. Sayısal Çıkış Modülü	75
IV.1.5.2. Analog Çıkış Modülü	75

IV.1.6.	İletişim Modülleri	76
IV.1.7.	Besleme Kaynağı	76
IV.1.8.	Özel Giriş / Çıkış Modülleri	77
IV.1.8.1.	Termokopul Giriş Modülü	77
IV.1.8.2.	Haberleşme Modülleri	77
IV.2.	KULLANIM AVANTAJLARI	78
V.	TEZ ÇALIŞMASI	79
V.1.	DENEYSEL ÇALIŞMA İÇİN İZLENEN YOL	79
V.2.	ASENKRON MOTORLARIN HAZIRLANMASI	80
V.3.	DENEY DÜZENİĞİNİN KURULMASI	83
V.3.1.	Deney Setinin İç Bağlantısı	84
V.3.2.	Deney Bağlantı Şeması	85
V.4.	VERİ SETİNİN OLUŞTURULMASI	86
V.4.1.	Güç Analizöründen Bilgisayara Veri Transferi	87
V.4.2.	Motor İç ve Dış Sıcaklıklarının Elde Edilmesi	92
V.5.	YAPAY SİNİR AĞI YÖNTEMİ İLE SİSTEMİN MODELLEMESİ	94
V.5.1.	Sistemin Parametrelerinin Belirlenmesi	94
V.5.2.	Sistemin Oluşturulması	96
V.5.3.	Yapay Sinir Ağı Modellerine Ait Elde Edilen Sonuçlar	99
V.5.3.1.	Dakikalık Akım THD Çıkışlı YSA Modeli Sonuçları	100
V.5.3.2.	Dakikalık Gerilim THD Çıkışlı YSA Sonuçları	106
V.5.3.3.	Saniyelik Akım THD Çıkışlı YSA Modeli Sonuçları	111
V.5.3.4.	Saniyelik Gerilim THD Çıkışlı YSA Modeli Sonuçları	116
VI.	SONUÇLAR VE TARTIŞMA	122
	KAYNAKLAR	127
	EKLER	133
	EK – A	134

EK – B.....	138
EK – C.....	141
EK – D.....	144
EK – E.....	147
ÖZGEÇMİŞ	150

ÖZET

ÜÇ FAZLI KAFESLİ ASENKRON MOTORLARDA TOPLAM HARMONİK DİSTORSİYONUNUN YAPAY SİNİR AĞLARI (YSA) YÖNTEMİ KULLANILARAK SAPTANMASI

Günümüzde asenkron motorlar gerek enerji tasarrufu sağlamak ve gerekse kolay kontrol imkanına sahip olmak için genellikle sürücü sistemler ile birlikte kullanılmaktadırlar. Sürücülerde kullanılan PWM inverter, her anahtarlama anında asenkron motorun besleme geriliminde harmoniklere sebep olmaktadır. Asenkron motorlarda stator gerilimi harmonikleri stator sargı yapısına ve adımına göre değişmektedir. Asenkron motorlarda stator sargıları kısaltılmış adımlı olarak sarıldıklarında bazı düşük seviye harmonikler bastırılmaktadır.

Bu çalışmada asenkron motorlarda toplam harmonik distorsiyonunu tahmin edebilen, farklı veri setleri ile eğitilmiş olan ve çıkışları akım ve gerilim toplam harmonik distorsiyonu olan, dört adet YSA modeli oluşturulmuştur

Yapay sinir ağından elde edilen sonuçlar ile deneysel çalışmadan elde edilen sonuçlar kıyaslanarak her sonuç için yaklaşım eğrileri çıkartılmış ve yapay sinir ağı sonuçlarının test sonuçlarına son derece yakın olduğu anlaşılmıştır.

Anahtar Kelimeler : Öğrenme, Eğitim, Yapay Sinir Ağları, Asenkron Motorlar, Stator Sargıları, AC Sürücü, Toplam Harmonik Distorsiyonu, Harmonik Tahmini.

Mart, 2008

Hıdır Selçuk, NOĞAY

ABSTRACT

ESTIMATION OF TOTAL HARMONIC DISTORTION IN THREE PHASE SQUIRREL CAGE INDUCTION MOTORS USING ARTIFICIAL NEURAL NETWORK (ANN) METHOD

Today in industry, squirrel cage induction motors are generally used with AC drive systems to save energy and to obtain easy control. PWM inverters used in AC drives systems give rise to harmonics in supply voltage of induction motors at each switching times. Stator voltage harmonics of induction motors vary depending on the construction of stator coil and pitches. If stator coil pitches are chorded, then some low order odd harmonics are suppressed in induction motors.

In this study, four ANN models are developed to predict total harmonics distortion in induction motors. The models, with current and voltage total harmonic distortion outputs are trained with different data sets and built.

Regression graphs for all results were obtained by comparing the results obtained from ANN models to those from the experimental study. The regression analysis show that ANN results agree well with the experimental results.

Key Words : Learning, Training, Artificial Neural Network, Induction Motors, Stator Winding, AC Drives, Total Harmonic Distortion, Harmonic Estimation.

March, 2008

Hıdır Selçuk, NOĞAY

YENİLİK BEYANI

ÜÇ FAZLI KAFESLİ ASENKRON MOTORLARDA TOPLAM HARMONİK DİSTORSİYONUNUN YAPAY SİNİR AĞLARI (YSA) YÖNTEMİ KULLANILARAK SAPTANMASI

Endüstride harcanan elektrik enerjisinin büyük bir kısmı asenkron motorlar tarafından tüketilmektedir. Günümüzde asenkron motorlar sanayide genellikle sürücüler ile çalıştırılmaktadırlar. Asenkron motorlar sürücüler ile birlikte kullanıldıklarında PWM inverter sebebiyle şebekede gerilim ve akım harmoniklerini arttırmaktadırlar. Harmoniklerin artmasının diğer bir sebebi ise motor sargı yapısı ve adımındır. Asenkron motorlarda stator sargıları kısaltılmış adımlı olarak sarıldıklarında bazı düşük seviye harmonikler bastırılmaktadır.

Bu çalışmada, üç fazlı kafesli asenkron motorların şebekeden çektiği akım ve uçlarındaki gerilim toplam harmonik distorsiyonlarını yapay sinir ağı yöntemiyle belirleyen, öğrenebilen, öğrenim sonucu üç fazlı, kafesli asenkron motorların herhangi çalışma parametrelerinde, herhangi sargı adımında ve herhangi PWM taşıyıcı frekansında matematiksel modele ihtiyaç duymadan şebekeye verdikleri ilk otuz harmonikten oluşan toplam harmonik distorsiyonunu minimum hatayla saptayabilecek karar verme mekanizmasına sahip bir sistem tasarlanmıştır. Oluşturulan bu sistem; gerek kullanılan verilerin deneysel olarak elde edilmesi açısından gerekse üç fazlı, kafesli, 1,1 kW, 90 S standart büyüklüğe sahip, dört kutuplu asenkron motorlarda toplam harmonik distorsiyonun sargı adımı, PWM taşıyıcı frekansı, motor iç ve dış sıcaklığı, stator akım ve gerilimlerine bakılarak saptanması açısından yapılmış ilk çalışmadır.

Kısaltılmış adımlı stator sargılı kafesli asenkron motorlar ve onların şebekeye verdiği harmonik etkiler, sistemde araç olarak kullanılmış olup, sistemin öğrenmesinin amaç olması hedeflenmiştir. Sistemin simülasyonu, en optimum sonuçları elde edecek şekilde tasarlanmış ve sistemi en iyi yansıtan model oluşturulmuştur.

Parametreleri farklı bir asenkron motorun şebekeye verdiği harmoniklerin belirlenebilmesi için aynı matematiksel modelin kullanılması söz konusu değil iken, yapay sinir ağı kullanılarak tasarlanmış bu yeni model tekrar eğitilerek farklı asenkron

motorlar için tekrar kullanılabilir. Kafesli asenkron motorlarda harmoniklerin önceden tahmin edebilecek bir sistemin elde edilmesini amaçlayan bu çalışmada, öğrenme işlemi gerçek zamanlı çalışmaya da imkan sağlar. Sistemin eğitimi sırasında özellikle en az veriyi kullanarak mümkün olan en iyi sonucun elde edilmesi hedeflenmiştir.

Bu çalışma sadece THD tahmini yapmakta değil özel harmonik değerlerinde tahmininin gerçekleştirilebilmesi için bir zemin oluşturacaktır. Ayrıca kullanılan asenkron motorlar daha da genişletilerek giriş değişkenleri daha da artırılarak çok daha büyük çapta bir tahmin programı oluşturabilmek bu çalışmadan sonra mümkün olabilir. Ayrıca bu çalışma aktif filtre tasarımı, motor tasarımı, sargı tasarımı konularında kullanılacak özgün bir bilgi ve uygulama kaynağı özelliğini göstermektedir.

Mart 2008

Yrd. Doç. Dr. Yaşar BİRBİR

H. Selçuk, NOĞAY

SEMBOL LİSTESİ

P	: Skaler giriş
w, A	: Ağırlık (girdilerin önemini belirten, başlangıçta rastgele atanan değer)
b	: Dengeleme sabiti (ofset)
n	: Toplama fonksiyonun çıkışı (net giriş)
a	: Skaler nöron çıkışı
m	: Katman sayısı
S	: Hassasiyet
t	: Varılmak istenen sonuç (Target)
θ_j	: Eşik değer (Dengeleme sabiti)
x	: Ağırlık ve dengeleme vektörü
k	: İterasyon sayısı
d	: Danışman
R	: Matlab ortamında YSA' nın genel girişleri
α	: Öğrenme Oranı
$\lambda_1, \lambda_2, ..\lambda_n$: Hessian matrisinin öz değerleri
$z_1, z_2, ..z_n$: Hessian matrisinin öz vektörleri
α	: Momentum katsayısı
A	: Değişken
Δw	: Ağırlıktaki değişim
e	: Hedef ve YSA çıkışları arasındaki hata
E	: Ağ hataları vektörü
E_m	: Çıktı katmanındaki m.yapay sinir hücresi için oluşan hata
FNET	: Aktivasyon fonksiyonu
$g(t)$: Ağın gradyeni
$x_1,..x_n$: Girdiler (Girişler)
η	: Öğrenme katsayısı
H	: Hessian matrisi
I	: Birim matris
J	: Jakobien matrisi
μ	: Marquardt parametresi

n	: Bir hücreye gelen toplam girdi
net, NET	: Ağa gelen net girdi
t(k)	: YSA çıkışı
TH	: Çıktı katmanına ait toplam hata
y(i)	: Gerçek çıkış
f	: Frekans (Hz)
f_v	: Frekansın v' ninci harmoniği
k_f	: Form faktörü
l	: Endüvi uzunluğu (cm)
m	: Faz sayısı
N	: Oyuk sayısı
n	: Dakikadaki devir sayısı
p	: Çift kutup sayısı
Q	: Kutup başına isabet eden oyuk sayısı
q	: Her bir kutup ve faza isabet eden oyuk sayısı
s	: Bobinlerde kırılenme miktarı, endüvi çevresinde bir uzaklık
T	: Bir periyot süresi (saniye)
v	: Endüvi çevre hızı (cm/s)
W	: Her bir faza isabet eden bobin sayısı ve her bir faza ait toplam sarım sayısı
ω	: Bir çift oyuğa yerleştirilmiş bir bobine ait sarım sayısı
ω_p	: Bir kutba yerleştirilen toplam sarım sayısı
y	: Bobin genişliği (cm)
Z	: Her bir faza ait toplam iletken sayısı, empedans
α	: Kutup örtme faktörü, açı cinsinden uzaklık
β	: Kırılenmiş bobinlerde endüklenen gerilim fazörleri arasındaki faz farkını π' den farklı yapan açı (Derece)
β_v	: Birinci gerilim oyuk fazörü ile bileşke gerilim fazörü arasındaki v'ninci harmonikteki açı, kırılenmiş bobinlerde endüklenen gerilim fazörleri arasındaki faz farkının v' ninci harmoniği
γ	: Oyuklar arasındaki elektriksel açı (Derece)
γ_{oluk}	: Oyuklar arasındaki geometrik açı
γ_v	: Birbirini takip eden oyuklar arasındaki faz farkının v' ninci harmoniği

δ	: Hava aralığı (mm)
λ	: Dalga uzunluğu
μ	: Alt harmonikleri tarif eden tam bir sayı, magnetik iletkenlik
ν	: Harmonik mertebe sayısı
ξ	: Sargı yayılma faktörü
$\xi_1, \xi_2, \dots, \xi_\nu$: Sargı yayılma faktörünün ν ' ninci harmoniğe kadar bilinen değeri
$\xi_s, \xi_{s\nu}$: Kirişlenme faktörü ve bu faktörün ν ' ninci harmoniği
Φ	: Mağnetik akı (Wb)
Ψ	: Faz akısı
ψ	: İletken sayısı, sinüs eğrisinin alan eğrisine nazaran faz farkı
τ_p	: Kutup taksimatı
ω	: Açısal hız (rad/s)

KISALTMA LİSTESİ

ART	: Adaptive Rezonance Theory
BPA	: Geri Yayılım Algoritması (Back Propagation Algorithm)
GA	: Genetik Algoritma
LMBP	: Levenberg-Marquardt Metodu
LVQ	: Learning Vector Quantization
MLP	: Çok Katmanlı Perceptronlar (Multilayered Perceptrons)
MSE	: Mean Squared Error
SOM	: Self Organizing Map
YSA	: Yapay Sinir Ağları
AA	: Alternatif Akım
DA	: Doğru Akım
A.S.M	: Asenkron Motor
PLC	: Programlanabilir Denetleyici
THD	: Toplam Harmonik Distorsiyonu

ŞEKİLLER LİSTESİ

SAYFA NO

Şekil II.1	Öğrenmenin Nörolojik Açıklaması.....	9
Şekil II.2	Geri Beslemeli Öğrenme Biçiminin Genel Yapısı	11
Şeki II.3	Bilgisayarlı Öğrenme Türleri	12
Şekil II.4	Biyolojik Sinir Hücresi ve Bileşenleri[1].....	14
Şekil II.5	Bir Biyolojik Sinir Hücresinin Yapısı.....	15
Şekil II.6	İki Biyolojik Hücrenin Bilgi Alış Verişi (Sinaps)	15
Şekil II.7	Biyolojik Sinir Sisteminin Blok Gösterimi	16
Şekil II.8	Yapay Sinir Hücresinin Yapısı	17
Şekil II.9	Doğrusal Aktivasyon Fonksiyonu	19
Şekil II.10	Hiperbolik Tanjant Aktivasyon Fonksiyonu	20
Şekil II.11	Sigmoid Aktivasyon Fonksiyonu	21
Şekil II.12	Signum Aktivasyon Fonksiyonu.....	21
Şekil II.13	Eşik Aktivasyon Fonksiyonu.....	22
Şekil II.14	İleri Beslemeli Ağ İçin Blok Diyagram	24
Şekil II.15	Geri Beslemeli Ağ İçin Blok Diyagram.....	24
Şekil II.16	Danışmanlı Öğrenme Yapısı.	26
Şekil II.17	Danışmansız Öğrenme Yapısı.	27
Şekil II.18	Takviyeli Öğrenme Yapısı.	27
Şekil II.19	Genel Bir Geri Yayılım Ağ Yapısı [2].....	31
Şekil II.20	İleri Beslemeli Bir Ağ İçin Geri Yayılım Akış Şeması [24]	32
Şekil II.21	Tek Katmanlı Hücrenin Matlab' da Kısaltılmış Şekli	33
Şekil II.22	Üç Katmanlı YSA İçin Kısaltılmış Gösterim.....	33
Şekil III.1	Asenkron Motorun Yapısı	49
Şekil III.2	Üç Fazlı, 36 Oyuklu, 4 Kutuplu El Tipi Sarım Şeması.....	52
Şekil III.3	Üç Fazlı, 36 Oyuklu, 4 Kutuplu Tam Kalıp Sarım Şeması	53
Şekil III.4	Üç Fazlı, 36 Oyuklu, 4 Kutuplu Tam Kalıp Kısa Adımlı Sarım Şeması	54
Şekil III.5	Gerilim/Frekans Değişim Eğrisi	60
Şekil III.6	Asenkron Motorun Faz Eşdeğer Devresi.....	61
Şekil III.7	Asenkron Motorun Her Faza İlişkin Harmonik Eşdeğer Devresi.....	62

Şekil III.8	Sinüs Üçgen Karşılaştırması İle Elde Edilen PWM Dalga Şekli.....	64
Şekil IV.1	PLC' nin Temel Yapısı.....	67
Şekil IV.2	Bir PLC Cihazında Sayısal Giriş Sinyallerinin İşlenmesi	68
Şekil IV.3	Dört Kanallı Analog Giriş Modülü Blok Diyagramı.....	70
Şekil IV.4	İşletim Sistemi.....	70
Şekil IV.5	Çift Mikrişlemcili PLC' nin Blok Diyagramı	71
Şekil IV.6	Tarama İşlemi.....	73
Şekil V.1	Asenkron Motorlarda Stator Sargıları.....	80
Şekil V.2	Kafesli Asenkron Motorun Yapı Ölçüleri	81
Şekil V.3	Stator Sarımları Yapılmış Deney Motorları	82
Şekil V.4	Deney Seti	83
Şekil V.5	Asenkron Motora Akuple Bağlı Elektromagnetik Fren Ünitesi.....	83
Şekil V.6	Deney Setinin Genel İç Bağlanıları	84
Şekil V.7	Deney Seti, Asenkron Motor ve Fren Ünitesinin Görünüşü	85
Şekil V.8	Deney Bağlantı Şeması	86
Şekil V.9	Cihaz Yönetim Sisteminin Ana Penceresi	88
Şekil V.10	Tablo Görünüm Penceresi	89
Şekil V.11	Periyodik Değerler İçin Parametre Seçimi	89
Şekil V.12	Periyodik Değerler Penceresi	90
Şekil V.13	Grafik Seçimi Penceresi	91
Şekil V.14	Grafik Penceresi.....	91
Şekil V.15	Güç Analizöründen Veri Transferi	92
Şekil V.16	Sıcaklık Ölçümleri İçin Bağlantı Şeması	93
Şekil V.17	Sıcaklık Analizi İçin Lab View Ön Paneli	93
Şekil V.18	Yapay Sinir Ağı Modellemesi İçin NFTOOLBOX Ana Penceresi.....	97
Şekil V.19	Çok Katmanlı Yapay Sinir Ağının Yapısı.....	98
Şekil V.20	Gizli Katmandaki Hücre Sayısının Seçimi.....	99
Şekil V.21	YSA Modeli İçin Elde Edilen Eğitim Sonuçları Penceresi.....	100
Şekil V.22	I THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi.....	101
Şekil V.23	Dakikalık Akım Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	102
Şekil V.24	Dakikalık Akım Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	103

Şekil V.25	Dakikalık Akım Çıkışlı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	104
Şekil V.26	YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması	105
Şekil V.27	YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata).....	105
Şekil V.28	YSA Modeli İçin Elde Edilen Eğitim Sonuçları Penceresi.....	106
Şekil V.29	V THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi	108
Şekil V.30	Dakikalık Gerilim THD Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi	108
Şekil V.31	Dakikalık Gerilim THD Çıkışlı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	109
Şekil V.32	Dakikalık Gerilim THD Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi	110
Şekil V.33	YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması	110
Şekil V.34	YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata).....	111
Şekil V.35	I THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi.....	112
Şekil V.36	Saniyelik akım THD Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	113
Şekil V.37	Saniyelik akım THD Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi	114
Şekil V.38	Saniyelik akım THD Çıkışlı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	115
Şekil V.39	YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması	115
Şekil V.40	YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata).....	116
Şekil V.41	V THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi	117
Şekil V.42	Saniyelik V THD Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	118
Şekil V.43	Saniyelik V THD Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	119
Şekil V.44	Saniyelik V THD Çıkışlı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi.....	120
Şekil V.45	YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması	120
Şekil V.46	YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata).....	121

TABLolar LİSTESİ

SAYFA NO

Tablo V.1	Kafesli Asenkron Motorun Etiket Deęerleri.....	81
Tablo V.2	Asenkron Motorun Yapı Ölçüleri I (mm).....	81
Tablo V.3	Asenkron Motorun Yapı Ölçüleri II (mm)	82
Tablo V.4	Oluřturulan Sistemlere Ait Ortak Giriř Parametreleri.....	94
Tablo V.5	YSA Modellerine Göre Sonuların Karřılařtırılması.....	99
Tablo V.6	Dakikalık I THD ıkıřlı YSA Modeli Sonuları.....	101
Tablo V.7	Dakikalık V THD ıkıřlı YSA Modeli Sonuları.....	106
Tablo V.8	Saniyelik I THD ıkıřlı YSA Modeli Monuları	112
Tablo V.9	Saniyelik V THD ıkıřlı YSA Modeli Sonuları	117

BÖLÜM I

GİRİŞ VE AMAÇ

I.1. GİRİŞ

İnsanoğlu, mağara duvarlarına hayvan figürleri çizerek başladığı veri işleme sürecine artık insan gözüyle görülemeyecek küçüklükteki “chip”lerle devam etmektedir. Bu gelişim yaklaşık 4000 yıl gibi uzun bir süreye yayılmasına karşın gerçek gelişim son 50 yıl içinde yaşanmıştır. Bu gelişim sürecinde artık kendi zekasıyla yetinmeyen insanoğlu makinalere de zeka verme uğraşı içindedir; artık amaç daha zeki, insan davranışlarını örnekleyebilen ve karar verebilen makineler üretmektir. Bu noktada karşımıza “Yapay Zeka” kavramı çıkmaktadır.

Yapay zeka üzerinde en fazla araştırma yapılan konu “yapay sinir ağları”dır. Yapay sinir ağları, temelde tamamen insan beyni örneklenerek geliştirilmiş bir teknolojidir. Bilindiği gibi; öğrenme, hatırlama, düşünme gibi tüm insan davranışlarının temelinde sinir hücreleri bulunmaktadır. İnsan beyninde tahminen 10^{11} adet sinir hücresi olduğu düşünülmektedir ve bu sinir hücreleri arasında sonsuz diyebileceğimiz sayıda sinaptik birleşme denilen sinirler arası bağ vardır. Bu sayıdaki bir birleşimi gerçekleştirebilecek bir bilgisayar sisteminin dünya büyüklüğünde olması gerektiği söylenmektedir. İnsan beyninin bu karmaşıklığı göz önüne alındığında, günümüz teknolojisinin 1.5 kg.’lık insan beynine oranla henüz çok geride olduğu söylenebilir.

Yapay sinir ağları, bilgi sınıflama ve bilgi yorumlamanın da içinde bulunduğu çok değişik problemlerin çözümünde kullanılmaktadır. Bir çok alanda olduğu gibi elektrik alanında, endüstrinin temel taşı olan elektrik motorlarında da yapay sinir ağları kullanılarak daha verimli, daha az kayıplı sistemlerin tasarımı hedeflenmektedir[1], [2].

Elektrik motorları sürücü sistemler olarak endüstrinin vazgeçilmez elemanlarından birisidir. Asenkron motorlar da, sağlam ve güvenilir yapıları, düşük

maliyet, ses ve ataletlerinin düşük olması, özellikle sincap kafesli tiplerinde az bakım gerektirmeleri, kirlı ve tehlikeli ortamlarda alıřtırılabilmeleri nedeniyle endüstride diđer elektrik motorlarına göre daha yaygın olarak kullanılmaktadırlar. Ayrıca geniř gü ve hız aralıđına sahip olmaları da asenkron motorların diđer motorlara karřı önemli bir üstünlüđüdür.

Elektrik enerji sisteminin ve bu sisteme bađlanan elektrik motorlarının arızasız ve güvenli bir řekilde alıřabilmesi için o sistemdeki dalga řeklinin sinüsoidal olması ve frekansının 50 Hz' lik tek frekansa sahip olması istenir. Ancak sisteme bađlanan bazı elemanlar ve bazı olaylar sonucunda akım ve gerilim büyüklükleri sinüsoidal özelliklerini kaybederler ve sistemde istenmeyen harmonikler oluşur.

Lineer olmayan yüklerin, elektrik dağıtım sistemlerinde, gerilim ve akım dalga biçiminde bozulma meydana getirdikleri uzun yıllardır bilinmektedir. Ancak günümüzde; eskiden beri var olan lineer olmayan yüklere ilaveten, gü elektroniđi elemanlarının hızla yaygınlaşması ise dalga řeklindeki bozulmaya duyarlı elemanların sayısındaki artış, bu konuda yapılacak alıřmaların önemini artırmıřtır. Akım ve gerilim dalgalarında oluşan bu harmonikler elektrik tesislerine ve bu tesislere bađlı tüketicilere zarar vermekte veya tamamen işlemez hale getirmektedir. Harmoniklerin deđerlerinin zararlardan önce saptanabilmesi bunun için alınacak önlemlerin de yolunu açacaktır. Harmonik deđerlerinin önceden tahmin edilebilmesi için mutlaka, sistemleri temsil edebilen yapay zeka modellerinin geliştirilmesi gerekmektedir. [39], [42], [3].

I.2. AMA

Bu alıřmada ama; uygulamalı öğretim, araçlar ise kısaltılmış adımlı özdeř gövdeye sahip kafesli asenkron motorlar, AC sürücü, PLC modülü, harmonik analizörü, deneysel veriler ve yapay sinir ađlarıdır. Özdeř gövdeye sahip, beř adet, üç fazlı, kafesli asenkron motor; aynı kesit alanına sahip, özdeř emaye tel ile bir adım kısaltılarak aynı sipir sayısında ve uzunluđunda tekrar sarılmışlardır. Kriřlenmiş stator sargılı kafesli asenkron motorlar AC sürücü üzerinden beslendiklerinde, 1 ile 15 kHz. arasındaki bütün PWM taşıyıcı frekansları için ayrı ayrı boş alıřmadan aşırı yüklü alıřmaya kadar yavaş yavaş yüklenerek elde edilen deneysel veriler, ilk otuz harmonikten oluşan, akım THD ve gerilim THD yüzde deđerlerinin tahmin edilebilmesi için modellenen yapay sinir ađlarını eğitmek için kullanılmıştır.

- Çalışmada kullanılan asenkron motorlar ile yapılan deneysel veriler, girişler ve çıkışlar olarak akım ve gerilim toplam harmonik distorsiyonu öğretilecek bilgiyi,
- Yapay Sinir Ağları bilgiyi öğrenecek kimseyi,
- Yapay sinir ağlarında kullanılan geri yayılım algoritması ise öğretme yöntemini

temsil etmektedir. Böylece, öğrenmenin üç temel unsuru bu çalışmada gerçekleşmiş olacaktır. Üç fazlı kafesli asenkron motorlarda, gerek sargı tasarımı, gerek filtre tasarımı ve gerekse motor tasarımında tasarımcılara yardımcı olunacak, harmoniklerin önceden tahmin etmekte kullanılacak olan deneysel verilere ulaşılacaktır. Ulaşılan deneysel verilerin doğruluğuna, tutarlılığına ve sistemi tam olarak yansıtmasına bağlı olarak eğitilecek olan YSA modelleri oluşturulmuş olacaktır.

Kafesli asenkron motorların hangi sargı adımında, hangi PWM taşıyıcı frekansı ile hangi akımda ve gerilimde toplam harmonik distorsiyonun nasıl olacağı hakkında herhangi bir matematiksel modele ihtiyaç duyulmadan akım ve gerilim için toplam harmonik distorsiyonlarının, güvenilir bir seviyede saptanabilmesi için, son derece tutarlı ve güvenilir bir veri seti oluşturmak ve bu veri seti ile tatmin edici saptamalar yapabilen YSA modellerine ulaşmak amaçlanmıştır.

Öğretimde; yapay sinir ağlarını oluşturma, ağın öğreneceği verilerin seçimi ve öğretim işlemi sırasında kullanılan yöntem, MatLab paket programında oluşturulan, Nftoolbox ve yazılımla gerçekleştirilmiştir.

BÖLÜM II

YAPAY SİNİR AĞLARI

II.1. AKIL VE ZEKA

Akıl kelimesi toplumda genellikle insanların zekâ düzeyini ifade etmek amacıyla kullanılmaktadır. Sıklıkla akıl kavramı zekâyla karıştırılmaktadır. Oysa akıl, düşünme, anlama, kavrama, idrak etme, karar verme ve önlem alma yetenekleridir. Akıl aynı zamanda muhakeme ve bilgi elde etme gücü olarak da tanımlanabilir.

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

Her insan doğuştan belirli bir zekâyı sahiptir. Zekâ, belirli bir konuda çalışarak, öğretilerek, eğitilerek edinilen bilgi ve birikimlerle, deneyimlere dayalı becerilerle geliştirilebilir. İlk kez karşılaşılan ya da ani olarak gelişen bir olaya uyum sağlayabilme, anlama, öğrenme, analiz yeteneği, beş duyunun, dikkatin ve düşüncenin yoğunlaştırılması zekâ ile gerçekleştirilmektedir. Zekâ yazılım veya tümleşik yongalarla taklit edilebilmektedir. Bu durumda zekâ ‘Yapay Zekâ’ olarak adlandırılmaktadır [2].

II.1.1. Yapay Zeka

İnsan beyni dünyanın en karmaşık makinası olarak kabul edilebilir. İnsan beyni sayısal bir işlemi birkaç dakikada yapabilmesine karşın; idrak etmeye yönelik olayları çok kısa bir sürede yapar. Örneğin yolda giden bir şoför, yolun kayganlık derecesini önündeki tehlikeden ne kadar uzak olduğunu, sayısal olarak değerlendiremezse dahi geçmişte kazanmış olduğu tecrübeler sayesinde aracın hızını

azaltır. Çünkü o saniyelerde ölçülebilecek kadar kısa bir sürede tehlikeyi idrak etmiş ve ona karşı koyma gibi bir tepki vermiştir. Bilgisayarlar çok karmaşık sayısal işlemleri anında çözümleyebilmelerine karşın, idrak etme ve deneyimlerle kazanılmış bilgileri kullanabilme noktasında çok yetersizdirler. Bu olayda insanı yada insan beynini üstün kılan temel özellik, sinirsel algılayıcılar vasıtası ile kazanılmış ve görevli olarak sınıflandırılmış bilgileri kullanabilmesidir. Uzman sistemler (US), Bulanık Mantık (BM), Genetik Algoritma (GA) ve Yapay Sinir Ağları (YSA) gibi yapay zeka alt dalları özellikle son yıllarda, geniş bir araştırma ve uygulama alanı bulmaktadır.

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

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

Genetik algoritmalar yapay zekânın gittikçe genişleyen bir kolu olan evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadır. Genetik algoritma Darwin' in evrim kuramı *doğada en iyinin yaşaması* kuralından esinlenerek oluşturulan, bir veri öbeğinden özel bir veriyi bulmak için kullanılan bir arama yöntemidir. Genetik algoritma geleneksel yöntemlerle çözümü zor veya imkânsız olan problemlerin çözümünde kullanılmaktadır. Herhangi bir problemin genetik algoritma ile çözümü, problemi sanal olarak evrimden geçirerek yapılmaktadır [2].

II.1.2. Yapay Sinir Ağları

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. O nedenle, yapay sinir ağlarının programlanması çok zor veya mümkün olmayan olaylar için geliştirilmiş adaptif bilgi işleme ile ilgilenen bir bilgisayar bilim dalı olduğu söylenebilir [1].

Günümüzde bilgi işleme büyük çoğunlukla sayısal bilgisayarlar ile gerçekleştirilmektedir. Bilgi işlem makinalarının değişik türlerini ortaya çıkarmayı

amaçlayan arařtırmalar, sayısal bilgisayar dnyasındaki bař döndürücü geliřmelerin gölgesinde kalmasına rađmen devam etmektedir. Bu arařtırmaların bir yönünü insanların ve diđer canlıların sahip olduđu yapıları, iřletim ilkelerini taklit eden bir bilgi iřletim sistemini geliřtirmek oluřturmaktadır.

Yapay sinir ađları, insan beyninden esinlenerek geliřtirilmiř, ađırlıklı bađlantılar aracılıđıyla birbirine bađlanan ve her biri kendi belleđine sahip iřlem elemanlarından oluřan paralel ve dađıtılmıř bilgi iřleme yapılarıdır. Yapay sinir ađları, bir bařka deyiřle biyolojik sinir ađlarını taklit eden bilgisayar programlarıdır.

Yapay sinir ađları, ađırlıklandırılmıř řekilde birbirlerine bađlanmış bir çok iřlem elemanlarından (nöronlar) oluřan matematiksel sistemlerdir. Bir iřlem elemanı, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu iřlem elemanı diđer nöronlardan sinyaller alır; bunları birleřtirir, dönüřtürür ve sayısal bir sonuç ortaya çıkartır. Genelde, iřlem elemanları kabaca gerçek nöronlara karřılık gelirler ve bir ađ içinde bir birlerine bađlanırlar; bu yapıda sinir ađlarını oluřturmaktadır[33].

Yapay sinir ađları bir programcının geleneksel yeteneklerini gerektirmeyen, kendi kendine öđrenme düzenekleridir. Bu ađlar öđrenmenin yanı sıra, ezberleme ve bilgiler arasında iliřkiler oluřturma yeteneklerine de sahiptir [2].

Yapay sinir ađları, insanlar tarafından gerçekleřtirilmiř örnekleri (gerçek beyin fonksiyonlarının ürünü olan örnekleri) kullanarak olayları öđrenebilen, çevreden gelen olaylara karřı nasıl tepkiler üretebileceđini belirleyen bilgisayar sistemleridir. İnsan beyninin fonksiyonel özelliklerine benzer řekilde;

- Öđrenme,
- İliřkilendirme,
- Sınıflandırma,
- Genelleme,
- Özellik belirleme,
- Optimizasyon

gibi konularda bařarılı bir řekilde uygulanmaktadır. Örneklerden elde ettikleri bilgiler ile kendi deneyimlerini oluřturur; ve daha sonra, benzer konularda benzer kararları verirler.

Yapay sinir ađları günümüzde birçok probleme çözümler üretebilecek yeteneđe sahiptir. Deđiřik řekillerde tanımlanmaktadır. Tanımların ortak birkaç noktası vardır. Bunların en bařında yapay sinir ađlarının birbirine hiyerarřik olarak bađlı ve paralel

olarak çalışabilen yapay hücrelerden oluşmaları gelmektedir. İşlem elemanları da denilen bu hücrelerin birbirlerine bağlandıkları ve her bağlantının bir değerinin olduğu kabul edilmektedir. Bilginin öğrenme yolu ile elde edildiği ve işlem elemanlarının bağlantı değerlerinde saklandığı dolayısıyla dağıtık bir hafızanın söz konusu olduğu da ortak noktaları oluşturmaktadır. İşlem elemanlarının bir birleri ile bağlanmaları sonucu oluşan ağa yapay sinir ağı denmektedir. Bu ağın oluşturulması biyolojik sinir sistemi hakkındaki bulgulara dayanmaktadır.

Teknik olarak da, bir yapay sinir ağının en temel görevi, kendisine gösterilen bir girdi setine karşılık gelebilecek bir çıktı seti belirlemektir. Bunu yapabilmesi için ağ, ilgili olayın örnekleri ile eğitilerek (öğrenme) genelleme yapabilecek yeteneğe kavuşturulur. Bu genelleme ile benzer olaylara karşılık gelen çıktı setleri belirlenir[1], [12].

II.1.3. Öğrenme

II.1.3.1. Öğrenmenin Psikolojik Temelleri

İnsanoğlunda doğuştan var olan içgüdüsel davranışlar yok denecek kadar azdır ve bu davranışlar çevreye uyum sağlamada yetersizdir. Bu nedenle, insanlar hayatları boyunca bir takım bilgileri öğrenmek mecburiyetinde kalmaktadırlar. Öğrenmenin tanımının nasıl yapılacağı öğrenmeye hangi açıdan bakıldığına göre değişmektedir. Günümüzde öğrenmeyi açıklayan değişik kuramlar vardır[15].

Öğrenme bir sistemin çevresine uyum sağlayabilmek için kendi kendini yeniden düzenlemesi olarak nitelendirilebilir. Öğrenme ile bireyler, ortamdaki değişimlere bağlı olarak, davranış yetilerini en iyi şekilde kullanmak suretiyle yaşamlarını sürdürme imkanı sağlamaktadırlar. Muhakeme etme ve problem çözme gibi “yüksek seviye” yeteneklerimizi deneyimler sayesinde geliştirebiliriz.

Araştırmacıların dahil olduğu psikolojik kurama göre öğrenmenin tanımı da değişmektedir. Öğrenme sadece bir davranış değişikliğinden ibaret değildir. Öğrenme bir etkileşim sürecidir. Öğretimde öğretmen kadar öğrenci de etkindir ve öğretim, aralarında sürekli etkileşim ve alış – verişle oluşmaktadır[14].

Psikologlar tarafından genel olarak üç öğrenme türünün olduğu söylenmektedir.

1. Alışkanlık
2. Çağrışımlı öğrenme

- Klasik şartlanma (şartlı refleks)
- Edimsel koşullanma (yaparak şartlanma)

3. Karmaşık Öğrenme (kavramsal öğrenme)

Alışkanlık en basit öğrenme türü olup, alışlagelmiş ve ciddi sonuçları olmayan bir uyarı göz ardı etmeyi öğrenme anlamına gelir. Bir uyarı artık bireyin ilgisini çekmemeye başladığında, tepki şiddeti azalmaktadır [15], [3].

Klasik ve edimsel koşullanma; bağlantılar kurmayı, yani birlikte ortaya çıkan belirli olayları öğrenmeyi gerektirir. Klasik şartlanma; bir uyarıya istem dışı bir yanıt almayı sağlayan yöntemdir.

Yaparak veya edimsel şartlanma; bireyin uyarıyı beklemeden istenilen sonuca ulaşmak için çevresinde aktif olarak harekete geçmesidir. Bir öğrenmenin oluşmasına imkan vermiş olan şartlar sona ererse, bireyin şartlı tepkileri de ortadan kalkacaktır. Bu olaya silinme adı verilmektedir.

Karmaşık öğrenme bağlantılar kurmaya eklenen her şeyi kapsar. Örneğin bir problemi çözerken strateji uygulamak, karmaşık öğrenme türündendir. Bu durumda iş, uyarı veya yanıtların kesin karakteristiğine göre değil ancak çözüm kurallarına göre çözülebilir.

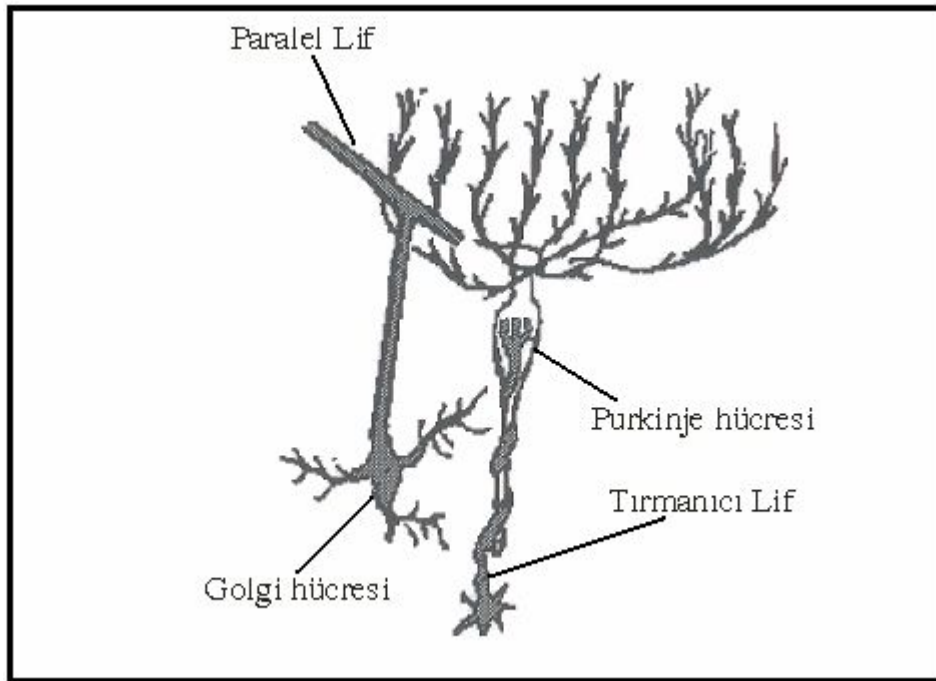
Sınama yanılma yolu ile öğrenmede birey bir dizi sınamadan sonra yavaş yavaş hatalarını düzeltir. Böylelikle davranış biçimi yavaş yavaş öğrenilmiş olur. Bunların dışında zoraki öğrenme yöntemi de vardır. Fakat dinlenme sürelerinin dengeli dağılması, öğrenmede başarılı olmanın temel koşuludur. İster çocuk, ister genç, ister yaşlı olsun sonunda insan bir makina değildir [3].

Bütün bu öğrenme yöntemlerinde çevre etkileyici bir rol oynamaktadır. En basit öğrenme biçimlerinin, karmaşık öğrenme türleri ile aynı kurallara uyduğu söylenemez. Öğrenen bireyin biyolojik karakteristikleri hesaba katılmadan, öğrenmenin ne olduğunu tam olarak anlamak mümkün değildir. Öğrenmenin sinir sistemi ve duyumsal – devrimsel yapısı, ona bazı ilişki türlerini tercihli olarak öğrenme imkanı verir.

Genellikle araştırmacılar, öğrenmenin nöral temelini sinir sistemindeki yapısal değişikliklerden ibaret olduğuna inanırlar. Bu değişimler sinapslar düzeyinde ortaya çıkmaktadır. İtki (impuls), bir nörondan diğerine göndericinin aksonu ile iletilir. Aksonlar sinaptik boşlukla ayrıldıkları için, gönderenin aksonu, bu boşluğa yayılan ve alıcı nöronu uyaran bir nörotransmitter salgılar. Daha özgül olarak nöral itki, göndericinin aksonu boyunca yol alırken nörotransmitteri serbest bırakmak için

aksonun ucundaki terminalleri tetikler ve bu transmitter daha sonra alıcı nörondaki resptorlar tarafından toplanır. Bütün bu yapıya sinaps adı verilir.

Beyincik korteksinde yer alan Purkinje hücreleri, dendritleri yardımıyla, giriş uyarıları alır; söz konusu bu giriş uyarıları, bir yandan beyinciğin derin çekirdeklerinden çıkan tırmanıcı liflerden ve diğer yandan Golgi hücrelerinin aksonları olan paralel liflerden gelir. Paralel liflerin sinapslarının etkinliği, bunların tırmanıcı liflerin sinapsları ile birlikte veya bağımsız çalışmasına bağlı olarak değişiklik gösterir. Bu değişiklik ise öğrenme olayına karşılık gelmektedir [3],[14].



Şekil II.1 Öğrenmenin Nörolojik Açıklaması

Önceden bir tepki oluşturmuş olan aynı uyarının Golgi hücrelerine verilmesi yeniden bir tepki oluşturmaz. Bu öğrenme türüne, sinapsların uzun vade durgunluğu adı verilir. Öğrenme ile ilgili anahtar fikirler şunlardır.

- Sinapstaki yapısal bir değişiklik, öğrenmenin nöral temelidir.
- Bu yapısal değişikliğin etkisi, sinapsı daha etkin hale getirmektedir.

Yapay sistemlerde de öğrenme, psikolojide olan tanıma benzer şekilde ele alınmaktadır. Sistem önceden bilmediği ve yapamadığı işlemleri, belirli bir eğitimden sonra gerçekleştirebiliyorsa, onun öğrendiği düşünülmektedir. Tanımın bu şekilde verilmesi, öğrenmenin hiyerarşik bir yapıya sahip olduğunu göstermektedir. Yapılan programlar da, psikologlar tarafından tanımlanan öğrenme türlerini taklit

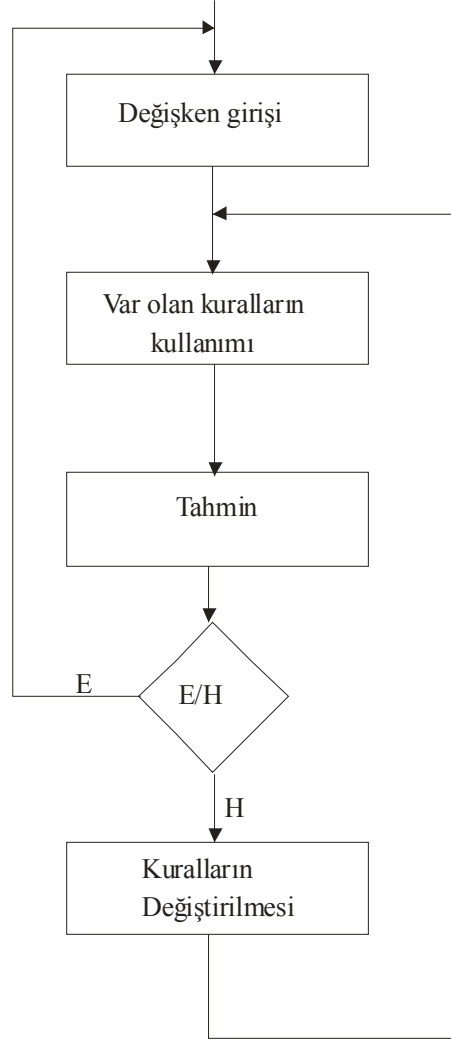
etmektedir. İlk modeller Pavlov' un şartlanmasına uygun olarak, fizyolojik öğrenme modeli temelinde yapılmakta idi. Daha sonra çağrışımlı öğrenme üzerine araştırmalar yoğunlaştı. Bu modellerde öğrenme, elemanları nörona benzer ağların kararlı duruma gelmesi şeklinde ele alınmaktadır. Çağrışımlı modelleri ise, kavramsal (cognitive) psikolojiye dayalı labirent modelleri izlemektedir. Günümüzde en çok araştırılan öğrenme şekli, örneklere göre öğrenme modelidir.

Öğrenme geri besleme mekanizmasının yardımı ile gerçekleşmektedir. Geri besleme, sistemin arzulan durumdan ne kadar uzakta olduğunu anlamakta yardımcı olur. Genellikle öğrenme sistemlerinde geri besleme süreci ve değişikliklerin algılanması söz konusudur[15], [3].

II.1.3.2. Bilgisayarlı Öğrenme

İnsan soyut olaylar veya örnekler yardımı ile öğretilir. Öğrenme yöntemlerinden bağımsız olarak insan, yeni ve farklı bir olayla karşılaşarak bilgi deposunu zenginleştirmektedir. Örneğin çocuktan, masanın üzerinde olan acı biberi alıp yememesini rica ediyorsunuz ve hatta görsel olarak ağzınızı büzerek gösteriyorsunuz bu onda daha da merak oluşturuyor ve biberi ağzına yalnız dokundurduktan sonra söylenenlerin ne olduğunu anlıyor. Çocuğa yeniden biberi teklif ettiğinizde, artık kesinlikle dokunmamaktadır. Bilgilerin bu biçimde elde edilmesine geri beslemeli öğrenme adı verilir. İnsan geri beslemeli şekilde edindiği bilgileri kullanarak yenilerine erişmektedir. Kısacası insan tecrübe yardımı ile öğrenir. Tecrübe ise önceki bilgilerle zıt olan yeni bir bilgiye rastlandığında edinilir. İnsan yalnızca yanlışlarını anladığında kendi kararını değiştirmektedir. Fakat bunun için de, yaptığının yanlış olduğunu gerçekten anlaması gerekir [3].

Bir bilgisayar programı da benzer şekilde çalışmaktadır. Yaşanan gerçekler temelinde mantıksal sonuçlandırma mekanizması da kullanarak yeni bilgilerin eklenmesiyle sistemin öğretilmesi gerçekleştirilir. Genel olarak bilgisayardaki basit öğrenme sürecini aşağıdaki şemayla gösterebiliriz. Şekil deki algoritmaya göre sistem, kullanıcıdan sistemin özelliklerini isteyerek (Evet/Hayır biçiminde) sonuçla ilgili belli bir karara varır. (Genelde ise bilgiler ikili biçimde verilmez). Kullanıcı, verilen kararları onaylamadığında sistemin karar verme mekanizmasını değiştirir. Sistem, aynı sınıfa dahil olan farklı nesnelere için tüm bilgileri elde ettikten sonra hata oranını düşürür ve artık kesin kararlar verir. Öğrenme süresince sistem, sürekli olarak kendini geliştirir. Bu gelişme başlangıç bilgileriyle ilişkilidir[3].

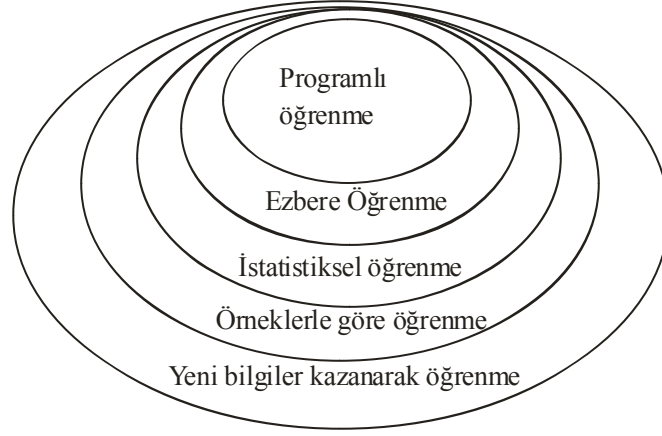


Şekil II.2 Geri Beslemeli Öğrenme Biçiminin Genel Yapısı

Bir öğrenme sisteminin oluşturulması aşaması aşağıdaki şekilde özetlenebilir:

- Sistem, yalnızca zıt bilgiyle karşılaştığında geri besleme mekanizması yardımıyla tabandaki değerlerini değiştirir.
- Sistem, girilen bilgi parametrelerini, daha önce kaydedilen değerlerle karşılaştırır ve en yakın benzerliği arayarak onları eşleştirmeye çalışır. Tüm bu görevleri, sistemin mantıksal çıkarım mekanizması üstlenmektedir.

Bilgisayarlı öğrenme insanlarınkine benzer şekilde hiyerarşik yapıya sahiptir. (Şekil II.3)



Şeki II.3 Bilgisayarlı Öğrenme Türleri

1. seviyede en basit tür olan “programlanmış öğrenme” söz konusudur. İlkel bilgisayar programları bu türdendir. Ardışık emirler neyin yapılması gerektiğini sıralı biçimde ifade etmektedir. Örneğin, kaynak yapan bir robot yalnızca emirleri gerçekleştirmekte, durum değişikliğini ise algılayamamaktadır.

2. seviye, “ezbere öğrenmeye” karşılık gelmektedir; her duruma ilişkin tepkiler bir bütün olarak tutulmaktadır. Burada sistem, kendini Pavlov’un şartlanmasına uygun olarak göstermekte ve durumunu değiştirebilmektedir. Yeni durum karşısında iyi sonuçlar üretildiğinde ödülleme, yanlış sonuçlarda ise ceza puanlarının verilmesi gerçekleştirilir.

3. seviyede, “istatistiksel öğrenme” söz konusudur. Burada sistemin birçok kez çalıştırılması (öğretilmesi) sonucu, tepkilere uygun olan bileşenler belirlenmektedir.

4. seviyede sistem, örneklere göre genelleştirme yapabilmekte ve “öğretmenli öğrenme” söz konusu olmaktadır.

5. seviyede ise sistem, yeni hipotezler ve durumlar oluşturarak kendi kendine (“öğretmensiz” veya “danışmansız”) öğrenebilmektedir[3].

Yapay zekanın kullanıldığı tüm alanlar için belirli öğrenme yöntemleri bulunabilir. İlk araştırmalarda, genel amaçlı öğrenme makinesi oluşturma fikri çok ilgi toplamıştır. Genel amaçlı bir öğrenme mekanizmasının olabilirliği düşüncesi daima cazip gelmiştir. Bu düşünceye göre insan, doğduğunda çok az şey bilir ve genel öğrenme sayesinde hemen hemen her şeyi kazanır. Tahminlere göre evrensel öğrenme kurallarını içeren sisteme, öğrenmesi için yeterli zaman tanınsaydı, oyunlar, doğal dil işleme vb. gibi zeka davranışları makina tarafından kolayca taklit edilebilirdi.

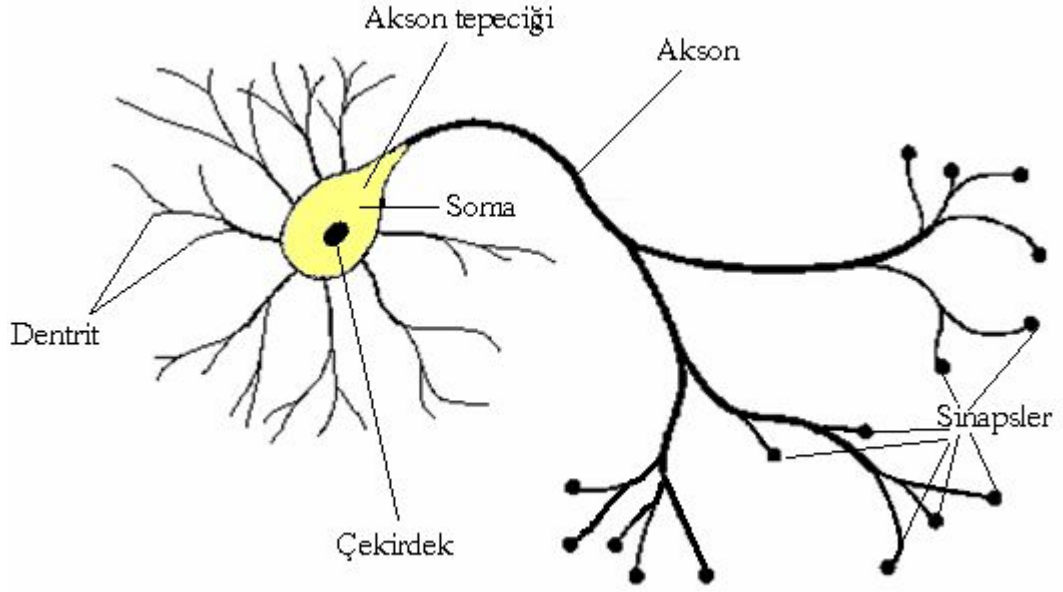
Yapay zeka alanındaki arařtırmalar, bu dūřüncenin artık geersiz olduėunu gōstermiřtir. Bu arařtırmalardan; bir organizmanın herhangi bir řey ōėrenebilmesi iin, zaten ok řey bilmesi gerektiėi sonucuna varılmıřtır. Yani ōėrenme, dūzenlenmiř bilgi ile bařlar. Bu organize olmuř bilgi, sonradan bŷyŷr ve daha iyi dūzenlenmiř bir hale gelir. ōėrenilecek řey hakkında gŷclŷ ipuları olmadıka, hibir řey ōėrenilemez. Fakat burada řuna da deėinmek gerekir ki, insan beyni farklı bōlgelerden oluřmakta ve bu bōlgeler arasında bir iřbirliėi bulunmaktadır. ōėrenme genellikle ayrık bir biimde ele alındıėı iin tek bir programın veya yōntemin bunu gerekleřtirebileceėi dūřŷnŷlmektedir. Doėal olarak bu da, zeka olgusunu gerektiren gōrevlerin yapay zeka ile gerekleřtirilmesinin ne derecede zor olduėunu gōstermektedir [3].

II.2. YAPAY SİNİR AėLARININ YAPISI VE TEMEL ELEMANLARI

II.2.1. Biyolojik Bir Beyin Sinir Hŷcresinin Yapısı

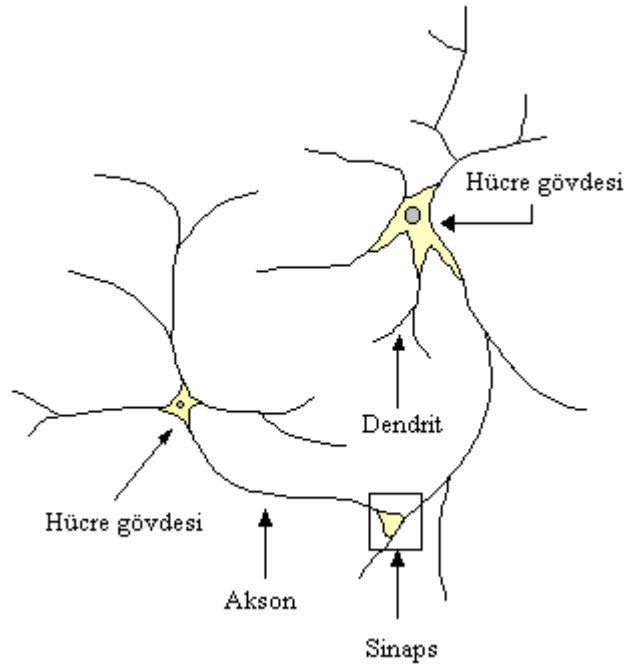
İnsan beyninin en temel parası, hatırlama, dūřŷnme, her harekette daha ōnceki deneyimlere bařvurma yeteneėini saėlayan kendine ōzgŷ sinir hŷcreleridir. Biyolojik sinir aėları beynimizde bulunan bir ok sayıda sinir hŷcresinin bir koleksiyonudur. Bir sinir aėı milyarlarca sinir hŷcresinin bir araya gelmesi ile oluřmaktadır. Sinir hŷcreleri birbirleri ile baėlanarak fonksiyonlarını yerine getirirler. Beynimizde 10^{10} adet sinir hŷcresi ve bunlarında 6×10^{13} , den fazla sayıda baėlantısının olduėu sōylenmektedir. İnsan beyni, ok hızlı alıřabilen mŷkemm el bir bilgisayar gibi gōrŷlebilir. İnsan beyninin alıřma frekansı 100 Hz' dir, aėırlıėı yetiřkin bir insanda yaklařık 1,3 kg. dır. Biyolojik sinir aėlarının performansları kŷŷmsenmeyecek kadar yŷksek ve karmařık olayları iřleyebilecek yetenektedir. Yapay sinir aėları ile bu yeteneėin bilgisayara kazandırılması amalanmaktadır[1],[30].

Biyolojik sinir aėları insan beyninin alıřmasını saėlayan en temel tařlardan birisidir. İnsanın bŷtŷn davranıřlarını ve evresini anlamasını saėlarlar. Biyolojik sinir aėları beř duyu organından gelen bilgiler iřıėında geliřtirdiėi algılama ve anlama mekanizmalarını alıřtırarak olaylar arasındaki iliřkileri ōėrenir. İnsan beyninin deėiřik bōlgeleri deėiřik fonksiyonları yerine getirmektedir. Duyu organlarından gelen bilgiler (sinyaller) beyin sinir sistemi sayesinde beyine tařınır ve beynin oluřturduėu kararlar da yine sinir sistemi tarafından vŷcudun organlarına eylem olarak gōnderilir.

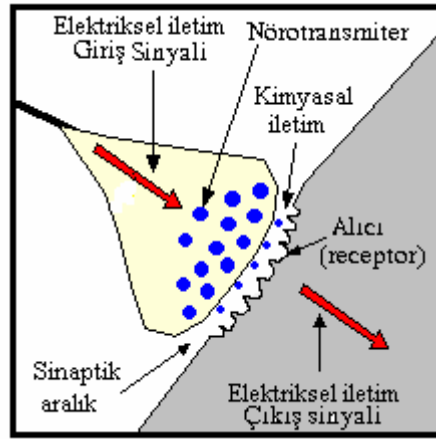


Şekil II.4 Biyolojik Sinir Hücresi ve Bileşenleri[1]

Şekil II.4' de gösterildiği gibi temel bir biyolojik sinir hücresi sinapslar, soma, axon, ve dentrite'lerden oluşmaktadır. Sinapsler sinir hücreleri arasındaki bağlantılar olarak görülebilir. Bunlar fiziksel bağlantılar olmayıp bir hücreden diğerine elektrik sinyallerinin geçmesini sağlayan boşluklardır. Bu sinyaller somaya giderler. Soma bunları işleme tabi tutar, sinir hücresi kendi elektrik sinyalini oluşturur ve axon aracılığı ile dentritlere gönderir. Dentritler ise bu sinyalleri sinapslere göndererek diğer hücrelere gönderilir. İki hücrenin birbirleri ile bilgi alış verişi sinaptik bağlantılarda nörotransmitterler yolu ile sağlanmaktadır. Şekil II.5' deki akson uçlarının her birisi başka bir hücre ile birleşmektedir. Şekil II.6' da iki biyolojik hücrenin nörotransmitterler yolu ile bilgi alış verişi gösterilmektedir[1].



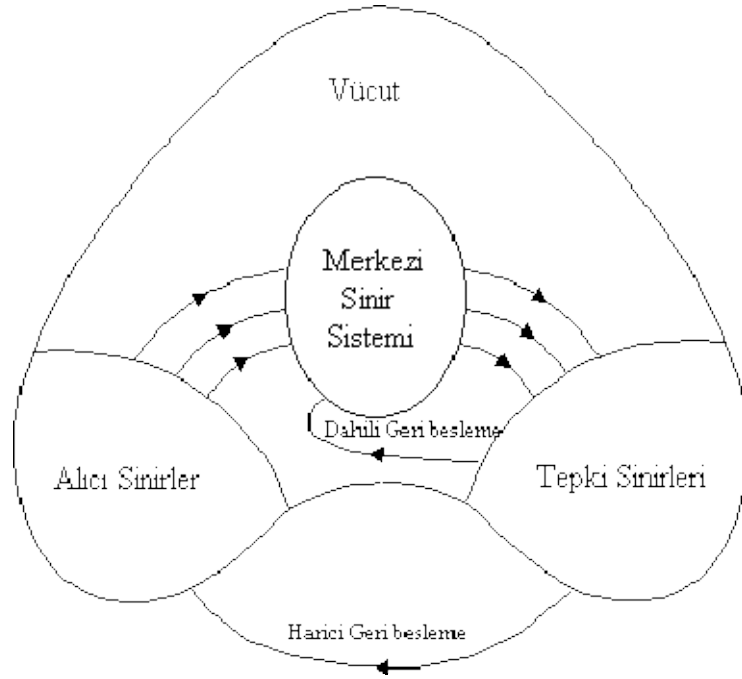
Şekil II.5 Bir Biyolojik Sinir Hücresinin Yapısı



Şekil II.6 İki Biyolojik Hücresinin Bilgi Alış Verişi (Sinaps)

Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin (merkezi sinir ağı) bulunduğu 3 katmanlı bir sistem olarak açıklanır. Alıcı sinirler (receptor) organizma içerisinden ya da dış ortamlardan algıladıkları uyarıları, beyne bilgi ileten elektriksel sinyallere dönüştürür. Tepki sinirleri (effector) ise, beynin ürettiği elektriksel darbeleri organizma çıktısı olarak

uygun tepkilere dönüştürür. Şekil II.7’ de bir sinir sisteminin blok gösterimi verilmiştir.



Şekil II.7 Biyolojik Sinir Sisteminin Blok Gösterimi

Merkezi sinir ağında bilgiler, alıcı ve tepki sinirleri arasında ileri ve geri besleme yönünde değerlendirilerek uygun tepkiler üretilir. Bu yönüyle biyolojik sinir sistemi, kapalı çevrim denetim sisteminin karakteristiklerini taşır. Yapay sinir ağları biyolojik hücrelerin bu özelliklerinden yararlanarak geliştirilmiştir[7].

II.2.2. Yapay Sinir Hücresi (İşlem Elemanı)

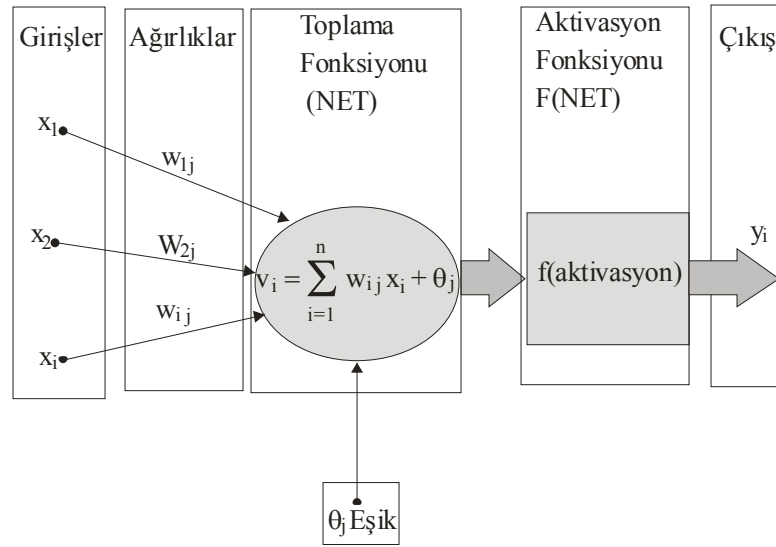
Yapay sinir ağları, birbirine bağlı çok sayıda işlem elemanlarından oluşmuş, genellikle paralel işleyen yapılar olarak adlandırılabilir. Yapay sinir ağlarındaki işlem elemanları basit sinirler olarak adlandırılabilir.

Yapay sinir ağları, insan beyni gibi, öğrenme hatırlama ve genelleme yeteneğine sahiptirler. İnsan beyninde öğrenme üç şekilde olur;

- Yeni aksonlar üreterek
- Aksonların uyarılmasıyla
- Mevcut aksonların güçlerini değiştirerek.

Her aksonun, üzerinden geçen işaretleri değerlendirebilecek yetenekte olduğu savunulmaktadır. Aksonun bu özelliği, bir işaretin belli bir sinir için ne kadar önemli olduğunu göstermektedir.

Yapay sinir ağlarının temel birimi işlem elemanı yada düğüm olarak adlandırılan yapay bir sinirdir. Bir yapay sinir, biyolojik sinirlere göre daha basit olmasına karşın biyolojik sinirlerin 4 temel işlevini taklit ederler. Şekil II.8’ de yapay bir sinir hücresi gösterilmektedir.



Şekil II.8 Yapay Sinir Hücresinin Yapısı

Girişler x_i sembolüyle gösterilmiştir. Bu girişlerin her biri ağırlık w ile çarpılır. Basitçe, bu ürünleri eşik değeri θ_j ile toplanır ve sonucu oluşturmak için aktivasyon fonksiyonu ile işlem yapılır ve y_i çıkışı alınır. Tüm yapay sinir ağları bu temel yapıdan türetilmiştir. Bu yapıdaki farklılıklar yapay sinir ağlarının farklı sınıflandırılmalarını sağlar. Bir yapay sinirin öğrenme yeteneği, seçilen öğrenme algoritması içerisinde ağırlıkların uygun bir şekilde ayarlanmasına bağlıdır [6], [2].

II.2.2.1. Girişler

Girişler ($x_1, x_2, \dots, x_3, \dots, x_n$) çevreden aldığı bilgiyi yapay sinir hücresine (işlem elemanına) getirir. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir. Girişler kendinden önceki sinirlerden veya dış dünyadan sinir ağına gelebilir. Bir sinir genellikle gelişigüzel bir çok girdiler alır[2].

II.2.2.2. Ağırlıklar

Ağırlıklar (w_1, w_2, \dots, w_i) bir yapay hücreye gelen bilginin önemini ve hücre üzerindeki etkisini belirleyen uygun katsayılardır. Her bir giriş kendine ait bir ağırlığa sahiptir. Şekil deki w_1 ağırlığı x_1 girişinin hücre üzerindeki etkisini göstermektedir. Ağırlıkların büyük yada küçük olması önemli yada önemsiz olduğu anlamına gelmez. Bir ağırlığın değerinin sıfır olması o ağ için en önemli olay olabilir. Eksi değerler önemsiz demek değildir. O nedenle artı veya eksi olması etkisinin pozitif veya negatif olduğunu gösterir. Sıfır olması ise herhangi bir etkisinin olmadığını gösterir. Ağırlıklar değişken veya sabit değerler olabilirler[2].

II.2.2.3. Toplama Fonksiyonu

Toplama fonksiyonu v_i , bir hücreye gelen net girdiyi hesaplar. Bunun için değişik fonksiyonlar kullanılmaktadır. En yaygın olanı ağırlıklı toplamı bulmaktır. Burada her gelen giriş (girdi) değeri kendi ağırlığı ile çarpılarak toplanır. Böylece ağa gelen net girdi bulunmuş olur. Aşağıdaki gibi formülize edilmektedir.

$$NET = \sum_i^n G_i A_i \quad (II.1)$$

Burada G girdileri, A ağırlıkları, n ise bir hücreye gelen toplam girdi (işlem elemanı) sayısını göstermektedir. Yalnız yapay sinir ağlarında daima bu formülün kullanılması şart değildir. Uygulanan yapay sinir ağı modellerinden bazıları kullanılacak toplama fonksiyonunu belirleyebilmektedir. Literatürde yapılan araştırmalarda toplama fonksiyonu olarak değişik formüllerin kullanıldığı görülmektedir. Bazı durumlarda toplama fonksiyonu bu kadar basit bir işlem yerine, en az (min), en çok (max), çoğunlukla veya birkaç normalleştirme algoritması gibi çok daha karmaşık olabilir [1], [2].

II.2.2.4. Aktivasyon Fonksiyonu

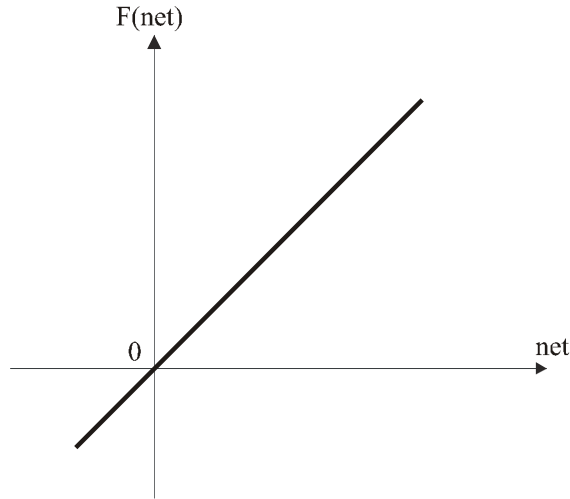
Bu fonksiyon, hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıkışı (çıkıyı) belirler. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonu olarak çıkıyı hesaplamak için de değişik formüller kullanılmaktadır. Bazı modeller (mesela çok katmanlı algılayıcı) bu fonksiyonun türevinin alınabilir bir fonksiyon olmasını şart koşturmaktadır. Toplama fonksiyonunda olduğu gibi aktivasyon fonksiyonunda da ağırlık işlem elemanlarının hepsinin aynı fonksiyonu

kullanması gerekmez. Bazı elemanlar aynı fonksiyonu diğerleri farklı fonksiyonları kullanabilirler. Bir problem için en uygun fonksiyonu, tasarımcı deneyimlerinin sonucunda belirlemektedir. Uygun fonksiyonu gösteren bir formül bulunmuş değildir [1], [2].

Aktivasyon fonksiyonları, doğrusal ve doğrusal olmayan biçimlerde (doğrusal, parçalı doğrusal, katı sınırlamalı, yumuşak sınırlamalı vb.).

1. Doğrusal Aktivasyon Fonksiyonu

Doğrusal bir problemi çözmek amacıyla kullanılan doğrusal hücre ve YSA' da ya da genellikle katmanlı YSA' nın çıkış katmanında kullanılan doğrusal aktivasyon fonksiyonu, hücrenin net girdisini doğrudan hücre çıkışı olarak verir. Doğrusal aktivasyon fonksiyonu denklem II.2 'deki gibi ifade edilir[7].



Şekil II.9 Doğrusal Aktivasyon Fonksiyonu

$$F(NET) = NET \quad (II.2)$$

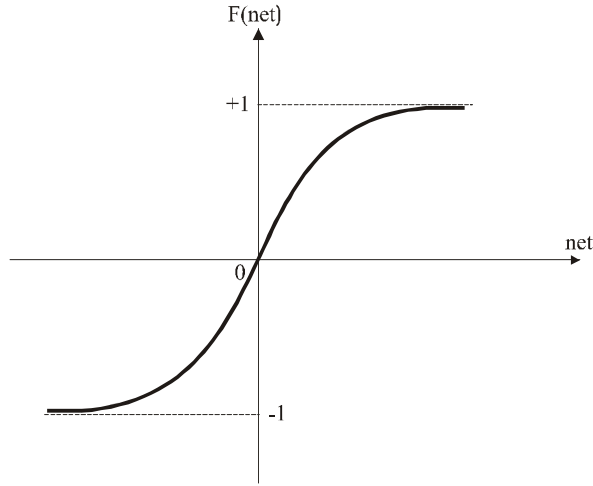
Burada NET işlem elemanına gelen NET girdi değerini göstermektedir. Bu değer toplama fonksiyonu kullanılarak belirlenmektedir[7].

2. Hiperbolik Tanjant Aktivasyon Fonksiyonu

Gelen NET girdi değerinin tanjant fonksiyonundan geçirilmesi ile hesaplanmaktadır. Türevi alınabilir, sürekli ve doğrusal olmayan bir fonksiyon olması nedeniyle doğrusal olmayan problemlerin çözümünde kullanılan YSA'larında

tercih edilir. Şekil II.10' da Hiperbolik tanjant aktivasyon fonksiyonu gösterilmektedir. Fonksiyonu aşağıdaki gibi ifade edilmektedir[7].

$$F(NE T) = \tanh(NE T) = \frac{e^{NE T} + e^{-NE T}}{e^{NE T} - e^{-NE T}} \quad (II.3)$$

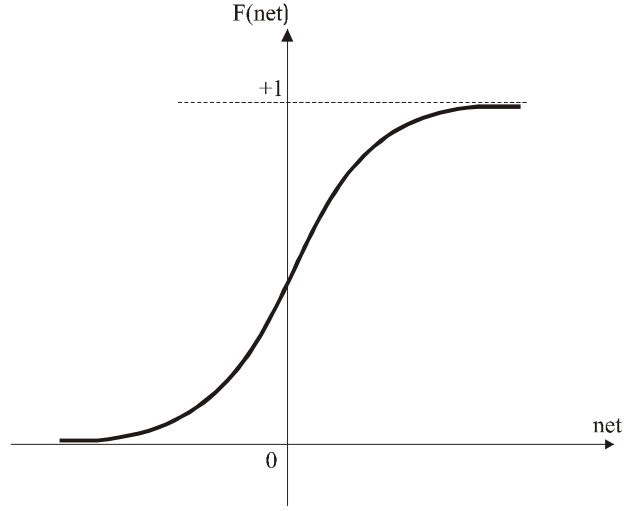


Şekil II.10 Hiperbolik Tanjant Aktivasyon Fonksiyonu

3. Logaritma Sigmoid Aktivasyon Fonksiyonu

Günümüzde en yaygın olarak kullanılan çok katmanlı algılayıcı modelinde genel olarak aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmaktadır. Bu fonksiyon şu formül ile gösterilmektedir. Bu fonksiyona sürekli eşikleme fonksiyonu da denmektedir. Şekil II.11' de Sigmoid aktivasyon fonksiyonu gösterilmektedir[7].

$$F(NE T) = \frac{1}{1 + e^{-NE T}} \quad (II.4)$$

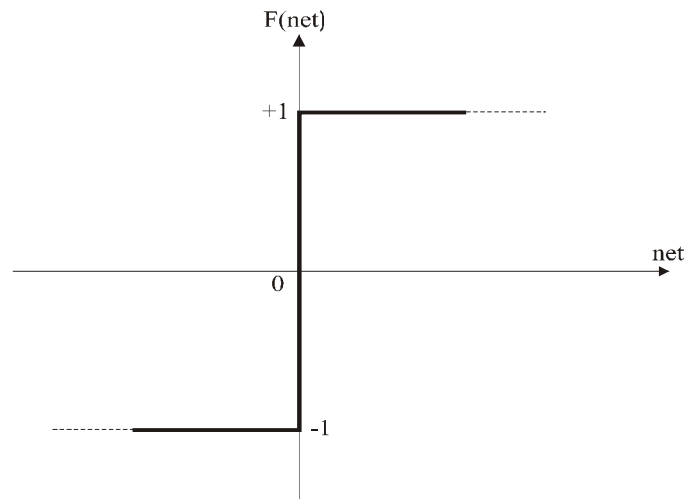


Şekil II.11 Sigmoid Aktivasyon Fonksiyonu

4. *Signum Aktivasyon Fonksiyonu*

Eşik aktivasyon fonksiyonunun -1 ile $+1$ arasında değişeni ise signum aktivasyon fonksiyonu olarak adlandırılır. Eşik aktivasyon fonksiyonlu hücreler, mantıksal çıkış verir ve sınıflandırıcı ağlarda tercih edilir. Şekil II.12’ de katı sınırlamalı aktivasyon fonksiyonu gösterilmektedir. Signum aktivasyon fonksiyonunu matematiksel olarak aşağıdaki gibi ifade edilir[24].

$$f(NET) = \text{sgn}(net) = \begin{cases} +1, net > 0 \\ -1, net < 0 \end{cases} \quad (\text{II.5})$$

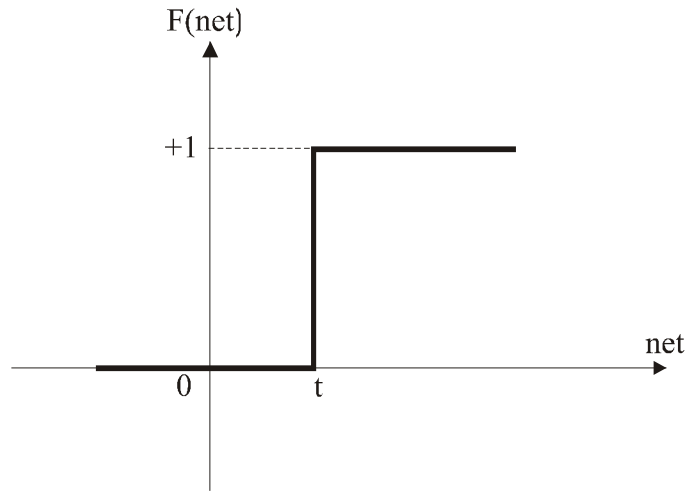


Şekil II.12 Signum Aktivasyon Fonksiyonu

5. Eşik Aktivasyon Fonksiyonu (Step)

Eşik aktivasyon fonksiyonu eğer net değeri eşik değeri (t)' den küçükse sıfır, eşik değerinden daha büyük veya eşit bir değer ise çıkışında $+1$ değeri verir. Aşağıdaki gibi ifade edilmektedir[7].

$$f(NE T) = \begin{cases} 1, NE T \geq t \\ 0, NE T < t \end{cases} \quad (II.6)$$



Şekil II.13 Eşik Aktivasyon Fonksiyonu

II.2.2.5. Ölçekleme ve Sınırlama

Yapay hücrelerde, aktivasyon fonksiyonun sonuçları ölçek veya sınır işlemlerinden geçebilir. Bu ölçeklendirme basitçe bir ölçek etmeni ile etkinlik değerinin çarpımının sonucudur. Sınırlandırma ise, ölçeklenmiş sonuçların enaz ve ençok sınırlarını aşmamasını sağlamaktadır[2].

II.2.2.6. Hücresinin Çıkışı

Aktivasyon fonksiyonu tarafından belirlenen çıkış değeridir. Üretilen çıkış dış dünyaya veya başka bir hücreye gönderilir. Hücre kendi çıkışını kendisine giriş olarak da gönderebilir. Bir işlem elemanın birden fazla çıkışı olmasına rağmen sadece bir çıkışı olmaktadır. Ağ şeklinde gösterildiğinde bir işlem elemanın birden fazla çıkışı varmış gibi görülmektedir. Bu sadece gösterim amacıyla. Bir yapay

sinir hücresinin tek bir çıkışı vardır. Aynı değer birden fazla işlem elemanına giriş olarak gitmektedir.

Her bir yapay sinir hücresinde (düğümde) bir çıkış işaretine izin verilir. Bu işaret diğer yüzlerce sinir hücresinin girişi olabilir. Bu durum biyolojik sinirde olduğu gibidir. Biyolojik sinirde de bir çok giriş varken sadece bir çıkış etkinliği vardır. Düğüm çıkışı aktivasyon fonksiyonunun sonucuna eşdeğerdir[1].

Yapay sinir ile biyolojik sinirler arasındaki benzerlik Tablo II.1' de gösterilmektedir.

Tablo II.1 Biyolojik Sinir Ağı ile Yapay Sinir Ağının Karşılaştırılması[2]

Biyolojik Sinir Ağı	Yapay Sinir Ağı
Sinir Sistemi	Sinirsel Hesaplama Sistemi
Sinir	İşlem Elemanı (Düğüm)
Sinaps	Ağırlıklar
Dendrit	Toplama İşlevi
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Hücrenin Çıkışı

II.3. YAPAY SİNİR AĞLARIN SINIFLANDIRILMASI

YSA'lar, genel olarak birbirleri ile bağlantılı işlemci birimlerden veya diğer bir ifade ile işlemci elemanlardan (neurons) oluşurlar. Her bir sinir hücresi arasındaki bağlantıların yapısı ağın yapısını belirler. İstenilen hedefe ulaşmak için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan bir öğrenme kuralına göre, hatayı sıfıra indirecek şekilde, ağın ağırlıkları değiştirilir. YSA'lar yapılarına ve öğrenme algoritmalarına göre sınıflandırılırlar.

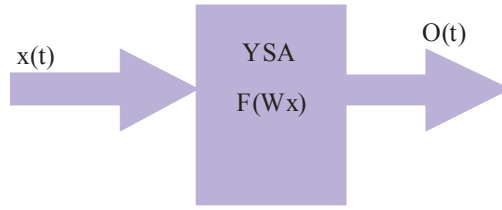
Yapay sinir ağı modelleri, bir ağın öğrenmesi gereken giriş-çıkış bilgilerine ve sinirler arası ilişkilerle ifade edilen mimarilerine göre farklılıklar göstermektedir. Farklı öğrenme türleri yapay sinir ağları modellerini belirlemektedir[3],[24].

II.3.1. Mimari Yapılarına Göre Sınıflandırılması

Yapay sinir ağları, yapılarına göre, ileri beslemeli (feedforward) ve geri beslemeli (feedback) ağlar olmak üzere iki şekilde sınıflandırılırlar.

II.3.1.1. İleri Beslemeli Ağlar

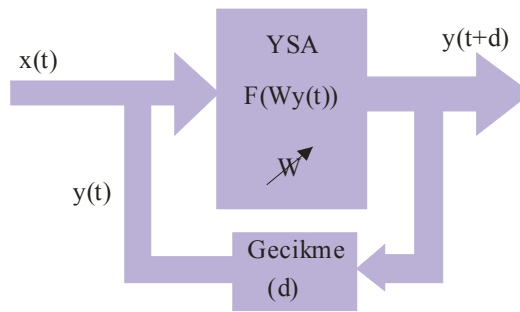
İleri beslemeli bir ağda işlem elemanları (İE) genellikle katmanlara ayrılmışlardır. İşaretler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir. İE' ler bir katmandan diğer bir katmana bağlantı kurarlarken, aynı katman içerisinde bağlantıları bulunmaz. Şekil II.14' de ileri beslemeli ağ için blok diyagram gösterilmiştir. İleri beslemeli ağlara örnek olarak çok katmanlı perseptron (Multi Layer Perseptron-MLP) ve LVQ (Learning Vector Quantization) ağları verilebilir[2], [24].



Şekil II.14 İleri Beslemeli Ağ İçin Blok Diyagram

II.3.1.2. Geri Beslemeli Ağlar

Bir geri beslemeli sinir ağı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Böylece, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Şekil II.15' de bir geri beslemeli ağ görülmektedir. Bu çeşit sinir ağlarının dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundur. Bu ağlar çeşitli tipteki zaman-serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir[24].



Şekil II.15 Geri Beslemeli Ağ İçin Blok Diyagram

II.3.2. Öğrenme Yaklaşımlarına Göre Sınıflandırılması

Yapay sinir ağlarında işlem elemanlarının bağlantılarının ağırlık değerlerinin belirlenmesi işlemine “ağın eğitilmesi” denir. Başlangıçta bu ağırlık değerleri rastgele olarak atanır. Yapay sinir ağları kendilerine örnekler gösterildikçe bu ağırlık değerlerini değiştirirler. Amaç ağa gösterilen örnekler için doğru çıktılar üretecek ağırlık değerlerini bulmaktır. Örnekler ağa defalarca gösterilerek 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 “ağın öğrenmesi” denir. Ağırlıkların değerlerinin değişmesi belirli kurallara göre yürütülmektedir. Bu kurallara “öğrenme kuralları” denir. Kullanılan öğrenme stratejisine göre değişik öğrenme kuralları geliştirilmiştir[1].

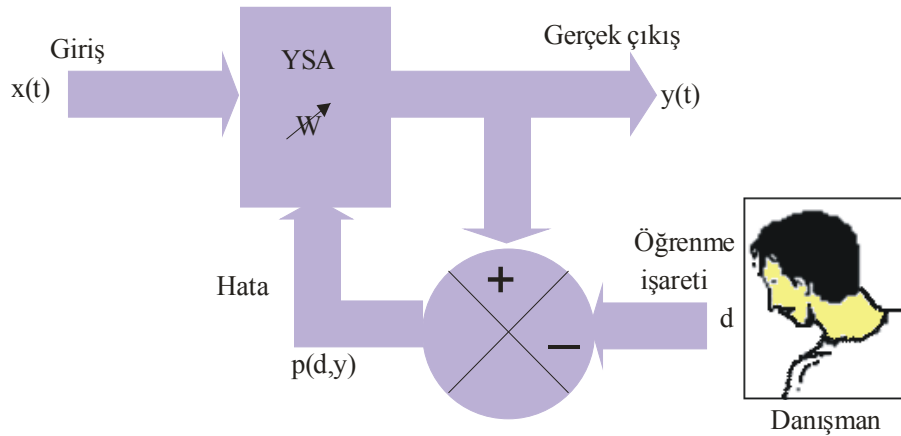
Yapay sinir ağlarında öğrenme olayının iki aşaması vardır. Birinci aşamada ağa gösterilen örnek için ağın üreteceği çıktı belirlenir. Bu çıktı değerinin doğruluk derecesine göre ikinci aşamada ağın sahip olduğu ağırlıklar değiştirilir. Ağın çıktısının belirlenmesi ve ağırlıkların değiştirilmesi öğrenme kuralına bağlı olarak farklı şekillerde olmaktadır. Ağın eğitimi tamamlandıktan sonra öğrenip öğrenmediğini (performansını) ölçmek için yapılan denemelere ise “ağın test edilmesi” denmektedir. Test etmek için ağın öğrenme sırasında görmediği örnekler kullanılır. Test etme sırasında ağın ağırlık değerleri değiştirilmez. Test örnekleri ağa gösterilir. Ağ eğitim sırasında belirlenen bağlantı ağırlıklarını kullanarak görmediği bu örnekleri için çıktılar üretir. Elde edilen çıktıların doğruluk değerleri ağın öğrenmesi hakkında bilgiler verir. Sonuçlar ne kadar iyi olursa eğitimin performansı da o kadar iyi demektir. Eğitimde kullanılan örnek setine “eğitim seti”, test için kullanılan sete ise “test seti” adı verilmektedir. Yapay sinir ağlarının bu şekilde bilinen örneklerden belirli bilgileri çıkartarak bilinmeyen örnekler hakkında yorumlar yapabilme (genelleme yapabilme) yeteneğine “Adaptif öğrenme” adı verilmektedir[1], [24].

Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Bir yapay sinir ağının eğitilebilmesi, gözlem ve eğitime göre ağdaki ağırlıkların, yöntem ve kuralların değiştirilmesi ile sağlanmalıdır. Bunun için genel olarak üç öğrenme metodundan ve bunların

uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

II.3.2.1. Danışmanlı Öğrenme (Supervised Learning)

Bu tip öğrenmede, YSA' ya örnek olarak bir doğru çıkış verilir. İstenilen ve gerçek çıkış arasındaki farka (hataya) göre işlem elemanları arasındaki bağlantıların ağırlıkları, en uygun çıkışı elde etmek için sonradan düzenlenir. Bu sebeple danışmanlı öğrenme algoritmasının bir “öğretmene” veya “danışmana” ihtiyacı vardır. Şekil II.16' da danışmanlı öğrenme yapısı gösterilmiştir. Widrow-Hoff tarafından geliştirilen Delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş Delta kuralı veya geri besleme (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir[2].

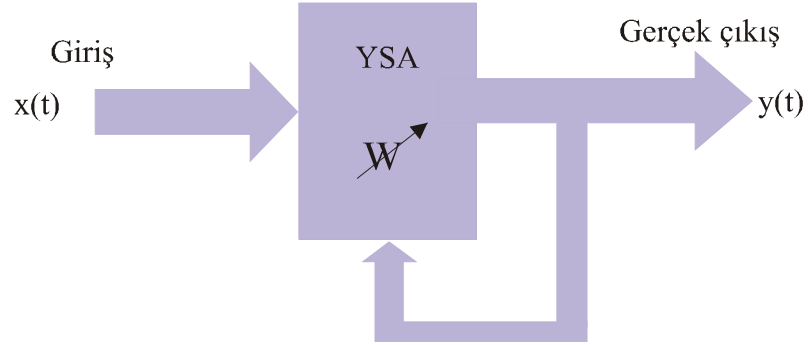


Şekil II.16 Danışmanlı Öğrenme Yapısı.

II.3.2.2. Danışmansız Öğrenme (Unsupervised Learning)

Bu tür öğrenme stratejisinde sistemin öğrenmesine yardımcı olan herhangi bir öğretmen yoktur. Girişe verilen örnekten elde edilen çıkış bilgisine göre ağ sınıflandırma kurallarını kendi kendine geliştirmektedir. Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek yoktur. Öğrenme süresince sadece giriş bilgileri verilir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler (patterns) oluşturmak üzere ayarlar. Şekil II.17' de danışmansız öğrenme yapısı gösterilmiştir. Grossberg tarafından geliştirilen ART

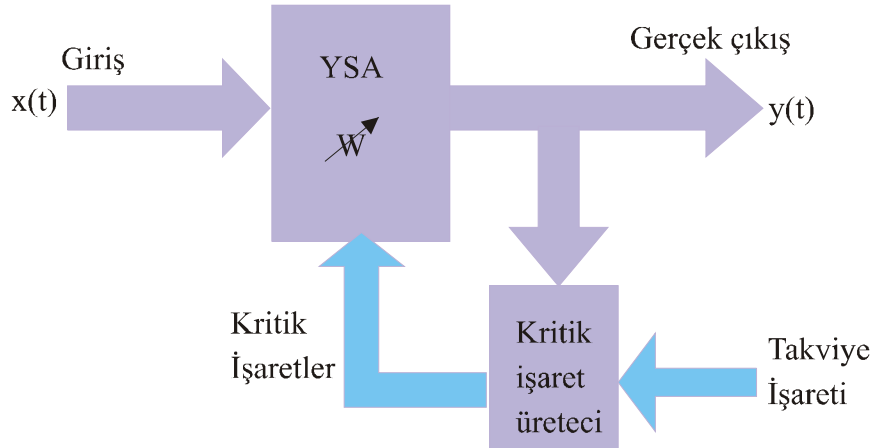
(Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir[2].



Şekil II.17 Danışmansız Öğrenme Yapısı.

II.3.2.3. Takviyeli öğrenme (Reinforcement learning)

Bu öğrenme kuralında da öğrenen sisteme bir öğretmen yardımcı olur. Fakat öğretmen her girdi seti için olması gereken (üretilmesi gereken) çıktı setini sisteme göstermek yerine, sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Şekil II.18’ de takviyeli öğrenme yapısı gösterilmiştir. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski’ nin geliştirdiği Boltzmann kuralı takviyeli öğrenmeye örnek olarak verilebilir[2].



Şekil II.18 Takviyeli Öğrenme Yapısı.

II.3.3. Öğrenme Kurallarına Göre Sınıflandırılması

Yapay sinir ağları gibi öğrenen sistemlerde öğrenme, öğrenme yaklaşımlarından hangisi uygulanırsa uygulansın bazı kurallara göre gerçekleştirilmektedir. Bu kurallar çevrimiçi ve çevrim dışı olmak üzere iki grupta toplanabilir.

II.3.3.1. Çevrimiçi (On-line) öğrenme kuralları

Çevrimiçi öğrenme kuralları gerçek zamanlı çalışabilmektedir. Bu kurallara göre öğrenen sistemler gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte diğer taraftan ise öğrenmeye devam etmektedir. ART (Adaptive Resonance Theory) ağının öğrenme kuralı ile Kohen öğrenme kuralı bu sınıfta bulunan öğrenme kurallarına örnek olarak verilebilir[1].

II.3.3.2. Çevrimdışı (Off-line) öğrenme kuralları

Çevrimdışı öğrenme kurallarına dayalı öğrenen sistemler kullanıma alınmadan önce örnekler üzerinde eğitilirler. Bu kuralları kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında artık öğrenme olmamaktadır. Sistemin öğrenmesi gereken yeni bilgiler söz konusu olduğunda sistem kullanımdan çıkarılmakta ve çevrimdışı olarak yeniden eğitilmektedir. Eğitim tamamlandıca sistem tekrar kullanıma alınmaktadır. Yapay sinir ağlarında yaygın olarak kullanılan “Delta Öğrenme Kuralı” bu tür öğrenmeye örnek olarak verilebilir[1].

II.4. ÖĞRENME KURALLARI

Öğrenme sistemlerinde kullanılan değişik öğrenme kuralları vardır. Yapay sinir ağlarında bu öğrenme kurallarının çoğu Hebb kuralına dayanmaktadır. Öğrenme algoritmalarının bir çoğu aslında Hebb, Delta, Kohonen ve Hopfield olmak üzere dört farklı öğrenme kuralından esinlenilerek geliştirilmiştir.

II.4.1. Hebb Kuralı

Bilinen en eski öğrenme kuralıdır. Diğer öğrenme kurallarının temelini oluşturmaktadır. Bu kurala göre bir işlem elemanı (hücre) diğer bir işlem elemanından bilgi alırsa ve her iki işlem elemanı da aktif ise (matematik olarak aynı işareti taşıyorsa) her iki hücrenin arasındaki bağlantı kuvvetlendirilmelidir. Başka bir deyişle bir hücre kendisi aktif ise bağlı olduğu hücreyi aktif yapmaya, pasif ise pasif yapmaya çalışmaktadır. Hebbien öğrenme kuralı bağlantı ağırlıklarını istenen

çıkışlara ve giriş çarpımlarına göre oransal olarak değiştirir. Hebb öğrenme kuralı danışmanlı öğrenme kurallarından sayılmaktadır. Çünkü öğrenme süreci boyunca doğru çıkışlar kullanılmaktadır.[28].

II.4.2. Hopfield Kuralı

Yapay sinir ağı elemanlarının bağlantılarının ne kadar kuvvetlendirilmesi veya zayıflaması gerektiği belirlenir. Eğer beklenen çıktı ve girdiler ikisi de aktif / pasif ise öğrenme katsayısı kadar ağırlık değerleri kuvvetlendir / zayıflatılmaktadır. 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[2], [28].

II.4.3. Delta Kuralı

Bu kural Hebb kuralının biraz daha geliştirilmiş şeklindedir. Bu kurala göre beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak gerekir. Bu yöntem yapay sinir ağının elemanlarının bağlantılarının ağırlık değerlerinin sürekli değiştirilmesi ilkesine dayanarak geliştirilmiştir. Ağırlık ürettiği çıktı ile üretilmesi gereken (beklenen) çıktı arasındaki hatanın karelerinin ortalamasını olabildiğince azaltmak hedeflenmektedir[1], [2].

II.4.4. Kohonen Kuralı

Kohonen tarafından geliştirilen bu yöntem biyolojik sistemlerdeki öğrenmeden esinlenmiştir. Bu kurala göre ağırlık 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[2].

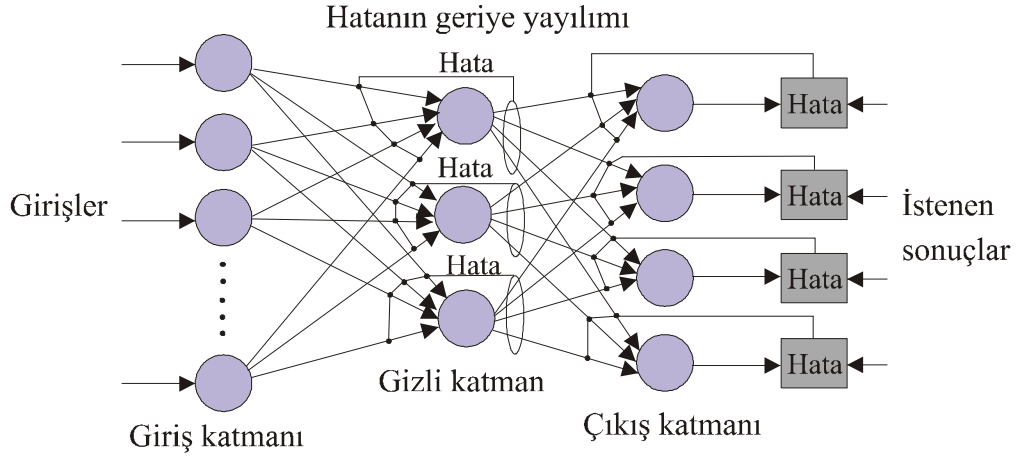
II.5. ÖĞRENME ALGORİTMALARI

Literatürde bulunan bir çok öğrenme algoritmasından sadece tez çalışmasında kullanılan, geri yayımlı öğrenme algoritması ve Levenberg-Marquardt öğrenme algoritması açıklanmıştır. Geri yayılım algoritması danışmanlı öğrenmede kullanılan en genel algoritmadır. Basit olması ve iyi bir öğrenme kapasitesine sahip olması birçok alana uygulanmasını sağlamıştır.

II.5.1. Geri Yayılım Öğrenme Algoritması

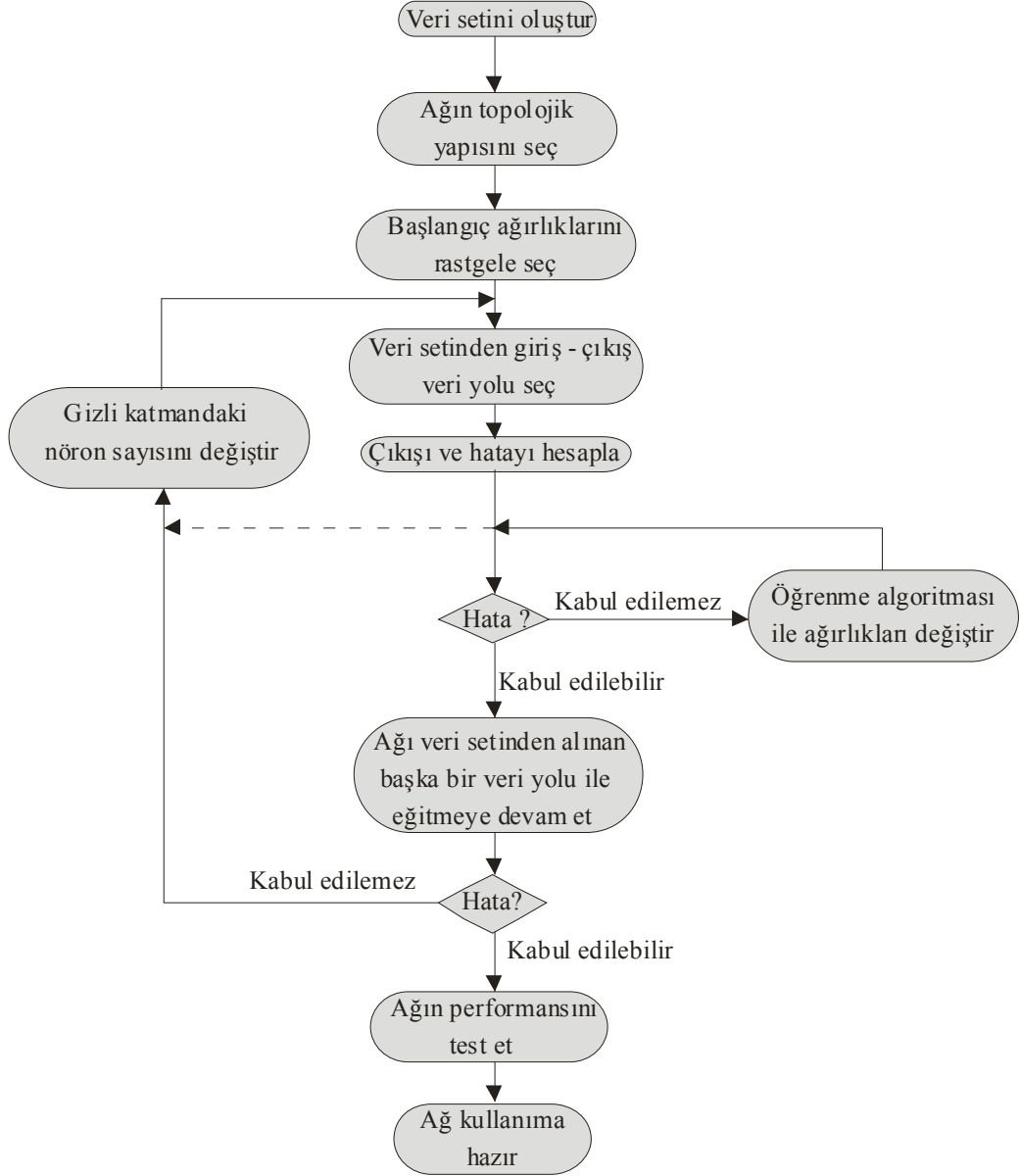
Geri yayılım ile öğrenen ağlar hiyerarşik yapıdadır. Giriş, çıkış ve en az bir gizli katman olmak üzere üç katmandan oluşurlar. Gizli katman ve gizli katmandaki düğüm sayısı değiştirilebilir. Düğüm sayısının artması ağın hatırlama yeteneğini artırmakla birlikte öğrenme işleminin süresini uzatmaktadır. Düğüm sayısının azaltılması eğitim süresini kısaltmakta fakat hatırlama yeteneğini azaltmaktadır. Giriş katmanındaki her bir düğüm gizli katmandaki her düğüme, gizli katman birden fazla ise bu katmandaki her bir düğüm kendisinden sonra gelen katmandaki her düğüme ve gizli katman çıkışındaki her düğüm çıkış katmanındaki her düğüme bağlıdır. Bir katmandaki hiçbir düğüm kendi katmanındaki diğer bir düğüme bağlı değildir. Her katmanın çıkış değerleri bir sonraki katmanın giriş değerleridir. Bu şekilde giriş değerlerinin ağın girişinden çıkışına doğru ilerlemesine ileri besleme denir. Şekil II.19 'da bir geri yayılım ağı örneği görülmektedir. Geri yayılım ağında hatalar, ileri besleme aktarım fonksiyonunun türevi tarafından, ileri besleme mekanizması içinde kullanılan aynı bağlantılar aracılığıyla, geriye doğru yayılmaktadır. Öğrenme işlemi, bu ağda basit çift yönlü hafıza birleştirmeye dayanmaktadır[2], [24].

Geri yayılım öğrenme yöntemi, türevi alınabilir aktivasyon fonksiyonlarını çok katmanlı herhangi bir ağa uygulayabilir. Bu işlem sistem hatasını veya maliyet işlevini azaltma esasına dayanan bir eniyileme (optimizasyon) işlemidir. Bu yöntemde ağırlık ayarlamaları yapıldığı için 'geri yayılım' ismi kullanılmıştır. Öğrenme fazında, giriş örnekleri ağa belli bir sırada sunulur. Her bir çalışma örneği çıktı (çıkış) örneği hesaplanana kadar katman katman ileri yayılır. Hesaplanan çıktı daha sonra olması beklenenle karşılaştırılıp aradaki fark 'hata' olarak bulunur. Hatalar, katman katman sinaptik ayarlamaların yapıldığı geri besleme bağlantılarında 'girişler' olarak kullanılır. Şekil II. 19 geri yayılım çalışması için değiştirilmiş bir çok katmanlı ileri beslemeli ağı göstermektedir. Geriye doğru olan bağlantılar sadece 'öğrenme fazı' için kullanılırken, ileri doğru olan bağlantılar hem öğrenme amacıyla hem de işlemsel fazlar için kullanılır[2].



Şekil II.19 Genel Bir Geri Yayılım Ağ Yapısı [2].

Geri yayımlı öğrenme kullanıldığında, sonraki katmanların hataları kullanılarak gizli katmanın ağırlıkları ayarlanır. Böylece çıkış katmanında hesaplanan hatalar son gizli katman ile çıkış katmanı arasındaki ağırlıklar ayarlanır. Aynı biçimde, bu işlemler ilk gizli katmana kadar tekrarlanır. Bu yolla hatalar katman katman ilgili katmanın ağırlık düzeltmeleri yapılarak geriye doğru yayılır. Tamamlanan çalışma süresi içinde 'toplam hata' en aza indirilinceye kadar bu işlemler tekrarlanır. Şekil II. 20' de ileri beslemeli çok katmanlı bir ağ için geri yayılım öğrenme algoritmasının akış şeması gösterilmektedir[2],[24]



Şekil II.20 İleri Beslemeli Bir Ağ İçin Geri Yayılım Akış Şeması [24]

Tez çalışmasında MATLAB R2006b programı kullanılmıştır. Geri yayımlı öğrenmede ağ değişkenleri ve modelleri Matlab’ daki gösterim şekli kullanıldığında aşağıdaki gibi ifade edilmektedirler.

p : Skaler giriş

w : Skaler ağırlık

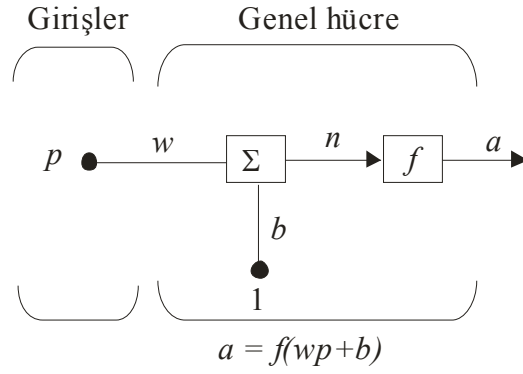
b : Dengeleme sabiti (offset)

n : Toplama fonksiyonunun çıkışı (net giriş)

a : Skaler nöron çıkışı

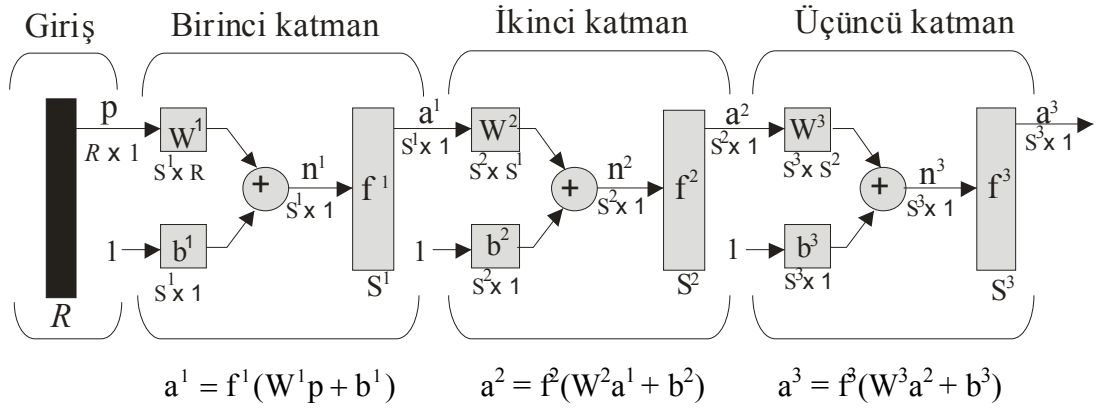
m : katman sayısı

S : Hassasiyet



Şekil II.21 Tek Katmanlı Hücrenin Matlab’ da Kısaltılmış Şekli

Şekil II.21’ de tek katmanlı bir hücre girişinin Matlab daki kısaltılmış görüntüsü gösterilmektedir. Üç tabakalı bir ağı kısaltılmış görüntüsü ise şekil II.22’ de gösterilmektedir[4].



Şekil II.22 Üç Katmanlı YSA İçin Kısaltılmış Gösterim

Şekil II.22’ ye göre üç katmanlı ileri beslemeli bir yapay sinir ağının çıkışı:

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) \quad (II. 7)$$

Geri yayılım algoritması aşağıdaki adımlardan oluşmaktadır.

1. *İleri hesaplama* : Ağ boyunca ileri doğru girişlerin yayılması.
2. *Geriye hesaplama* : Ağ boyunca geriye doğru hassasiyetlerin (sensitivities) yayılması
3. *Ağırlıkların dağılımı* : Ağırlıklar ve dengeleme katsayılarının hatanın karesel farkı (mean square error) alınarak *en dikey iniş* “steepest descent algorithm” algoritması kullanılarak güncellenmesi [10].

II.5.1.1. İleri hesaplama

Çok katmanlı ağlarda bir katmanın çıkışı sonra gelen katmanın girişi olmaktadır ve denklem II.8' deki gibi ifade edilmektedir.

$m = 0, 1, 2, \dots, M-1$, için

$$a^{m+1} = f^{m+1}(W^{m+1} a^m + b^{m+1}) \quad (\text{II.8})$$

M , ağdaki katman sayısıdır. İlk katmana dışardan gelen girişler :

$$a^0 = p \quad (\text{II.9})$$

Nöronların son katmandaki çıkışları :

$$a = a^M \quad (\text{II.10})$$

olmaktadır[4].

II.5.1.2. Geriye hesaplama

Geri yayılım öğrenme algoritmasının ikinci basamağı olan geriye doğru hesaplamada hassasiyetlerin (sensitivities) ağ boyunca geriye doğru yayılımı hesap edilmektedir. Geri yayılım algoritmasında ortalama hata karesi (mean square error) kullanılmaktadır. Geri yayılım algoritmasını oluşturan, girişler ile çıkışlar arasındaki bağıntılardır.

$$\{p_1, t_1\}, \{p_2, t_2\}, \{p_3, t_3\}, \dots, \{p_q, t_q\} \quad (\text{II.11})$$

p_q , ağın girişi, t_q , ağın girişlerine karşılık gelen ağın varmak istediği sonuçlardır veya veri setinde girişlere karşılık gelen hedef (target) çıkışlardır[5]. Algoritmada amaç ortalama hata karesini en aza indirmektir. Ortalama hata karesi (Mean square error) :

$$F(x) = E [e^2] = E [(t - a)^2] \quad (\text{II.12})$$

Burada x , ağın ağırlık ve dengeleme (ofset) vektörüdür. Eğer ağ çok katmanlı bir ağ olursa çıkışları da çoklu çıkışlar olacağından yukardaki ifade şu şekilde genelleştirilir[8], [4].

$$F(x) = E [e^T e] = E [(t - a)^T (t - a)] \quad (\text{II.13})$$

K , iterasyon sayısı olmak üzere ortalama hata karesinin genelleştirilmiş son şekli aşağıdaki gibi olmaktadır.

$$\hat{F}(x) = [(t_{(k)} - a_{(k)})^T (t_{(k)} - a_{(k)})] = e_{(k)}^T e_{(k)} \quad (\text{II.14})$$

Ortalama hata karesi için en dikey iniş algoritması (steepest descent algorithm)

:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m} \quad (\text{II.15})$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m} \quad (\text{II.16})$$

α , öğrenme oranıdır. Çok katmanlı ağlarda hata, gizli katmanlarda ağırlıkların doğrusal olarak değişen belirgin bir fonksiyonu olmadığı için yukardaki kısmi türev hesapları tek katmanlı doğrusal ağlarda (ADALINE) olduğu gibi kolay hesaplanamamaktadır. Bu yüzden türev alma işlemini kolaylaştırmak için zincir kuralı uygulanmaktadır. Zincir kuralı için F , n 'nin belirgin bir fonksiyonu ise, F fonksiyonunun üçüncü değişken olan w değişkenine göre türevi :

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (\text{II.17})$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i} \times \frac{\partial n_i^m}{\partial b_i^m} \quad (\text{II.18})$$

Yukardaki eşitliklerin her ikisinde ikinci terimlerinin türevleri, ağırlık (m) katmanının girişleri, ağırlıkların ve dengelemelerin (ofsets) belirgin bir fonksiyonu olduğu için kolayca hesaplanabilir[4].

$$n_i^m = \sum_{j=1}^{s^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \quad (\text{II.19})$$

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \quad \frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (\text{II.20})$$

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m} \quad (\text{II.21})$$

Yukardaki denklemde s_i^m , F fonksiyonunun, m katmanındaki ağ girişinin i' ninci elemanındaki değişim için hassasiyeti göstermektedir ve aşağıdaki gibi basitleştirilebilir[4], [8].

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} \quad (\text{II.22})$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m \quad (\text{II.23})$$

$$\mathbf{s}^m = \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{S^m}^m} \end{bmatrix} \quad (\text{II.24})$$

Hassasiyetleri hesaplanırken aşağıdaki Jakobiyen (Jacobian) matrisinden yararlanılmaktadır.

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \equiv \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{S^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{S^m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{S^{m+1}}^{m+1}}{\partial n_{S^m}^m} \end{bmatrix} \quad (\text{II.25})$$

i ve j matrisin elemanları olmak üzere :

$$\begin{aligned}\frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial \left(\sum_{l=1}^{s^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \\ &= w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} \dot{f}^m(n_j^m)\end{aligned}\quad (\text{II.26})$$

f fonksiyonunun türevi :

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m} \quad (\text{II.27})$$

Bu durumda Jakobiyen matrisi aşağıdaki gibi yazılabilir[4].

$$\frac{\partial n^{m+1}}{\partial n^m} = W^{m+1} F^m(n^m) \quad (\text{II.28})$$

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \quad (\text{II.29})$$

Zincir kuralını kullanarak hassasiyet için bağıntıyı matris formunda tekrar yazacak olursak :

$$\begin{aligned}\mathbf{s}^m &= \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}\end{aligned}\quad (\text{II.30})$$

Hassasiyetler ağ boyunca son katmandan ilk katmana doğru geriye doğru yayılırsa :

$$\mathbf{s}^M \longrightarrow \mathbf{s}^{M-1} \longrightarrow \dots \longrightarrow \mathbf{s}^2 \longrightarrow \mathbf{s}^1 \quad (\text{II.31})$$

Son katman için elde edilen hassasiyet[4] :

$$s_i^M = \frac{\partial \hat{F}}{\partial n_i^M} = \frac{\partial (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})}{\partial n_i^M} = \frac{\partial \sum_{j=1}^{S^M} (t_j - a_j)^2}{\partial n_i^M} = -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M} \quad (\text{II.32})$$

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = \dot{f}^M(n_i^M) \quad (\text{II.33})$$

$$s_i^M = -2(t_i - a_i) \dot{f}^M(n_i^M) \quad (\text{II.34})$$

Matris formunda ifade edilirse :

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}) \quad (\text{II.35})$$

II.5.1.3. Ağırlıkların ve Dengelemelerin Dağılımı

Geri yayılım algoritmasının son basamağı olan ağırlıkların ve dengelemelerin (ofsets veya biases) güncellenmesi, en dikey iniş algoritması (steepest descent algorithm) kullanılarak aşağıdaki formüller ile gerçekleştirilmektedir. (k), iterasyon sayısı, (m) katman sayısıdır[4], [10].

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \quad (\text{II.36})$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \quad (\text{II.37})$$

Matris formunda ifade edilecek olursa[4], [6] :

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (\text{II.38})$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad (\text{II.39})$$

II.5.2. Levenberg- Marquardt Algoritması (LMBP)

Levenberg – Marquardt algoritması Newton yöntemine bir yaklaşımdan ibarettir. $F(x)$ fonksiyonuna sahip olduğumuzu ve x parametresine göre minimize etmek istediğimizi düşünelim. Newton metodunda şöyle olacaktır[4].

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k \quad (\text{II.40})$$

$\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ ve $\mathbf{g}_k \equiv \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ olmak üzere $F(\mathbf{x})$ fonksiyonu :

$$F(x) = \sum_{i=1}^N v_i^2(x) = \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \quad (\text{II.41})$$

Gradyanın j ' ninci elemanı :

$$[\nabla F(x)]_j = \frac{\partial F(x)}{\partial x_j} = 2 \sum_{i=1}^N v_i(x) \frac{\partial v_i(x)}{\partial x_j} \quad (\text{II.42})$$

Matris formunda ifade edilirse :

$$\nabla F(x) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \quad (\text{II.43})$$

Jakobiyan matrisi :

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(x)}{\partial x_1} & \frac{\partial v_1(x)}{\partial x_2} & \dots & \frac{\partial v_1(x)}{\partial x_n} \\ \frac{\partial v_2(x)}{\partial x_1} & \frac{\partial v_2(x)}{\partial x_2} & \dots & \frac{\partial v_2(x)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(x)}{\partial x_1} & \frac{\partial v_N(x)}{\partial x_2} & \dots & \frac{\partial v_N(x)}{\partial x_n} \end{bmatrix} \quad (\text{II.44})$$

Bir sonraki adımda Hesiyan (Hessian) matrisi ifade edilir. k ve j matrisin elemanları olmak üzere :

$$[\nabla^2 F(x)]_{k,j} = \frac{\partial^2 F(x)}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(x)}{\partial x_k} \frac{\partial v_i(x)}{\partial x_j} + v_i(x) \frac{\partial^2 v_i(x)}{\partial x_k \partial x_j} \right\} \quad (\text{II.45})$$

Hesiyan matrisi matris formunda ifade edilecek olursa :

$$\nabla^2 F(x) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \quad (\text{II.46})$$

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(x) \nabla^2 v_i(x) \quad (\text{II.47})$$

$\mathbf{S}(\mathbf{x})$ ihmal edilebilecek kadar küçük olduğu için sıfır kabul edilir ve bu durumda Hesiyan matrisi :

$$\nabla^2 F(x) \cong 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) \quad (\text{II.48})$$

Eğer II.40 denkleminde II.48 ve II.43 denklemleri yerlerine koyulursa *Gauss – Newton* yöntemi elde edilir [10]:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - [2\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} 2\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \\ &= \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \end{aligned} \quad (\text{II.49})$$

Gauss – Newton yönteminde ikinci türev hesaplamalarına gerek duyulmadığı için standart *Newton* yönteminden daha avantajlıdır. *Gauss – Newton* yönteminin bir dezavantajı, $\mathbf{H} = \mathbf{J}^T \mathbf{J}$ matrisi tersinir (invertible) değildir. Bu problem Hesiyan matrisi yaklaşımı kullanılarak çözülebilir.

$$\mathbf{G} = \mathbf{H} + \mu\mathbf{I} \quad (\text{II.50})$$

Bu matrisin tersinir olabilmesi için, $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ \mathbf{H} matrisinin özdeğerleri ve $\{z_1, z_2, \dots, z_n\}$ ise matrisin özvektörleri olduğu varsayılırsa :

$$\mathbf{G}\mathbf{z}_i = [\mathbf{H} + \mu\mathbf{I}]\mathbf{z}_i = \mathbf{H}\mathbf{z}_i + \mu\mathbf{z}_i = \lambda_i\mathbf{z}_i + \mu\mathbf{z}_i = (\lambda_i + \mu)\mathbf{z}_i \quad (\text{II.51})$$

\mathbf{G} ' nin özvektörleri ile \mathbf{H} ' nin özvektörleri aynı olduğu için \mathbf{G} ' nin özdeğerleri $(\lambda_i + \mu)$ olur. Bütün i değerleri için $(\lambda_i + \mu) > 0$ olduğu sürece \mathbf{G} matrisi pozitif ve tersinir (invertible) olur. *Levenberg – Marquardt* algoritmasında *Gauss – Newton* yöntemi şu şekilde de ifade edilebilir.

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \quad (\text{II.52})$$

Levenberg – Marquardt algoritmasının çok yararlı bir özelliği, μ_k arttırılırsa algoritma düşük öğrenme oranı ile en dikey iniş algoritmasına (steepest descent algorithm) yaklaşır. μ_k 'nın büyük olduğu değerler için :

$$\mathbf{x}_{k+1} \cong \mathbf{x}_k - \frac{1}{\mu_k} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) = \mathbf{x}_k - \frac{1}{2\mu_k} \nabla F(\mathbf{x}) \quad (\text{II.53})$$

μ_k sifira doğru azaltılırsa algoritma Gauss – Newton algoritması olur. Algoritma μ_k ile başlar. μ_k küçük değerler alır (örneğin $\mu_k = 0,01$). Eğer bir adım $F(\mathbf{x})$ in küçük değerleri için yeterli gelmez ise μ_k , bir katsayı ile çarpılarak adım tekrarlanır. Bu katsayı birden büyük olmalıdır ($\nu > 1$). En sonunda $F(\mathbf{x})$ azaltılırsa en dikey iniş (steepest descent) algoritması yönünde küçük bir yaklaşım yapmış oluruz. Eğer bir adım $F(\mathbf{x})$ in çok daha fazla azalmasına neden olmuş ise o zaman μ_k , bir sonraki adımda algoritma Gauss – Newton algoritmasına yaklaşması için (ν) faktörüne bölünerek hızlı bir yaklaşım sağlanmış olunur. Algoritma, Newton yönteminin hızı ile en dikey inişin (steepest descent) yakınsaması arasında hoş bir uzlaşma sağlar[4].

Çok katmanlı bir ağı Levenberg – Marquardt algoritması kullanarak eğitmek için ortalama hata karesi (mean square error) kullanılır. Eğer her hedef eşit olasılık ile ortaya çıkar ise, ortalama hata karesi, eğitim setinde Q hedefleri üzerinden hataların karelerinin toplamı ile orantılıdır demektir.

$$\begin{aligned} F(\mathbf{x}) &= \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q) \\ &= \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q = \sum_{q=1}^Q \sum_{j=1}^{S^M} (e_{j,q})^2 = \sum_{i=1}^N (v_i)^2 \end{aligned} \quad (\text{II.54})$$

$e_{j,q}$, q ' nuncu giriş / hedef (input / target) için hatanın j ' ninci elemanıdır.

LMBP algoritmasında kilit rolü olan adım, Jakobiyen (Jacobian) matrisinin hesaplanmasıdır. Bu hesaplamayı gerçekleştirmek için geri yayılım algoritmasının bir değişimi kullanılmaktadır. Standart bir geri yayılım algoritmasında hataların karelerinin ağırlık ve dengelemelere göre türevi alınır. Jakobiyen matrisini

oluşturmak için hataların karelerinin türevi yerine hataların türevi alınması gerekmektedir. Hata vektörü :

$$\begin{aligned}\mathbf{v}^T &= [v_1 \quad v_2 \quad \cdots \quad v_N] \\ &= [e_{1,1} \quad e_{2,1} \quad \cdots \quad e_{S^M,1} \quad e_{1,2} \quad \cdots \quad e_{S^M,Q}]\end{aligned}\quad (\text{II.55})$$

Parametre vektörü :

$$\begin{aligned}\mathbf{x}^T &= [x_1 \quad x_2 \quad \cdots \quad x_n] \\ &= [w_{1,1}^1 \quad w_{1,2}^1 \quad \cdots \quad w_{S^1,R}^1 \quad b_1^1 \quad \cdots \quad b_{S^1}^1 \quad w_{1,1}^2 \quad \cdots \quad b_{S^M}^M]\end{aligned}\quad (\text{II.56})$$

$N = Q \times S^M$ ve $n = S^1(R+1) + S^2(S^1+1) + \cdots + S^M(S^{M-1}+1)$, dur. Bu denklemleri II.43 Jakobiyen matrisi eşitliğinde yerlerine koyarsak çok katmanlı bir ağı eğitilmesinde kullanılan Jakobiyen matrisi aşağıdaki gibi yazılabilir[4].

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \cdots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \cdots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \end{bmatrix}\quad (\text{II.57})$$

Jakobiyen matrisinin terimleri geri yayılım algoritmasına basit bir değişim uygulayarak hesap edilebilir.

$$\frac{\partial \hat{F}(x)}{\partial x_l} = \frac{\partial e_q^T e_q}{\partial x_l}\quad (\text{II.58})$$

Jakobiyen matrisinin elemanlarının hesap edilmesi LMBP algoritması için gereklidir.

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l} \quad (\text{II.59})$$

Geri yayılım algoritmasında II.17' deki eşitlikten hassasiyet, eşitlik II.21' de belirtilmişti. LMBP algoritması için Marquardt hassasiyeti (*Marquardt sensitivity*) adı altında yeni bir hassasiyet tanımlanmaktadır.

$$\tilde{s}_{i,h}^m \equiv \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \quad (\text{II.60})$$

Eşitlik II.55' den $h = (q-1)S^M + k$

Jakobiyen matrisinin elemanları ise :

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = s_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m \times a_{j,q}^{m-1} \quad (\text{II.61})$$

Eğer $x_l =$ dengeleme (bias veya ofset) olursa :

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial b_i^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m \quad (\text{II.62})$$

Marquardt hassasiyetleri, son katmanda yapılabilecek bir değişiklik ile standart geri yayılım algoritmasındaki hassasiyetler ile aynı bağıntılar yardımıyla hesaplanabilir[4].

Son katmandaki Marquardt hassasiyetleri :

$$\begin{aligned} \tilde{s}_{i,h}^m &= \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} = \frac{\partial (t_{k,q} - a_{k,q}^M)}{\partial n_{i,q}^M} = -\frac{\partial a_{k,q}^M}{\partial n_{i,q}^M} \\ &= \begin{cases} -\dot{f}^M(n_{i,q}^M) & \text{for } i = k \\ 0 & \text{for } i \neq k \end{cases} \end{aligned} \quad (\text{II.63})$$

p_q girişi ağa uygulandığında ve bu girişe karşılık gelen ağ çıkışı a_q^M hesaplandığında, Levenberg – Marquardt geri yayılımı aşağıdaki eşitlik ile başlatılır.

$$\tilde{\mathbf{S}}_q^M = -\dot{\mathbf{F}}^M(\mathbf{n}_q^M) \quad (\text{II.64})$$

$\dot{\mathbf{F}}^M(n^M)$ eşitlik II.29' da geri yayılım algoritmasında tanımlanmaktadır. $\tilde{\mathbf{S}}_q^M$ matrisinin her sütunu, Jakobiyen matrisinin bir satırını üretmek için, eşitlik II.30 kullanılarak ağ boyunca geriye yayılır. Sütunlar ayrıca aşağıdaki eşitlik ile geriye yayılabilir :

$$\tilde{\mathbf{S}}_q^m = \dot{\mathbf{F}}^m(\mathbf{n}_q^m)(\mathbf{W}^{m+1})^T \tilde{\mathbf{S}}_q^{m+1} \quad (\text{II.65})$$

Her katman için toplam hassasiyet matrisleri her giriş için hesaplanan ilave matrisler ile bulunur :

$$\tilde{\mathbf{S}}^m = \left[\tilde{\mathbf{S}}_1^m \mid \tilde{\mathbf{S}}_2^m \mid \dots \mid \tilde{\mathbf{S}}_Q^m \right] \quad (\text{II.66})$$

Ağa sunulan her giriş için \mathbf{S}^M hassasiyet vektörleri geriye doğru yayılır. Bunun nedeni hataların karelerinin toplamalarının türevlerinden başka her bir hatanın da türevlerinin hesaplanmasıdır. Ağa uygulanan her giriş için \mathbf{S}^M hataları olacaktır (ağ çıkışının her elemanı için). Her hata için Jakobiyen matrisinde bir satır olacaktır.

Hassasiyetlerin geri yayılımından sonra Jakobiyen matrisi II.61 ve II.62 eşitlikleri kullanılarak hesap edilir.

Levenberg – Marquardt geri yayılım algoritmasının (LMBP) iterasyonu aşağıdaki gibi ifade edilebilir [4], [10], [27].

1. Eşitlik II.9 ve II.10 kullanılarak ağa bütün girişler sunulur ve karşılık gelen çıkışlar ile hatalar hesaplanır, $e_q = t_q - a_q^M$. Toplam hata karesi, $F(x)$, bütün girişler için II.54 eşitliği kullanılarak hesap edilir.
2. Jakobiyen matrisi hesap edilir, (eşitlik II.57). Hassasiyetler II.64 ve II.65 eşitlikleri kullanılarak hesap edilir. II.66 eşitliği kullanılarak Marquardt hassasiyetlerinde ki tek matrisler büyütülür. Jakobiyen matrisinin elemanları II.61 ve II.62 eşitlikleri kullanılarak hesaplanır.
3. $\Delta \mathbf{x}_k$ 'yı elde etmek için II.52 eşitliği çözülür.
4. Hataların kareleri toplamı $\mathbf{x}_k + \Delta \mathbf{x}_k$ kullanılarak tekrar hesap edilir. Eğer bu yeni hata kareleri toplamı birinci adımda hesaplanandan daha

küçük çıkarsa μ , ν ' ye bölünür ve $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k$ olur ve birinci adıma geri dönülür. Eğer hataların kareleri toplamı azaltılmaz ise bu takdirde μ , ν ile çarpılır ve üçüncü adıma geri dönülür.

Algoritmada hataların kareleri toplamı hedef alınan refrans hataya indirildiği zaman başarıya ulaşıldığı varsayılmaktadır[5].

II.6. ÜSTÜNLÜKLERİ VE DEZAVANTAJLARI

II.6.1. Üstünlükleri

1. YSA öğrenmenin bir modeli olarak düşünülebilir. Örneklerden faydalanarak öğrenme işlemini gerçekleştirirler. YSA yardımıyla bir kararın verilebilmesi için önce ağı eğitilmesi ve performansının test edilmesi gerekmektedir.
2. YSA' larda bilgi ağı bağlantılarının değerleri ile ölçülmekte ve bağlantılarda saklanmaktadır.
3. YSA eksik bilgileri işleyerek sonuca varabilir. Yapay sinir ağları eğitildikten sonra eksik bilgiler ile çalışabilir ve gelen yeni örneklerde eksik bilgi olmasına rağmen sonuç üretebilirler.
4. Hatalara karşı toleranslıdırlar. Yapay sinir ağlarının eksik bilgiler ile çalışabilme yetenekleri hatalara karşı toleranslı olmalarını sağlamaktadır.
5. Sadece nümerik bilgiler ile çalışabilmektedirler. Sembolik ifadeler ile gösterilen bilgilerin nümerik gösterime çevrilmeleri gerekmektedir [27], [1].

II.6.2. Dezavantajları

1. Yapay sinir ağlarının 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 ve aynı zamanda sadece tek bir bilgiyi işleyebilmektedir. Paralel işlemleri seri makinalarda yapmak ise zaman kaybına yol açmaktadır. Bunun yanı sıra bir ağı nasıl

oluşturulması gerektiğini belirleyecek kuralların olmaması da başka bir dezavantajdır.

2. YSA yapısına uygun olarak yerel çözümlere takılabilir. Uygun ağ yapısının belirlenmesi genellikle deneme yanılma yolu ile gerçekleştirildiğinden bu YSA' nın bir dezavantajı olarak görülmektedir.
3. YSA' nın bir çoğunda öğrenme katsayısının, gerekli katman ve katmanlardaki hücre sayılarının belirlenmesinde kesin bir kural olmamaktadır. Gizli katmandaki nöron sayısı, bir tanıma işleminin doğruluğunu ve eğitim hızını etkilemektedir. Örneğin karmaşık örüntüler, gizli katmanda nöron sayısı az olduğunda bulunamamaktadır. Fakat sayının çok büyük olması hesaplama yükünü korkunç şekilde artırmaktadır.
4. YSA' nın eğitilmesinde örnekler önem taşımakta ve örneklerin seçiminde genel bir kural olmamaktadır.
5. Bilgiler ağa “gömülü” olmasından dolayı yorumlanması ve çözümün elde edilmesinde YSA' nın davranışının takibi zordur. Ağın davranışı açıklanamaz. Bir probleme çözüm üretildiği zaman bunun nasıl ve neden üretildiği konusunda bir bilgi bulmak mümkün değildir. Bu ise ağın sonucuna olan güveni azaltmaktadır.
6. Ağın eğitiminin ne zaman bitirileceğine karar vermek içinde geliştirilmiş bir yöntem yoktur. Ağın örnekler üzerindeki hatasının belirli değerin altına indirilmesi eğitimin tamamlanması için yeterli görülmektedir. Fakat sonuçta optimum (en iyi) öğrenmenin gerçekleştiği söylenememektedir. Sadece iyi çözümler üreten bir ağ oluşturu denilmektedir. Optimum sonuçları veren bir mekanizma henüz geliştirilmemiştir.
7. Yapay sinir ağları sadece rakamsal bilgiler ile çalışmaktadırlar. Problemin rakamsal gösterime dönüştürülmesi gerekmektedir. Bu ise kullanıcının becerisine bağlıdır [27], [2].

BÖLÜM III

ASENKRON MOTORLAR

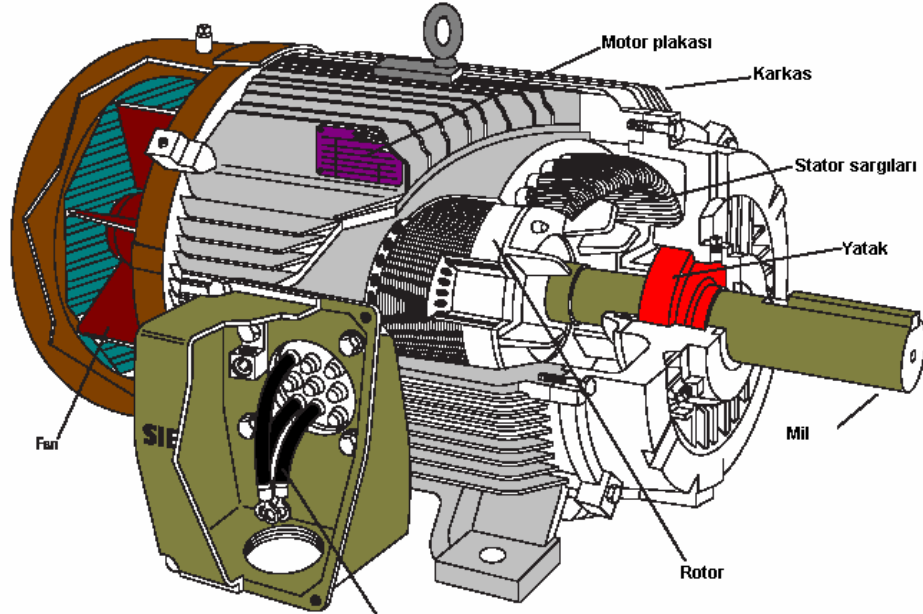
Endüstride kullanılan elektrik makinalarından en sağlam olanı, en az arıza yapanı, en ucuz olanı ve en çok kullanılanı asenkron makinalardır. Çalışma ilkesi bakımından bu makinalara “İndüksiyon Makinaları” da denmektedir. Bu makinalar endüstride genellikle motor olarak çalıştırılırlar; fakat belirli koşulların sağlanması halinde generatör olarak da kullanılabilirler.

Asenkron makinalar bir fazlı, iki fazlı, ve çok fazlı olarak yapılırlar. Günümüzde, fabrikalarda, evlerde, bürolarda bir ve çok fazlı asenkron motorlar büyük ölçüde kullanılmaktadır. Motor olarak çalıştıklarında yarıiletken elemanlar ile kontrolleri yapılabilmektedir[3].

Asenkron motorlar rotorların yapısına göre sincap kafesli ve bilezikli olmak üzere iki sınıfa ayrılır. Faz sayılarına göre bir fazlı, iki fazlı, üç fazlı ve çok fazlı olarak yapılabilmektedir. Tez çalışmasında, üç fazlı sincap kafesli asenkron motorlar kullanıldığı için sadece bu sınıfta olan asenkron motorlar açıklanmaktadır.

III.1. SİNCAP KAFESLİ ASENKRON MOTORLAR

Sincap kafesli asenkron motorların statorlarında normal alternatif akım sargıları bulunur. Şekil III.1’ de sincap kafesli bir asenkron motorun yapısı gösterilmektedir. Rotor sargısı, rotora açılan oluklara yerleştirilir. Rotor olukları, otomatik tezgahlarda kesilerek açılır. Rotor sargısı, oluklara yerleştirilen çıplak çubuklar ve bu çubukları her iki baştan kısa devre eden kısa devre halkalarından oluşur. Kısa devre halkaları ile rotor çubuklarının rotordan, bozulmaksızın çıkarıldığını düşündüğümüzde sincap kafesine benzeyen kısa devre bir sargı sistemi elde olunur. Sincap kafesli motor deyimi rotor sargılarının bu biçimde yapılması nedeni ile kullanılmaktadır. Normal çalışma koşulları altında, rotor çubuklarında endüklenen gerilim 10 Volt’ un altındadır. Bu nedenle, rotor çubukları rotor sac paketlerinden yalıtılmaz.



Şekil III.1 Kalkınan Motorun Yapısı

Rotor saç paketi de, statorda olduğu gibi 0,5 mm.' lik yüzeyleri yalıtılmış silisyumlu saçlardan yapılır. Rotor saç paketi statordan hava aralığına geçen akının magnetik yolunu oluşturur. Sac paketi, rotor sargılarını taşır ve magnetik yoldan endüklenen momenti mile iletir. Küçük motorlarda rotor saç paketi doğrudan doğruya mil üzerine oturtulur. Paketi mile sabitlemek için kamalar kullanılır. Büyük motorlarda ise rotor saç paketi mile sabitlenmiş olan yıldız göbek üzerine ya sıkı geçme ya da kamalar ile sabitlenir.

Mil yüksek H – (Standart büyüklüğü) 56 – 316 mm olarak tanımlanan küçük güçlü motorlarda, yuvarlak çubuklu rotor olukları kullanılır. Kısa devre rotor sargıları, özellikle yol almada, mekanik ve ısıl zorlanmalara dayanabilmelidir. Büyük güçlerde de imal edilmekle beraber, küçük güçlerde imal olunan motorlar genellikle sincap kafesli motorlardır. Bu motorlar dayanıklı az arıza yapan ve en ucuz motorlardır. Sincap kafesli motorlarda, rotor sargısının kutup sayısı, stator döner alanının karşıt endükleme etkisi nedeni ile stator kutup sayısına eşittir. Hava aralığı aynı olan değişik kutuplu sincap kafesli motorların, enerji dönüşümü görüş açısından, rotorları değiştirilebilir. Bununla beraber, rotor kafesinin çubukları farklı ise, motorun moment, yol alma akımı vb. gibi karakteristikleri değişir.

Motorlar değişik kutup sayısında, örneğin $2p = 2, 4, 6, \dots$, için yapılır. Yapılan hesaplardan, rotor saç paketinin mekanik dayanıklılığı bakımından, $2p = 2$ kutuplu

motorların sınır gücü olarak 12.000 kW ve $2p = 4$ kutuplu motorların sınır gücü için yaklaşık 20.000 kW bulunmaktadır[39].

III.1.1. Stator Sargı Yapıları

Tez çalışmasında üç fazlı alternatif akım motorları kullanılmıştır. Üç fazlı sargı, aralarında makina içinde uzayda elektriksel olarak 120° faz farkı olan eşit üç adet bir fazlı sargıdan oluşur. Her stator oluşuna bir bobin yanı yerleştirilerek sarılan üç fazlı sargılara bir tabakalı sargılar, her olukta üst üste iki bobin yanı bulunan üç fazlı sargılara da iki tabakalı (tam kalıp) sargılar denmektedir. Bir tabakalı sargılar yarım kalıp ve el tipi olmak üzere iki gruba ayrılmaktadır.

Sarım şeması çiziminde kullanılan semboller :

x : Oluk sayısı

$2P$: Tek kutup sayısı

P : Çift kutup sayısı

C : Kutupta faz başına düşen oyuk sayısı (Renk sayısı)

m : Faz sayısı

Y_x : Oyuk adımı

α : Oyuk başına düşen elektriki açı

Kullanılan formüller :

Kutupta faz başına düşen oyuk sayısı (Renk sayısı):

$$C = \frac{x}{2P \times m} \quad (\text{III.1})$$

Uzun oyuk adımı :

$$Y_x = \frac{x}{2P} + q \quad \text{veya} \quad Y_x = \frac{x + q}{2P} \quad (\text{III.2})$$

Normal oyuk adımı :

$$Y_x = \frac{x}{2P} \quad (\text{III.3})$$

Kısa oyuk adımı :

$$Y_x = \frac{x}{2P} - q \text{ veya } Y_x = \frac{x - q}{2P} \quad (\text{III.4})$$

Oyuk başına düşen elektriki açısı :

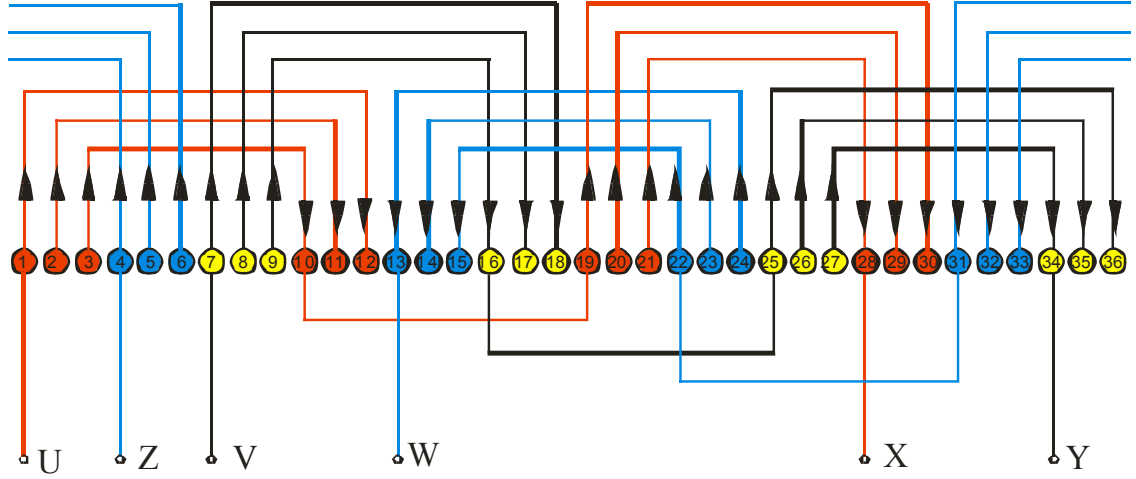
$$\alpha = \frac{360.P}{x} \text{ veya } \alpha = \frac{180.2P}{x} \quad (\text{III.5})$$

III.1.1.1. Dört Kutuplu Dengeli El Tipi Sargı

Üç fazlı dengeli el tipi sarımda kutup sayılarına göre sargılar değişik özellikler göstermektedir. Bu sargı yapıları bir tabakalı farklı adımlı sargılar sınıfına girmektedir. Dört kutuplu dengeli el tipi sargılar şu özellikleri gösterirler :

1. İki katlı olurlar,
2. Her katta her üç fazdan bobin grupları bulunur.,
3. Aynı faza ait bobin gruplarından birinin çıkış ucu, diğerinin giriş ucu ile birleştirilir.(Çıkış - Girişe bağlanır.)
4. Faz girişleri arasındaki elektriki açısı, 120° ' dir ve geometriki açıya eşit değildir.

Stator sarımları çizilirken her bir oyuk küçük bir daire ile, bobin kenarları ise bir çizgiyle ifade edilir. Sarım şemasının karışmaması için sargıların, oyukların altında kalan kısımları çizilmez. Şekil III.2' de 36 oyuklu dengeli el tipi sarım şeması gösterilmektedir. U, V, W uçları enerji girişi, X, Y, Z uçları da sırasıyla enerji çıkışıdır.



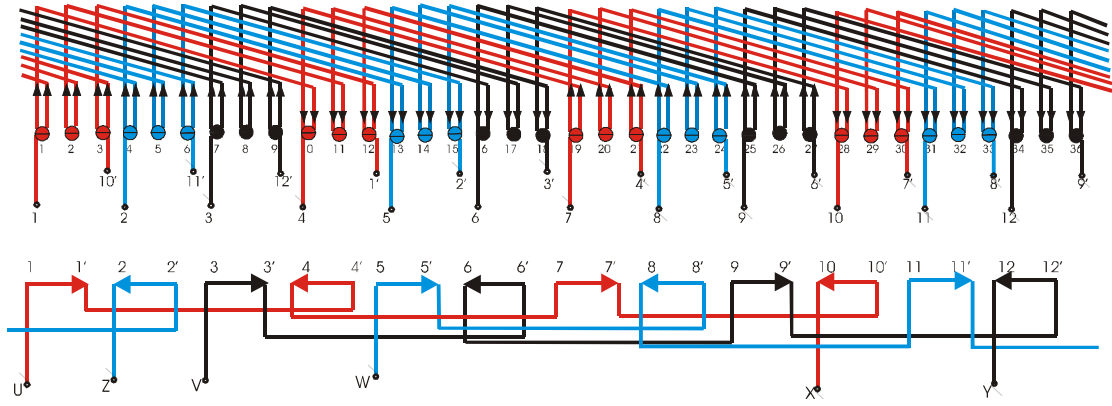
Şekil III.2 Üç Fazlı, 36 Oyuklu, 4 Kutuplu El Tipi Sarım Şeması

Alternatif akım sarımlarında, sarım şemalarında kutuplaşmanın oluşması için daima enerjinin iki fazdan girip bir fazdan çıktığı kabul edilmektedir. Kutuplaşmanın düzgün çıkması bakımından genellikle U – V uçları giriş, W ucu çıkış olarak alınmaktadır. Buna göre U ucundan akımın girdiği varsayılarak bobinler üzerinde akım yönleri ok işaretleri ile belirtilir. Akımın girişini belirten ok bobinin üst kısmına, çıkışını belirten ok bobinin alt kısmına yerleştirilir. Oklandırma sonunda kutuplar oluşur.

III.1.1.2. Tam Kalıp Sargı (İki Tabakalı)

Şekil III.3' de üç fazlı, 36 oyuklu, 4 kutuplu bir asenkron motorun tam kalıp sarım şeması gösterilmektedir. Bobin sayısının oyuk sayısına eşit olduğu bu tip sargılara Tam Amerikan veya Tam Gabare de denilmektedir. Bir oyukta iki bobin kenarı bulunduğu için tam kalıp sargılar, iki tabakalı sargılar sınıfına girmektedir.

Tam kalıp sargılarda adım kısaltma imkanı vardır. Tam kalıp sarım için gerekli olan hesaplama el tipi sarımdaki gibi yapılmaktadır.



Şekil III.3 Üç Fazlı, 36 Oyuklu, 4 Kutuplu Tam Kalıp Sarım Şeması

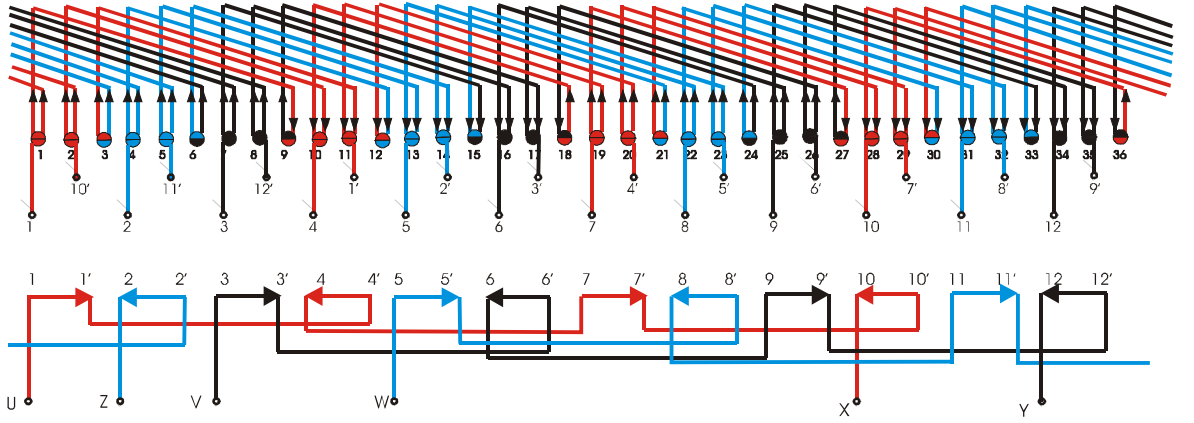
Tam kalıp sarımda işlem sırası şu şekilde olmaktadır.

1. Oyuklar birer daire şeklinde gösterilir.
2. Daireler ortalarında çizilen bir çizgi ile ikiye bölünür.
3. C değerine göre önce üst yarım daireler sırası ile renklendirilir.
4. Y_x değeri kadar atlandıktan sonra alt yarım dairelerde renklendirilir.
5. Giriş kenarlarını temsil eden uzun çizgiler dairenin sol tarafına üst yarım dairenin renginde, çıkış kenarlarını temsil eden kısa çizgiler alt yarım dairenin renginde dairenin sağ tarafına çizilir.
6. Bobin giriş kenarları ile çıkış kenarları üst taraflarından birleştirilir. Bobin giriş ucu sol tarafa uzun çizgi ile çizilir ve üssüz rakam ile belirtilir. Çıkış uçları da sağ tarafa kısa çizgi ile çizilir ve üslü rakamlar ile gösterilir.

Sarım şeması çizimi tamamlandıktan sonra uç bağlantıları yapılır. Sarım şemasının altına her biri bir bobin grubunu temsil eden bobin grubu renginde kısa çizgiler çizilir. Çizgiler sağa doğru başlayan ve birbirine ters yönlü, akım yönünü temsil eden oklar ile yönlendirilir. Bobin grubunun girişini temsil eden çizginin sol tarafına üssüz, çıkışını temsil eden sağ tarafına üslü rakamlar verilir. Aynı renkte (aynı faza ait) çizgiler ok yönünde birleştirilir. Birinci faza ait bobin grubunu temsil eden birinci çizgiden itibaren bir çizgi (bir renk) atlanarak ikinci fazın girişi yapılır ve aynı şekilde ok yönünde birleştirilir. Daha sonra ikinci fazın girişinden itibaren bir çizgi atlanarak üçüncü fazın girişi yapılır ve çizgiler ok yönünde birleştirilir. Oklandırma yöntemine göre kutuplandırma yapılır [42].

İki tabakalı (tam kalıp) sargıların en büyük avantajı; bobinlerin birbirine eşit yapılabilmesi ve bobin genişliğine uygun büyüklük verilerek magnetik harmonik

alanların ve bunların meydana getirdiği kayıpların önlenmesidir. İki tabakalı sargıda, sargı başlarının şekilleri ve durumları havalandırmaya daha elverişlidir. Şekil III. 4' de kısaltılmış adımlı tam kalıp sarım şeması gösterilmektedir. 36 oyuklu, 4 kutuplu tam kalıp bir stator sarım şemasında oyuk adımı $Y_x, 9$ (1-10) bulunur. İki bobin yanı arasındaki açı 180° dir. Her oyuk arasındaki elektriki açı $\alpha, 20^\circ$ bulunur. Oyuk adımını bir oyuk kısaltmak istediğimizde (1 - 9), bu durumdaki bobin yanları arasındaki açı 160° olacaktır ve bağlantı şeması şekil III. 4' de olduğu gibidir [41].



Şekil III.4 Üç Fazlı, 36 Oyuklu, 4 Kutuplu Tam Kalıp Kısa Adımlı Sarım Şeması

III.1.2. Stator Sargılarında Endüklenen Gerilim ve Harmonikler

Stator sargılarında endüklenen gerilimlerin ifadesi ve harmoniklerin incelenebilmesi için ilk önce periyodik fonksiyonların değişimlerinin Fourier Serileri ile gösterilmeleri gerekmektedir.

III.1.2.1. Periyodik Fonksiyonların Fourier Serileri ile Gösterilişi

Elektrik makinalarında rastlanan gerilim, akım, magnetik akı gibi fiziki büyüklükler zaman veya uzaklığın periyodik fonksiyonu olarak değişirler. Bu periyodik değişimler sinüs formunda olmayabilir. Sinüs formunda olmayan ve periyodisitesi $T = 2\pi$ olan bir periyodik fonksiyon, Fourier' e göre argümanları $x, 2x, 3x, 4x, \dots, vx$ olan sinüs ve cosinüs terimlerinden oluşan seriler ile gösterilir.

$$\begin{aligned}
f(x) &= A_1 \sin x + A_2 \sin 2x + A_3 \sin 3x + \dots + A_v \sin vx + \dots \\
&+ B_0 + B_1 \cos x + B_2 \cos 2x + B_3 \cos 3x + \dots + B_v \cos vx + \dots \\
&= \sum_{v=1}^{v=\infty} A_v \sin vx + \sum_{v=0}^{v=\infty} B_v \cos vx
\end{aligned} \tag{III.6}$$

B_0 , verilen fonksiyonun ortalama değerini belirten sabit bir sayıdır. $A_1, A_2, A_3, \dots, A_v$, serideki sinüs terimlerinin genlikleridir. Peryodik bir fonksiyonun x eksenine simetrik olan eğrilerden oluşması durumunda $B_0 = 0$ olur. Bu durumda Fourier serisi eşitlik (III.7) ' de olduğu gibi yazılır[42].

$$\begin{aligned}
f(x) &= A_1 \sin x + A_3 \sin 3x + A_5 \sin 5x + \dots + A_v \sin vx + \dots \\
&+ B_1 \cos x + B_3 \cos 3x + \dots + B_v \cos vx + \dots
\end{aligned} \tag{III.7}$$

III.1.2.2. Bir Sargı İletkeninde Endüklenen Gerilim

Bir çift oluğa yerleştirilmiş olunan stator sargı iletkenlerinde endüklenen elektromotor kuvvetinin birinci harmoniğinin ani değeri :

$$e_1(t) = B_1(x).l.v \tag{III.8}$$

Burada $v = 2.\tau_p.f$ olduğundan, eşitlik (III.7)' de yerine koyularak ve Fourier serisindeki ilk terime dönüşüm yapılarak aşağıdaki eşitlik elde edilir.

$$e_1(t) = B_{1\max}.l.2\tau_p.f.\sin \omega t \tag{III.9}$$

Endüklenen gerilimin birinci harmoniğinin genliği :

$$E_{1\max} = B_{1\max}.l.2.\tau_p.f \tag{III.10}$$

Birinci harmoniğin efektif değeri :

$$E_{1\text{eff}} = \frac{E_{1\max}}{\sqrt{2}} = \sqrt{2}.B_{1\max}.l.\tau_p.f \tag{III.11}$$

v ' ninci harmoniğin efektif değeri :

$$E_{eff} = \frac{E_{v\max}}{\sqrt{2}} = \sqrt{2} \cdot B_{v\max} \cdot l \cdot \tau_p \quad (\text{III.12})$$

Magnetik akı cinsinden yazılacak olursa aşağıdaki eşitli elde edilir [42], [44].

$$E_{veff} = \frac{\pi}{\sqrt{2}} \cdot f_v \cdot \Phi_{v\max} \quad (\text{III.13})$$

III.1.2.3. Bir Çap Bobininde Endüklenen Gerilim

Bir çift oluğa yerleştirilmiş ve bobin yanları arasındaki açı 180° olan bir bobinde endüklenen gerilimin birinci ve v ' ninci harmoniklerinin efektif değerlerini yazmak için eşitlik (III.12), bir bobinde bulunan iletken sayısı 2ω ile çarpılır. Gerilim denklemlerini Φ_{\max} cinsinden yazabilmek için k_f form faktörü kullanılır. Bu faktör $\frac{\Phi_{v\max}}{\Phi_{\max}}$ oranına eşittir. Bu durumda gerilimin birinci ve v ' ninci harmonikleri aşağıdaki (III.14) ve (III.15) eşitliklerinde olduğu gibi yazılabilir[44].

$$E_1 = E_{1eff} = 4,44 \cdot f_1 \cdot k_{f1} \cdot \omega \cdot \Phi_{\max} \quad (\text{III.14})$$

$$E_v = E_{veff} = 4,44 \cdot f_v \cdot k_{fv} \cdot \omega \cdot \Phi_{\max} \quad (\text{III.15})$$

III.1.2.4. Krişlenmiş Bir Bobinde Endüklenen Gerilim

İki bobin yanı arasındaki açının 180° ' den farklı olduğu düşünüldüğünde yada adım kısaltması yapıldığında endüklenen gerilimin hesabı için bobinin krişlenme faktörü kullanılır. Genel olarak v ' ninci harmonik için krişlenme faktörü :

$$\xi_{sv} = \frac{|E|}{|E_{1v}| + |E_{2v}|} \quad (\text{III.16})$$

Genel olarak bu krişlenme faktörü her bir harmonik için ayrı ayrı hesap edilebilir. Eşitlik (III.14) ve eşitlik (III.15) deki bağıntılar krişlenme faktörü de göz önünde tutularak aşağıdaki gibi tekrar yazılır[42].

$$E_1 = E_{1eff} = 4,44 \cdot f_1 \cdot k_{f1} \cdot \xi_{s1} \cdot \omega \cdot \Phi_{max} \quad (III.17)$$

$$E_v = E_{veff} = 4,44 \cdot f_v \cdot k_{fv} \cdot \xi_{sv} \cdot \omega \cdot \Phi_{max} \quad (III.18)$$

III.1.2.5. Endüviye Yayılmış Bir Sargıda Endüklenen Gerilim

Bobinlerde endüklenen gerilimlerin ani değerlerinin toplamı söz konusu sargıda endüklenen toplam gerilimin ani değerini vermektedir. Endüvi çevresine yayılmış bir sargıda endüklenen elektromotor kuvvetleri birbirinin aynısı olup aralarında faz farkları vardır. E_v , bir faza ait, q çift oyuğa yerleştirilmiş, q adet kısmi bobinde endüklenen gerilimlerin toplamıdır. q adet fazörün geometrik toplamlarının cebirsel toplamlarına oranı yayılma faktörü ξ_v ' yi vermektedir. Yayılma faktörü aşağıdaki gibi ifade edilir.

$$\xi_v = \frac{\sum_{i=1}^q E_{vi}}{\sum_{i=1}^q E_{vi}} = \frac{2 \cdot R \cdot \sin \frac{q \cdot \gamma_v}{2}}{2 \cdot q \cdot l \cdot \sin \frac{\gamma_v}{2}} = \frac{\sin q \cdot \frac{\gamma_v}{2}}{q \cdot \sin \frac{\gamma_v}{2}} \quad (III.19)$$

Yayılma faktörü hesaba katılarak sargıda endüklenen gerilimin v ' ninci harmonik ifadesi :

$$E_v = q \cdot \xi_v \cdot E_v \quad (III.20)$$

E_v ' nin eşitlik (III.18)' deki değeri yerine konulursa,

$$E_v = 4,44 \cdot f_v \cdot k_{fv} \cdot \xi_{sv} \cdot \xi_v \cdot q \cdot \omega \cdot \Phi_{max} \quad (III.21)$$

Elde olunur. Burada ω bir çift oluktaki sarım sayısıdır. Bir çift kutup altındaki bir faza ait toplam sarım sayısı $\frac{W}{P} = q \cdot \omega$ olduğundan, eşitlik (III.21) aşağıdaki şekilde yazılabilir.

$$E_v = 4,44 \cdot f_v \cdot k_{fv} \cdot \xi_{sv} \cdot \xi_v \cdot \frac{W}{P} \cdot \Phi_{max} \quad (III.22)$$

W , bir faz veya sargı kolunun toplam sarım sayısıdır[42], [44].

III.1.2.6. Kirişlenme ile Bazı Harmoniklerin Yok Edilmesi

Bir çap bobininde bobin yanları arasında bir kutup taksimatına eşit bir uzaklık vardır. Bobin yanlarında endüklenen gerilimler arasındaki faz farkı π' ye eşittir. Bu faz farkı v' ninci harmonikte $v.\pi'$ ye eşittir. Kirişlenmiş bobinde ise bobin yanları arasındaki y uzaklığı kutup taksimatından farklı olacağından, bobin yanları arasındaki faz farkı γ_1 , $\frac{y}{\tau_p}.\pi$ olacaktır. Kirişlenmiş bobinde, bobin yanlarında endüklenen gerilimlere ait fazörler arasındaki faz farkını π' den farklı yapan miktarı β ile gösterilecek olursa, β açısının v' ninci harmoniği,

$$\beta_v = v.\pi - v.\frac{y}{\tau_p}\pi \quad (III.23)$$

bu açının birinci harmoniği,

$$\beta_1 = \pi - \frac{y}{\tau_p}\pi \quad (III.24)$$

olarak yazılır.

v' ninci harmonik için kirişleme faktörü aşağıdaki gibi hesap edilir.

$$\xi_{sv} = \cos \frac{v.\beta_1}{2} \quad (III.25)$$

Denklem (III.24), (III.25) de yerine konulduğunda bilinen trigonometrik kurallara dayalı olarak aşağıdaki eşitlik elde edilir.

$$\xi_{sv} = \sin \frac{v\pi}{2} . \sin \frac{v\pi}{2} . \frac{y}{\tau_p} \quad (III.26)$$

Burada, $\sin \frac{v\pi}{2}$, v' nin 1, 5, 9 ve 13 değerleri için +1 ve v' nin 3, 7, 11, ve 15

değerleri için de -1 olmaktadır. $y = \tau_p$ olması halinde $\xi_{sv} = \left(\sin \frac{v\pi}{2} \right)^2$ olacağından, v' nin her bir tek değeri için ξ_{sv} daima +1 dir.

Bu durumda v' nin tek değerleri de göz önüne alınarak v' ninci harmonik için sargı faktörü şu şekilde yazılabilir.

$$\xi_{sv} = \pm \sin \frac{v \cdot \pi}{2} \cdot \frac{y}{\tau_p} \quad (\text{III.27})$$

Alanda üçüncü harmoniğin olmadığı ve kuvvetli bir beşinci harmoniğin olduğu varsayılırsa, faz sargısında endüklenen gerilimde beşinci harmoniği yok etmek için bobinler $\frac{1}{5} \cdot \tau_p$ oranında kırışlenir. Başka bir deyişle bobin genişliği $y = \frac{4}{5} \cdot \tau_p$ olarak seçilecek olursa, $\xi_{s5} = 0$ olur. Bunu denemek için, denklem (III.27)' de 'y' yerine $\frac{4}{5} \tau_p$ ve 'v' yerine de 5 yazılırsa :

$$\begin{aligned} \xi_{s5} &= + \sin 5 \cdot \frac{\pi}{2} \cdot \frac{4}{5} \cdot \frac{\tau_p}{\tau_p} \\ &= + \sin 2\pi \\ &= 0 \end{aligned} \quad (\text{III.28})$$

Olur. Buradan görüleceği üzere, $\xi_{s5} = 0$ olması demek, endüklenen gerilimde beşinci harmoniğin bulunmaması demektir. Buna benzetilerek bobinlerin kırışlenmesi yolu ile istenilen her hangi bir harmonik yok edilebilir[44], [45].

III.2. ASENKRON MOTORLARIN KONTROLU

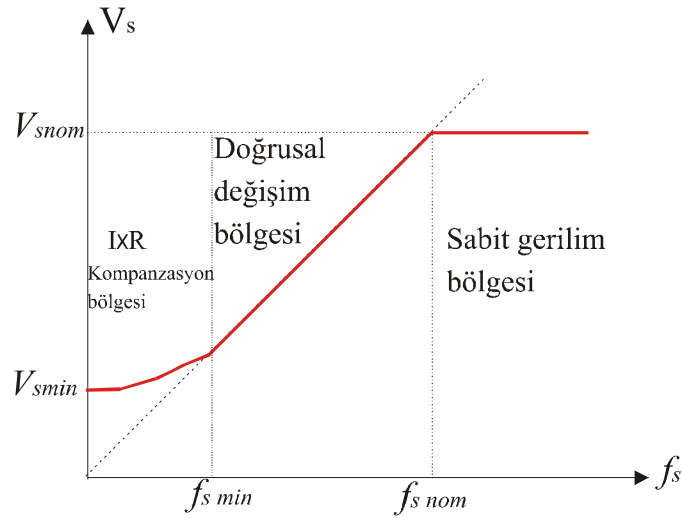
Tez çalışmasında stator geriliminin genlik ve frekansının değiştirilmesi ile hız kontrolünün sağlandığı *skalar kontrol yöntemi* (A.A sürücüsü kullanılarak) kullanıldığı için sadece bu yöntem açıklanacaktır. Asenkron makinelerin değişken hızlı tahrik sistemlerinin kontrolünde stator geriliminin genlik ve frekansının değişimine dayalı yöntemler aşağıdaki gibi iki temel kısma ayrılırlar [53].

1. Skalar kontrol yöntemleri
2. Vektörel kontrol yöntemleri

III.2.1.1. Skalar Kontrol Yöntemleri

Asenkron motorların hız kontrolünde stator geriliminin genlik ve frekansının değiştirilmesi en uygun yöntemdir. Asenkron motorun sürekli rejimde geçerli olan moment ifadesine göre $R_s = 0$ olması koşulu altında gerilim / frekans (V_s / f_s) oranının sabit tutulması ile düşük hızlar dışında makinanın hızının geniş bir aralıkta kontrol edilebildiği görülmektedir. Bu durumda makinanın oluşturabileceği maksimum moment değeri tüm hız kontrol aralığında sabit kalacaktır. Makinanın bağlı olduğu kaynaktan çektiği akım yine belli bir yük için tüm hız aralığı boyunca

değişmeyecektir. Ancak bu yöntemde, stator geriliminin genliğinin az olduğu düşük hız bölgelerinde stator direnci önemli hale gelir. Motora nominal yükünün uygulanması durumunda, bu çalışma bölgesinde gerekli moment değerinin sağlanması için makinaya uygulanan gerilime oranla oldukça büyük değerde olan $R_s I_s$ gerilim düşümünün de karşılanması gerekmektedir. Bu değer yol alma sırasında önemli olup motorun başlangıçta üretmesi gereken yol alma momentinin değerini etkilemektedir. Bu nedenle özellikle düşük hız bölgelerinde gerilimin genliği, bahsedilen gerilim düşümünü kompanse edebilecek şekilde gerilim/frekans oranının belirlediği değerden daha yüksek seçilmelidir. I.R kompanzasyonu olarak adlandırılan bu yöntem asenkron makinalar için sürücü imal eden firmalar tarafından yaygın olarak kullanılmaktadır. Şekil III.5’ de stator geriliminin genlik ve frekansının değişimleri gösterilmektedir[53].



Şekil III.5 Gerilim/Frekans Değişim Eğrisi

III.2.2. Sinüzoidal Olmayan Beslemenin Asenkron Motora Etkisi

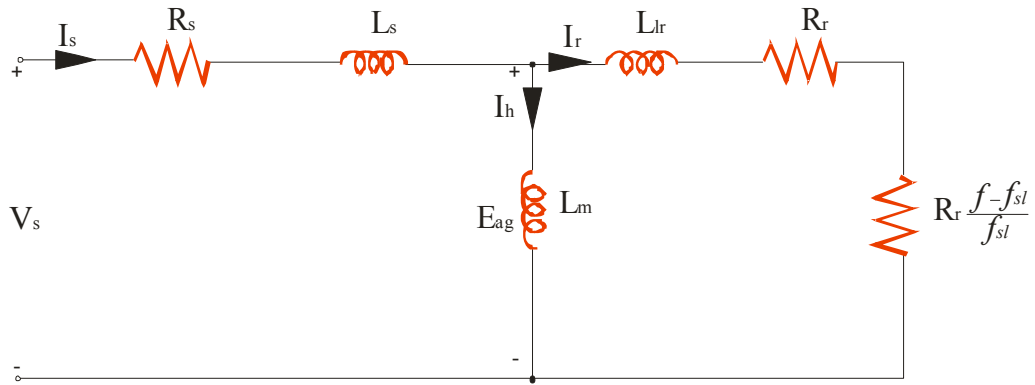
Asenkron motorların üç fazlı dengeli ve sinüzoidal gerilimlerle beslendikleri varsayılmaktadır. Pratikte, değişken frekanslı kontrolörler, her fazda birbirinin aynı, sadece aralarında 120° şer derece faz farkı olan üç fazlı gerilim veya akımlar oluştururlar. Ancak oluşturulan dalga şekli saf sinüs biçiminde olmayıp temel frekans bileşeninin yanı sıra yüksek frekanslı harmonikler içerirler.

III.2.2.1. Motor Harmonik Akımları

Birinci dereceden bir yaklaşım olarak gerilimin harmonikli olması durumunda akım, şekil III.6' daki faz eşdeğer devresinden yararlanılarak her bir h harmoniğ için i_h harmonik akım bileşeni hesaplanabilir. Herhangi bir h harmoniğinde, gerilimin bileşenleri (v_{ah} , v_{bh} , v_{ch}) tarafından üretilen akı hava aralığında $\omega_{sh} = h\omega_s$ hızı ile döner. $H=6n-1$ ($n=1, 2, 3, \dots$) harmoniğinin ürettiği akının, temel bileşenlere göre ters yönde dönecektir. Böylece bu harmonikler rotora göre ters yönde dönen bir akı oluştururlar. $h = 6n + 1$ ($n = 1, 2, 3, \dots$) şeklindeki harmonikler ise rotorla aynı yönde dönen bir akı oluştururlar. Hız kontrolü için değişken frekanslı çalışmada motor oldukça düşük bir kayma değerinde döner ; böylece, yine birinci dereceden bir yaklaşımla, rotor hızının temel bileşen senkron hızıyla aynı olduğu varsayılabilir. ($\omega_r \cong \omega_s$). Bu nedenle şekil III.6' daki eşdeğer devrenin bir h harmoniğinde, o harmonik frekansına ilişkin senkron hıza karşı düşen rotor kayması :

$$s_h = \frac{\omega_{sh} \pm \omega_r}{\omega_{sh}} \cong \frac{h \pm 1}{h} \approx 1 \quad (III.33)$$

Şeklinde dir. h harmonik frekansındaki yaklaşık eşdeğer devre şekil III.7' de verilmiştir. Bu devre faz eşdeğer devresinde L_m ihmal edilerek elde edilir. Motorun bir gerilim kaynağından beslendiği ve v_s stator harmonik gerilimlerinin bilinmesi durumunda, buna karşı düşen akım harmonikleri her bir harmonik için teker teker şekil III.7' deki harmonik eşdeğer devre kullanılarak bulunabilir.



Şekil III.6 Asenkron Motorun Faz Eşdeğer Devresi

Akımın harmonik bileşenlerini hesaplamak için mıknatıslama bileşenleri gelenlikle ihmal edilebilir. Harmonik akım bileşenleri harmonik frekansında kaçak reaktanslarca belirlenir:

$$I_h = \frac{V_h}{h\omega(L_{ls} + L_{lr})} \quad (\text{III.34})$$

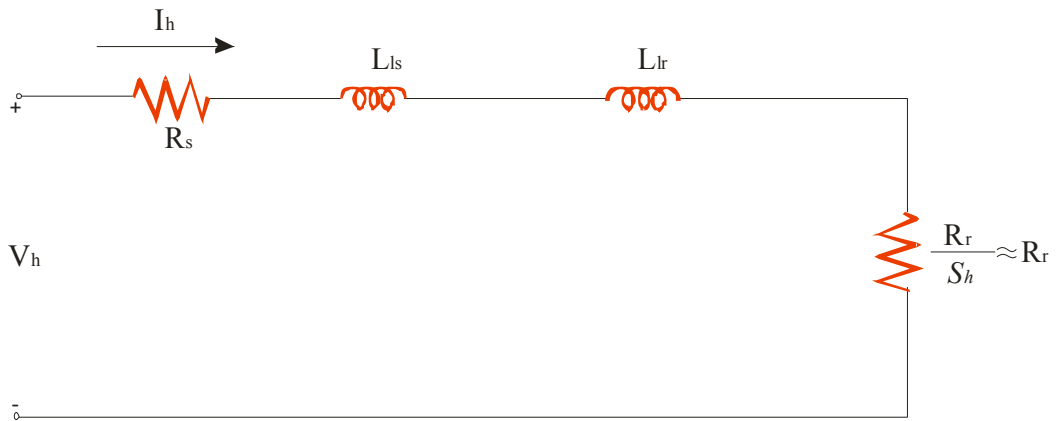
PWM kıyıcının anahtarlama frekansı artırılarak harmonik gerilimlerinin olduğu çevirici çıkış frekansı yükseltilir. Böylelikle harmonik akımlarının genliği düşürülebilir. Ancak harmonik akımlarının bu hesaplama yöntemi kaçak reaktans ve dirençler frekansla değiştiği için sadece birinci dereceden bir yaklaşıma dayanmaktadır[54].

III.2.2.2. Harmonik Kayıplar

Harmonik akımlarından dolayı stator ve rotor sargılarının her bir fazında oluşan ek bakır kayıpları yaklaşık olarak aşağıdaki gibi hesaplanır.

$$\Delta P_{cu} = \sum_{h=2}^{\infty} (R_s + R_r) \cdot I_h^2 \quad (\text{III.35})$$

R_s ve R_r harmonik frekanslarla doğrusal olmayan bir şekilde artar. Harmonik frekansı eddy (demir) akım ve histeresize ilişkin ek çekirdek kayıplarının hesaplanması oldukça uzun işlemler gerektirir. Bu ve diğer ek kayıplar motorun geometrisi, kullanılan manyetik malzeme ve 50 hz' lik frekans için optimize edilebilen lamel kalınlıklarına bağlıdır.



Şekil III.7 Asenkron Motorun Her Faza İlişkin Harmonik Eşdeğer Devresi

III.2.2.3. Moment Salınımları

Statoru besleyen kaynaktaki harmoniklerin varlığı momentte salınım bileşenlerine neden olmaktadır. Düşük frekanslarda oluşan salınım momentleri performansı bozan hız salınımlarına ve motor milinde malzeme yorgunluğuna yol açar. Bu moment salınımlarının rotor hızı üzerindeki etkileri rezonans etkisi oluşmayacağı varsayımıyla aşağıdaki gibi ifade edilir.

$$\text{Hız salınımlarının genliği} = k \frac{msg}{f_{salınlı} \times e} \quad (\text{III.36})$$

Burada msg : moment salınımlarının genliği, e : eylemsizlik, k : motor değişkenleri arasındaki ilişkidir. Belli bir moment salınım genliği yüksek frekanslarda ihmal edilebilir seviyede hız salınımlarına neden olabilir.

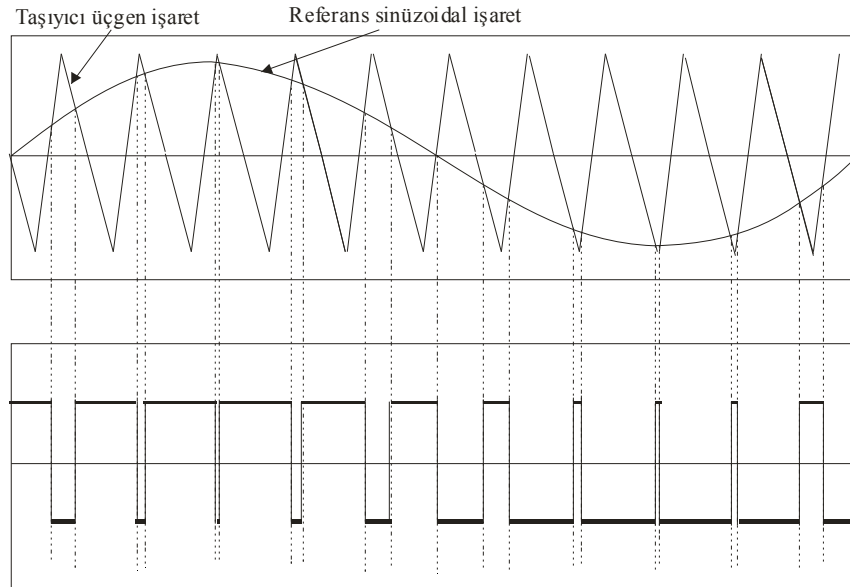
III.2.3. Darbe Genişlik Modülasyonu (PWM)

Değişken hızlı tahrik sistemlerinin endüstriyel uygulamalarında eviriciler kullanılmaktadır. Eviriciler sabit doğru gerilim veya akımdan beslenirler. Bu kaynaktan motor sargılarına uygulanmak üzere sinüzoidal akımlar elde edilmesi için en ekonomik ve uygulanması en kolay yöntem, giriş genliğinin sabit tutulduğu, fakat bu genliğin uygulanma süresinin ya da darbe genişliğinin çıkışta istenilen sinüzoidal işarete göre modüle edilerek, darbe dizisi biçimindeki işaretlerin oluşturulduğu yöntemdir. Bu modülasyon işlemi, darbe genişliklerini istenilen sinüzoidal işaretlere uygun olarak modüle ettiği için darbe genişlik modülasyonu veya PWM olarak adlandırılır. Şekil III.8' de PWM ile elde edilen darbeler ve sinüzoidal gerilim gösterilmiştir. PWM dalga şekli değişik yöntemlerle üretilebilir. PWM dalga şekilleri oluşturabilmek için bazı yöntemlere başvurulur. Kullanılan yöntemlerden sinüs - üçgen karşılaştırması yöntemi en yaygın olanı ve ac sürücülerde en çok kullanılanı olduğu için sadece bu yöntem açıklanacaktır[53].

III.2.3.1. Sinüzoidal PWM (SPWM)

Çıkışta elde edilmek istenen sinüzoidal işaret ile frekansı güç devresindeki anahtarlama frekansına eşit olan bir üçgen dalga karşılaştırılarak kesişme noktalarında darbe üretilmektedir. Şekil III.8' de üst kısımda referans sinüs ve bununla karşılaştırılacak olan üçgen dalga, alt kısımda ise karşılaştırma sonucunda üretilen PWM dalga şekli verilmektedir. Üretilen PWM işarete anahtarların açık ve

kapalı olma sürelerinin toplamı, üçgen işaretin periyotuna eşittir. PWM dalga şeklinin temel bileşeni ise teorik olarak sinüzoidal gerilim referansına eşdeğerdir. Bu şekilde sinüzoidal bir işaretin zamana göre değişen genliği, evirici deki anahtarların açık veya kapalı olma sürelerine, dolayısıyla oluşan darbelerin uzunluklarına karşı düşürülmüş olur. Elde edilen bu PWM işaretleri eviricide yer alan güç anahtar elemanlarına uygulanarak PWM şeklinde gerilimler motor sargılarına uygulanır. Böylece elde edilen dalga şeklinde temel bileşen dışında kalan harmonik etkilerini azaltmak anahtarlama frekansının artırılması ile mümkün olur. Sargılara uygulanan dikdörtgen darbe geriliminin sargılardan akıtıldığı akım ise sargı endüktanslarının filtre etmesi nedeniyle oldukça sinüzoidal bir biçimdedir.



Şekil III.8 Sinüs Üçgen Karşılaştırması İle Elde Edilen PWM Dalga Şekli

PWM gerilim dalga şekilleri çoğunlukla IGBT tipindeki güç anahtar elemanlarının iletme sokulması ile elde edilen “1” ve kesime götürülmesi ile elde edilen “0” seviyelerinden oluşur. Evirici katında yer alan IGBT’lerin iletme geçme ve kesime gitme süreleri toplamı yani anahtarlama periyodunun tersi anahtarlama frekansını verir. Anahtarlama frekansını eviricide yer alan tüm IGBT’ler için eşit ve sabit değerdedir. Buna karşın asenkron makinanın stator sargılarına uygulanacak PWM gerilim dalga şeklinin genlik ve frekansı bu IGBT’lerin iletme gitme ve kesime gitme süreleri değiştirilerek oluşturulur. Anahtarlama frekansının artması, stator sargılarından geçen akımların harmonik bileşenlerinin azalmasına neden olmakla birlikte, IGBT’lerin anahtarlama kayıplarını arttırmaktadır. Motor faz

sargularına uygulanan stator PWM geriliminin dalga şekline ve yüke bağı olarak oluşan stator faz akımlarında istenen temel bileşenin yanı sıra harmonik akım bileşenlerini de içerdığı görülür. Akımda en yüksek genlikli temel bileşene ek olarak, dengeli sistemlerde 3, 5, 7, 9, 11, ... vb. tek harmonikler de yer alacaktır.

Anahtarlama frekansının artması ile akımdaki harmonikler azaltılırken, özellikle problem yaratan temel bileşene yakın harmonikler IGBT'lerin uygun açılıp kapanmaları ile kontrol edilebilir, ortadan kaldırılabılır veya minimum hale getirilebilir. Amaç çıkış dalga şeklinin sinüse yaklaştırılması ve böylece ek kayıpların ve moment salınımlarının azaltılmasıdır. Bu sorunun çözümü hızlı anahtarlama elemanlarının kullanılması ile mümkündür[56], [54], [53].

BÖLÜM IV

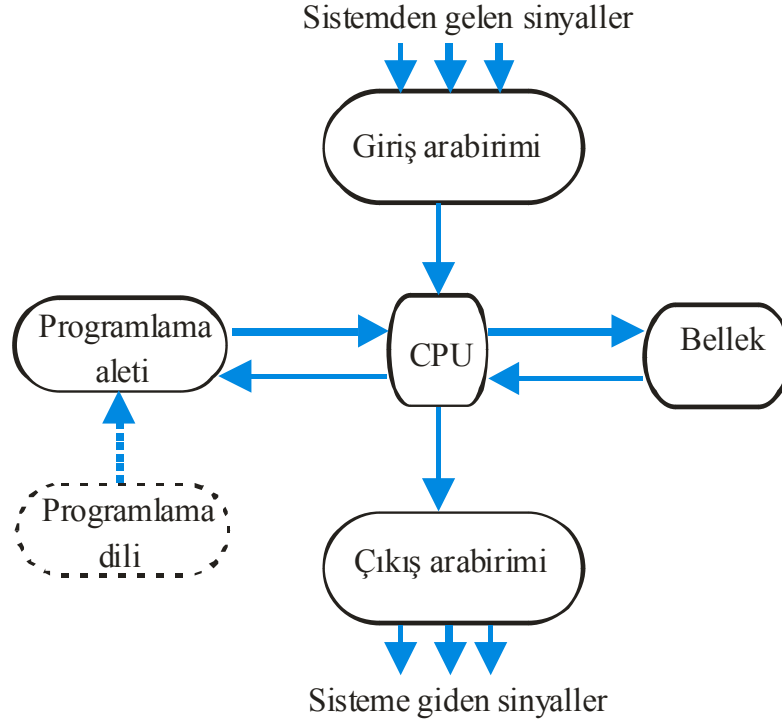
PROGLAMLANABİLİR LOJİK KONTROLÖR (PLC)

Endüstriyel uygulamaların her dalında yapılan genel amaçlı kumanda ve otomasyon çalışmalarının bir sonucu olan PLC tekniği, kullanıcılara A' dan Z' ye her türlü çözümü getiren komple bir, teknoloji alt grubudur. PLC'lerin ortaya çıkarılma amacı, röleli kumanda sistemlerinin gerçekleştirdiği fonksiyonların mikroişlemcili kontrol sistemleri ile yerine getirilebilmesidir. Lojik temelli röle sistemlerine alternatif olarak dizayn edildiklerinden PROGRAMLANABİLİR LOJİK KONTROLÖR (Programmable Logic Controller) adi verilmiştir. Günümüzde endüstrisinde el değmeden gerçekleştirilen otomatik üretimlerde PLC'ler kullanılmaktadır.

IV.1. YAPISI

PLC' ler şekil IV.1' de gösterilen temel kısımlardan oluşmaktadırlar. PLC' lerde temel olarak şu birimler bulunmaktadır.

- Giriş arabirimi (Input interface),
- Merkezi İşlem Birimi (CPU, Central Processing Unit),
- Bellek,
- Programlama dili,
- Programlama aleti,
- Çıkış arabirimi (Output Interface)[57], [46].



Şekil IV.1 PLC' nin Temel Yapısı

IV.1.1. Giriş Arabirimi

Giriş arabirimi kontrol sistemi ile kontrol edilen sistem arasında bir bağlantı sağlar. bu arabirimin esas fonksiyonu, ortamdaki gelen sinyaller alıp CPU' nun değerlendirebileceği şekle dönüştürmektir. Giriş arabirimleri sembolük olarak genellikle "I" (Input) gösterimi ile adlandırılırlar. Giriş arabiriminin yaptığı temel fonksiyonel işlemler şunlardan oluşmaktadır.

- Giriş arabiriminin terminallerine bağlı devrelerde gerilim olup olmadığını algılamak,
- Yüksek hızlı giriş darbelerini kullanarak önceden belirlenmiş değerlere kadar artırarak yada azaltarak saymak,
- Denetim bilgisi olarak, sistemin çeşitli noktalarında kontağın açılıp – kapanmasını algılayıp bunları mikroişlemcinin değerlendirmesine sunmak,
- Çeşitli birimlerden (termokupul, basınç sezici gibi) gelen analog sinyalleri alıp bunları CPU' nun değerlendirebileceği şekle getirmek,
- ASCII kodunda iletişim yapabilmek,
- Disk kodlayıcısı ve çeşitli anahtarlar tarafından üretilen kodları almak.

Giriş arabirimleri aşağıdaki gibi sınıflara ayrılır.

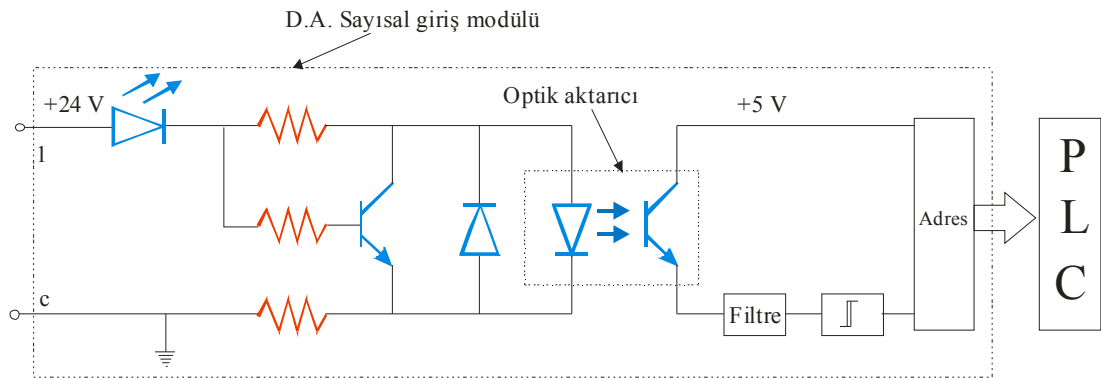
- Sayısal Giriş Arabirimleri,
- Analog Giriş Arabirimleri [57].

IV.1.1.1. Sayısal Giriş Modülü

PLC' lerde bir sayısal sinyal girişi için ayrıca bir sayısal giriş terminali bulunmaktadır. PLC' nin sayısal girişleri için kaynak olarak kullanılmakta olan elemanlar aşağıdaki gibi tasnif edilebilir.

- Durum anahtarları,
- Disk kodlayıcısı (Thumbell Switch),
- Ani Temaslı – Basmalı tip (Push) butonlar,
- Fotoelektrik gözler (algılayıcılar),
- Sınır anahtarlar,
- Yaklaşım anahtarları,
- Sıvı seviye anahtarları,
- Kontaktör kontakları,
- Röle kontakları,
- Basınç anahtarları.

Şekil IV.2 ' de bir PLC' nin sayısal giriş modülünün yapısı gösterilmektedir. Şekildeki yapı sadece bir sayısal sinyalin girilebilmesi için gereken modülü göstermektedir. Daha fazla sayısal giriş yapılması gerektiğinde bu yapı giriş sayısı kadar tekrarlanır.



Şekil IV.2 Bir PLC Cihazında Sayısal Giriş Sinyallerinin İşlenmesi

Standarda uygun olarak giriş modülüne doğrudan uygulanabilecek gerilim seviyeleri şunlardır : 24 V, 48 V, 100 V, 120 V, 200 V, 240 V doğru veya alternatif

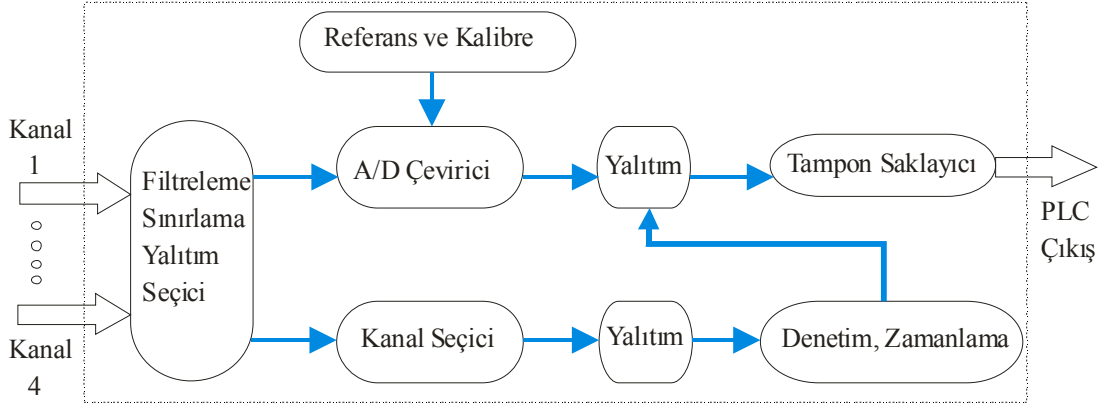
akım olabilir. PLC giriş devresine gelen bir sinyali mantık -1 kabul edebildiği bir alt sınır ve mantık – 0 kabul edebildiği bir üst sınır değeri vardır. Giriş bilgisinin doğru olarak algılanabilmesi için sinyal gerilim seviyesinin bu değerler arasında olması gerekir.

PLC, kontrol edilen sistemin durumunu bu sistemden sayısal girişe gelen sinyal ile algılayabilir. Bunun için durumu ifade eden kontağın bu gerilimi iletiyor veya iletmiyor olması gerekmektedir. Söz konusu gerilim bilgisi sayısal giriş modülüne şu şekilde girilir. PLC' nin mevcut olan güç kaynağının + ucundan alınan besleme gerilimi (örneğin 24 V) sahada (kontrol ortamında) mevcut olan durumu algılayabilmek üzere yerleştirilen elemanın kontak giriş terminallerinden birine, bu kontağın çıkış terminali ise PLC' nin giriş modülüne bağlanır.

Elektronik devre elemanlarının çeşitli nedenlerden dolayı arızalanma riskinden dolayı dış ortamdan gelen bilgi sinyalleri şekli IV.2' de gösterildiği gibi PLC' nin CPU' suna doğrudan değil optik bir aktarıcı üzerinden gönderilir. Sayısal giriş modülüne uygun olmayan bir bilgi sinyali girişinde genellikle bozulan eleman optik aktarıcı LED' i olur[60], [58], [57].

IV.1.1.2. Analog Giriş Modülü

Bir analog giriş modülü analog çoklayıcı aracılığı ile birden fazla veri girişi için kullanılabilir. Genellikle, analog çoklayıcıda süzgeç ve sınırlama işlemleri de yapılır. Sınırlama belli bir düzeyde olmayan AA sinyallerin A/D çeviriciye ulaşmasını engeller. CPU ile analog giriş modülü arasındaki elektriksel izolasyon işlemi yapılır. Modül içersindeki zamanlama ve denetim işlemleri CPU' nun tarama çevriminden (periyotundan) ayrı bir şekilde, kanal seçimini ve giriş verilerini tampon belleğe yazılışını denetler. Bu yolla belleğe yazma süreçleri ile CPU tarafından bellekten veri okuma süreçlerinin zamansal çalışması önlenir. A/D çeviricisinin kalibrasyonu modül içersinde üretilen bir referans geriliminin ayarlanması ile yapılır. Tipik bir PLC analog giriş modülünün blok diyagramı şekil IV.3' de gösterilmektedir.

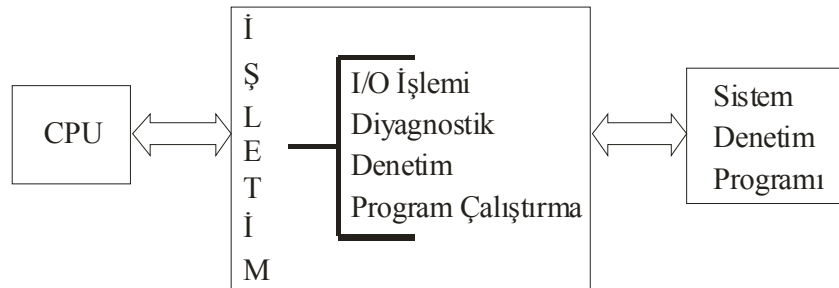


Şekil IV.3 Dört Kanallı Analog Giriş Modülü Blok Diyagramı

PLC' nin analog girişleri için kaynak olarak kullanılan elemanlara örnek olarak termokopullar, potansiyometreler, akış algılayıcıları, basınç algılayıcıları gösterilebilir. PLC giriş arabirimleri veriyi genellikle paralel bir biçimde alırlar, fakat seri veri kabul eden çeşitleri de vardır[57].

IV.1.2. Merkezi İşlem Birimi

Merkezi işlem birimi (CPU) bellekle birlikte bir PLC' ye işlem kabiliyeti kazandıran kısım olup, bellekte saklı bulunan programın gerektirdiği bütün aritmetik, mantık ve veri işleme gibi fonksiyonları gerçekleştirir. CPU çok büyük tümleşik devre teknolojisi ürünü olan bir mikroişlemcidir. PLC' lerde mikroişlemcinin çalışması işletim sistemi adı verilen bir program sistemi tarafından yönetilmektedir. İşletim sistemi üretici firma tarafından kullanıcının müdahalesine imkan tanımayan şekilde hazırlanmış olan bir PLC yönetim yazılımıdır. Şekil IV.4' de işletim sisteminin genel işlevleri ve denetim uygulama programı ile ilişkisi gösterilmektedir [58], [57].

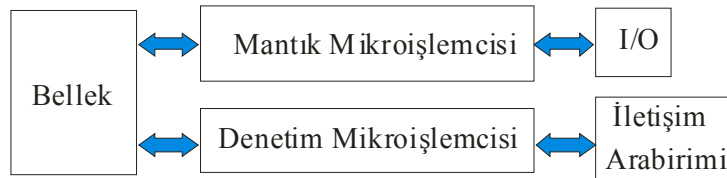


Şekil IV.4 İşletim Sistemi

IV.1.2.1. Diyagnostikler

Şekil IV.4’ de görüldüğü gibi, işletim sistemi diyagnostik prosedürleri de içermektedir. Diyagnostikler başlangıçta kendinden çalışan veya kullanıcı tarafından çalıştırılan olmak üzere iki genel gruba ayrılır. Birinci tür diyagnostikler genelde işletim sisteminin kullanıldığı belleği test ederler. Kullanıcı tarafından çalıştırılan diyagnostikler ise bütün belleği ve iletim kapıları gibi diğer unsurları da teste tabi tutarlar.

CPU bütün işlemleri bir mikroişlemci üzerinden gerçekleştirir. Günümüzde bu işlemler birden fazla mikroişlemci tarafından gerçekleştirilebilmektedir. Şekil IV.5’ de yaygın olarak kullanılan çift mikroişlemcili yaklaşım şekli gösterilmiştir. Buradaki mikroişlemcilerden biri denetim diğeri de mantık işlemlerini sürdürür. Denetim mikroişlemcisi denetim çevrimindeki denklemleri çözen ve operatörlerle etkileşim gibi daha karmaşık veri ve hesap işlemlerini yapar. Mantık mikroişlemcisi ise zamanlama, mantık ve sayma gibi işlemlerle birlikte uygulama programının mantığı ile belirlenen bir tarama gerçekleştirir. Tarama işlemi; programı gerçekleştirme esnasında girişlerin durumunu okumak ve buna göre çıkışların durumunu yenileme işlemleridir. Mantık mikroişlemcisi, programın bir yerinde kendi yeteneğinin üstüne çıkan bir işlemle karşılaşırsa bunu denetim mikroişlemcisine aktarır ve taramaya devam eder.



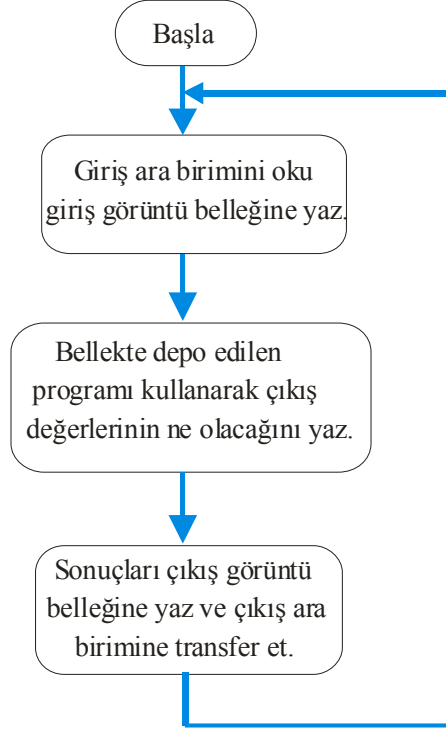
Şekil IV.5 Çift Mikroişlemcili PLC’ nin Blok Diyagramı

Uygulama programının yürütülmesi sırasında CPU bütün girişleri okur, denetim mantığının gerektirdiği işlemleri yaparak çıkış terminallerini iletme geçirir veya iletimden keser.

Temel işlem bilgileri bellekte kelimeler denilen bit grupları halinde depolanır. Bellekte depolanan her kelime ya bir komut, ya bir komutun parçası veya verinin bir kısmıdır. Veri ya referans verisi ya da giriş arabirimi üzerinden alınıp depolanan sinyallerin değerleri olabilir. PLC’ de CPU ve belleğin birlikte çalışması sırasıyla aşağıdaki işlemlerin gerçekleşmesini sağlamaktadır.

1. Kontrol edilen girişleri okumak. Bu işlem giriş arabiriminden görüntü belleğine alınan bilginin incelenmesiyle yapılır ve bu bilgiler burada saklanır. Bu bellek bölümünün içeriği bir sonraki taramaya kadar değişmez.
2. PLC' ye yazılan sistemi kontrol etmeye yönelik programa göre adım adım sırayla komut işlemleri gerçekleştirilir. Bu işlemler yapılırken bir önceki adımda hesaplanan ara değerler daha sonraki adımlarda kullanılır. Hesaplama sürecinde giriş birimindeki değer değişimleri değerlendirilmez.
3. PLC' deki programın yürütülmesi tamamlandıktan sonra program mantığında hesaplanan değerler çıkış görüntü belleğine yazılır ve bu veriler çıkış arabirimine gönderilir. Çıkış birimine verilerin transfer edilmesi gerçekleştikten sonra, yeni mevcut durum üzerinde yeni kararlar verebilmek için tekrar birinci adıma dönülür. Çıkış görüntü belleği ve çıkış arabirimindeki mevcut değerler bir sonraki çevrime kadar değiştirilmezler.

CPU, bir sonraki işlem için gerekli komutlar ve referans verileri aramak üzere sürekli olarak bellekte depolanan programa göz atar. Ayrıca merkezi işlem birimi, dışardan alınan verileri gelecek kullanımlar için belleğe depolar veya bazı karar verme çalışmaları sırasında ara işlemler için bellekte bulunan verilerden yararlanır. Bu işlem sırası şekil IV.6' da bir akış diyagramı olarak gösterilmektedir [59], [57].



Şekil IV.6 Tarama İşlemi

IV.1.3. Bellek

PLC belleği genellikle RAM ve ROM bellekten oluşmaktadır. Genel program çalışmalarında RAM (Random Access Memory – Rastgele Erişimli Bellek) kullanılır. Buradan bilgi okunabilir veya geçici depolama için yazılabilir. ROM (Read Only Memory – Sadece Okunur Bellek) sabit bellek çeşididir. Bu belleğe üretim sırasında bir kere yazılmış olan sistem bilgileri, cihazın ömrü boyunca sabit kalır. PLC, çalışmaları sırasında bu bellekten yalnız bilgi okuyabilir. PLC belleğinde aşağıda belirtilen dört ayrı çeşit bilgi bulunur.

İşletim sistemi : İşletim sistemi genel amaçlı bir mikrobilgisayarı PLC haline getiren programdır. Çevre birimleri ile iletişimi sağlar. Denetim uygulama programını yürütür.

Denetim uygulama programı : Belirli bir denetim işlemi için yazılmış programdır. Merdiven programı şeklinde veya daha değişik bir programlama dilinde yazılmış olabilir.

Veri tabloları : Bu alanda denetim işleminin gerçekleştirilmesi sırasında sürdürülecek matematik işlemler için sabitler, zaman ve uyarma işlemleri için ön değerler ve uygulama programının kullanılacağı diğer sabitler depolanır. Burada

ayrıca, son olarak okunan sistem giriş değerleri ve son olarak dışarı verilen sistem çıkış değerleri bulunur.

Yaz – boz alanı : Burada program tarafından kullanılacak geçici değerler tutulur. Veri tablolarının kullanılması yerine yaz – boz alanının kullanılması verilere daha kısa bir sürede ulaşılmasını sağlar. Yaz – boz alanı aşağıda gösterilen iki ayrı bölüm olarak kullanılır.

- *Giriş görüntü bölümü* : Bu bölüm, programın yürütülmesi sırasında giriş birimindeki sinyallerin (var/yok) durumlarının saklandığı özel bir bellek alanıdır. Her bir okuma aşamasına gelindiğinde giriş birimindeki değerler yeniden alınır ve eski değerlerin yerine konulur.
- *Çıkış görüntü bölümü* : Bu bölüm, kontrol programının yürütülmesi sürecinde hesaplanan değerlerin saklandığı özel bir bellek alanıdır. Kullanıcı programının yürütülmesi tamamlandığında, bu bölümün içeriği çıkış birimine transfer edilir. Bir sonraki transfere kadar çıkış biriminde tutulmaya devam eder[60],[57].

IV.1.4. Programlama Aletleri

Programlama aleti, programcı ve PLC arasında bağlantıyı sağlamaktadır. Programcı, gerekli kontrol işlemlerini planlayıp, seçilen PLC için özel olan program biçimine aktarır. Bir programlama aleti üzerinden bilgiler programlama dilinin belli bir versiyonu için, standart olan semboller, harfler ve sayılar kullanılarak PLC' ye girilebilir. Programlama aleti mikroişlemci tabanlı özel bir el cihazı olabileceği gibi kişisel bir bilgisayar da olabilir. Bu birim, programın yazılması PLC' ye aktarılması ve gerektiğinde PLC' nin çalışması sırasında; giriş / çıkış bellek durumlarının gözlenmesi veya bazı parametrelerin değiştirilmesi olanaklarını sağlar[57].

IV.1.5. Çıkış Arabirimi

Analog çıkış modülleri genellikle hem doğru akım hem de alternatif akım gerilimleri veya akımları üretebilecek şekilde imal edilirler. Program tarafından üretilen sayısal değerler ise çeşitli elemanları denetlemek veya sürmek üzere analog gerilim veya akımlara çevrilebilir. Kontrol edilecek elemanlar analog ve sayısal olabileceklerinden dolayı çıkışlar da;

- Sayısal Çıkış Modülü,
- Analog Çıkış Modülü,

Olarak iki gruba ayrılır.

Bir selenoid valfin açılıp kapanması işlemi sayısal çıkış ile, oransal kontrollü bir valfin pozisyonu ise analog çıkış ile kontrol edilir. Çıkış arabirimleri genellikle sembolik olarak “Q” (Output) olarak adlandırılırlar.

Çıkış arabiriminin temel işlemleri şunlardır.

- Çıkış arabiriminin terminallerine bağlı devrelere güç uygulamak veya kesmek,
- Bazı işlem adımlarında dış birimleri denetleyecek analog sinyalleri üretmek (mesela, oransal valf kontrolü),
- ASCII kodunda iletişim yapabilmek,
- Kod sinyalleri ile kontrol edilecek birimler için uygun kodlar üretmek[57].

IV.1.5.1. Sayısal Çıkış Modülü

Sayısal çıkış arabirimi giriş arabiriminin tersi bir görev yapar. Bu, CPU’ dan aldığı sinyaller ile kontrol edilen sistemdeki kontaktör, röle, selenoid gibi kumanda elemanlarını sürmeye uygun donanım birimidir. Çıkış arabirimi de giriş arabirimi gibi modül biçimindedir. Bu modül, gerektiğinde yeni çıkışlar oluşturmak amacı ile genişletilebilir.

Sayısal çıkış modülü röle, triyak veya transistör çıkışlı olabilir. Özellikle çalışma esnasında yüksek hızlı açma – kapama gerektiren durumlarda, doğru akımda transistör, alternatif akımda ise triyaklı olan çıkışlar tercih edilmektedir. Çıkış modülünden büyük akımlar (örneğin röle çıkışlı modül için maksimum 10 -16 A) çekilmez. Çıkışa ait maksimum akım değerleri üretici firma kullanım kitapçıklarından öğrenilir[57].

IV.1.5.2. Analog Çıkış Modülü

Analog elemanları etkileyecek sinyaller PLC dahilinde kodlar olarak ortaya çıkmaktadır. Bu sinyaller değişik değerlere sahip olduklarından kodları da çok bitlidir. Arabirimler sözcük temelli olan (çok bitli ikili kod) bu kodları genellikle paralel olarak çıkışa verirler. Analog çıkış modülü söz konusu kodları analog gerilim veya akım sinyallerine dönüştürür. Kullanılan veri (kod) uzunluğu genellikle 10 bit veya daha azdır. Bir çıkış modülü genellikle 4 ayrı veriyi işleyip iletebilir. Analog çıkış modüllerin çıkışı doğru akım veya alternatif akım olabilmektedir. Yaygın

olarak akım modülü için 4- 20 mA akım, gerilim çıkışı için ise 0 – 10 V gerilim veren modüller kullanılmaktadır[60].

IV.1.6. İletişim Modülleri

PLC' lerde iletişim modülleri üç kısımdan oluşmaktadır.

- Paralel giriş,
- Paralel çıkış,
- Seri giriş/çıkış.

Paralel giriş : CPU' nun paralel formatta çok bitli çıkış veren elemanlarına bağlantısını sağlar. Paralel giriş modülü 16 bitlik olan 4 ayrı kanaldan oluşan bir elemandır. Bu elemanların her birisi seçici girişi olan üç durumlu bir yarı iletken kapı vasıtasıyla kontrol edilirler. Aksi halde her bir girişin ayrı bir modüle bağlanması gerekir.

Paralel çıkış : Bu modüllerin yapısı temelde giriş modüllerine benzemektedir. Bir analog çoklayıcı kullanılır. Paralel çıkış arabiriminde çıkış elemanının bir zamanlayıcı sinyali ile bilgi alabilecek türden olması gerekmektedir.

Seri giriş / çıkış : Bu arabirim seri formatta veri çıkışı veya girişi olan elemanların CPU ile bağlantısını sağlar. Seri giriş / çıkış arabirimine örnek olarak mil şifre kodlayıcılar gösterilebilir. Bu tür bir kodlayıcının ürettiği darbe dizisinde her bir darbenin kaçırılmadan sayılması gerekir. Dolayısıyla da arabirim içersinde hızlı darbeleri sayabilen bir sayıcı yer alır. Bu sayıcıya girece darbe freansı 100 Hz. ile 50 kHz. arasında değişebilir. Minimum darbe genişliği 10 -20 ms arasında olmalıdır. Arabirim ile CPU arasındaki iletişim iki yönlüdür. Modül CPU' dan ön değer ve diğer denetim verilerini kabul eder. CPU' ya ise sayılan değeri ve endeks işaretinin durumlarını bildirir. CPU uygulama programına göre sayıcıyı etkinleştirir ve reset eder[57].

IV.1.7. Besleme Kaynağı

Besleme kaynağı tüm sistemin çalışmasında ara rol oynamaktadır. PLC' lerde besleme kaynağı iki kısımdan oluşmaktadır.

- Pil veya akülü besleme kaynağı (dahili),
- Şebekeden beslenen gerilim kaynağı.

Pil veya akülü besleme kaynağı sistemin enerjisi kesildiğinde RAM' ların kesintisiz olarak beslenmesini sağlayarak uygulama programlarının silinmesini önler.

PLC şebekeden besleme gerilimine bağlandığında kaynaktan beslenmeye başlar ve gerekirse aküyü şarj eder. Besleme gerilim kaynağı, sisteme (monitör, giriş – çıkış birimleri, CPU, RAM, ROM) enerji verildiğinde PLC için gerekli çeşitli düzeydeki DC ve AC gerilimleri sağlar[60].

IV.1.8. Özel Giriş / Çıkış Modülleri

Aşağıdaki işlemleri yapabilmek için özel I / O modüllerine ihtiyaç vardır.

- PID denetim veya ileri düzeyde hassas konum denetimi,
- Sadece birkaç ms' lik gecikmelerin kabul edilebileceği, çok hızlı değişmelerin olduğu uygulamalar,
- Yoğun ASCII transferi gerektiren uygulamalar,
- Yoğun matematik işlemler gerektiren uygulamalar.

Bu modüller genellikle daha komplike olup, ana CPU ile sadece parametre ve veri yenileme amaçları için ilişki kurar, zamanlama ve denetimi kendi dahili yapılarında gerçekleştirirler[57].

IV.1.8.1. Termokopul Giriş Modülü

Standart tip I/O modüllerine ek olarak bazı tip modüller de direkt olarak algılayıcıdan sinyal kabul ederler. Termokopul giriş modülü bunlara bir örnektir. Bu modül termokopulların çıkışlarını doğrudan kabul eder ve soğuk eklem kompanzasyonu modül içerisinde gerçekleştirilir. Çalışma ilkesi standart analog giriş modüllerinkine benzer. Özelliği ise çok düşük seviyedeki (maksimum 43 mV) sinyalleri işleyebilmesidir. Termokopul giriş modülüne gelen sinyal süzülür, yükseltilir ve bir ADC (Analog dijital dönüştürücü) yardımı ile sayısal formata çevirilir. Bu sayede, istenen bir ortamın sıcaklığını istenilen sıcaklık değerinde sabit tutulması için gerekli kontrol işlemleri yapılabilir ve istenirse sıcaklık değerinde bir gösterege yazdırılabilir[59],[57].

IV.1.8.2. Haberleşme Modülleri

ASCII Haberleşme Modülü : CPU ile dış birimler arasında sayısal veri alışverişini sağlar. Örnek olarak; video terminallere veya yazıcılara bağlantı sağlayan ara birimler verilebilir. Modül genellikle bir RAM tampon bellek ve bir mikroişlemci ile birlikte gerekli bütün iletişim birimlerini içerir. Genel olarak bu modülde ya RS

232C ya da RS 422 ve 20 mA' lik çevrim akımına müsaade edilen iki seri iletişim kapısından birisi bulunur.

Yerel Giriş / Çıkış Uyum Modülü : Bu modüller büyük PLC sistemlerde alt sistemlerin kontrol edilebilmesine imkan sağlar. Uzaktan kontrolü yapılacak sistemin standart giriş - çıkışlarından gerekli olanları kullanılır. Bu modülün kullanıldığı kasa rafında DA güç kaynağı, dahili sürücü devreler, haberleşme ile işlemci modülü bulunur[57].

IV.2. KULLANIM AVANTAJLARI

PLC'lerin, daha önce kullanılan konvansiyonel sistemler ile karşılaştırıldığında bir çok avantajı vardır. Eski sistemlerin getirdiği birtakım zorluklar bugün PLC'lerin yaygınlaşması ile aşılmıştır. PLC sistemleri önceki sistemlere göre daha az yer kaplamaktadır. Dolayısıyla kontrol sisteminin yer aldığı dolap yada pano boyutları oldukça küçülmektedir. Sınırlı alanlarda kontrol mekanizmasının kurulması imkanı ortaya çıkmıştır. Sistem için sarf edilen kablo maliyetleri nispeten daha azalmıştır. Ayrıca PLC sisteminin kurulmasının kolay olması ve kullanıcıya, kurulu hazır bir sistemin üzerinde değişiklik ve ilaveleri kolayca yapabilme esnekliğinin sağlanması, PLC'lerin giderek yaygınlaşmasına ve endüstride her geçen gün daha fazla kullanılmalarına neden olmuştur. Bu avantajlar ile proje maliyetleri de azaltılarak, proje mühendislerine de ticari açıdan büyük faydalar sağlamıştır[58].

Günümüzde bir PLC ile çok çeşitli makinaların kontrolünü yapmak mümkündür. PLC' ye yazılan programla uygulamaya özgü çalışma şekillerini oluşturmak mümkündür. Klasik kumanda sistemiyle oluşturulmuş kontrol sisteminin çalışma şeklinde değişiklik yapılması durumunda kumanda panosunun yeniden düzenlenmesi gerekir. Oysaki, PLC ile kontrol edilen bir düzeneğin çalışma şeklini değiştirmek için, donanımı değiştirmeden sadece PLC programını değiştirmek yeterli olacaktır[60].

BÖLÜM V

TEZ ÇALIŞMASI

V.1. DENEYSEL ÇALIŞMA İÇİN İZLENEN YOL

Tez çalışmasında araştırma aşamasından sonra deneysel çalışmaya geçilmiştir.

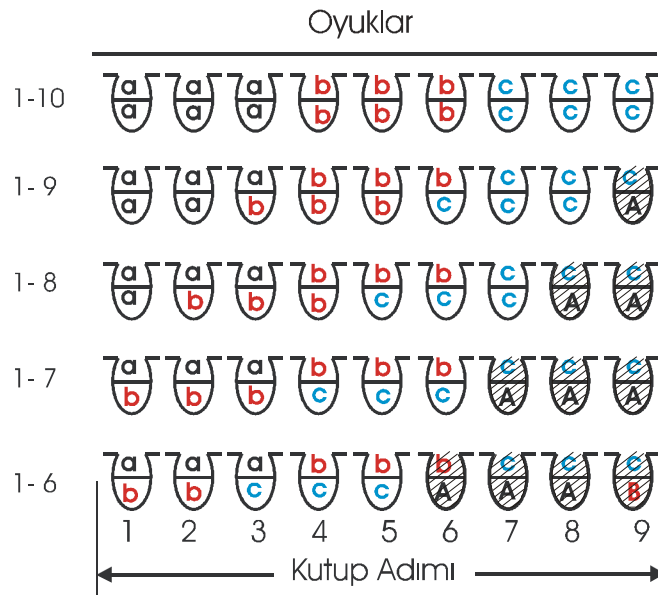
Deneysel çalışma sırasıyla aşağıdaki işlemler yapılarak gerçekleştirilmiştir.

1. Deneysel çalışmada kullanılacak asenkron motorların hazırlanması,
2. Deney setinin oluşturulması ve deney bağlantısının kurulması,
3. Veri setinin oluşturulması,
4. Y.S.A yöntemi ile sistemin modellemesi ve eğitilmesi,
 - a. Parametrelerin belirlenmesi,,
 - b. Aktivasyon fonksiyonunun seçimi,
 - c. Gizli katman sayısının seçimi,
 - d. Gizli katmanlardaki hücre sayısının seçimi,
5. Y.S.A sonuçlarının deney sonuçları ile karşılaştırılması ve değerlendirme.
 - a. Örnekleme zamanı bir dakika olan gerilim Toplam Harmonik Distorsiyonu (THD) çıkışlı modelin sonuçları ve deneysel sonuçlar ile karşılaştırılması,
 - b. Örnekleme zamanı bir dakika olan akım THD çıkışlı modelin sonuçları ve deneysel sonuçlar ile karşılaştırılması
 - c. Örnekleme zamanı bir saniye olan gerilim THD çıkışlı modelin sonuçları ve deneysel sonuçlar ile karşılaştırılması
 - d. Örnekleme zamanı bir saniye olan akım THD çıkışlı modelin sonuçları ve deneysel sonuçlar ile karşılaştırılması

V.2. ASENKRON MOTORLARIN HAZIRLANMASI

Özdeş yapıya sahip statoru gövdeye çakılmış, beş adet asenkron motorun stator sargıları bir adım kısaltılarak tekrar sarılmıştır. Şekil V.1' de asenkron motorların sargı adımı kısaltılmış (kirişlenmiş) tam kalıp sarımlarının oyuklardaki pozisyonu gösterilmektedir. Şekilde 'a', 'b' ve 'c' harfleri stator fazlarına karşılık gelmektedir. Harflerin büyük olması sargıların yönlerini göstermektedir. Taralı olan oyuklar kör oyukları temsil etmektedir. Kutup adımı dokuz olup faz başına düşen oyuk sayısı üç tür. Motor adımları oyukların yan taraflarında gösterilmektedir. 1-10 adımlı motor 180° sargı adımına sahiptir. Her motor 20° azaltılarak sarılmıştır. Motorlarda adım kısaltması yapılsa da sipir uzunluğu kısaltılmadığı için kullanılan iletkenin uzunluğunda bir değişiklik yoktur. Aynı kesitte bobin her motorda ve her oyukta aynı sayıda bobin kenarı gelecek şekilde sarılmıştır. Motorun fabrikadan çıkışı el tipi sarımlı olup etiket değerleri aşağıdaki tablo V.1' de verilmektedir. Deneysel çalışmada kullanılan asenkron motorların stator bağlantı şemaları Ek – A' da gösterilmektedir.

Deneylerde kullanılan kafesli asenkron motorların yapısal gövde boyutları şekil V.2' de harfler ile gösterilmektedir. Tablo V.2 ve tablo V.3' de şekil V.2' deki harflerin karşılığı olan ölçüler rakamsal olarak gösterilmektedir. Bütün ölçüler milimetre (mm) olarak verilmiştir. Şekil V.3' de ise stator sargıları yapılmış olan statorların fotoğrafı gösterilmektedir.

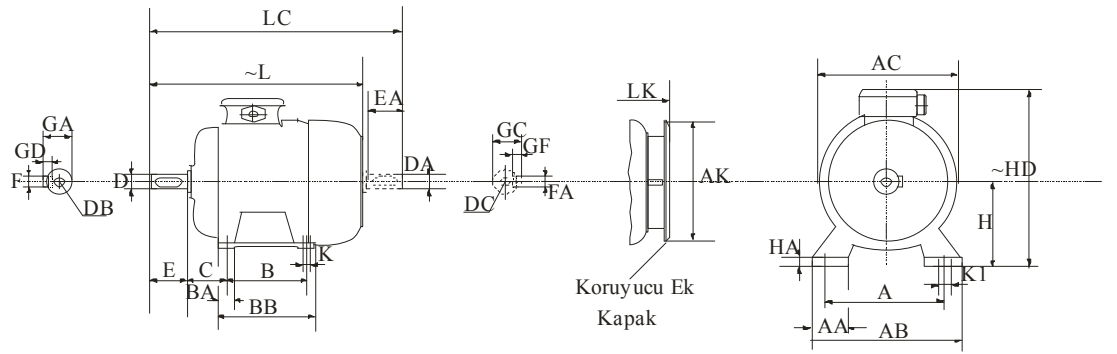


Şekil V.1 Asenkron Motorlarda Stator Sargıları

Tablo V.1 Kafesli Asenkron Motorun Etiket Değerleri

<i>Volt</i>	<i>Frekans</i>	<i>Akım</i>	<i>Güç</i>	<i>Cosφ</i>	<i>n d/d</i>
Δ 220	50 Hz	4,7 A	1,1 kW	0,80	1380
Y 380	50 Hz	2,7 A	1,1 kW	0,80	1380

Kullanılan motorlar 90 S yapı tipinde olup, stator sargıları için kullanılan emaye kalınlığı : 2 X 0,48 mm ve sipir sayısı 66 sipirdir.



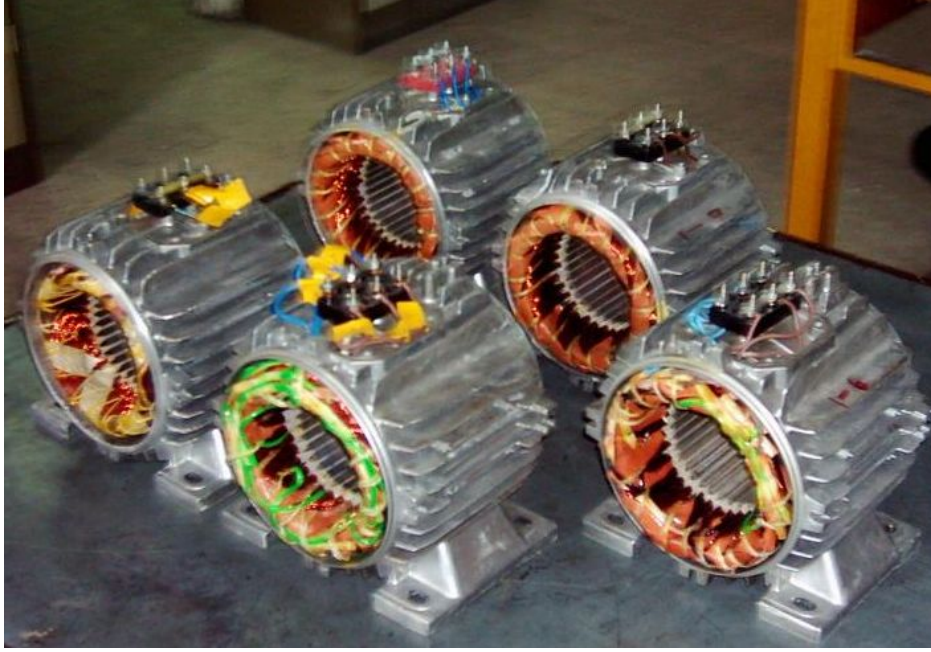
Şekil V.2 Kafesli Asenkron Motorun Yapı Ölçüleri

<i>H</i>	<i>~HD</i>	<i>HA</i>	<i>A</i>	<i>AA</i>	<i>AB</i>	<i>AC</i>	<i>AK</i>	<i>K</i>	<i>KI</i>	<i>F × GD</i>
						ϕ	ϕ	ϕ		<i>FA × GF</i>
90	216	12	140	43	180	176	151	10	15	8 × 7

Tablo V.2 Asenkron Motorun Yapı Ölçüleri I (mm)

Tablo V.3 Asenkron Motorun Yapı Ölçüleri II (mm)

<i>~L</i>	<i>LC</i>	<i>~LK</i>	<i>C</i>	<i>E</i>	<i>DB</i>	<i>D</i>	<i>GA</i>	<i>BB</i>	<i>B</i>	<i>BA</i>
				<i>EA</i>	<i>DC</i>	<i>DA</i>	<i>GC</i>			
308	363	337	56	50	M8	24	27	130	100	35



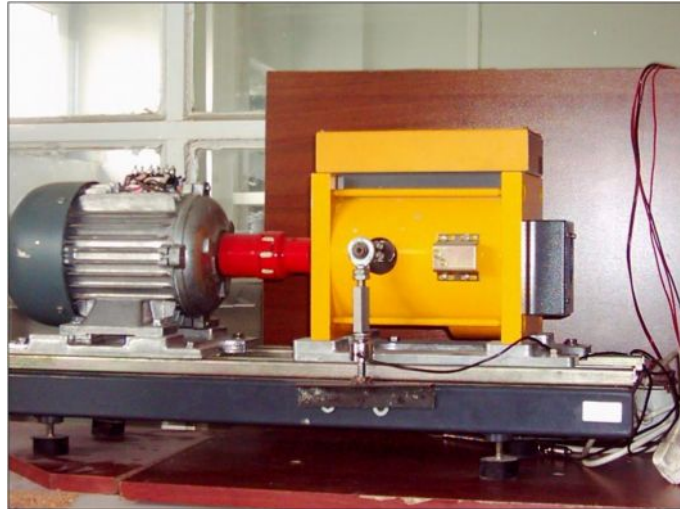
Şekil V.3 Stator Sarımları Yapılmış Deney Motorları

V.3. DENEY DÜZENEĞİNİN KURULMASI



Şekil V.4 Deney Seti

Şekil V.4’ de deney setinin fotoğrafı gösterilmektedir. Deney seti AC sürücü kontrol paneli (1), güç analizörü (2), RS-485 dönüştürücü (3), PLC ekranı (4), DC ayarlı potansiyometre (5), ve start - stop butonlarından (6) oluşmaktadır. Deney setinde PLC modüllerinin kullanılmasının amacı deneysel işlemlerimizi kolaylaştırması ve tezin eğitimsel yönüne de katkı sağlamaktır.

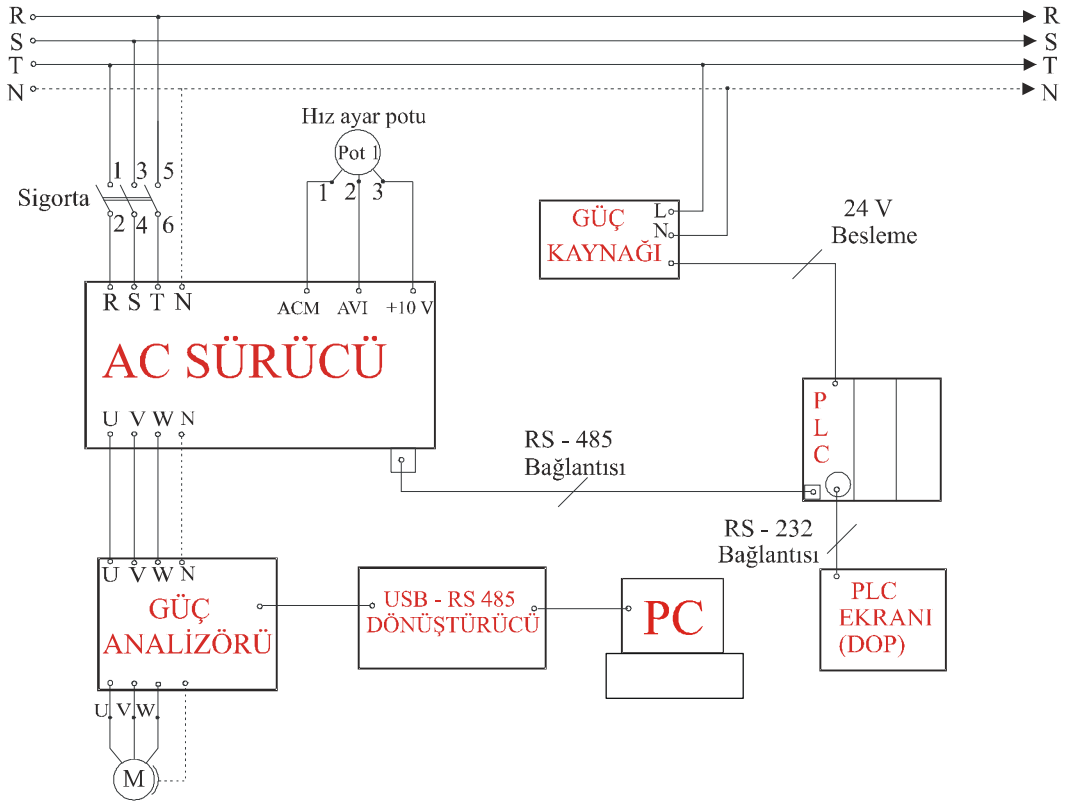


Şekil V.5 Asenkron Motora Akuple Bağlı Elektromagnetik Fren Ünitesi

Şekil V.5’ de asenkron motora akuple bağlı olan elektromagnetik fren ünitesinin fotoğrafı gösterilmektedir.

V.3.1. Deneysel Setin İç Bağlantısı

Deneysel setin iç bağlantısında PLC giriş ve çıkış modülleri, AC sürücü, güç analizörünün akım ve gerilim bağlantıları bulunmaktadır. Şekil V.6’ da deneysel setin genel iç bağlantısı gösterilmektedir. PLC modüllerinin giriş çıkış bağlantıları üretici firmanın kullanım klavuzundaki bağlantı kurallarına uygun bir şekilde üretici firma tarafından yapılmıştır. PLC için kullanılan programlama dili, ‘Ladder’ (merdiven) diyagramı yöntemi ile üretici firmanın kendi programlama kodları kullanılarak yapılmış olup 440 adım da gerçekleştirilmiştir.



Şekil V.6 Deneysel Setin Genel İç Bağlantıları

Deneysel seti üç faz, faz – nötr 220 V’ luk gerilim ile beslenmektedir. Üç faz gerilimi ilk olarak alternatif akım sürücüsünü beslemektedir. Üç faz iletkeni, sürücüden güç analizörünün akım girişlerine ve güç analizöründen de asenkron motora bağlıdır. Güç analizörü ile bilgisayar RS – 485 dönüştürücü vasıtası ile bir

dakikalık örnekleme zamanı ile haberleşmektedir. Deneysel çalışmada güç analizörünün kendi hafızasına saniyelik olarak kaydedilmiş olan veriler, Modbus haberleşme protokolü kullanılarak bilgisayara aktarılmıştır. Güç kaynağı PLC modüllerine uygun gerilim verilebilmesi için kullanılmaktadır. PLC modülleri ile PLC monitörü RS – 232 dönüştürücü ara kablosu ile haberleşmektedir. Şekil V.7’ de deney seti, asenkron motor ve elektromagnetik fren ünitesi toplu olarak gösterilmektedir.

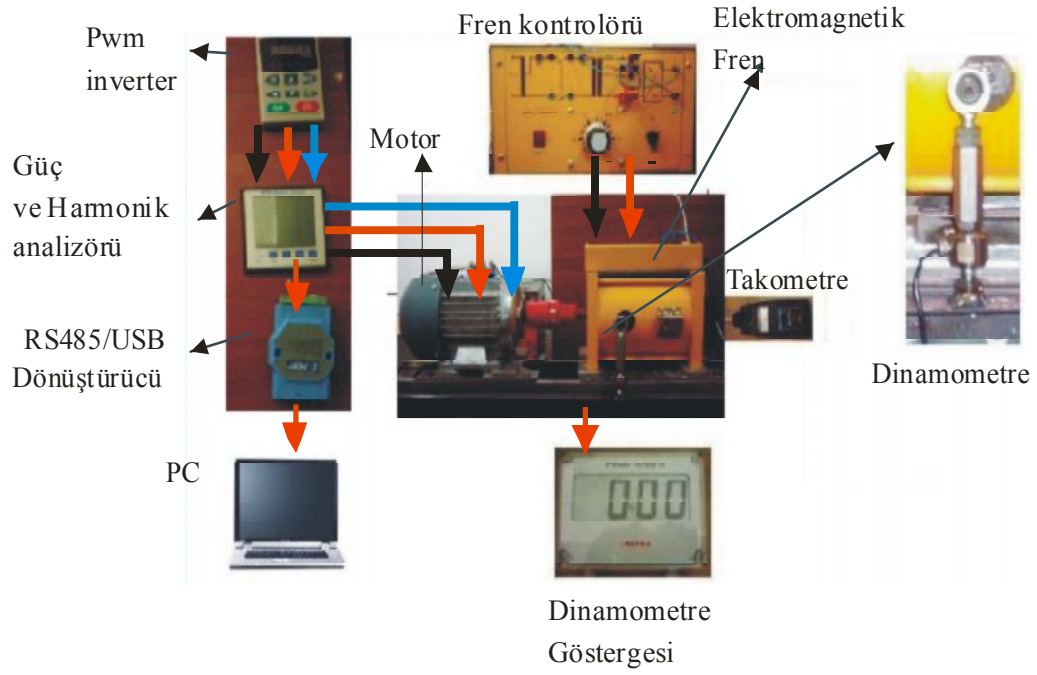


Şekil V.7 Deney Seti, Asenkron Motor ve Fren Ünitesinin Görünüşü

V.3.2. Deney Bağlantı Şeması

Şekil V.8’ den anlaşılacağı gibi deney bağlantı şemasında, asenkron motorlar AC sürücü ile beslenmektedir. Deneyslerde elektromagnetik fren kullanılmıştır. Motorların yük momenti, dinamometre vasıtasıyla ölçülmüştür.

Asenkron motorların iç ve dış sıcaklıkları K tipi termokopul kullanılarak elde edilmiştir. Lab View paket programı vasıtası ile NI Field Point şebeke modülü ve NI Field Point Termokopul modülü kullanılarak aynı bilgisayara saniyelik veri akışı sağlanarak sıcaklık bilgileri elde edilmiştir.



Şekil V.8 Deney Bağlantı Şeması

V.4. VERİ SETİNİN OLUŞTURULMASI

Kafesli asenkron motorlarda toplam harmonik distorsiyonunu tahmin eden bir yapay sinir ağı modeli oluşturmak için ilk önce yapay sinir ağını eğitmek üzere kullanılacak olan verilerin elde edilmesi gerekmektedir. Veri seti giriş ve çıkış verilerinden oluşmaktadır. Tez çalışmasında veri seti deneysel olarak elde edilmiştir. Doğru ve tutarlı verilerden oluşan bir veri seti elde etmek için ilk önce deney düzeneğinin kurulması gerekmektedir. Beş farklı stator sargılı, SPWM ile beslenen sincap kafesli, üç fazlı asenkron motorlar hafif yükten aşırı yüke kadar yüklenmiştir. Her yük akımı seviyesinde faz gerilimleri ve faz akımları, aktif güç, reaktif güç, görünür güç, ve harmonikler tek bir güç analizörü vasıtasıyla ölçülmüştür. Alınan veriler RS – 485 dönüştürücü vasıtasıyla bilgisayara aktarılmıştır. Bu işlem PWM' in her taşıyıcı frekansında ve her motor için tekrarlanmıştır. Yapay sinir ağı modelinin eğitilebilmesi ve güvenilir tahminler üretmesi veri setinin doğruluğuna ve sistemi tam olarak yansıtmasına bağlı olmaktadır. Tez çalışmasında örnekleme zamanı 1 dakika olan ve 1 saniye olan iki ayrı veri seti oluşturulmuştur. Verilerin veri seti haline gelebilmeleri için mutlaka yapay sinir ağına gönderilecek şekilde getirilmeleri gerekmektedir. Yapay sinir ağı verileri bir matris gibi algılayarak her bir girişe ait olan verileri sıra ile alır. Bunun için verilerin bir matris haline getirilip MATLAB programı vasıtasıyla transpozeleri alınmıştır. İlk oluşturulan veri seti örnekleme

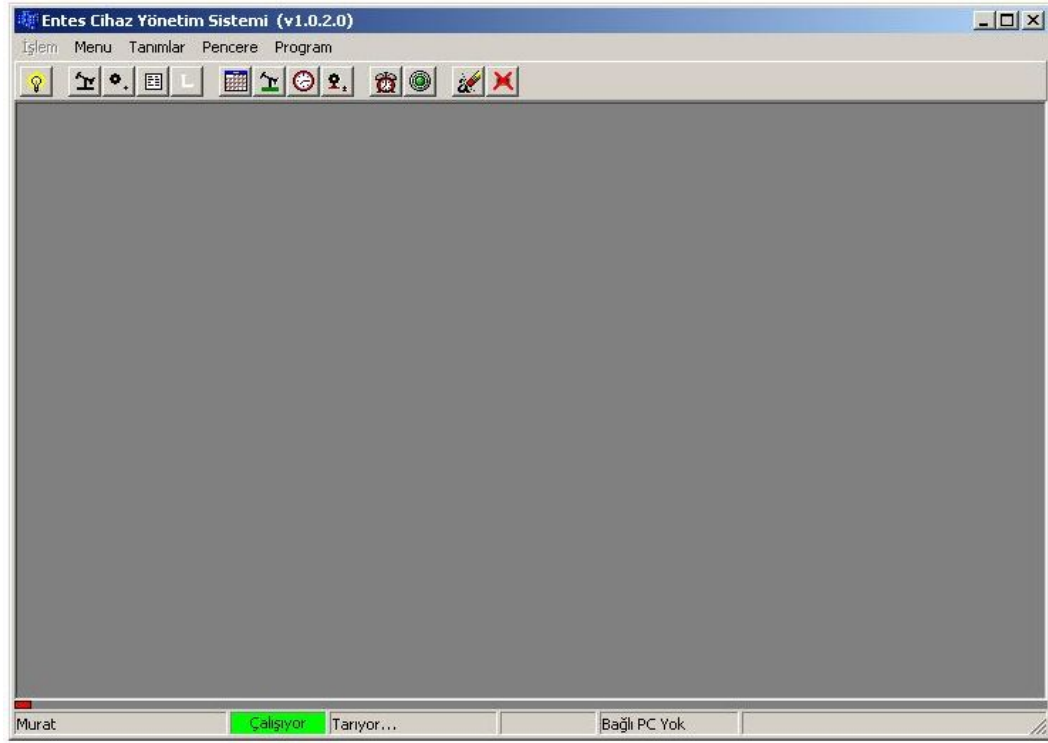
zamanı 1 dakika olması sebebiyle 663 adet veriden oluşmaktadır. İkinci oluşturulan veri seti örnekleme zamanı 1 saniye olan 3940 veri den oluşmaktadır. İkinci oluşturulan veri setinde giriş olarak motorun iç ve dış sıcaklıkları da eklenmiş olup ayrıca fabrikadan alınan el tipi sarımlı motor da deneylere dahil edilerek sistemin yapay sinir ağı tarafından daha fazla tanınmasına yardımcı olunmuştur. Sıcaklık ölçümlerinde Lab View programının sıcaklık modülleri kullanılarak motorun iç ve dış sıcaklıkları K tipi termokopul ile ölçülmüş ve bilgisayara saniyede bir aktararak elde edilmiştir. Sıcaklık verileri için deney bağlantı şeması Şekil V.16' da gösterilmektedir. Sıcaklıkların analizi için Lab View ön paneli Şekil V.17' de gösterilmektedir. İlk veri setinde motorlara sabit sinüs dalgası verilerek de deneyler yapılmış olup el tipi sarımlı motor verilere dahil edilmemiştir. Deneylerde kullanılan alternatif akım sürücüsünün parametre ayarları aşağıda verilmektedir.

<i>Kontrol Yöntemi</i>	: <i>V/F Kontrol (Skaler)</i>
<i>Maksimum Çıkış Frekansı</i>	: <i>50 Hz.</i>
<i>Maksimum Çıkış Gerilimi</i>	: <i>380 V.</i>
<i>Orta Nokta Frekansı</i>	: <i>25 Hz. (Fabrika Ayarı)</i>
<i>Orta Nokta Gerilimi</i>	: <i>3.4 V. (Fabrika Ayarı)</i>
<i>Minimum Çıkış Frekansı</i>	: <i>0.50 Hz.</i>
<i>Minimum Çıkış Gerilimi</i>	: <i>3.4 V. (Fabrika Ayarı)</i>
<i>Frekans Üst Sınırı</i>	: <i>% 100</i>
<i>Frekans Alt Sınırı</i>	: <i>% 0</i>
<i>Hızlanma Zamanı</i>	: <i>10 s</i>
<i>Yavaşlama Zamanı</i>	: <i>10 s</i>
<i>PWM Taşıyıcı Frekansı</i>	: <i>1 – 15 Khz</i>
<i>Motor Yönü</i>	: <i>İleri</i>
<i>PLC Modu</i>	: <i>Bir program çevrimini çalıştır.</i>
<i>Motor Ortalama Akımı</i>	: <i>% 100</i>
<i>Motor Yüksüz Akımı</i>	: <i>% 40 (Fabrika değeri)</i>
<i>Motor Kutup Sayısı</i>	: <i>4</i>

V.4.1. Güç Analizöründen Bilgisayara Veri Transferi

Güç analizöründen bilgisayara veri transferi için ilk önce RS – 485 dönüştürücünün sürücüsünün bilgisayara kurulması gerekmektedir. Sürücü

kurulduktan sonra veri aktarımı için güç analizörünü üreten firma tarafından yazılımı yapılmış olan paket program vasıtası ile veri aktarılır. Şekil V.9’ da veri transferi için kullanılan paket programının ana penceresi gösterilmektedir. İlk olarak ana pencere açılır.



Şekil V.9 Cihaz Yönetim Sisteminin Ana Penceresi

Ana pencere cihaz yönetim sisteminin başlangıç sayfasıdır. Ana pencere üzerinde ‘Tanımlar’ kısmından güç analizörüne ait aşağıdaki tanımlamalar yapılır.

- Bölge Tanımları,
- Cihaz Tanımları,
- Cihaz Cinsi Tanımları,
- Kullanıcı Tanımları,
- Bağlantı Noktası Tanımları,
- Bağlantı Noktası
- Haberleşme Ayrıntıları,

Çalışmada kullanılan güç analizörünün deney esnasındaki tablo görünümü için ‘Menü’ kısmından ‘Tablo Görünüm’ açılarak deney takibi yapılabilmektedir. Şekil V. 10’ da tablo görünüm gösterilmektedir.



Şekil V.10 Tablo Görünüm Penceresi

Deney esnasında verilerin dakikada bir ve istenir ise saniye de bir kaydı yapılmaktadır. Daha önceden deneyin yapıldığı dakikaya kadar olan verileri elde etmek için 'Menü' kısmından 'Periyodik Değerler' penceresi açılır. Şekil V. 11' de periyodik değerler penceresi gösterilmektedir. Periyodik değerler penceresinden deneylerin yapılmaya başlandığı tarih ve saat ile deneylerin bitirildiği son tarih ve saat girilir. İlk deney tarihi ile son deney tarihi arasındaki alınan bütün değerler şekil V.12' de gösterildiği gibi bir dakikalık örnekleme zamanı ile elde edilir. Elde edilen tablonun sol tarafında yapılan deneye ait değerler alındığı tarih ve saati verilmektedir.

Şekil V.11 Periyodik Değerler İçin Parametre Seçimi

Entes Cihaz Yönetim Sistemi (v1.0.2.0) - [Periyodik Değerler (01.01.2005 00:00-02.01.2008 00:00)]

İşlem Menu Tanımlar Pencere Program

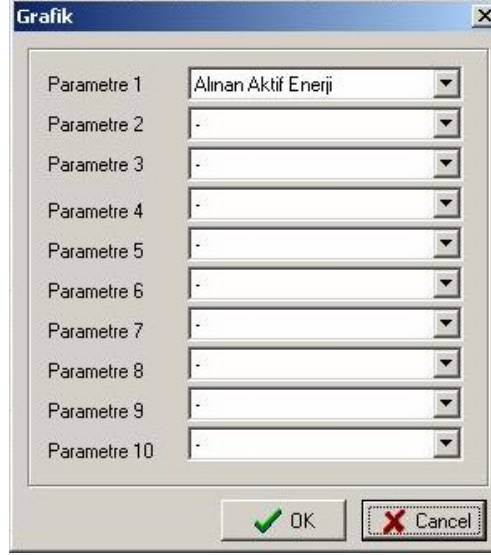
MPR-63 Grafik

Tarih	Saat	Röle Durumu	Alınan Aktif Enerji	Çıkan Aktif Enerji	İndüktif Reaktif Enerji	Kapasitif Reaktif Enerji	Gerilim LN1	Gerilim LN2	Gerilim LN3	Gerilim LN	Gerilim L12	Gerilim L23
12.05.2006	18:00	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
12.05.2006	20:20	0	0	0	0	0	3.0	3.0	3.0	3.0	0.0	0.0
12.05.2006	20:21	0	0	0	0	0	56.2	57.5	56.5	56.7	98.2	98.8
12.05.2006	20:22	0	0	0	0	0	229.4	228.1	228.1	228.5	396.5	394.8
12.05.2006	20:23	0	0	0	0	0	228.5	227.3	227.2	227.6	395.2	393.4
12.05.2006	20:24	0	0	0	0	0	2.4	2.1	2.6	2.3	0.0	0.0
12.05.2006	20:33	0	0	0	0	0	1.3	2.1	2.4	1.9	0.0	0.0
12.05.2006	20:34	0	0	0	0	0	1.2	2.1	2.5	1.9	0.0	0.0
12.05.2006	20:35	0	0	0	0	0	1.1	2.1	2.5	1.9	0.0	0.0
12.05.2006	20:36	0	0	0	0	0	1.3	2.1	2.4	1.9	0.0	0.0
12.05.2006	20:37	0	0	0	0	0	107.9	108.9	107.9	108.2	187.7	187.8
12.05.2006	20:38	0	0	0	0	0	196.1	195.4	194.9	195.4	339.4	338.1
12.05.2006	20:39	0	0	0	0	0	226.5	225.0	225.1	225.5	391.4	389.5
12.05.2006	20:40	0	0	0	0	0	226.0	224.2	224.2	224.8	390.2	388.0
12.05.2006	20:41	0	0	0	0	0	225.4	224.6	224.0	224.6	390.2	388.2
12.05.2006	20:43	0	0	0	0	0	226.4	224.7	224.8	225.3	391.0	388.9
12.05.2006	20:44	0	0	0	0	0	225.4	224.0	223.4	224.2	389.8	387.3
12.05.2006	20:45	0	0	0	0	0	2.3	2.1	2.3	2.2	0.0	0.0
12.05.2006	20:46	0	0	0	0	0	224.9	227.1	225.2	225.7	391.9	392.0

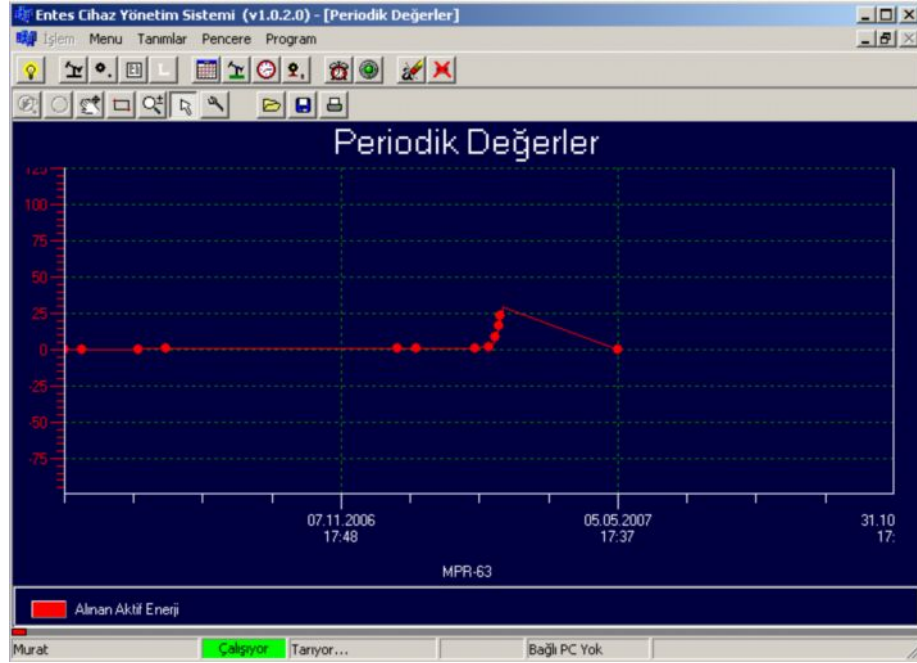
Murat Çalışıyor Tarıyor... Bağlı PC Yok

Şekil V.12 Periyodik Değerler Penceresi

Şekil V.12’ deki periyodik değerler tablosundan istediğimiz değerin zamana göre grafiği elde edilebilir. Bunun için periyodik değerler penceresinin sağ üst köşesinden ‘Grafik’ seçilir ve şekil V.13’ deki grafik penceresi açılır. Grafik penceresinden zamana göre değişimini elde etmek istediğimiz ölçülen parametre değerlerinden istenileni seçilebilir. Şekil V. 13’ de örnek olarak ‘Alınan Aktif Enerji’ seçilmiştir. Şekil V. 14’ de alınan aktif enerjinin zamana göre değişimi grafiksel olarak gösterilmektedir.



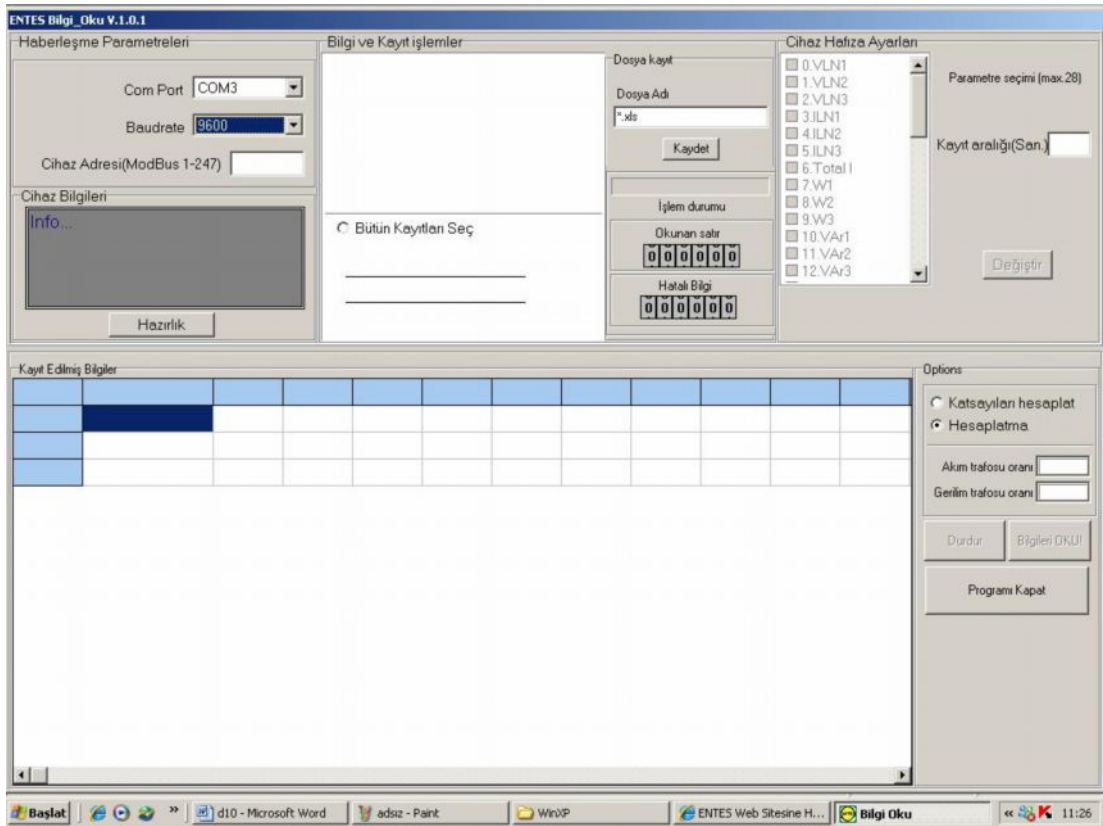
Şekil V.13 Grafik Seçimi Penceresi



Şekil V.14 Grafik Penceresi

Güç analizörü ile saniyelik değerler ölçmek ve bu ölçülen değerleri elde etmek için ilk önce 'modbus' haberleşme protokolunun kullanıldığı paket programının bilgisyara kurulması gerekmektedir. Daha sonra veri okuma programından şekil V.15' deki pencere açılır. Bu pencere üzerinden ilk olarak 'Com Port' seçilerek bağlantı noktası belirlenir. 'Boudrate' seçilerek band genişliği güç analizörünün üzerindeki ile aynı değere getirilir. Bilgi ve kayıt işlemler ara penceresinde gözükün bütün kayıtlar 'bütün kayıtları seç' ile seçilir. Toplam harmonik distorsiyonunun

hesaplatılabilmesi için ‘Katsayıları Hesaplat’ komutu kullanılır. Pencerenin alt kısmında bütün kayıtlar saniyeri ile birlikte görünür. Kaydetme işlemine istenilen zamanda ‘durdur’ komutu ile son verilir. Bu program deneylerin hepsinin yapıldıktan sonra güç analizöründe depolanmış verileri almaya yönelik olduğu için deney esnasında çalıştırılmamaktadır. Deneyler yapıldıktan sonra deney esnasında güç analizörüne kaydedilen veriler, o andaki kayıt zamanları ile birlikte elde edilir. Durdurma işleminden sonra veriler ‘Kaydet’ komutu seçilerek istenilen yere excel dosyası olarak, istenilen başlık verilerek kaydedilebilir.

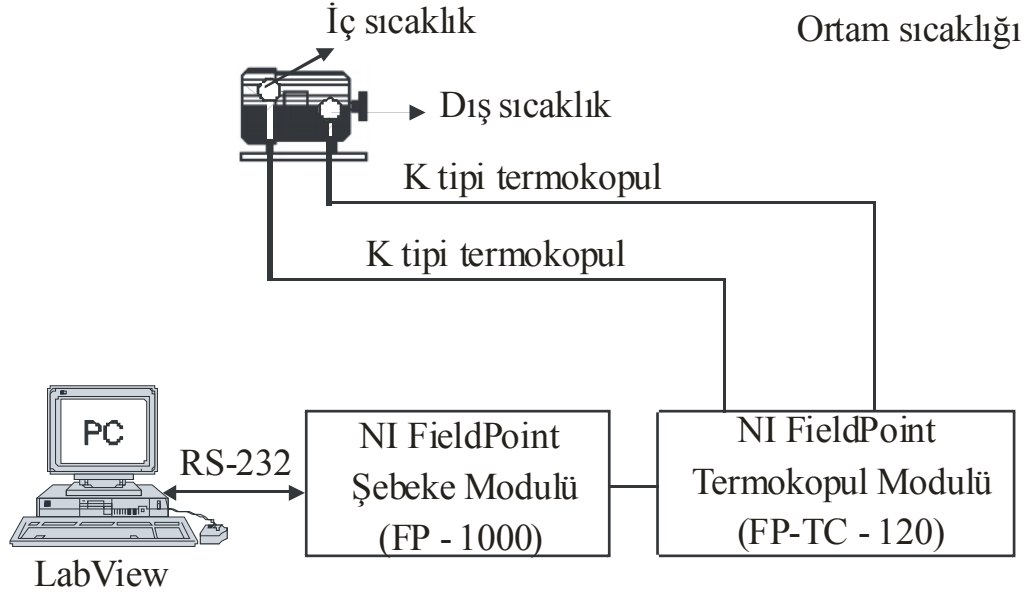


Şekil V.15 Güç Analizöründen Veri Transferi

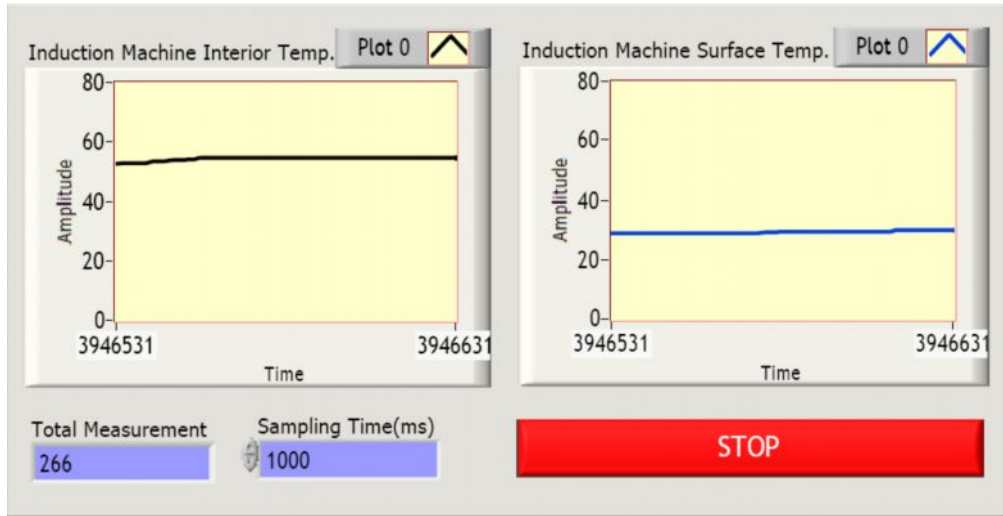
V.4.2. Motor İç ve Dış Sıcaklıklarının Elde Edilmesi

Deneyisel çalışmada kullanılan asenkron motorların iç ve dış sıcaklıkları aynı noktadan ölçülmüştür. Sıcaklık ölçümleri için Lab View paket programının sıcaklık modülleri kullanılmıştır. Şekil V.16’ da sıcaklık ölçümleri için kullanılan modüller ve bağlantı şeması gösterilmektedir. Sıcaklık verileri Lab View ön paneli

kullanılarak bilgisayara transfer edilmiştir. Bunun için kullanılan paket programın ön paneli şekil V.17’ de gösterilmektedir.



Şekil V.16 Sıcaklık Ölçümleri İçin Bağlantı Şeması



Şekil V.17 Sıcaklık Analizi İçin Lab View Ön Paneli

V.5. YAPAY SİNİR AĞI YÖNTEMİ İLE SİSTEMİN MODELLEMESİ

V.5.1. Sistemin Parametrelerinin Belirlenmesi

Tablo V.4 Oluşturulan Sistemlere Ait Ortak Giriş Parametreleri

	<i>Sembolü</i>	<i>Birimi</i>	<i>Tanımı</i>
1	V_{L1N}	Volt	Birinci faza ait faz – nötr gerilimi
2	V_{L2N}	Volt	İkinci faza ait faz – nötr gerilimi
3	V_{L3N}	Volt	Üçüncü faza ait faz – nötr gerilimi
4	V_{L1L2}	Volt	Birinci faza ait fazlararası gerilim
5	V_{L1L3}	Volt	İkinci faza ait fazlararası gerilim
6	V_{L2L3}	Volt	Üçüncü faza ait fazlararası gerilim
7	I_1	Amper	Birinci fazın akımı
8	I_2	Amper	İkinci fazın akımı
9	I_3	Amper	Üçüncü fazın akımı
10	P_1	WAT	Birinci fazın aktif gücü
11	P_2	WAT	İkinci fazın aktif gücü
12	P_3	WAT	Üçüncü fazın aktif gücü
13	Q_1	VAr	Birinci fazın reaktif gücü
14	Q_2	VAr	İkinci fazın reaktif gücü
15	Q_3	VAr	Üçüncü fazın reaktif gücü
16	S_1	VA	Birinci fazın görünür gücü
17	S_2	VA	İkinci fazın görünür gücü
18	S_3	VA	Üçüncü fazın görünür gücü
19	TF	Kherz	PWM taşıyıcı frekansı
20	BA	-	Bobin adımı
21	$T_{iç}$	C°	Motorun iç sıcaklığı
22	$T_{dış}$	C°	Motorun gövde sıcaklığı

Tez çalışmasında üç fazlı kafesli asenkron motorlarda toplam harmonik distorsiyonun tahmin edilebilmesi için iki farklı veri seti ile dört ayrı yapay sinir ağı modeli oluşturulmuştur. Örnekleme zamanı bir dakika olan veri seti ile akım THD çıkışlı ve gerilim THD çıkışlı olmak üzere iki adet, örnekleme zamanı bir saniye olan veri seti ile de akım THD ve gerilim THD çıkışlı iki adet YSA modelleri oluşturulmuştur. Bir dakikalık örnekleme zamanına sahip veri seti ile eğitilmiş olan YSA modellerinde 20 adet giriş ve tek çıkış kullanılmıştır. Bir saniyelik örnekleme

zamanına sahip YSA modellerinde ise toplam 22 adet giriş ve tek çıkış kullanılmıştır. Motor iç ve dış sıcaklıkları bir saniyelik örnekleme zamanı ile ölçülmüş ve bir saniyelik örnekleme zamanına sahip veri setinde kullanılmıştır. Çıkışlar akım için ve gerilim için olmak üzere ilk otuz harmonikten oluşan toplam harmonik distorsiyonun ölçü aletinin ölçtüğü çıkış değeri olarak temel alınmıştır. Toplam harmonik distorsiyonu, güç analizöründe birinci harmoniğin yüzde değerlerinin ilk otuz harmonik için toplamı olarak ‘fast fourier transformation’ (FFT) yöntemi ile hesaplanarak tespit edilmektedir.

Oluşturulan yapay sinir ağı modelleri iki katmanlı olup gizli katmanda kullanılan nöron sayısı 20 dir. Dakikalık veri seti ile elde edilen Y.S.A modelleri için kullanılan deneysel veriler 663 adettir. Saniyelik veri seti ile elde edilen YSA modeli için kullanılan veriler toplam 3940 adet tir. Modelleme için kullanılan verilerin % 70’ i ağı eğitilmesi sırasında , geriye kalan % 30’ luk veri ise ağı sınanması için kullanılmıştır. Veriler, oluşturulan yapay sinir ağında daha doğruya yakın sonuçlar verebilmesi için düzenlenmiş ve ağı anlayabileceği nümerik veriler haline getirilmiştir. Eğitim ve sınama süreçlerinde kullanılması için verilerin ayrılma işlemine geçilmeden evvel veri seçimi, rasgele yapılmıştır. Böylece tüm sistemi yansıtacak giriş ve çıkış parametrelerinden oluşan verilerle ağı eğitilmiş, en iyi sonucu alabilmek için yine rasgele seçilmiş verilerle de sınanmıştır. Deneysel verilerin, ağı tarafından işlenebilmesi, sinir hücrelerinde hesaplanan toplam fonksiyonun yaratacağı olumsuz etkileri önleyebilmek ve yapay sinir ağının performansını iyi yönde etkileyebilmek için normalize edilmeleri gerekir Aşağıda [-1,1] aralığı için kullanılacak normalizasyon formülü verilmiştir.

$$X_{yeni} = 2 \frac{X - x_{min}}{x_{max} - x_{min}} - 1 \quad (V.1)$$

Normalize işlemi, giriş ve çıkış verilerinin ağı tarafından işlendiklerinde aşırı salınımları sonucu ortaya çıkacak olumsuz etkileri ortadan kaldırmaya yardımcı olur. Burada oluşturulan iki yapay sinir ağı modelinde de minimum-maksimum normalizasyon yöntemi kullanılmış, giriş ve çıkış değerlerinin tamamı -1 ile +1 değerleri arasında normalize edilmiştir. MatLab yazılım programında “premnmx” komutu yardımıyla normalizasyon işlemi gerçekleştirilmiştir. Bu komut sonucu ölçeklendirilmiş normalize değerler, “postmnmx” komutu ile gerçek ilk değerlere

program sonunda dönüştürülmüştür. Sistemin modellenmesinde, oluşturulan yapay sinir ağında kullanılacak aktivasyon fonksiyonunun belirlenmesinde kesin bir kural yoktur. Bu nedenle, sistemin davranışına bağlı olarak gizli katman ve çıkış katmanındaki yapay sinir hücrelerinde kullanılan aktivasyon fonksiyonları, eğitimde elde edilen en az hata değerine göre belirlenmiştir. Oluşturulan yapay sinir ağı modellerinde, en az hata değerine tanjant hiperbolik aktivasyon fonksiyonu kullanılarak ulaşılmıştır.

Yapay sinir ağının eğitimi için performans ölçütü olarak hataların kareleri ortalamasını kullanan Levenberg-Marquardt algoritması çok uygundur. Verilen girişlere karşılık ağın hesapladığı çıkışlar ile gerçek (istenen) çıkış karşılaştırılır. Ağın çıkışı ile gerçek çıkışı arasındaki fark hata olarak hesaplanır. Bu hatanın toplamının ortalaması minimize edilmek istenir. Bu minimize edilmek istenen değer MSE (Mean Squared Error) olarak tanımlanır ve ağın daha küçük ağırlık ve performans değerlerine sahip olmasına neden olduğundan öğrenme performansını etkileyen önemli faktörlerden biridir. Bu çalışmada, en iyi sonuç karesel ortalama hata fonksiyonu kullanılarak alınmıştır. MSE aşağıdaki gibi formüle edilir.

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

Burada, $e(k)$: Hedef çıkış ve YSA çıkışları arasındaki hatayı, $y(k)$: Gerçek çıkışı, $t(k)$: YSA çıkışını ifade eder.

Sistemin oluşturulmasında kullanılan algoritma türü Geri yayılım algoritmasıdır. MatLab yazılım programında “trainlm” ile kullanılan algoritma ileri beslemeli ağ yapısındadır.

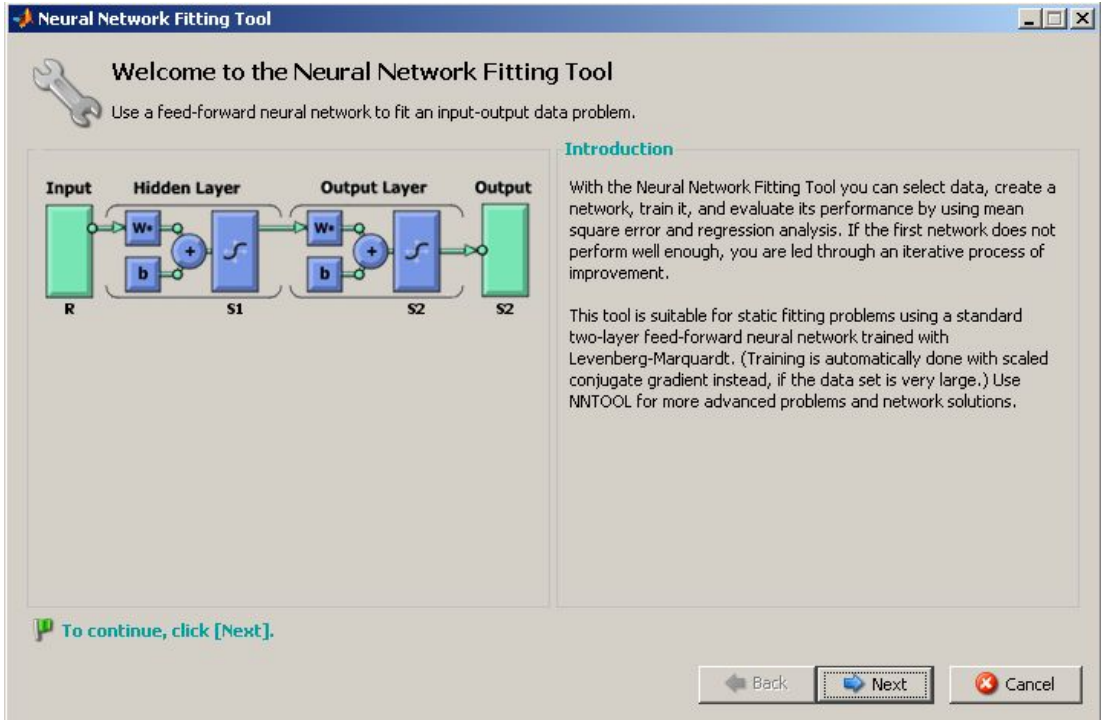
V.5.2. Sistemin Oluşturulması

Kafesli asenkron motorlarda toplam harmonik distorsiyonunun tahmin edilmesi, doğrusal olmayan ve karmaşık matematiksel denklemleri içeren bir modele ihtiyaç duyar. Doğrusal olmayan sistemleri açıklamada gösterdikleri başarılarla ön plana çıkan yapay sinir ağları, deneysel çalışmalardan elde edilen verileri kullanarak modelleme yapmada güçlü bir araçtır.

Oluşturulan sistemin başarısı, uygulanacak olan yaklaşımlar ve deneyimlerle yakından alakalıdır. Aşağıda aşamalı olarak sistemin oluşumu ve denemelerle elde edilen en iyi sonuçlar anlatılmıştır.

İlk aşama olarak; uygun yapay sinir ağı yapısının seçimi, sistemin çözümüne bağlı olarak seçilmelidir. Bu çalışmada kullanılan ağ yapısı, tahmin (prediction) sınıfına girmektedir. Uygulanan giriş değerlerinden tahmini çıkış değerleri bulunmaya çalışılır. Çok katmanlı olarak oluşturulan tahmin yapma yeteneğine sahip dört adet yapay sinir ağı modelleri oluşturulmuştur.

İkinci aşama olarak, uygulama başarısını belirleyen en önemli faktörlerden birisi olan öğrenme algoritmasının seçimidir. Bu çalışmada; oluşturulan ağ modelleri ileri beslemeli olup, geri yayılım algoritması olan Levenberg-Marquardt algoritması kullanılmıştır. Çalışmada Levenberg – Marquardt öğrenme algoritmasının kullanıldığı Matlab paket programında ki “nftoolbox” (Neural Network Fitting Tool) kullanılmıştır. Şekil V.18’ de “Neural Network Fitting Tool” ana penceresi gösterilmektedir.

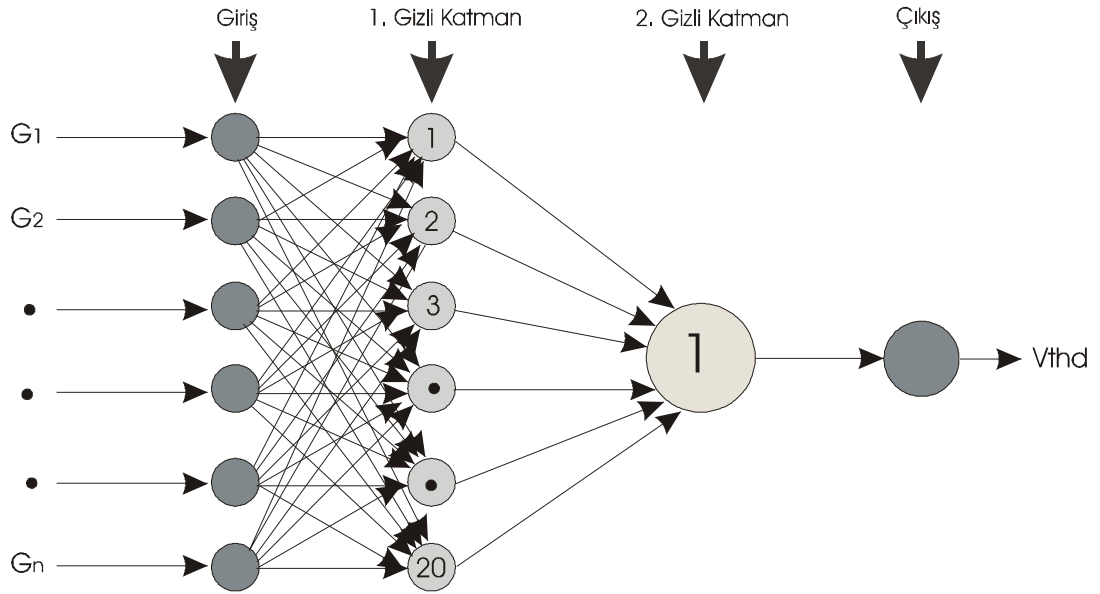


Şekil V.18 Yapay Sinir Ağı Modellemesi için NFTOOLBOX Ana Penceresi

Üçüncü aşama ise, uygulamada kullanılacak sistemin davranışı göz önünde bulundurularak seçilecek olan aktivasyon fonksiyonudur. Tez de kullanılan bütün yapay sinir ağı modellerinde, sonuçları önemli oranda etkileyecek fonksiyon tanjant hiperbolik fonksiyon seçilmiştir.

Giriş parametreleri, transfer edilirken fonksiyonun aktif olan bölgesinden aktarılmasını sağlayan normalizasyon tekniği kullanılarak bütün sinir ağı modelleri $[-1,1]$ aralığında normalize edilmiştir.

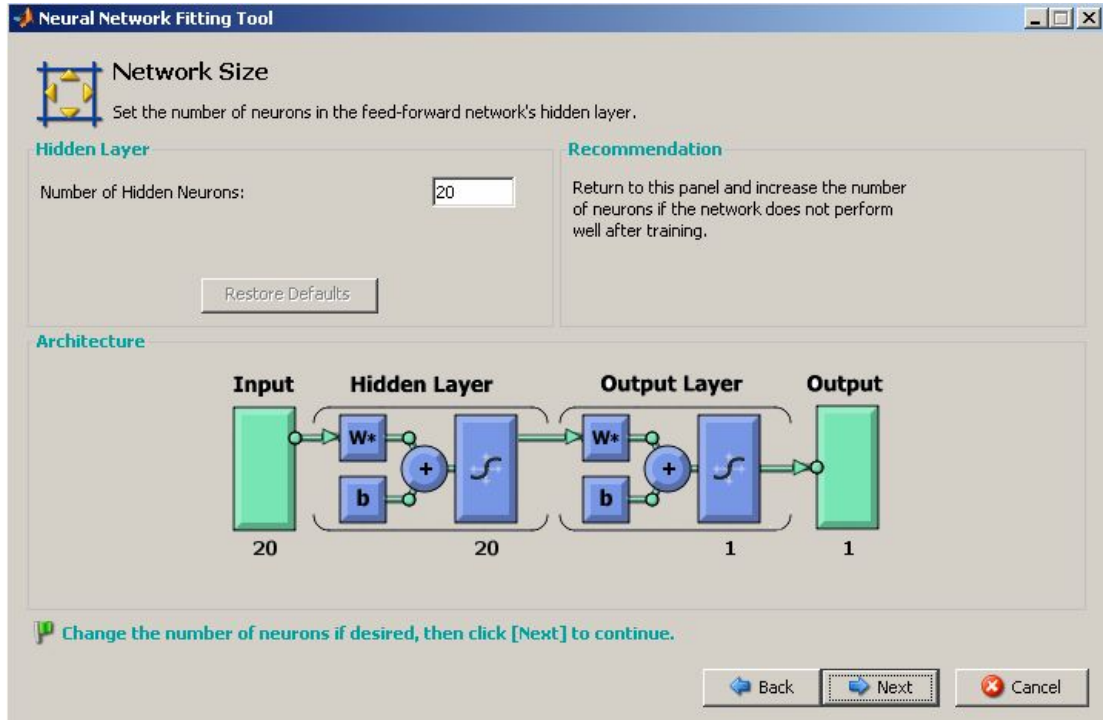
Tez çalışmasında kullanılan çok katmanlı yapay sinir ağı modellerinin yapısı aşağıda açıklanmıştır.



Şekil V.19 Çok Katmanlı Yapay Sinir Ağının Yapısı

Tez çalışmasında kullanılan bütün ağ modelleri; giriş, iki gizli katman ve çıkış olmak üzere dört katmandan oluşmaktadır. İlk gizli katmandaki yapay sinir hücre sayısı 20, ikinci gizli katmanda ise, bir adettir. Gizli katmanların sayısı kullanılan paket programda şekil V.20' de gösterilen 'nftool' penceresi açılarak seçilir. İlk gizli katman bu pencere kullanılarak 20 seçilmiştir ikinci gizli katman ise programın kendisi tarafından bir olarak belirlenmiştir. İkinci gizli katmanın bir olarak seçilmesinin sebebi direk çıkış katmanı olarak kullanılmasıdır. Girdi ve çıktılara ait değerler $[-1,1]$ aralığında minimum-maksimum normalize edilmiştir.

Sonuçların doğru değerlendirilebilmesi için bütün öğrenen yapay sinir ağı modellerinde öğrenme algoritmaları ve aktivasyon fonksiyonları aynı seçilmiştir. Ayrıca aktivasyon fonksiyonu olarak, tanjant hiperbolik fonksiyonu kullanılmıştır.



Şekil V.20 Gizli Katmandaki Hücre Sayısının Seçimi

Oluşturulan YSA modellerine ait MatLab ortamında geliştirilen ileri beslemeli yapay sinir ağına ait program kodlamaları, (M – File olarak) Ek - B, C, D ve E’ de verilmiştir.

V.5.3. Yapay Sinir Ağı Modellerine Ait Elde Edilen Sonuçlar

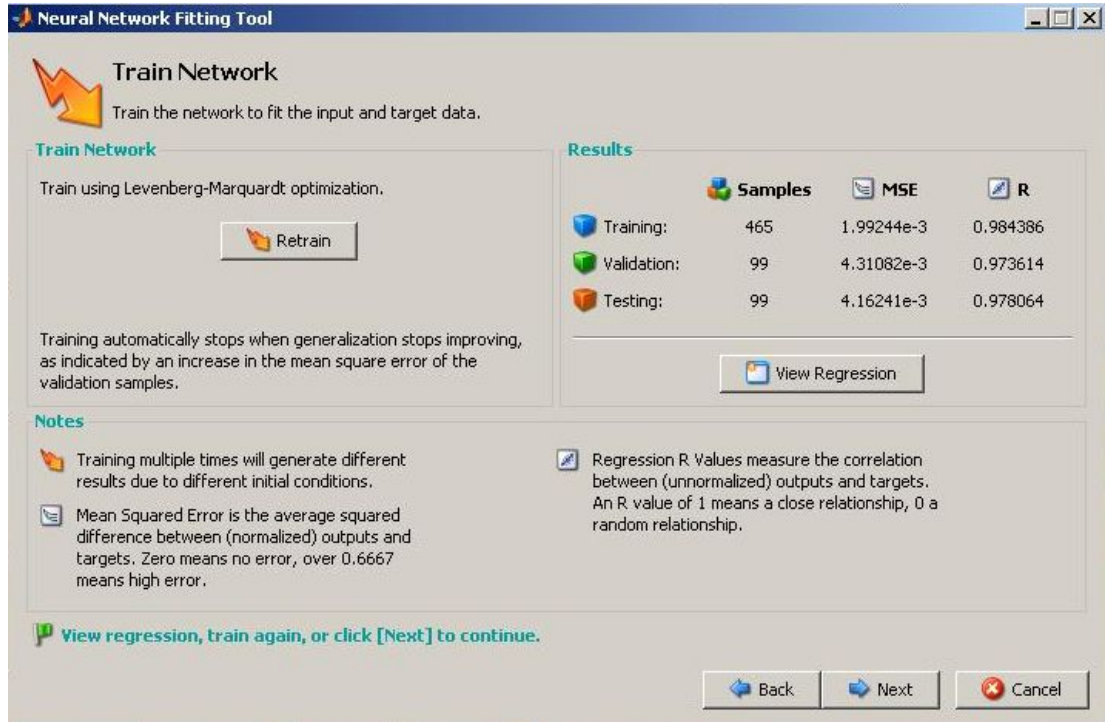
Oluşturulmuş yapay sinir ağı modellerinin test verileriyle sınanması sonucunda, bir saniyelik örnekleme zamanı ile eğitilen yapay sinir ağı modelinin daha iyi öğrendiği, istenilen sonuçlara bir dakikalık eğitim seti ile elde edilen YSA modellerine göre daha yakın değerler elde ettiği görülmüştür.

Tablo V.5 YSA Modellerine Göre Sonuçların Karşılaştırılması

	<i>Örnekleme (Sn)</i>	<i>Çıkış</i>	<i>MSE (Test)</i>	<i>R (Regression) (Test)</i>
YSA 1	60	I THD	0,00416241	0,978064
YSA 2	60	V THD	0,0149844	0,961758
YSA 3	1	I THD	0,0048809	0,96924
YSA 4	1	V THD	0,00017334	0,99716

V.5.3.1. Dakikalık Akım THD Çıkışlı YSA Modeli Sonuçları

Eğitim sırasında hesaplanan MSE değerleri ile eğitim adımı arasındaki değişim, bir dakika örnekleme zamanına sahip olan veri seti ile eğitilerek oluşturulan YSA modeli için Şekil V.22’ de gösterilmiştir. Eğitim adım sayısının artışıyla 1 değerinden referans alınarak başlayan MSE değeri, onyedinci eğitim adımına doğru hızla bir düşüş göstermekte ve onyedinci eğitim adımı sonunda 0.00133261 değerinde sonlanmaktadır.



Şekil V.21 YSA Modeli İçin Elde Edilen Eğitim Sonuçları Penceresi

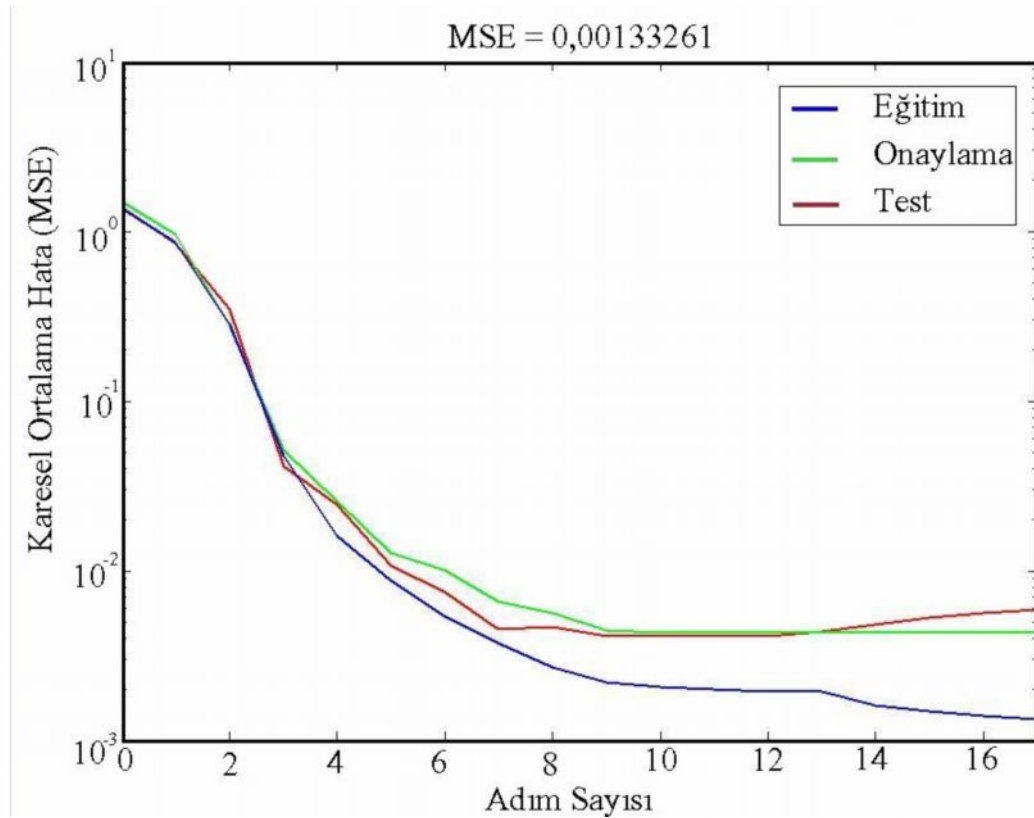
Şekil V.21’ de dakikalık akım THD çıkışına sahip YSA modeli için elde edilen eğitim sonrası NFTOOL sonuç penceresi gösterilmektedir. YSA modelinde eğitim için kullanılan veriler 465 adettir. Geriye kalan veri setinin % 30’ luk kısmının % 15 ‘i (99 adet) modellenen sistemin eğitilebildiğini doğrulamak (onaylamak) için kullanılmıştır. Diğer veri setinin % 15’ lik kısmı (99 adet) ise YSA modelini test etmek amacıyla kullanılmıştır. Tablo V.6’ da bir dakika örnekleme zamanına sahip veri seti ile oluşturulan akım THD çıkışına sahip YSA modelinden elde edilen sonuçlar gösterilmektedir.

Tablo V.6 Dakikalık I THD Çıkışlı YSA Modeli Sonuçları

	<i>Veri Sayısı</i>	<i>MSE</i>	<i>Yaklaşım (R)</i>
<i>Eğitim</i>	465	0,00199244	0,984386
<i>Onaylama</i>	99	0,00431082	0,973614
<i>Test</i>	99	0,00416241	0,978064

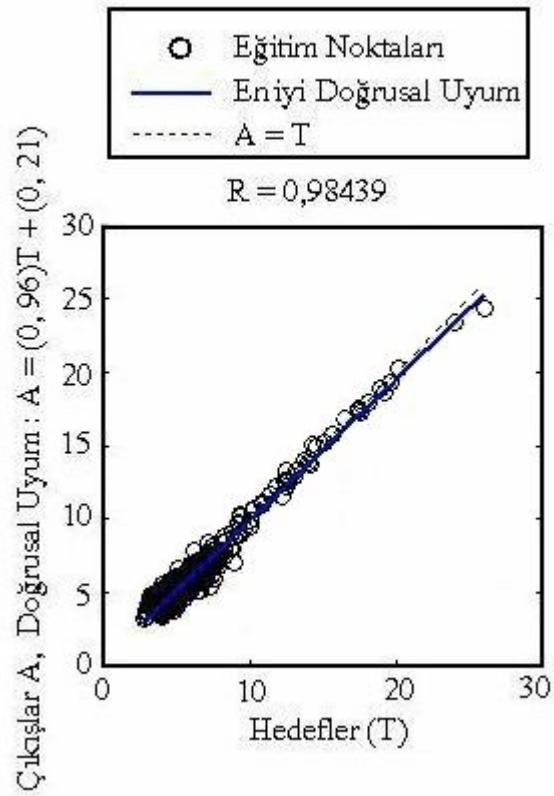
Karesel ortalama hata (MSE) değerinin sıfıra yakın olması tahminin ne kadar iyi olduğunun bir göstergesidir. Benzer şekilde yaklaşım oranının da bire yakın olması gerekmektedir. Yaklaşım oranı modelin % kaç tahmin edebildiği konusunda fikir vermektedir. Tablo V. 6' da eğitim için kullanılan 465 veri % 98,4386 oranında bir tahmin yapmıştır.

Dakikalık akım çıkışlı YSA modeline ait gerçek sonuçlar (deneysel sonuçlar) ile YSA sonuçları arasındaki yaklaşım eğrileri; eğitim, onaylama, ve test için sırasıyla şekil V.23, V.24 ve V.25' de gösterilmektedir.

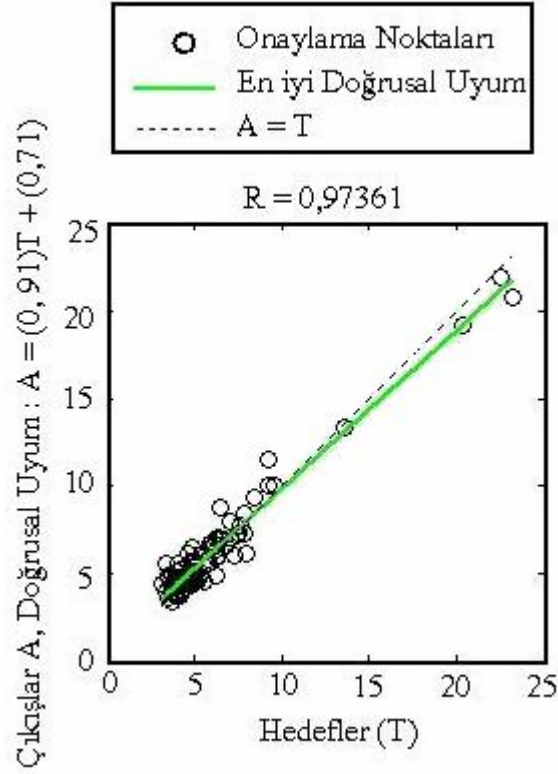


Şekil V.22 I THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi

Bir dakika örnekleme zamanına sahip veri seti ile oluşturulmuş akım THD çıkışına sahip YSA modelinde eğitim adım sayısına göre karesel ortalama mutlak hata değeri değişimi şekil V.22’ de gösterilmektedir. Eğitim adım sayısının değeri, kullanılan paket program tarafından rastgele seçilmiş olup değişim değerlerinin görülmesi için onyeddi olarak belirlenmiştir. Oluşturulan bu model için, en iyi sonucun 17 eğitim adımında elde edildiği ve 0,00133261 karesel ortalama hata payı ile sistemi en iyi şekilde yansıttığı görülmüştür.

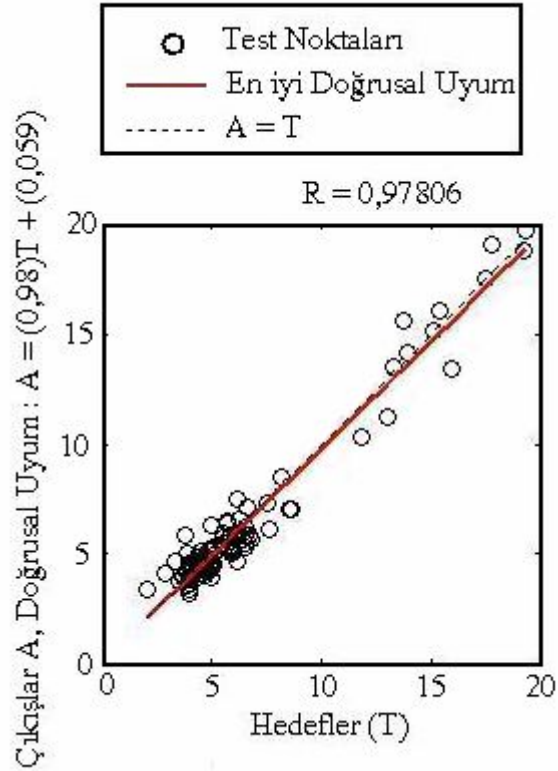


Şekil V.23 Dakikalık Akım Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi



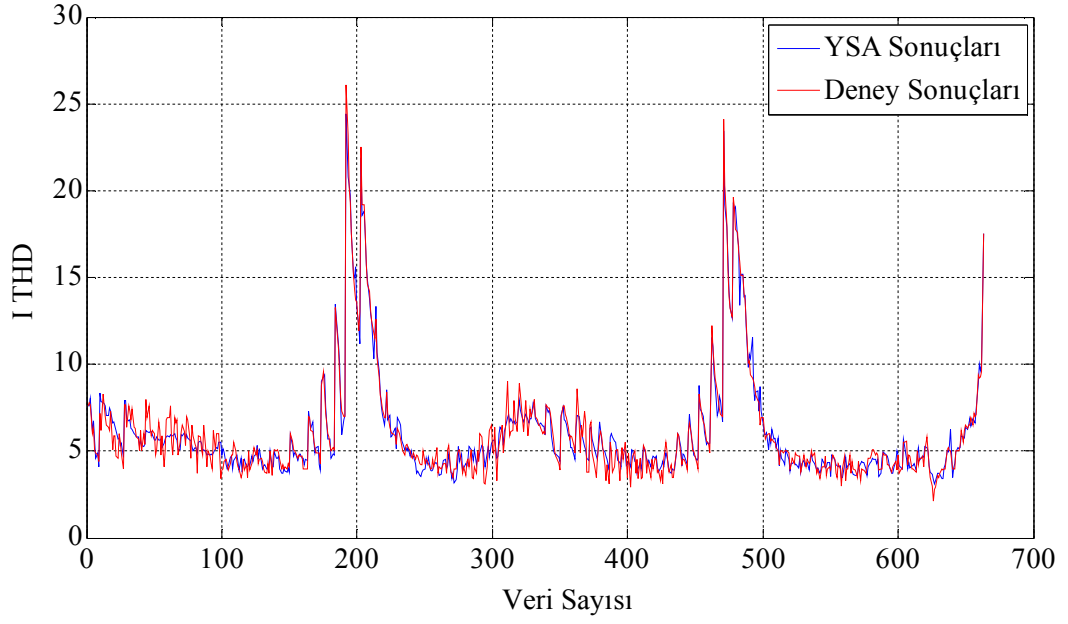
Şekil V.24 Dakikalık Akım Çıkışı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi

Şekil V. 26' da oluşturulan YSA modelinin çıkışları (YSA sonuçları) ile veri setinde kayıtlı olan, deneysel olarak elde edilmiş gerçek sonuçlar (Deney Sonuçları) her veri için akım THD çıkışına göre kıyaslanmıştır. Kırmızı ile gösterilen sonuçlar deney sonuçlarıdır. Mavi ile gösterilenler ise oluşturulan YSA modelinin ürettiği tahminleri temsil etmektedir. % 100' lük bir tahminin olabilmesi için bu iki eğrinin tam olarak üst üste binmesi gerekmektedir. Şekilden de anlaşılacağı üzere bu iki sonuç tam olarak üst üste örtüşmemektedir. Fakat yukardaki tahmin oranları dikkate alındığında birbirlerine çok yakın sonuçlar olduğu görülebilir.



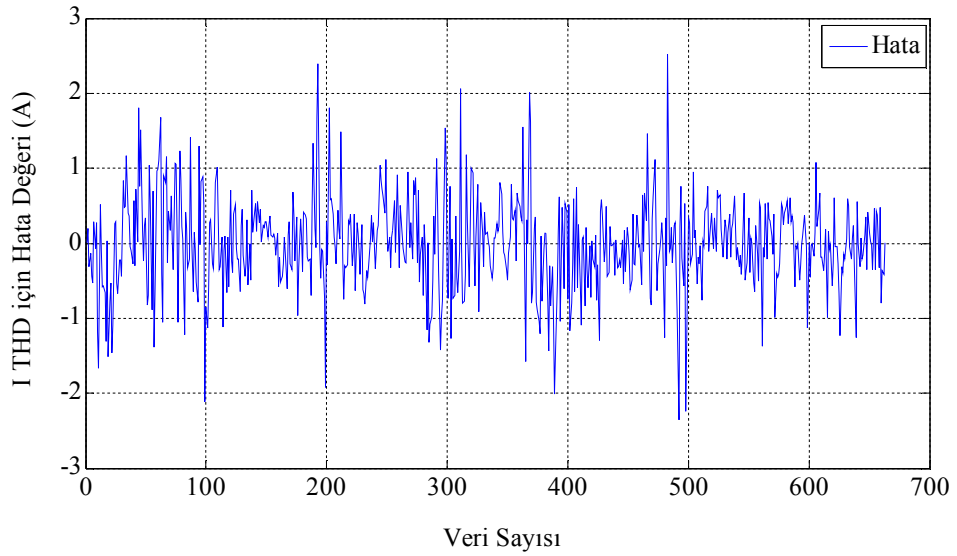
Şekil V.25 Dakikalık Akım Çıkışı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi

Şekil V. 26' da oluşturulan YSA modelinin çıktıları (YSA sonuçları) ile veri setinde kayıtlı olan, deneysel olarak elde edilmiş gerçek sonuçlar (Deney Sonuçları) her veri için akım THD çıkışına göre kıyaslanmıştır. Kırmızı ile gösterilen sonuçlar deney sonuçlarıdır. Mavi ile gösterilenler ise oluşturulan YSA modelinin ürettiği tahminleri temsil etmektedir. % 100' lük bir tahminin olabilmesi için bu iki eğrinin tam olarak üst üste binmesi gerekmektedir. Şekilden de anlaşılacağı üzere bu iki sonuç tam olarak üst üste örtüşmemektedir. Fakat yukardaki tahmin oranları dikkate alındığında bir birlerine çok yakın sonuçlar olduğu görülebilir.



Şekil V.26 YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması

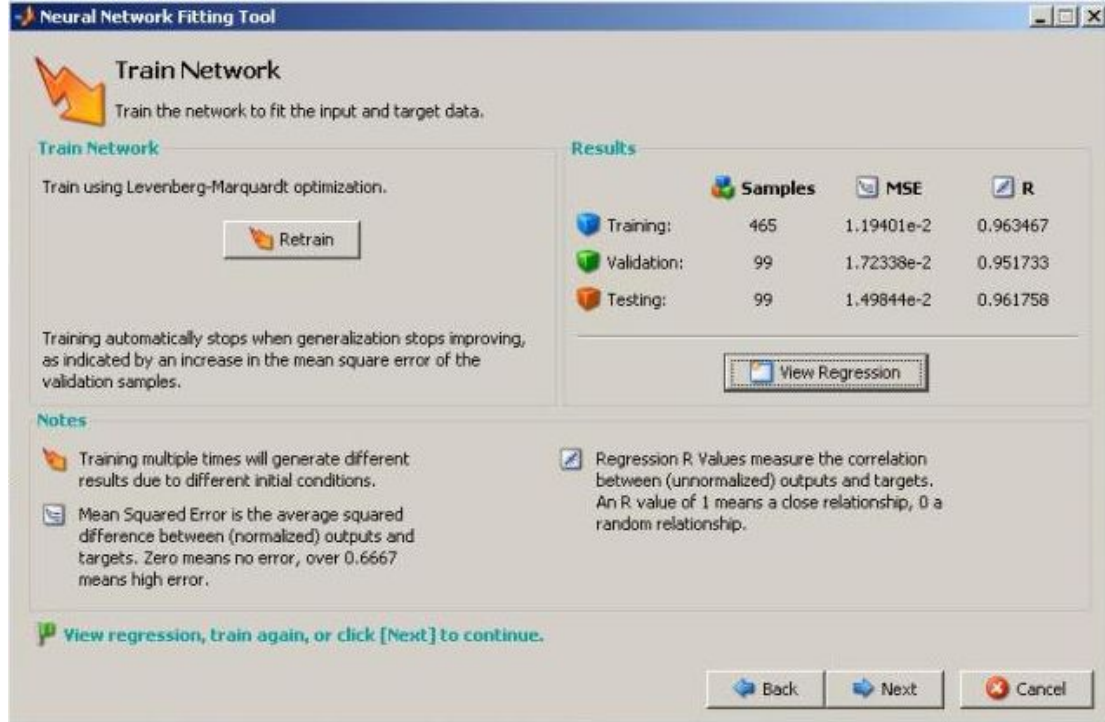
Sonuçların tam olarak üst üste örtüşmemesinin sebebi olan, gerçek sonuç ile YSA' nın sonuçlarının birbirinden çıkartılarak elde edilen, asıl değere ne kadar yaklaşıldığını bize bildiren hata eğrisi şekil V.27' de gösterilmektedir.



Şekil V.27 YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata)

V.5.3.2. Dakikalık Gerilim THD Çıkışlı YSA Sonuçları

Çıkışı gerilim THD olan YSA modeli için; eğitim sırasında hesaplanan MSE değerleri ile eğitim adımı arasındaki değişim, Şekil V.229’ da gösterilmiştir. Eğitim adım sayısının artışıyla 1 değerinden referans alınarak başlayan MSE değeri, ondokuzuncu eğitim adımına doğru düşüş göstermekte ve ondokuzuncu eğitim adımı sonunda 0.0086623 değerinde sonlanmaktadır.



Şekil V.28 YSA Modeli İçin Elde Edilen Eğitim Sonuçları Penceresi

Şekil V.28’ de dakikalık gerilim THD çıkışına sahip YSA modeli için elde edilen eğitim sonrası NFTOOL sonuç penceresi gösterilmektedir. YSA modelinde eğitim için kullanılan veriler akım çıkışlı modelde olduğu gibi 465 adettir. Geriye kalan veri setinin % 30’ luk kısmının % 15 ‘i (99 adet) modellenen sistemin eğitilebildiğini doğrulamak (onaylamak) için kullanılmıştır. Diğer veri setinin % 15’ lik kısmı (99 adet) ise YSA modelini test etmek amacıyla kullanılmıştır. Tablo V.7’ de gerilim THD çıkışına sahip YSA modelinden elde edilen sonuçlar gösterilmektedir.

Tablo V.7 Dakikalık V THD Çıkışlı YSA Modeli Sonuçları

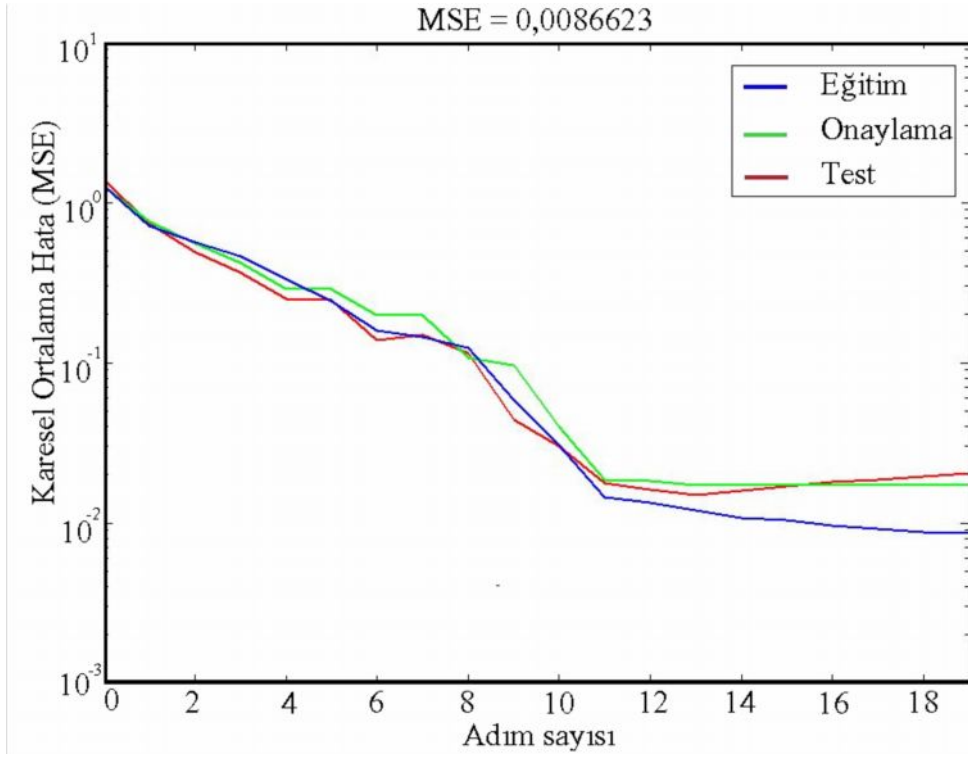
	Veri Sayısı	MSE	Yaklaşım (R)
--	-------------	-----	--------------

<i>Eđitim</i>	465	0,0119401	0,963467
<i>Onaylama</i>	99	0,0172338	0,951733
<i>Test</i>	99	0,0149844	0,961758

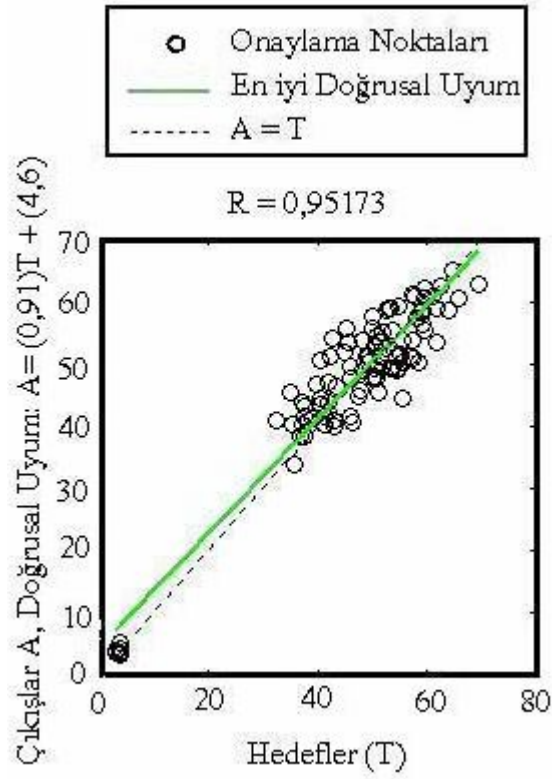
Karesel ortalama hata (MSE) deęeri akım ıkıřlı modelde olduęu gibi sifıra yakın olmasına raęmen akım ıkıřlı YSA modeline kıyasla daha kt tahmin rettięini bildirmektedir. Benzer řekilde yaklařım oranının da bire yakın olması gerekmektedir. Tablo V. 7’ de eđitim iin kullanılan 465 veri % 96,3467 oranında bir tahmin yapmıřtır.

Gerilim ıkıřlı YSA modeline ait gerek sonular ile YSA sonuları arasındaki yaklařım eđrileri; eđitim, onaylama, ve test iin sırasıyla řekil V.30, V.31 ve V.32’ de gsterilmektedir.

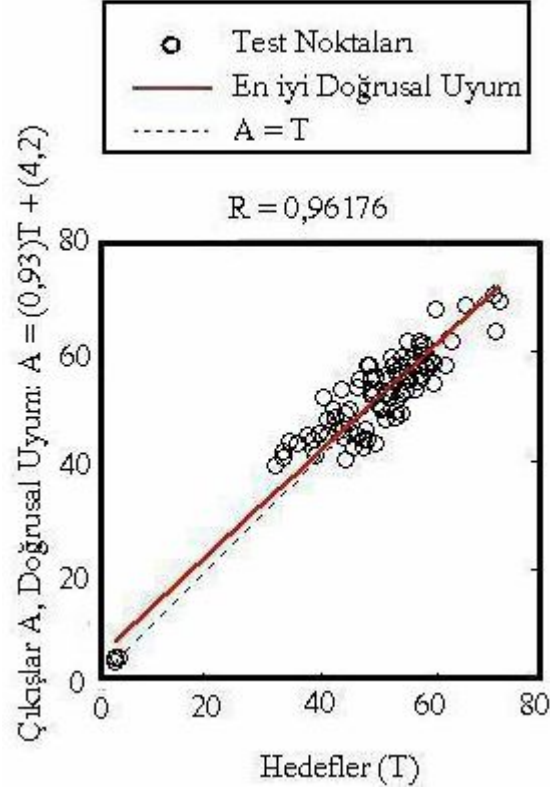
Gerilim THD ıkıřına sahip YSA modelinde eđitim adım sayısı ile karesel ortalama hata deęeri deęiřimi řekil V.29’ da gsterilmektedir. Eđitim adım sayısının deęeri, kullanılan paket program tarafından, deęiřim deęerlerinin grlmesi iin ondokuz olarak belirlenmiřtir. Oluřturulan bu model iin, en iyi sonucun 19 eđitim adımında elde edildięi ve 0,0086623 karesel ortalama hata payı ile sistemi en iyi řekilde yansıttıęı grlmřtir.



Şekil V.29 V THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi

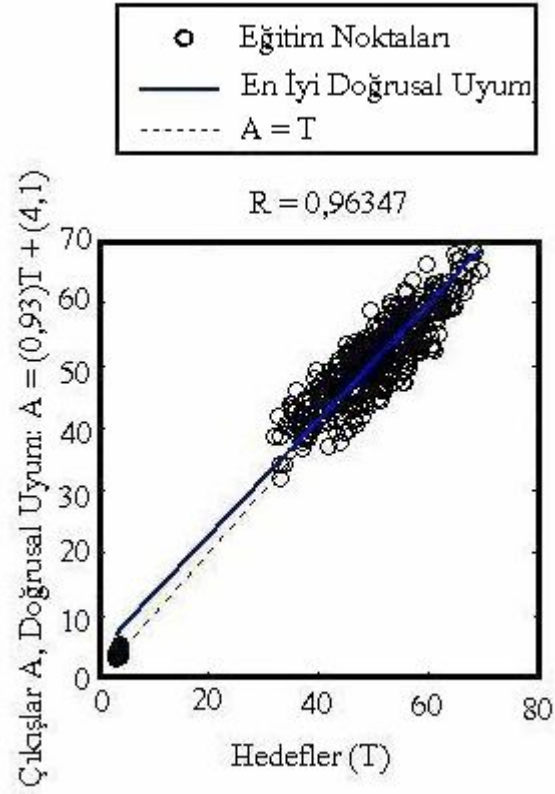


Şekil V.30 Dakikalık Gerilim THD Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi

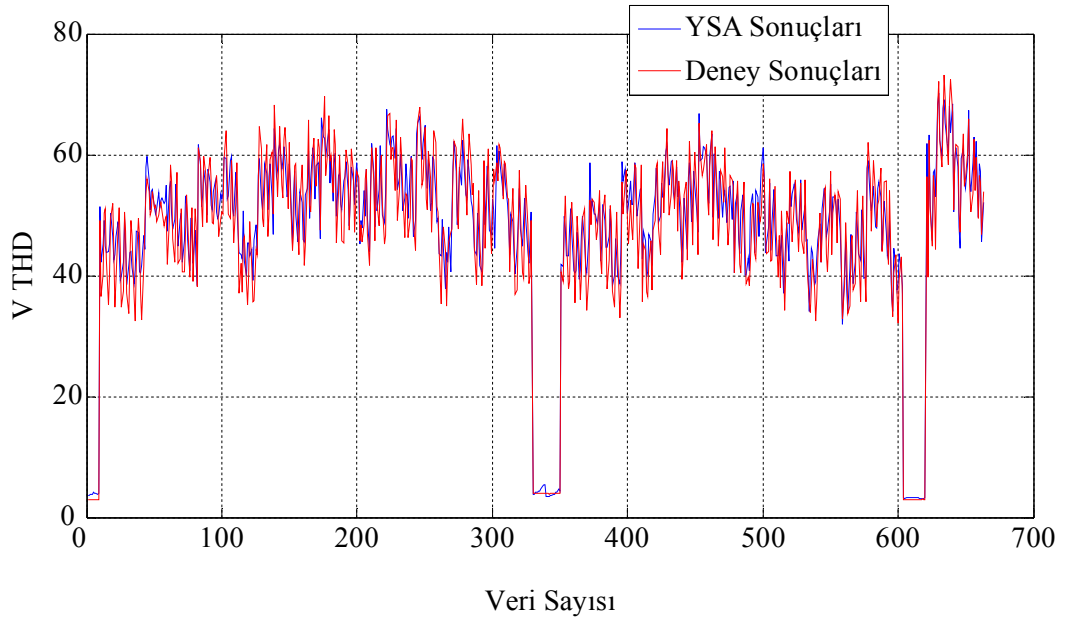


Şekil V.31 Dakikalık Gerilim THD Çıkışlı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi

Şekil V. 33' de oluşturulan YSA modelinin çıkışları ile gerçek sonuçlar (Deney Sonuçları) her veri için gerilim THD çıkışına göre kıyaslanmıştır. Kırmızı ile gösterilen sonuçlar deney sonuçlarıdır. Mavi ile gösterilenler ise oluşturulan YSA modelinin ürettiği tahminleri temsil etmektedir. Akım çıkışlı modelde olduğu gibi % 100' lük bir tahminin olabilmesi için bu iki eğrinin tam olarak üst üste binmesi gerekmektedir. Şekilden anlaşılacağı üzere bu iki sonuç tam olarak üst üste örtüşmemektedir. Tablo V.7' de belirtilen sonuçlar dikkate alındığında birbirlerine yakın sonuçlar olduğu görülebilir.

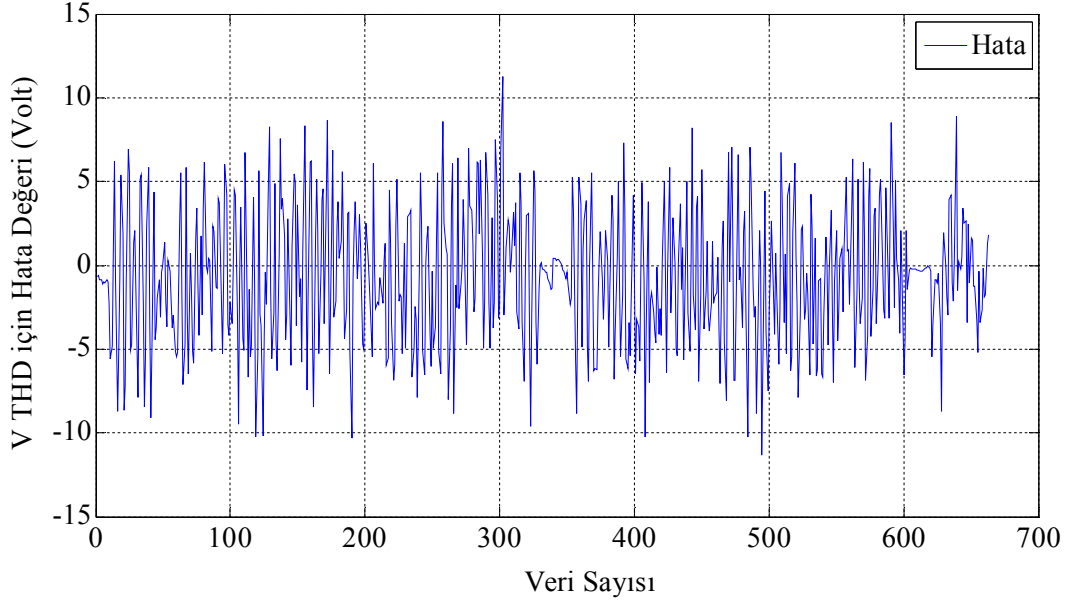


Şekil V.32 Dakikalık Gerilim THD Çıkışı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi



Şekil V.33 YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması

Gerçek sonuç ile YSA' nın sonuçlarının birbirinden çıkartılarak elde edilen, akım THD çıkışlı dakikalık YSA modeli için hata eğrisi şekil V.34' de gösterilmektedir.



Şekil V.34 YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata)

Sistemde giriş parametresi olarak kullanılan; stator sargı adımı ve PWM taşıyıcı frekansı değer değişimleri çıkıştaki ilk otuz harmonikten oluşan toplam harmonik distorsiyonun % değerini önemli ölçüde etkilemektedir.

V.5.3.3. Saniyelik Akım THD Çıkışlı YSA Modeli Sonuçları

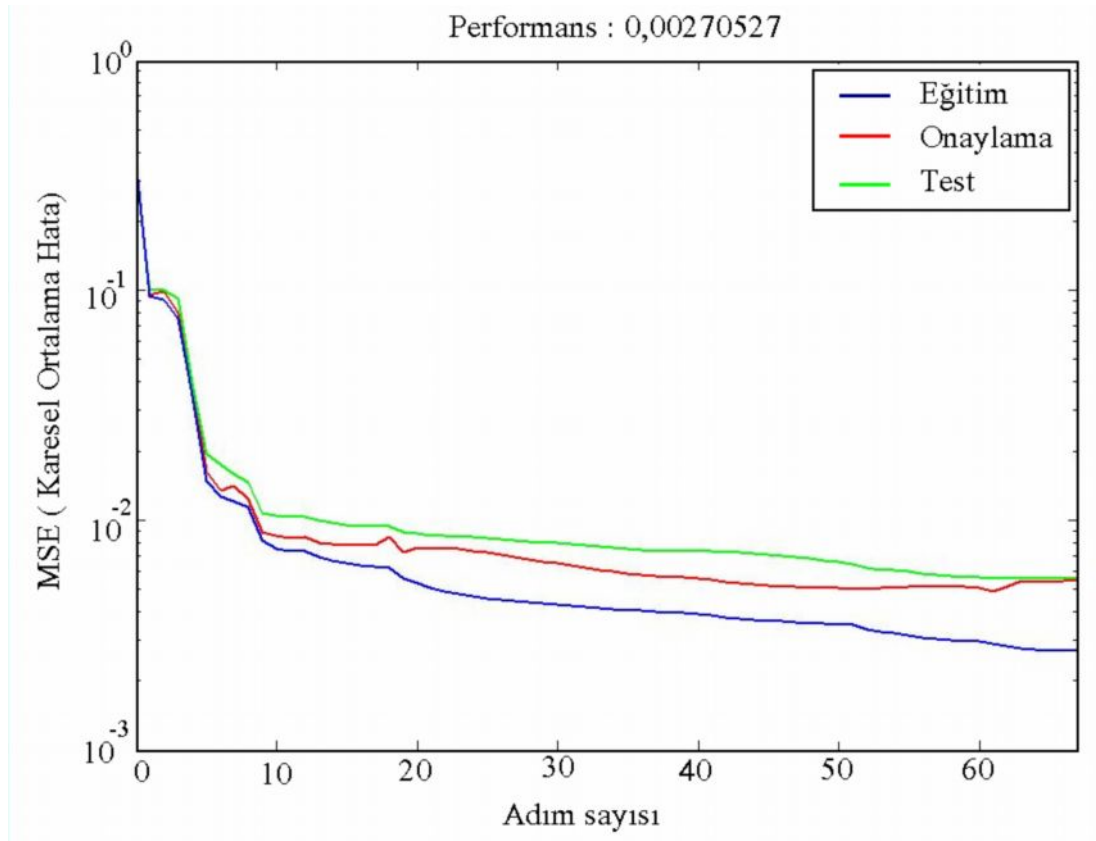
Bir saniye örnekleme zamanı ile oluşturulan veri seti ile elde edilen akım THD çıkışına sahip YSA modelinde, eğitim sırasında hesaplanan MSE değerleri ile eğitim adımı arasındaki değişim Şekil V.2235' de gösterilmiştir. 1 değerinden referans alınarak başlayan MSE değeri, eğitim adımı arttıkça düşmektedir. En iyi sonucun alındığı yetmişinci eğitim adımında, 0.00270527 ortalama karesel hata değerine ulaşılmıştır.

Eğitim için kullanılan veriler 2758 adettir. Geriye kalan veri setinin % 30' luk kısmının % 15 'i (591 adet) modellenen sistemin eğitilebildiğini doğrulamak (onaylamak) için kullanılmıştır. Diğer veri setinin % 15' lik kısmı (591 adet) ise YSA modelini test etmek amacıyla kullanılmıştır. Tablo V.8' de akım THD çıkışına

sahip bir saniye örnekleme zamanı ile elde edilmiş olan, veri seti kullanılarak modellenen, YSA modelinden elde edilen sonuçlar gösterilmektedir.

Tablo V.8 Saniyelik I THD Çıkışlı YSA Modeli Monuçları

	<i>Veri Sayısı</i>	<i>MSE</i>	<i>Yaklaşım (R)</i>
<i>Eğitim</i>	2758	0,002904	0,97882
<i>Onaylama</i>	591	0,0055928	0,96774
<i>Test</i>	591	0,0048809	0,96924

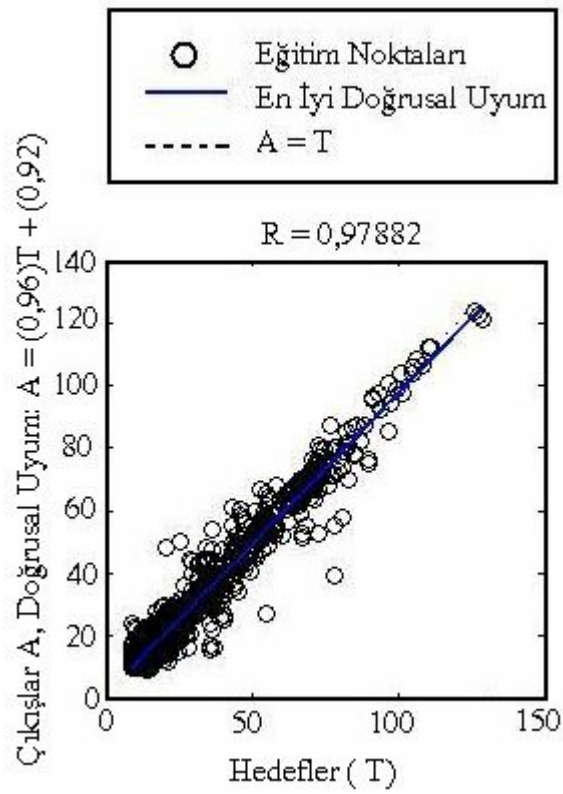


Şekil V.35 I THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi

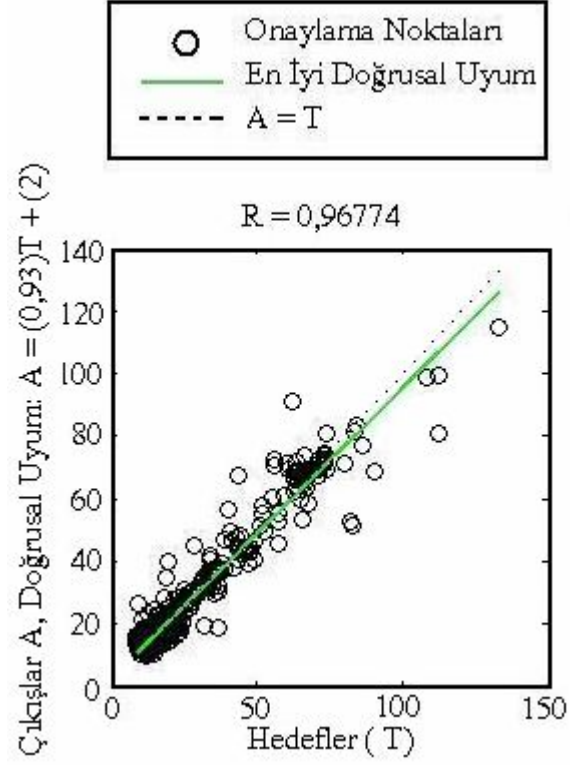
Karesel ortalama hata (MSE) değeri eğitim, onaylama ve test için sıfıra oldukça yakındır. Yaklaşım oranları ise üretilen tahmini sonuçların ne kadar güvenilir olduğunu bize bildirmektedir. Tablo V. 7’ de eğitim için kullanılan veri % 97,882 oranında bir tahmin üretmiştir. Eğitim adım sayısının değeri, kullanılan paket program tarafından, değişim değerlerinin görülmesi için yetmiş olarak belirlenmiştir.

Oluşturulan bu model için, en iyi sonucun 70 eğitim adımında elde edildiği ve 0.00270527 karesel ortalama hata payı ile sistemi en iyi şekilde yansıttığı görülmüştür.

Çıkışı akım THD olan, bir saniye örnekleme zamanı ile elde edilmiş veri seti ile modellenen YSA modeline ait sonuçlar ile deney sonuçları (gerçek sonuçları) arasındaki yaklaşım eğrileri; eğitim, onaylama, ve test için sırasıyla şekil V.36, V.37 ve V.38' de gösterilmektedir.



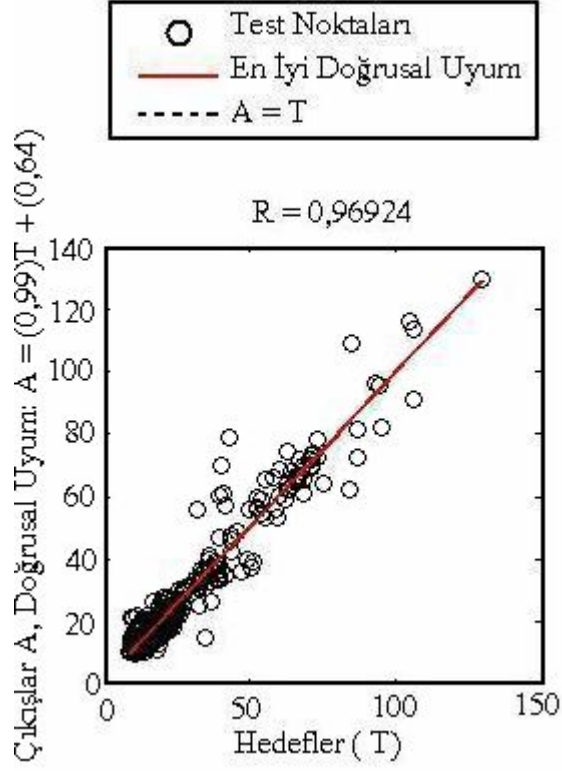
Şekil V.36 Saniyelik akım THD Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi



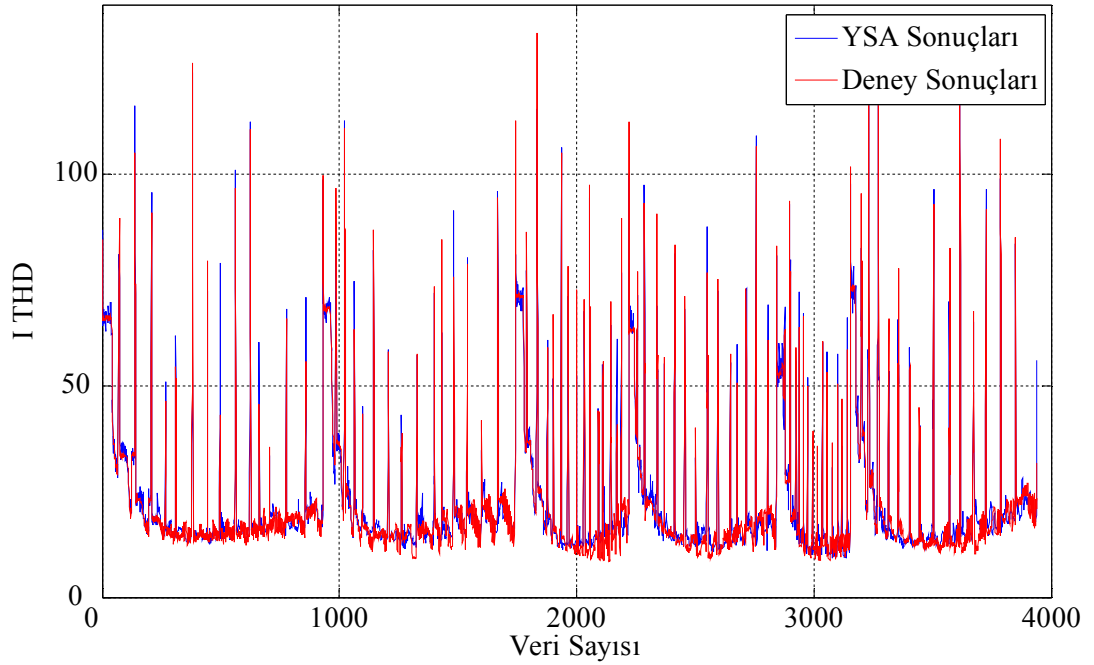
Şekil V.37 Saniyelik akım THD Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi

Oluşturulan YSA modelinin çıkışları ile gerçek sonuçlar (Deney Sonuçları) her veri için akım THD çıkışına göre kıyaslanmıştır. Kırmızı ile gösterilen sonuçlar deney sonuçlarıdır. Mavi ile gösterilenler ise oluşturulan YSA modelinin ürettiği tahminleri temsil etmektedir. Diğer YSA modellerinde olduğu gibi % 100' lük bir tahminin olabilmesi için bu iki eğrinin tam olarak üst üste örtüşmesi gerekmektedir.

3940 adet veri seti kullanılarak eğitilen ve oluşturulan YSA modelinin amacı, öğrenmeyi artırmak ve oluşan hata payını en aza indirmektir. Veri sayısı arttıkça öğrenme için, gerekli zaman da artmaktadır.

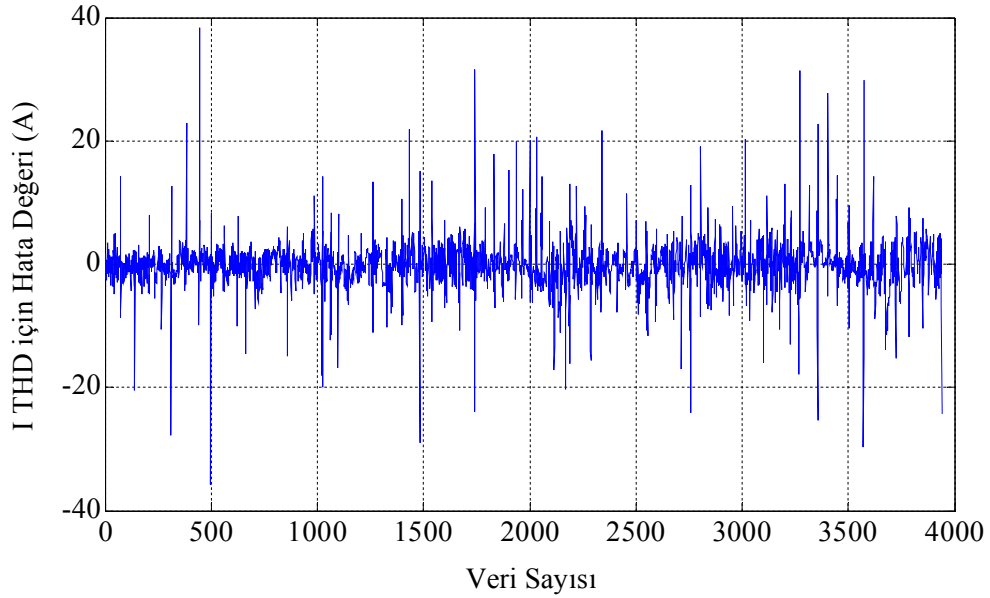


Şekil V.38 Saniyelik akım THD Çıkışı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi



Şekil V.39 YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması

Gerçek sonuç (Deneysel sonuç) ile YSA' nın sonuçlarının birbirinden çıkartılması ile elde edilen, çıkışı akım THD olan, bir saniye örnekleme zamanına sahip veri seti kullanılarak oluşturulmuş YSA modeli için hata eğrisi şekil V.34' de gösterilmektedir.



Şekil V.40 YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata)

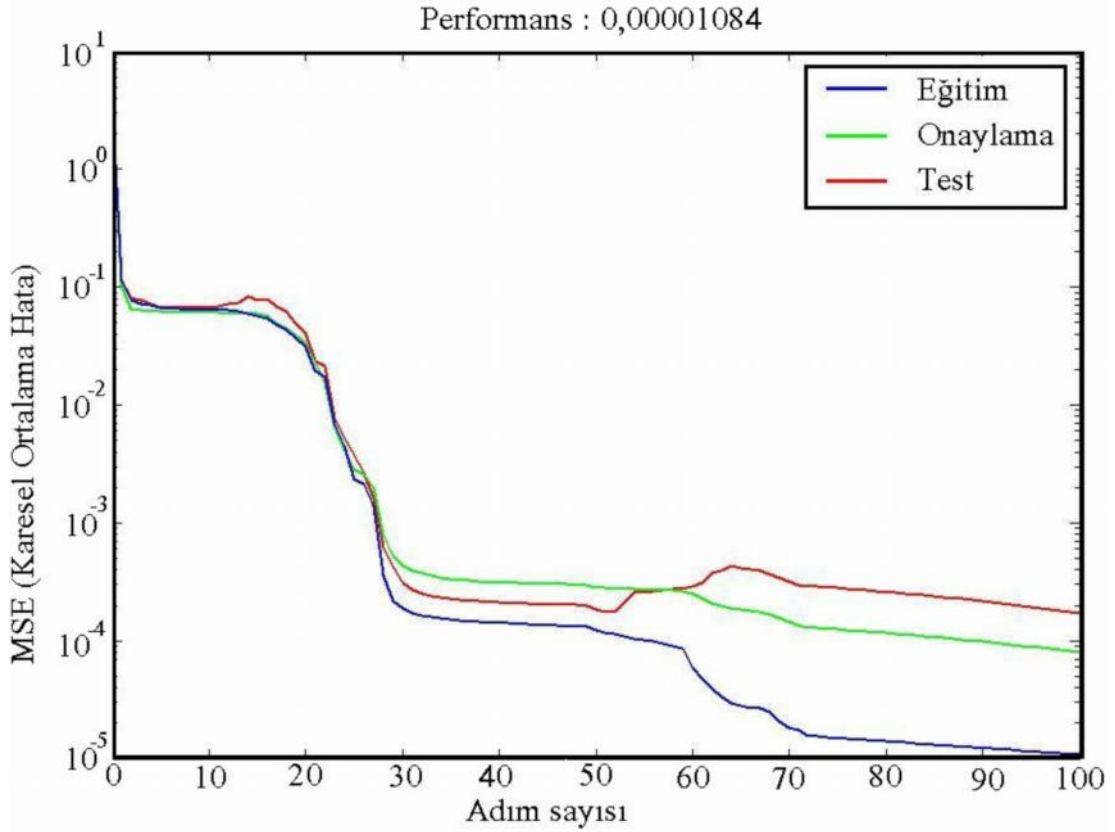
V.5.3.4. Saniyelik Gerilim THD Çıkışlı YSA Modeli Sonuçları

Gerilim THD çıkışına sahip YSA modelinde, eğitim sırasında hesaplanan MSE değerleri ile eğitim adımı arasındaki değişim Şekil V.2241' de gösterilmiştir. 1 değerinden başlayan MSE değeri, eğitim adımı arttıkça, diğer YSA modellerinde olduğundan daha ivmeli bir şekilde düşmektedir. En iyi sonuç; yüzüncü eğitim adımında, 0.00001084 ortalama karesel hata değerine ulaşılmıştır.

Eğitim için kullanılan veriler bu model için de diğer modellerde olduğu gibi toplam verinin % 70' ini oluşturmaktadır ve 2758 adettir. Geriye kalan veri setinin % 30' luk kısmının % 15 'i (591 adet) modellenen sistemin eğitilebildiğini doğrulamak (onaylamak) ve ezberlemediği kısmını pekiştirmek için kullanılmıştır. Diğer veri setinin % 15' lik kısmı (591 adet) ise YSA modelini test etmek amacıyla kullanılmıştır. Tablo V.9' de YSA modelinden elde edilen sonuçlar gösterilmektedir.

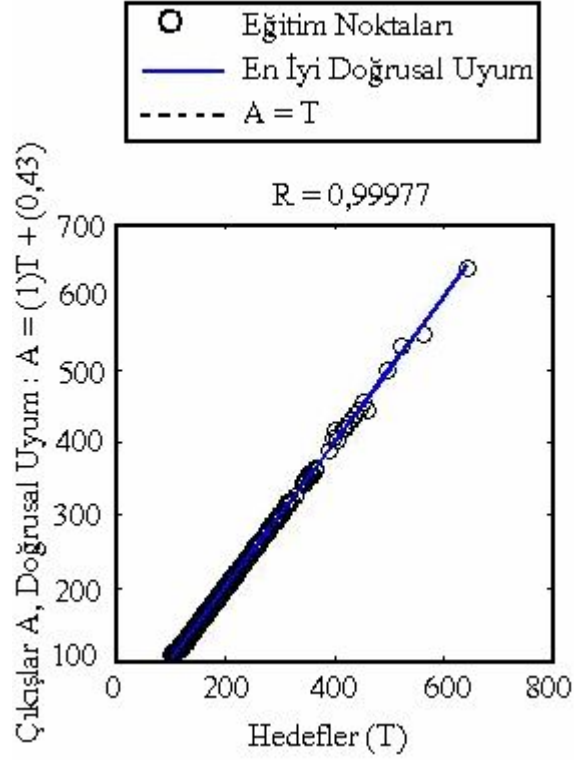
Tablo V.9 Saniyelik V THD Çıkışlı YSA Modeli Sonuçları

	<i>Veri Sayısı</i>	<i>MSE</i>	<i>Yaklaşım (R)</i>
<i>Eğitim</i>	2758	0,00001084	0,99977
<i>Onaylama</i>	591	0,00008051	0,99813
<i>Test</i>	591	0,00017334	0,99716

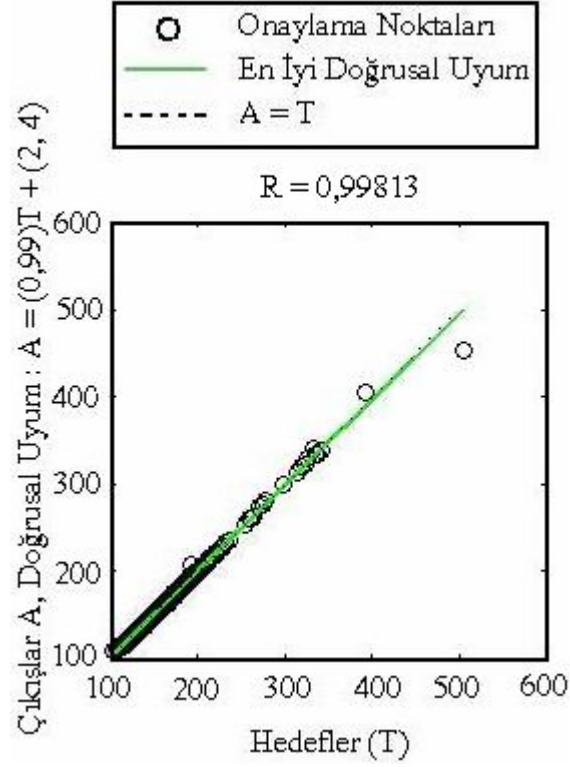


Şekil V.41 V THD Çıkışlı YSA Modeline Ait Adım Sayısına Göre MSE Değişimi

Karesel ortalama hata (MSE) değeri eğitim, onaylama ve test için oluşturulan diğer YSA modellerinden daha bariz bir şekilde sifira yakındır. Tablo V. 7' de eğitim için kullanılan veri % 99,977 oranında bir tahmin üretmiştir. En iyi sonucun yüzüncü eğitim adımında elde edildiği ve 0.00001084 karesel ortalama hata payı ile sistemi en iyi şekilde yansıtmaktadır. YSA modeline ait sonuçlar ile deney sonuçları (gerçek sonuçları) arasındaki yaklaşım eğrileri; eğitim, onaylama, ve test için sırasıyla şekil V.42, V.43 ve V.44' de gösterilmektedir.



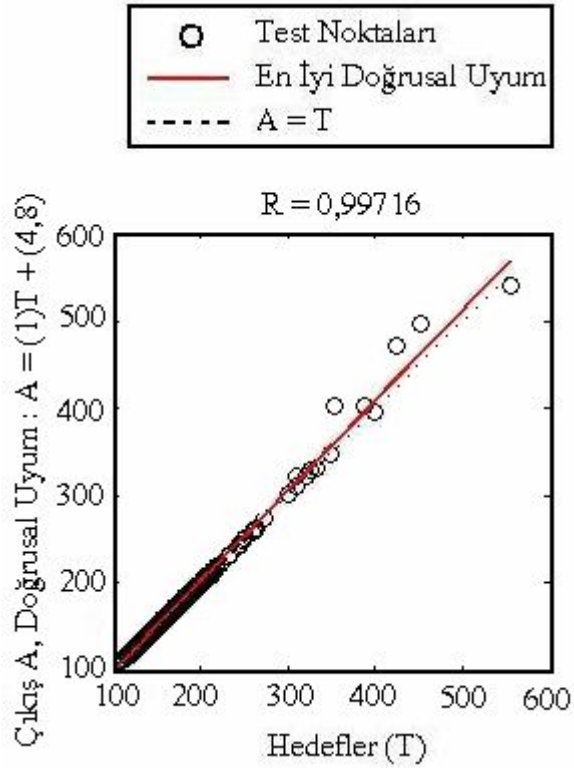
Şekil V.42 Saniyelik V THD Çıkışlı YSA Modelinin Eğitim Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi



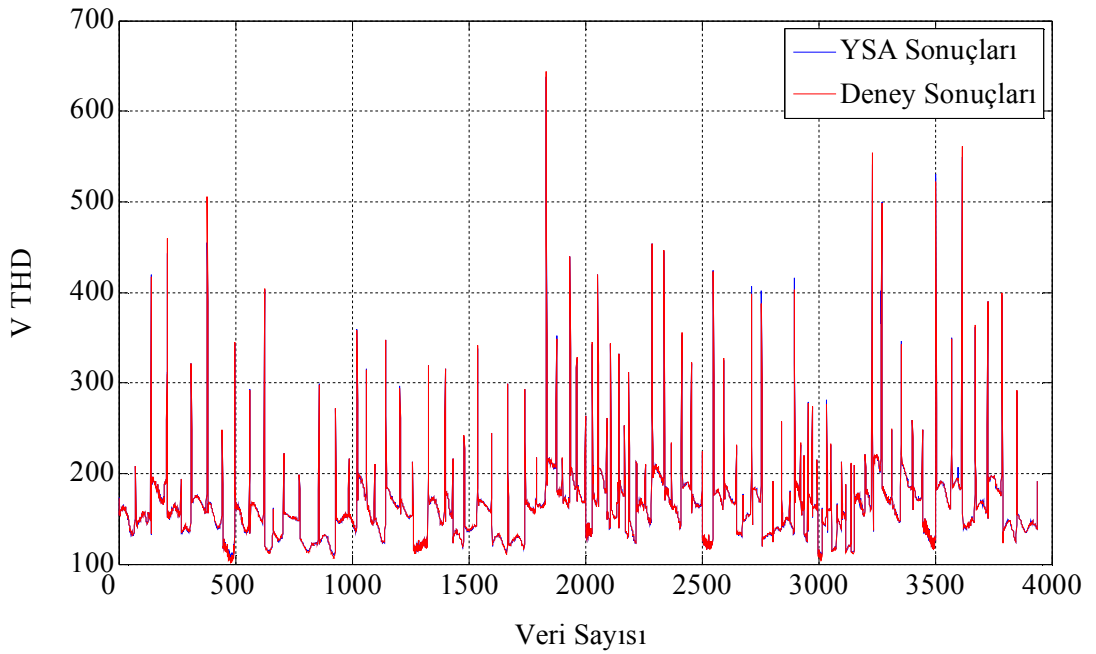
Şekil V.43 Saniyelik V THD Çıkışlı YSA Modelinin Onaylama Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi

Oluşturulan YSA modelinin çıktıları ile gerçek sonuçlar (Deney Sonuçları) her veri için gerilim THD çıkışına göre diğer modellerde olduğu gibi kıyaslanmıştır. Kırmızı ile gösterilen sonuçlar gerçek sonuçlardır (deney sonuçları). Mavi ile gösterilenler ise oluşturulan YSA modelinin ürettiği tahminleri temsil etmektedir. Diğer YSA modellerinde olduğu gibi % 100' lük bir tahminin olabilmesi için bu iki eğrinin tam olarak üst üste örtüşmesi gerekmektedir. Şekil V.45 ayrıntılı olarak incelendiğinde bu iki eğrinin hemen hemen tam olarak bir biriyle örtüştüğünü söylemek mümkündür.

Şekil V.45' deki grafikten de görülebileceği üzere kırmızının mavinin tam üstüne denk gelmesi ile mavi ile gösterilen YSA sonuçları grafikte çok az gözükmemektedir. Üretilen tahminin bu denli yüksek olabilmesinin başlıca sebebi; genişlerin sistemi çok iyi yansıtmasındandır.

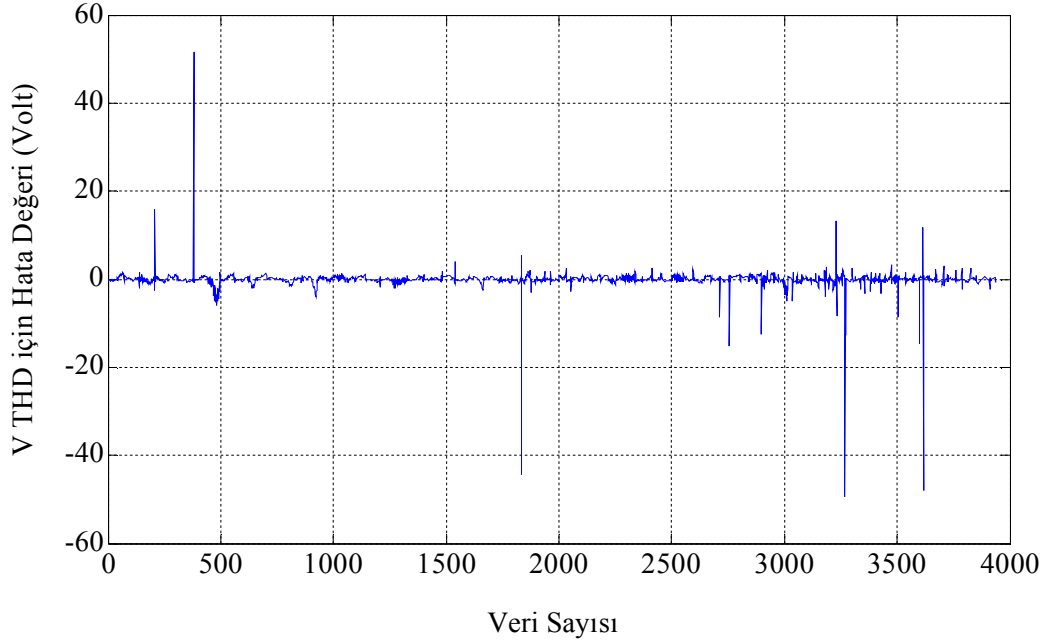


Şekil V.44 Saniyelik V THD Çıkışı YSA Modelinin Test Sonuçları ile Gerçek Sonuçlar (Hedefler) Arasındaki Yaklaşım (R) Eğrisi



Şekil V.45 YSA Sonuçları ile Gerçek Sonuçların Karşılaştırılması

Gerçek sonuç (Deneysel sonuç) ile YSA' nın sonuçlarının birbirinden çıkartılması ile elde edilen, çıkışı gerilim THD olan, bir saniye örnekleme zamanına sahip veri seti kullanılarak oluşturulmuş YSA modeli için hata eğrisi şekil V.46' da gösterilmektedir.



Şekil V.46 YSA Sonuçları ile Gerçek Sonuçlar Arasındaki Fark (Hata)

Bu modelde görüldüğü gibi veri setinin saniyelik örnekleme zamanı ile oluşturulması sonucunda fazla sayıda veri içermesinin de etkisiyle (3940 adet) ağ sonuçları daha iyi öğrenmiş ve daha 20. adımda sonuçların diğer eğitim adımlarına göre ezberlemeden daha iyi sonuçlar verdiği görülmüştür.

BÖLÜM VI

SONUÇLAR VE TARTIŞMA

Farklı stator sargı adımına sahip üç fazlı kafesli asenkron motorların sürücü sistemler ile birlikte kullanıldığında, şebekede var olan, ilk otuz harmonikten oluşan toplam harmonik distorsiyonlarının hem akım ve hemde gerilim için yapay zeka yöntemi kullanılarak saptandığı bu çalışmada, öğrenme hedeflenmiştir. Kullanılan bütün kafesli asenkron motorlarda özdeş ve aynı kesit alanına sahip emaye tel kullanılmış ve adım kısaltması yapılmasına rağmen sipir uzunluğunun değişmemesine özen gösterilmiştir. Stator sargı adımlarında bir adım kısaltılma yapılmış, özdeş gövdeye sahip beş adet, üç fazlı, 4 kutuplu, kafesli asenkron motorlar, 1 Khz ile 15 Khz arasında her PWM taşıyıcı frekansı için boş çalışmadan aşırı yüklü çalışma rejimine kadar yüklenerek iki ayrı veri seti oluşturulmuştur. Motorların yüklenmesi elektromagnetik fren ünitesi vasıtası ile gerçekleştirilmiştir. Oluşturulan veri setleri, güç analizörünün dakikada bir kaydettiği veriler ve saniyede bir kaydettiği verilerdir. Saniyede bir kaydedilen veriler de, motor iç ve dış sıcaklıkları da eş zamanlı (bir saniye örnekleme zamanı ile) olarak LABVIEW paket programının sıcaklık modülleri vasıtasıyla ölçülmüştür. Deneysel olarak oluşturulan iki farklı veri seti, MATLAB paket programının NFTOOLBOX' ı kullanılarak oluşturulan, dört adet, farklı çıkışlara sahip, yapay sinir ağlarına öğretilmiştir. Öğrenim, dakikalık (örnekleme zamanı bir dakika olan) veri seti ile 20 adet giriş parametresi ve bir adet çıkış parametresi ile yapılmıştır. Saniyelik veri seti ile sıcaklık değerleri de giriş olarak kullanılmış ve toplam 22 adet giriş ve bir adet çıkış ile öğrenim gerçekleştirilmiştir. Dakikalık veri seti ile iki adet, ve saniyelik veri seti ile de iki adet olmak üzere toplam dört adet YSA modeli oluşturulmuştur. Oluşturulan YSA modellerinden dakikalık olanların birisinin çıkış parametresi akımın ilk otuz harmonikten oluşan toplam harmonik distorsiyonu, diğeri ise gerilimin ilk otuz harmonikten oluşan toplam harmonik distorsiyonudur. Aynı

şekilde saniyelik veri seti ile oluşturulan YSA modellerinin de çıkışlarının birisi akım THD ve diğeri de gerilim THD dir.

Veri setinin deneysel olarak elde edilebilmesi, ancak tutarlı, doğru ve hassas verilere ulaşıldığında YSA modeli en doğru tahminleri vereceğinden, uzun süren çalışmalar sonucu gerçekleştirilmiştir. Bütün deneylerde motorlar aynı yöne doğru çalıştırılmış ve tam olarak soğutulduktan sonra diğerk taşıyıcı frekansına geçilmiştir. Motorların yüklenmesi esnasında oluşan titreşimler, kullanılan dinamometrenin doğru değerler ölçmesine engel olduğundan dinamometre ile ölçülen kuvvet değerleri ilk aşamada veri setinden çıkartılmıştır. Daha doğru tahminlerin üretilebilmesi veya YSA modelinin daha fazla eğitilebilmesi için sistemin tam olarak yansıtılabilmesi gerektiği düşünülmüş ve ilk çalışmalarda beş tane olan giriş parametresi daha sonra yirmiye çıkartılmıştır. Amaç güvenilir tahminler üreten YSA modelinin iskeletine ulaşmaktır. İkinci (saniyelik) veri setine fabrikadan alınan klasik motor da eklenmiş ayrıca motorların aynı noktalarından eş zamanda verilerin kaydedildiği iç ve dış sıcaklıkları da giriş parametrelerine eklenerek daha doğru tahminler saniyelik gerilim çıkışlı YSA modelinde üretilmiştir.

Sistemin modellenmesi esnasında, ağıın tasarımı için;

1. Uygun olan yapının seçilmesi,
2. Seçilen yapıya uygun giriş parametrelerinin tayin edilmesi,
3. Giriş parametrelerinin, sinir ağlarının anlayacağı veri türüne çevrilmesi, eğitim ve sınama amaçlı ayrılma oranının tayini,
4. Bu yapıya uygun öğrenme algoritmasının seçimi,
5. Uygun algoritmanın uygun parametrelerinin seçimi,
6. Giriş, çıkış ve gizli katman sayılarının tespiti,
7. Bu katmanlarda kullanılacak yapay sinir hücresi sayısının belirlenmesi,
8. Seçilen nöronlarda kullanılacak aktivasyon fonksiyonunun belirlenmesi,
9. Parametreleri etkileyen ağırlıkların tespiti,
10. Uygun eğitim adım sayısının tespiti,
11. Eğitim, sınama ve onaylama setlerinde kullanılacak normalizasyon yönteminin tespiti,

sistemin modellenmesi aşamasında, dikkat edilen noktalar olmuştur. En iyi sonucu alabilmek ve yapılan hatanın minimuma indirilebilmesi için öğretim yöntemi ve tekniği, deneme yanılma metodu kullanılarak tespit edilmiştir.

Aynı sistemi yansıtan dört farklı veri seti ile eğitilmiş farklı çıkışlara sahip yapay sinir ağı modelleri arasında karşılaştırma yapabilmek için; kullanılan algoritmalar, gizli katmanlarındaki hücre sayısı ve aktivasyon fonksiyonları aynı alınmıştır. Algoritma olarak geri yayılım algoritması olan Levenberg-Marquardt algoritması kullanılmıştır. Gizli katmanındaki yapay sinir hücre sayısı yirmi olarak seçilmiş ve aktivasyon fonksiyonu olarak da tanjant hiperbolik fonksiyonun daha iyi sonuçlar verdiği tespit edilmiştir.

Sonuç olarak :

- İlk aşamadaki toplam 663 adet, bir dakika örnekleme zamanı ile oluşturulan deneysel verinin, 465 adedi ağ modellerinin eğitiminde kullanılmıştır. Eğitim aşamasında giriş ve çıkış parametreleri ağ modellerine maksimum-minimum normalizasyon yöntemi kullanılarak $[-1,1]$ aralığında öğretilmiştir. Geriye kalan 198 adet deneysel verinin yarısı (99 adet) sınama amaçlı diğer yarısı da YSA modelinin verileri ezberleyip ezberlemediğini ölçmek için eğitilebildiğini doğrulamak yada onaylamak amaçlı kullanılmış olup sadece giriş parametreleri ağ modellerine öğretilip çıkış parametresinin yapay sinir ağı tarafından hesaplanması istenmiştir.
- İkinci aşama olarak toplam 3940 adet, bir saniye örnekleme zamanı ile oluşturulan deneysel verinin 2758 adedi ağ modellerinin eğitiminde kullanılmıştır. Eğitim aşamasında giriş ve çıkış parametreleri $[-1,1]$ aralığında ağ modellerine öğretilmiştir. Geriye kalan 1182 adet deneysel verinin yarısı (591 adet) sınama amaçlı diğer yarısı da YSA modelinin eğitilebildiğini onaylayabilmek amaçlı kullanılmıştır. Aynı şekilde sadece giriş parametreleri ağ modellerine öğretilip çıkış parametresinin yapay sinir ağı tarafından hesaplanması istenmiştir.

Eğitim sonucunda; dakikalık veri seti ile oluşturulan akım THD çıkışlı YSA modeli 465 adet veri ile % 98,4386 oranında eğitilebilmiştir. Sınama için kullanılan 99 veri ile % 97,8064 oranında bir tahmin yapmıştır. Yapılan tahminin karesel ortalama hata değeri ya da YSA' nın test performansı 0,00416241 dir. MSE değerinin sıfıra yakın olması istenen bir durumdur. Akım çıkışlı modelin onyedinci eğitim adımında en iyi performansına ulaştığı görülmüştür.

Dakikalık veri seti ile oluşturulan gerilim THD çıkışlı YSA modeli 465 adet veri ile % 96,3467 oranında eğitilebilmiştir. Sınama için kullanılan 99 veri ile % 96,1758 oranında bir tahmin yapmıştır. Ağın test performansı 0,0149844 dir. Dakikalık veri seti ile eğitilen iki YSA modeli arasında çok küçük bir fark vardır. Gerilim çıkışlı model ondokuzuncu eğitim adımında en iyi performansı sergilediği görülmüştür. Dakikalık YSA modellerinden akım çıkışlı modelin sistemi gerilim çıkışlı modele göre daha iyi öğrendiği ve yansıttığı bir gerçektir. Fakat iki YSA modeli arasında çok ciddi bir fark olmayıp her ikiside güvenilir saptamalar yapmıştır.

Örnekleme zamanı bir saniye olan (saniyelik) veri seti ile eğitilmiş olan YSA modelleri 3940 adet verinin % 70 'i olan 2758 adet veri ile eğitilmiştir. Saniyelik iki adet YSA modelinden akım çıkışlı olanı % 97,882 oranında eğitilebilmiştir. Sınama (test) için kullanılan 591 adet veri ile % 96,924 oranında tahmin gerçekleşmiştir. Bu ağın test performansı 0,0048809 olup sıfıra oldukça yakın ve güvenilirdir.

Saniyelik veri seti ile eğitilmiş olan gerilim çıkışlı diğer YSA modeli, akım çıkışlı ile benzer şekilde 2758 adet veri ile eğitilmiş olup % 99,977 oranında öğrenim gerçekleştirmiştir. Test için kullanılan 591 adet veri ile % 99,716 oranında tahmin gerçekleştirilmiştir. Bu ağın test performansı 0,00017334 olup son derece güvenilirdir.

Saniyelik veri seti ile eğitilen ve tahmin yaptırılan YSA modellerinin daha iyi eğitilebildiği görülmüştür. Bununla birlikte dakikalık veri setinde yirminci adıma yaklaşırken iyi sonuca varılırken saniyelik veri setinde yetmişinci eğitim adımında doğru sonuca ulaşılmıştır. Saniyelik veri setindeki performansın daha yüksek ve daha iyi tahmin yapılabilmesi veri setinin çok daha fazla verilerden oluşmuş olması olabilir. Bir saniye örnekleme zamanına sahip bir veri setinin sistemi daha iyi temsil etmesi beklenen bir durumdur.

Bu çalışmada ; kafesli asenkron motorlarda ilk otuz harmonikten oluşan toplam harmonik distorsiyonunun, bir yapay zeka yöntemi olan yapay sinir ağı modeli kullanılarak, herhangi bir matematiksel modele ihtiyaç duymadan saptanması gerçekleştirilmiştir. YSA modelinin giriş değişkeni olarak stator sargı adımı, AC sürücünün PWM taşıyıcı frekansı, faz gerilimleri, faz akımları, faz güçleri seçilmiştir. Oluşturulan YSA modeli tamamen deneysel veriler ile eğitilmiş olup saniyelik veri seti ile % 99,716 oranında son derece güvenilir tahminler gerçekleştirilmiştir.

Gerçekleştirilen bu çalışma sadece THD tahmini yapmakta değil özel harmonik değerlerinde tahmininin gerçekleştirilebilmesi için bir zemin oluşturmaktadır. Ayrıca kullanılan asenkron motorlar daha da genişletilerek giriş değişkenleri daha da artırılarak çok daha büyük çapta bir tahmin programı oluşturabilmek bu çalışmadan sonra mümkün olabilir. Ayrıca bu çalışma aktif filtre tasarımı, motor tasarımı, sargı tasarımı konularında kullanılabilecek özgün bir kaynak teşkil etmektedir.

KAYNAKLAR

- [1] ÖZTEMEL, E. : “*Yapay Sinir Ağları*”, Papatya Yayıncılık, İstanbul, (2003).
- [2] ELMAS, Ç. : “*Yapay Sinir Ağları*”, Seçkin Yayıncılık, Ankara, (2003).
- [3] NABİYEYEV, V.V. : “*Yapay Zeka*”, Seçkin Yayıncılık, Ankara (2003).
- [4] HAGAN, T. M. ; DEMUTH, B. ; BEALE, H.: “*Neural Network Design*” *University of Colorado Bookstore*, USA, (1996).
- [5] EFE, M., KAYNAK, O. : “*Yapay Sinir Ağları ve Uygulamaları*”, Boğaziçi Üniversitesi, (2000).
- [6] FREEMAN, A. J. ; SKAPURA, M. D. : “*Neural Networks: Algorithms, Applications, and Programming Techniques*”, *Addison – Wesley Publishing Company*, California USA, (1991).
- [7] HAM, M., F. ; KOSTANIC, I. : “*Principles of Neurocomputing for Science and Engineering*”, *Mcgraw – Hill Companies*, New York, USA, (2001).
- [8] MADAN, M. ; JIN L. ; G.,HOMMA N. : “*Static and Dynamic Neural Networks : From Fundamentals to Advanced Theory*” , *IEEE Press, Wiley – Interscience*, USA, (2003).
- [9] BISHOP, M. C. : “*Neural Networks for Pattern Recognition*”, *Oxford University Press*, USA, (2003).
- [10] DEMUTH, H.; BEALE, M. : “*Neural Network Toolbox For Use With Matlab*”, *Mathworks Inc. , USA*, (2000)
- [11] VEELNTURF, J. P. L. : “*Analysis and Applications Artificial Neural Network*”, *Prentice Hall International Ltd.*, USA, (1995).
- [12] FAUSETT, L. : “*Fundamentals of Neural Networks : Architectures, Algorithms and Applications*”, USA, (1995)
- [13] BOX, G. E. P. ; JENKINS, G. : “*Time Series Analysis, Forecasting and Control*”, *Golden – Day*, San Francisco, CA, (1970).
- [14] HESAPÇIOĞLU, M. : “*Öğretim İlke ve Yöntemleri*”, *Beta Basım Yayım Dağıtım A.Ş.*, İstanbul, (1998).
- [15] SELÇUK, Z. : “*Gelişim ve Öğrenme*”, *Nobel Yayın Dağıtım Ltd. Şti.*, Ankara, (2000).

- [16] LEE, Y. C. ; LEE, J. W. ; WANG, Y. N. ; GU, C. J. : “ Effect of Voltage Harmonics on the Electrical and Mechanical Performance of a Three – Phase Induction Motor” *Industrial and Commercial Power Systems Technical Conference*, Atlanta, Canada, IEEE, **(1998)** 88-94.
- [17] CHADHRY, S. R. ; AHMED – ZAÏD, S. ; DEMERDASH, A. : “An Artificial – neural Network Method for the Identification of Saturated Turbo Generator Parameters Based on a Coupled Finite Element/State Space Computational Algorithm”, *Transactions on Energy Conversion*, IEEE Volume 10, **(1995)** 625 – 633.
- [18] ABDESLAM, O. D. ; WIRA, P. ; FLIELLER, D. ; MERCKLE, J. : “ Power Harmonic Indetification and Compensation with an Artivicial Neural Network Method”, *IEEE International Symposium on Industrial Electronics*, Volume 3, **(2006)** 1732 – 1737.
- [19] KONISHI, M. ; TORIGOE, T. ; NISHI, T. ; IMAI, J. : “Application of Neural Network to Fault Diagnosis of Electro – Mechanical System” *Memoirs of the Faculty of Engineering Okayama University*, Vol. 39, , **(2005)** 21 -27.
- [20] VIHAROS, J. ; MONOSTORI, L. ; VINCZE, T. : “ Training and Application of Artificial Neural Networks with Incomplete Data”, *Lecture Notes of Artificial Intelligence, The Fifteenth International Conference on Industrial & Engineering Application of Artificial Inteeligence & Expert Systems, Springer Computer Science, Springer – Verlag, Heidelberg, Australia, (2002)* 649 – 659.
- [21] DANSON, M. F. ; BOWYER, P. : “Estimating Live Fuel Moisture Content From Remotely Sensed Reflectance” , *Remote Sensing of Environment*, Volume 92, **(2004)** 309 – 321.
- [22] LU, W. ; KEYHANI, A. ; FARDOUN, A. : “ Neural Network Based Modeling and Parameter Indentification of Switched Reluctance Motors”, *IEEE Transactions on Energy Conversion*, Vol. 18, **(2003)**.
- [23] HEGAZY, G. Y. ; FOUUDA, S. S. ; SALAMA, A. M. M.; CHIKANI, Y. A. : “ The Effect of Modelling on the Accuracy of the Estimation of Harmonic Voltages in Distribution Systems”, *Conference Proceedings, Canadian Conference on Electrical and Computer Engineering*, vol.1, Canada, **(1994)** 131 – 135.

- [24] BOSE, K. B. : “*Modern Power Electronics and AC Drives*”, Printice Hall PTR, Upper Saddle River, NJ 07458, USA (2002).
- [25] YUSOF, Y. ; YATIM, M. H. A. : “ Simulation and Modeling of Stator Flux Estimator For Induction Motor Using Artificial Neural Network Techniques”, *National Power and Energy Conference*, Bangi, Malaysia, (2003).
- [26] MORENO, J. M. ; PERACAUOLA, J.; ESQUIVEL, A. : “ Neural Network Based Approach for The Computation of Harmonic Power in a Real – Time Microprocessor Based Vector Control For an Induction Motor Drive, ISIE 2000, *International Symposium on Industrial Electronics*, Volume 1, (2000) ,277 – 282.
- [27] ERTAN, B. H. ; ÜÇTUĞ, Y. M. : “*Modern Electrical Drives*” Springer - Verlag, New York, USA, (2000).
- [28] ABDI, H.; VALENTIN, D.; EDELMAN, B. : “*Neural Networks Series : Quantitative Applications in the Social Sciences*”, Sage Publications, London, (1999).
- [29] RINGROSE, J. M.; NEGNEVITSKY, M. :“Harmonic Source Monitoring in Power Systems Using State Estimation and Neural Networks”, *Elsevier, Australia*, (1994).
- [30] DE WILDE, P.: “*Neural Network Models : Theory and Projects*”, Springer – Verlag, New York, USA, (1997).
- [31] SIDDIQUE, A. ; YADAVA, G.; SIN, B. : “Applications of Artificial Intelligence Techniques for Induction Machine Stator Fault Diagnostics: Review” *SDEMPED 2003 ~ Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, Allandale, GA, USA (2003).
- [32] VASANASONG, E. ; SPOONER, E. D. : “*Neural Network for Estimation of Harmonic Currents Produced by Grid – Connected PV Systems*” Elsevier, The University of New South Wales, (2001).
- [33] YURTOĞLU, H. : “*Yapay Sinir Ağları Metodolojisi ile Öngörü Modellemesi : Bazı Makro Ekonomik Değişkenler İçin Türkiye Örneği*”, Uzmanlık Tezi, Devlet Planlama Teşkilatı, Ankara, (2005).
- [34] AL-RASHIDI, H. A.; GASTLI, A.; AL-BADI, A.: “ *Optimisation of Variable Speed Induction Motor Efficiency Using Artificial Neural Networks*”, Elsevier, (2000).

- [35] MCKINNON, D. ; SEYOUM, D. ; GRANTHAM, C. : “ *Rapid Determination of Fundamental and Harmonic RMS Quantities in a Three Phase System*”, Elsevier, The University of New South Wales, **(2001)**.
- [36] WILDI, T. : “ *Electrical Machines, Drives and Power Systems*”, Speria Enterprises Ltd., New Jersey, USA, **(2002)**.
- [37] CATHEY, J. J. : “*Electric Machines Analysis and Design Applying Matlab*”, Mc Graw – Hill Companies, Singapore, **(2001)**.
- [38] ARRILLAGA, J. ; SMITH, B. C. ; WATSON, N. R. ; WOOD, A. R. : “*Power System Harmonic Analysis*”, John Wiley & Sons Ltd., England, **(2000)**.
- [39] SARIOĞLU, K. : “*Elektrik Makinalarının Temelleri : Asenkron makinalar*”, Çağlayan Kitabevi, İstanbul, **(1983)**.
- [40] ATEŞ, M. H.; PEŞİNT, M. A. :“*Elektrik Makinalarının Esasları*”, Gazi Üniversitesi Basın – Yayın Yüksekokulu Matbaası, Ankara, **(1990)**.
- [41] HEILES, .; / Çeviri: ÜNALAN, E. : “ *Elektrik Makinalarının Sargıları ve Bunların Yapılması*”, İstanbul Teknik Üniversitesi Matbaası, İstanbul, **(1977)**.
- [42] BODUROĞLU, T. : “ *Elektrik Makinaları Dersleri : Teori, Hesap ve Konstrüksiyon, Cilt II, Kısım 1, Döner Alternatif Akım Makinalarına Giriş*”, İstanbul Teknik Üniversitesi, Beta Basım Yayım Dağıtım A.Ş, İstanbul, **(1988)**.
- [43] BODUROĞLU, T. : “ *Elektrik Makinaları Dersleri : Teori, Hesap ve Konstrüksiyon, Cilt II, Kısım 3, Asenkron Makinaların Hesap ve Konstrüksiyonu*”, İstanbul Teknik Üniversitesi, Beta Basım Yayım Dağıtım A.Ş, İstanbul, **(1994)**.
- [44] İKİZLİ, O. : “*Elektrik Makinalarında Dağılma, Kayıplar ve Isınma’ nın Hesapları*”, İstanbul Teknik Üniversitesi, Şirketi Mürettibiye Basımevi, İstanbul, **(1962)**.
- [45] BİRİR, Y. ; NOGAY, H. S. : “Harmonic Variations in Three-phase Induction Motors Fed by PWM Inverter with Different Stator Coil Pitches”, *Proceedings of the 6th WSEAS Int. Conf. on Applications of Electrical Engineering*, (AEE’07), İstanbul, May., 27 -29, **(2007)**.
- [46] BİRİR, Y. ; NOGAY, H. S. : “ Design and Implementation of PLC-Based Monitoring Control System for Three-Phase Induction Motors Fed by PWM

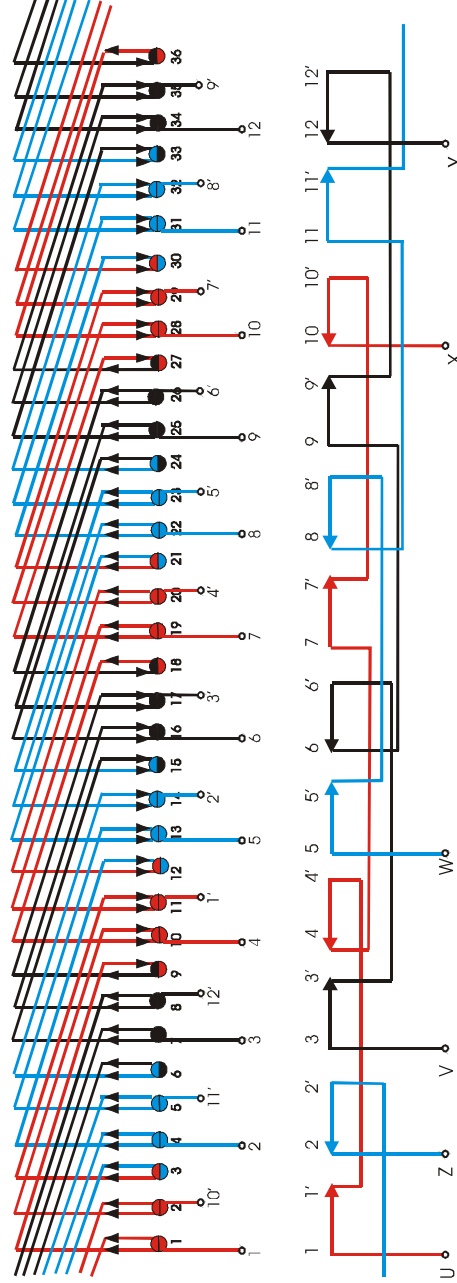
- Inverter”, *Proceedings of the 6th WSEAS Int. Conf. on Applications of Electrical Engineering*, (AEE’07), İstanbul, May., 27 -29, **(2007)**.
- [47] BİRİR, Y. ; NOGAY, H. S., TOPUZ, V. : “Estimation of Total Harmonic Distortion in Short Chorded Induction Motors Using Artificial Neural Network”, *Proceedings of the 6th WSEAS Int. Conf. on Applications of Electrical Engineering*, (AEE’07), İstanbul, May., 27 - 29, **(2007)**.
- [48] BİRİR, Y. ; NOGAY, H. S.; TASKIN, S. : “Prediction of Current Harmonics in Induction Motors with Artificial Neural Network”, *International Aegean Conference on Electrical Machines and Power Electronics*, (ACEMP’07), Electromotion’ 07 Joint Conference of IEEE, Bodrum, Eyl., 10 - 12, **(2007)**.
- [49] SARIOĞLU, K. : “*Elektrik Makinalarının Temelleri : Enerji Dönüşümü, Makina Modelleri*”, Matbaa Teknisyenleri Basımevi, İstanbul, **(1975)**.
- [50] BODUROĞLU, T. : “ *Elektrik Makinaları Dersleri : Teori, Hesap ve Konstrüksiyon, Cilt II, Kısım 2, Asenkron Makinalar* ”, İstanbul Teknik Üniversitesi, Beta Basım Yayım Dağıtım A.Ş, İstanbul, **(1988)**.
- [51] MERGEN, F. ; ZORLU, S. :“ *Elektrik Makinaları II : Asenkron Makinalar*”, Birsen Yayınevi, İstanbul, **(2005)**.
- [52] GÜZELBEYOĞLU, N. : “ *Elektrik Makinaları I, II* ”, İ.T.Ü Elektrik – Elektronik Fakültesi Ofset Baskı Atölyesi, İstanbul, **(1998)**.
- [53] SARIOĞLU, K. M.; GÖKAŞAN, M.; BOĞOSYAN, S. :“ *Asenkron Makinalar ve Kontrolü*”, Birsen Yayınevi, İstanbul, **(2003)**.
- [54] MOHAN, N. ; UNDELAND, M. T.; ROBBINS, P. W. ; / Çeviri: TUNCAY, N., GÖKAŞAN. M.; BOĞOSYAN, S. : “ Güç Elektroniği : Çeviriciler, Uygulamalar ve Tasarım”, Literatür: Yayıncılık, Dağıtım Paz. Şti., İstanbul, **(2003)**.
- [55] MERGEN, F. ; KOCABAŞ, D. A.: “ Elektrik Makinalarında Sargılar”, Birsen Yayınevi, İstanbul, **(2005)**.
- [56] BOSE, K. B. : “ *Power Electronics and Motor Drives : Advances and Trends*”, Elsevier Inc., USA, **(2006)**.
- [57] ÖZCAN, M.; ÖZKAN, O. A. : “ *Otomasyon Sistemlerinde PLC ve Uygulamaları*”, Nobel Basımevi, Ankara, **(2004)**.
- [58] PARR, E. A.: “*Programmable Controllers an Engineers Guide*”, Elsevier Inc., Britain, **(2003)**.

- [59] HACKWORTH, R. J.; HACKWORTH, D. F.: “*Programmable Logic Controllers: Programming Methods and Applications*”, Prentice Hall Health, (2003).
- [60] BEYAZIT, H.: “*Uygulamalı PLC Programlama ve Operatör Panel Konfigürasyonu*”, Nobel Yayın Dağıtım, Bursa, (2005).
- [61] BAYINDIR, S. N.: “*Elektronik Filtre Devreleri Tasarımı*”, D.E.Ü. Mühendislik Fakültesi Basım Ünitesi, İzmir, (1994).
- [62] NOGAY, H. S.; BIRBIR, Y.: “*Designation of Harmonic Estimation ANN Model Using Experimental Data Obtained From Different Produced Induction Motors*” 9th WSEAS Int. Conf. on Neural Networks (NN' 08), Sofia, Bulgaria, May., 2 - 4, (2008).
- [63] BIRBIR, Y. ; NOGAY, H. S., OZEL, Y. : “*Estimation of Low Order Odd Current Harmonics in Short Chorded Induction Motors Using Artificial Neural Network*”, 9th WSEAS Int. Conf. on Neural Networks (NN' 08), Sofia, Bulgaria, May., 2 - 4, (2008).

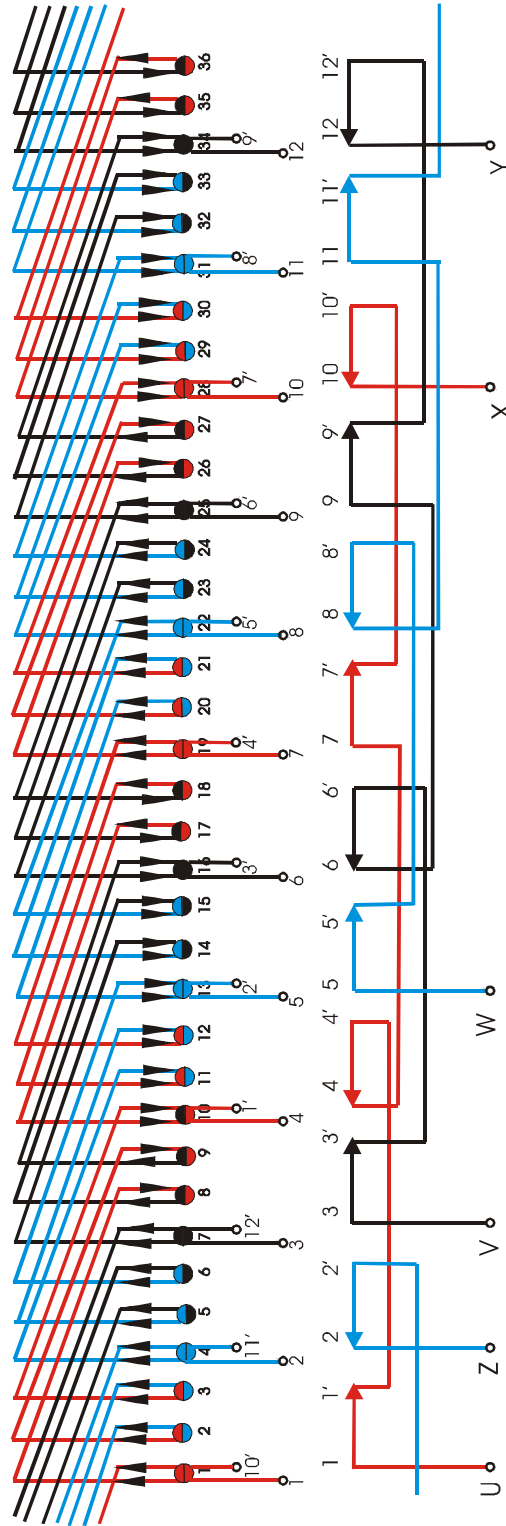
EKLER

Üç fazlı kafesli asenkron motorların stator sargıları.

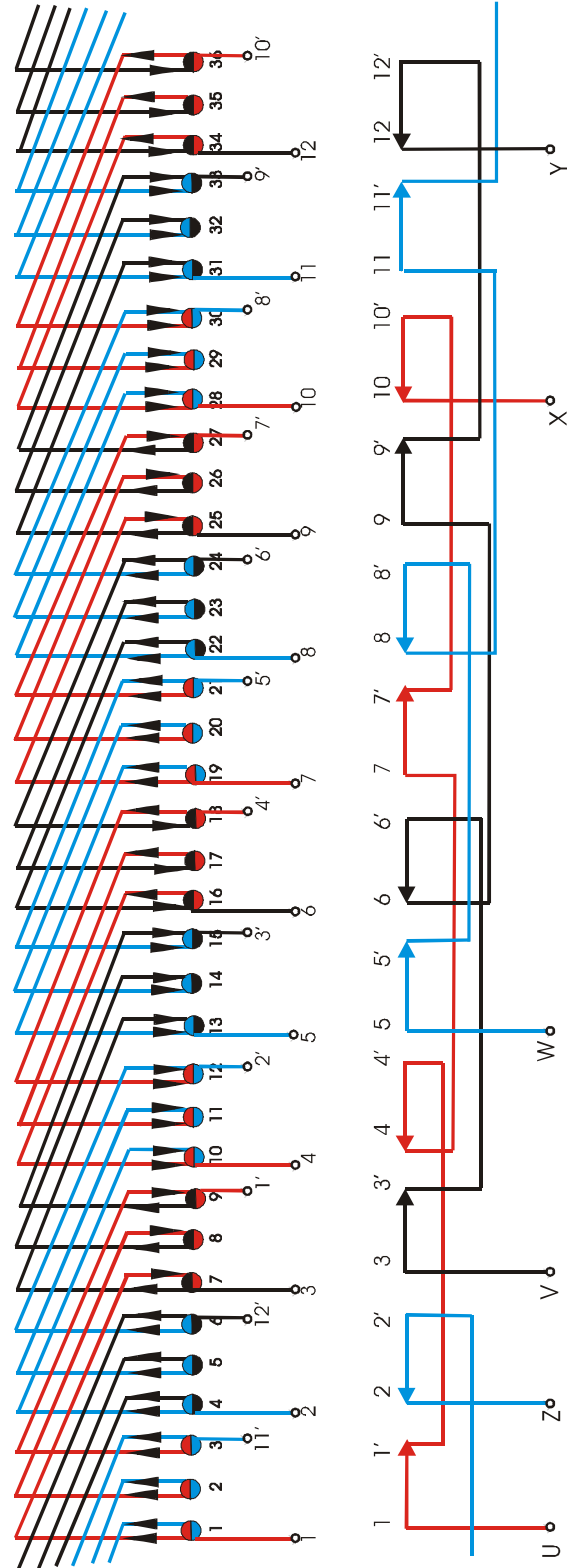
(1-9) Sargı adimli motorun sarım şeması



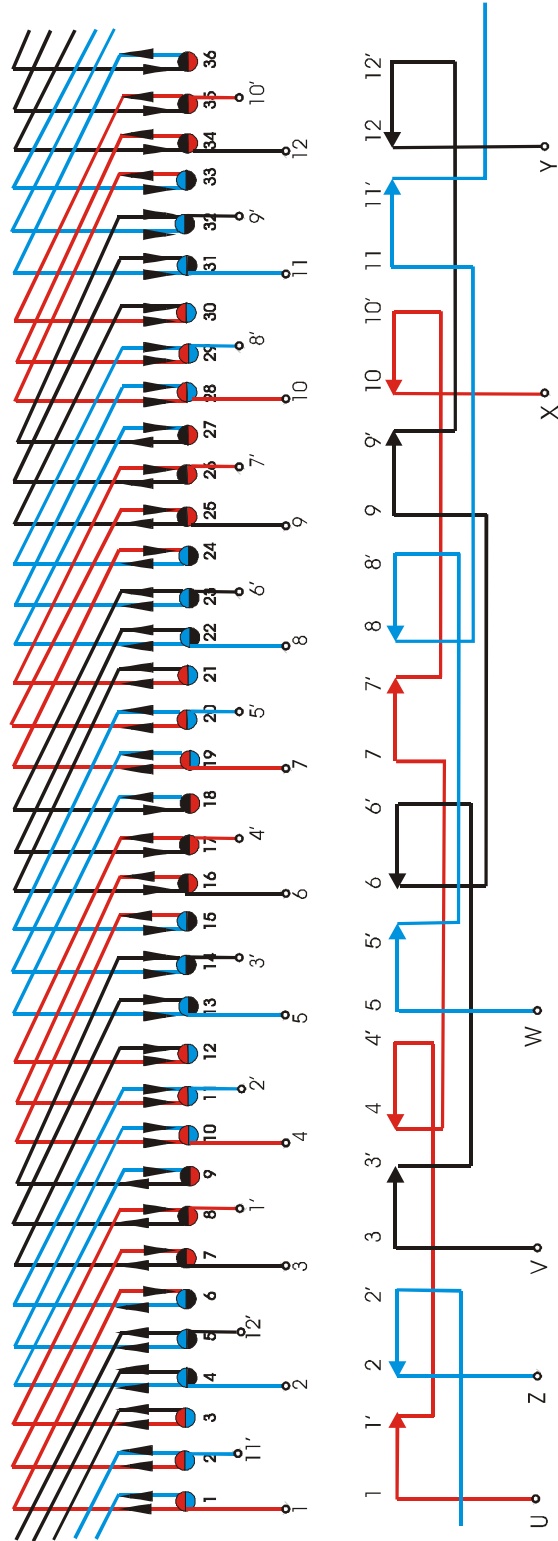
(1-8) Sargı adımlı motorun sarım şeması



(1-7) Sargı adımlı motorun sarım şeması



(1-6) Sargı adimli motorun sarım şeması



EK – B

MatLab paket programında NFTOOLBOX ile oluşturulan akım THD çıkışlı dakikalık YSA modeli için kaydedilen kodlamalar (M – file);

```
function [net,ps,ts] = fitwithnet(p,t)
%FITWITHNET Creates and trains a neural network to fit input/target data.
% [NET,PS,TS] = FITWITHNET(P,T) takes:
% P - RxQ matrix of Q R-element input samples
% T - SxQ matrix of Q S-element associated target samples
% arranged as columns, and returns these results:
% NET - The trained neural network
% PS - Settings for preprocessing network inputs with MAPMINMAX.
% TS - Settings for postprocessing network outputs with MAPMINMAX.
% For example, to create an network with this function:
% load housing
% [net,ps,ts] = fitwithnet(p,t);
% To test the network on the original or new data:
% pn = mapminmax('apply',p,ps); % Preprocess inputs
% an = sim(net,pn); % Apply network
% a = mapminmax('reverse',an,ts); % Postprocess outputs
% e = t - a; % Compare targets and outputs
% To reproduce the results you obtained in NFTOOL:
% [net,ps,ts] = fitwithnet(input',output');
% Random Seed for Reproducing NFTool results
rand('seed',2.058311164E9)
% Normalize Inputs and Targets
[normInput,ps] = mapminmax(p);
[normTarget,ts] = mapminmax(t);
% Create Network
numInputs = size(p,1);
numHiddenNeurons = 20; % Adjust as desired
numOutputs = size(t,1);
net = newff(minmax(normInput),[numHiddenNeurons,numOutputs]);
% Divide up Samples
testPercent = 0.15; % Adjust as desired
validatePercent = 0.15; % Adjust as desired
[trainSamples,validateSamples,testSamples] =
dividevec(normInput,normTarget,testPercent,validatePercent);

% Train Network
[net,tr] =
train(net,trainSamples.P,trainSamples.T,[],[],validateSamples,testSamples);
% Simulate Network
[normTrainOutput,Pf,Af,E,trainPerf] =
sim(net,trainSamples.P,[],[],trainSamples.T);
[normValidateOutput,Pf,Af,E,validatePerf] =
sim(net,validateSamples.P,[],[],validateSamples.T);
[normTestOutput,Pf,Af,E,testPerf] =
sim(net,testSamples.P,[],[],testSamples.T);
```

```

% Reverse Normalize Outputs
trainOutput = mapminmax('reverse',normTrainOutput,ts);
validateOutput = mapminmax('reverse',normValidateOutput,ts);
testOutput = mapminmax('reverse',normTestOutput,ts);
% Plot Regression
figure
postreg({trainOutput,validateOutput,testOutput}, ...
{t(:,trainSamples.indices),t(:,validateSamples.indices),t(:,testSamples.indices)});

```

YSA yapısı :

Architecture:

```

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
targetConnect: [0 1]
numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

```

Subsistemct structures:

```

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
targets: {1x2 cell} containing 1 target
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

```

Functions:

```

adaptFcn: 'trains'
gradientFcn: 'calcjx'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'trainlm'

```

Parameters:

```

adaptParam: .passes
gradientParam: (none)
initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .max_fail, .mem_reduc,

```


.min_grad, .mu, .mu_dec, .mu_inc,
.mu_max, .show, .time

Weight and bias values:

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

EK – C

MatLab paket programında NFTOOLBOX ile oluşturulan gerilim THD çıkışlı dakikalık YSA modeli için kaydedilen kodlamalar ('M – file);

```
function [net,ps,ts] = fitwithnet(p,t)
%FITWITHNET Creates and trains a neural network to fit input/target data.
% [NET,PS,TS] = FITWITHNET(P,T) takes:
% P - RxQ matrix of Q R-element input samples
% T - SxQ matrix of Q S-element associated target samples
% arranged as columns, and returns these results:
% NET - The trained neural network
% PS - Settings for preprocessing network inputs with MAPMINMAX.
% TS - Settings for postprocessing network outputs with MAPMINMAX.
% For example, to create an network with this function:
% load housing
% [net,ps,ts] = fitwithnet(p,t);
% To test the network on the original or new data:
% pn = mapminmax('apply',p,ps); % Preprocess inputs
% an = sim(net,pn); % Apply network
% a = mapminmax('reverse',an,ts); % Postprocess outputs
% e = t - a; % Compare targets and outputs
% To reproduce the results you obtained in NFTOOL:
% [net,ps,ts] = fitwithnet(input',outputv');
% Random Seed for Reproducing NFTool results
rand('seed',1.368365304E9)
% Normalize Inputs and Targets
[normInput,ps] = mapminmax(p);
[normTarget,ts] = mapminmax(t);
% Create Network
numInputs = size(p,1);
numHiddenNeurons = 20; % Adjust as desired
numOutputs = size(t,1);
net = newff(minmax(normInput),[numHiddenNeurons,numOutputs]);
% Divide up Samples
testPercent = 0.15; % Adjust as desired
validatePercent = 0.15; % Adjust as desired
[trainSamples,validateSamples,testSamples] =
dividevec(normInput,normTarget,testPercent,validatePercent);
% Train Network
[net,tr] =
train(net,trainSamples.P,trainSamples.T,[],[],validateSamples,testSamples);
% Simulate Network
[normTrainOutput,Pf,Af,E,trainPerf] =
sim(net,trainSamples.P,[],[],trainSamples.T);
[normValidateOutput,Pf,Af,E,validatePerf] =
sim(net,validateSamples.P,[],[],validateSamples.T);
[normTestOutput,Pf,Af,E,testPerf] =
sim(net,testSamples.P,[],[],testSamples.T);
% Reverse Normalize Outputs
```

```

trainOutput = mapminmax('reverse',normTrainOutput,ts);
validateOutput = mapminmax('reverse',normValidateOutput,ts);
testOutput = mapminmax('reverse',normTestOutput,ts);
% Plot Regression
figure
postreg({trainOutput,validateOutput,testOutput}, ...
{t(:,trainSamples.indices),t(:,validateSamples.indices),t(:,testSamples.indices)});

```

YSA Yapısı:

Architecture:

```

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
targetConnect: [0 1]
numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

```

Subsistemct structures:

```

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
targets: {1x2 cell} containing 1 target
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

```

Functions:

```

adaptFcn: 'trains'
gradientFcn: 'calcjx'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'trainlm'

```

Parameters:

```

adaptParam: .passes
gradientParam: (none)
initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .max_fail, .mem_reduc,

```

.min_grad, .mu, .mu_dec, .mu_inc,
.mu_max, .show, .time

Weight and bias values:

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

EK – D

MatLab paket programında NFTOOLBOX ile oluşturulan gerilim THD çıkışlı saniyelik YSA modeli için kaydedilen kodlamalar (M – file);

```
M-File
function [net,ps,ts] = fitwithnet(p,t)
%FITWITHNET Creates and trains a neural network to fit input/target data.
% [NET,PS,TS] = FITWITHNET(P,T) takes:
% P - RxQ matrix of Q R-element input samples
% T - SxQ matrix of Q S-element associated target samples
% arranged as columns, and returns these results:
% NET - The trained neural network
% PS - Settings for preprocessing network inputs with MAPMINMAX.
% TS - Settings for postprocessing network outputs with MAPMINMAX.
% For example, to create an network with this function:
% load housing
% [net,ps,ts] = fitwithnet(p,t);
% To test the network on the original or new data:
% pn = mapminmax('apply',p,ps); % Preprocess inputs
% an = sim(net,pn); % Apply network
% a = mapminmax('reverse',an,ts); % Postprocess outputs
% e = t - a; % Compare targets and outputs
% To reproduce the results you obtained in NFTOOL:
% [net,ps,ts] = fitwithnet(s0x0131caklikligirisler',ci25temmuz');
% Random Seed for Reproducing NFTool results
rand('seed',4.83392781E8)
% Normalize Inputs and Targets
[normInput,ps] = mapminmax(p);
[normTarget,ts] = mapminmax(t);
% Create Network
numInputs = size(p,1);
numHiddenNeurons = 10; % Adjust as desired
numOutputs = size(t,1);
net = newff(minmax(normInput),[numHiddenNeurons,numOutputs]);
% Divide up Samples
testPercent = 0.15; % Adjust as desired
validatePercent = 0.15; % Adjust as desired
[trainSamples,validateSamples,testSamples] =
dividevec(normInput,normTarget,testPercent,validatePercent);
% Train Network
[net,tr] =
train(net,trainSamples.P,trainSamples.T,[],[],validateSamples,testSamples);
% Simulate Network
[normTrainOutput,Pf,Af,E,trainPerf] =
sim(net,trainSamples.P,[],[],trainSamples.T);
[normValidateOutput,Pf,Af,E,validatePerf] =
sim(net,validateSamples.P,[],[],validateSamples.T);
[normTestOutput,Pf,Af,E,testPerf] =
```

```

sim(net,testSamples.P,[],[],testSamples.T);
% Reverse Normalize Outputs
trainOutput = mapminmax('reverse',normTrainOutput,ts);
validateOutput = mapminmax('reverse',normValidateOutput,ts);
testOutput = mapminmax('reverse',normTestOutput,ts);
% Plot Regression
Figure
postreg( {trainOutput,validateOutput,testOutput}, ...
{t(:,trainSamples.indices),t(:,validateSamples.indices),t(:,testSamples.indices)});

```

YSA Yapısı:

Architecture:

```

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
targetConnect: [0 1]
numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

```

Subsystem structures:

```

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
targets: {1x2 cell} containing 1 target
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

```

Functions:

```

adaptFcn: 'trains'
gradientFcn: 'calcjx'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'trainlm'

```

Parameters:

```

adaptParam: .passes
gradientParam: (none)
initParam: (none)
performParam: (none)

```

trainParam: .epochs, .goal, .max_fail, .mem_reduc,
.min_grad, .mu, .mu_dec, .mu_inc,
.mu_max, .show, .time

Weight and bias values:

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

EK – E

MatLab paket programında NFTOOLBOX ile oluşturulan gerilim THD çıkışlı saniyelik YSA modeli için kaydedilen kodlamalar (M – file);

```
M - File
function [net,ps,ts] = fitwithnet(p,t)
%FITWITHNET Creates and trains a neural network to fit input/target data.
% [NET,PS,TS] = FITWITHNET(P,T) takes:
% P - RxQ matrix of Q R-element input samples
% T - SxQ matrix of Q S-element associated target samples
% arranged as columns, and returns these results:
% NET - The trained neural network
% PS - Settings for preprocessing network inputs with MAPMINMAX.
% TS - Settings for postprocessing network outputs with MAPMINMAX.
% For example, to create an network with this function:
% load housing
% [net,ps,ts] = fitwithnet(p,t);
% To test the network on the original or new data:
% pn = mapminmax('apply',p,ps); % Preprocess inputs
% an = sim(net,pn); % Apply network
% a = mapminmax('reverse',an,ts); % Postprocess outputs
% e = t - a; % Compare targets and outputs
% To reproduce the results you obtained in NFTOOL:
% [net,ps,ts] = fitwithnet(sicaklikligirisler',cv25temmuz');
% Random Seed for Reproducing NFTool results
rand('seed',9.31316785E8)
% Normalize Inputs and Targets
[normInput,ps] = mapminmax(p);
[normTarget,ts] = mapminmax(t);
% Create Network
numInputs = size(p,1);
numHiddenNeurons = 20; % Adjust as desired
numOutputs = size(t,1);
net = newff(minmax(normInput),[numHiddenNeurons,numOutputs]);
% Divide up Samples
testPercent = 0.15; % Adjust as desired
validatePercent = 0.15; % Adjust as desired
[trainSamples,validateSamples,testSamples] =
dividevec(normInput,normTarget,testPercent,validatePercent);
% Train Network
[net,tr] =
train(net,trainSamples.P,trainSamples.T,[],[],validateSamples,testSamples);
% Simulate Network
[normTrainOutput,Pf,Af,E,trainPerf] =
sim(net,trainSamples.P,[],[],trainSamples.T);
[normValidateOutput,Pf,Af,E,validatePerf] =
sim(net,validateSamples.P,[],[],validateSamples.T);
[normTestOutput,Pf,Af,E,testPerf] =
sim(net,testSamples.P,[],[],testSamples.T);
```



```

% Reverse Normalize Outputs
trainOutput = mapminmax('reverse',normTrainOutput,ts);
validateOutput = mapminmax('reverse',normValidateOutput,ts);
testOutput = mapminmax('reverse',normTestOutput,ts);
% Plot Regression
figure
postreg({trainOutput,validateOutput,testOutput}, ...
{t(:,trainSamples.indices),t(:,validateSamples.indices),t(:,testSamples.indices)});

```

YSA Yapısı:

Architecture:

```

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
targetConnect: [0 1]
numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

```

Subsystem structures:

```

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
targets: {1x2 cell} containing 1 target
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

```

Functions:

```

adaptFcn: 'trains'
gradientFcn: 'calcjx'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'trainlm'

```

Parameters:

```

adaptParam: .passes
gradientParam: (none)
initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .max_fail, .mem_reduc,

```

.min_grad, .mu, .mu_dec, .mu_inc,
.mu_max, .show, .time

Weight and bias values:

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

ÖZGEÇMİŞ

Hıdır Selçuk Noğay, 1975 Isparta – Senirkentde doğdu. 1999 yılında Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümü’nden mezun oldu. 1999-2000 öğretim yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümünde Araştırma Görevlisi olarak çalışmaya ve M.Ü. Fen Bilimleri Enstitüsü Elektrik Eğitimi Programında Yüksek Lisans programına başladı. Yüksek lisans eğitimini 2002 yılında tamamlayarak aynı yıl M.Ü. Fen Bilimleri Enstitüsü Elektrik Eğitimi Programında Doktora eğitimine başladı. Halen aynı bölümde Araştırma Görevlisi olarak görev yapmaktadır.

İstanbul, 07.01.2008