

**T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ENDÜSTRİ 4.0 PERSPEKTİFİNDE YAPAY SİNİR AĞLARI
KONTROLLÜ REAKTİF GÜÇ KOMPAZASYONU**

Mehmet Bedri DOĞRUYOL

ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ŞANLIURFA
2020**

**T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**ENDÜSTRİ 4.0 PERSPEKTİFİNDE YAPAY SİNİR AĞLARI
KONTROLLÜ REAKTİF GÜÇ KOMPAZASYONU**

Mehmet Bedri DOĞRUYOL

ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**ŞANLIURFA
2020**

Dr. Öğr. Üyesi Nurettin BEŞLİ danışmanlığında Mehmet Bedri DOĞRUYOL'un hazırladığı “Endüstri 4.0 Perspektifinde Yapay Sinir Ağları Kontrollü Reaktif Güç Kompanzasyonu” konulu bu çalışma 14/01/2020 tarihinde aşağıdaki jüri tarafından oy birliği ile Harran Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Ana Bilim Dalı'nda YÜKSEK LİSANS tezi olarak kabul edilmiştir.

İmza

Danışman : Dr. Öğr. Üyesi Nurettin BEŞLİ

.....

Üye : Dr. Öğr. Üyesi Faruk KÜRKER

.....

Üye : Dr. Öğr. Üyesi Abdulkadir GÜMÜŞÇÜ

.....

Bu tezin Elektrik Elektronik Mühendisliği Ana Bilim Dalında Yapıldığını ve Enstitümüz Kurallarına Göre Düzenlendiğini Onaylarım.

Doç. Dr. İsmail HİLALİ
Enstitü Müdürü

Not: Bu Tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı 5486 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

Sayfa No

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
ŞEKİLLER DİZİNİ	iv
ÇİZELGELER DİZİNİ	v
SİMGELER DİZİNİ	vi
1. GİRİŞ	1
2. ÖNCEKİ ÇALIŞMALAR	3
3. MATERYAL ve YÖNTEM	7
3.1. Endüstri 4.0	7
3.1.1. Endüstrinin Tarihsel Gelişimi	8
3.1.2. Endüstri 4.0 Kavramı	9
3.2. Yapay Sinir Ağları	15
3.2.1. Yapay Sinir Ağlarına Giriş	15
3.2.2. Yapay Sinir Ağlarının Yapısı ve Temel Elemanları	17
3.2.3. Yapay Sinir Ağlarının Sınıflandırılması	23
3.2.4. Yapay Sinir Ağlarının Temel Öğrenme Kuralları	26
3.3. Reaktif Güç Kompanzasyonu	30
3.3.1. Reaktif Güç ve Güç Katsayısı	30
3.3.2. Reaktif Güç Kompanzasyonu	31
3.3.3. Reaktif Güç Tüketicileri ve Sistemin Tasarlanması	32
3.3.4. Kompanzasyon Sistemlerinin Avantajları ve Dezavantajları	34
3.3.5. Reaktif Güç Kompanzasyonunda Kapasite Hesabı	34
3.3.6. Reaktif Güç Kompanzasyon Çeşitleri	37
3.3.7. Kompanzasyon Sistem Tasarımında Kullanılan Geleneksel Yöntemler	39
3.3.8. Şönt Reaktörler	42
3.3. YSA'nın Tasarımı	45
4. ARAŞTIRMA BULGULARI ve TARTIŞMA	54
4.1. Gerçek Veriler Kullanıcı Arayüzü	54
4.2. Kompanzasyon Yokken Kullanıcı Arayüzü	55
4.3. Statik Kompanzasyon Kullanıcı Arayüzü	56
4.4. Yapay Zeka Kullanıcı Arayüzü	58
5. SONUÇLAR ve ÖNERİLER	60
KAYNAKLAR	62
ÖZGEÇMİŞ	65
EKLER	66
EK 1 Veri Okuma Kaynak Kodu	66
EK 2 Yapay Sinir Ağı Modeli Oluşturma Kaynak Kodları	67
EK 3 Arayüz Program Kaynak Kodu	68

ÖZET

Yüksek Lisans Tezi

ENDÜSTRİ 4.0 PERSPEKTİFİNDE YAPAY SİNİR AĞLARI KONTROLLÜ REAKTİF GÜÇ KOMPANZASYONU

Mehmet Bedri DOĞRUYOL

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Anabilim Dalı

Danışman: Dr. Öğretim Üyesi Nurettin BEŞLİ
YIL: 2020, Sayfa: 76

Elektrik enerjisinde verim; hizmet kalitesini ve mevcut üretimi eksiltmeden aynı işin daha az enerji kullanılarak yapılabilmesidir. Reaktif Güç Kompanzasyonu enerji tesislerinde verimi arttırmak ve enerji tasarrufunu sağlamak için en önemli yoldur. Sanayinin tarihsel süreçte geçirdiği üç büyük endüstri devriminin sonucunda günümüz teknolojisi artık insan gücüne eskisi kadar gereksinim duymayan, iletişim ağları ile veri akışlarının sağlandığı, bilgi ve otomasyonun üretim süreçlerinde bütünleşik olarak kullanıldığı dördüncü sanayi devrimindedir. Endüstri 4.0, bilgisayar yazılım ve donanımları ile uygulamalardan elde edilen veri ya da bilgilerin kullanılmasına imkân sağlamaktadır. Tasarlanacak zeki sistemlerin oluşturulması sürecinde Yapay zekâ yöntemleri çoğunlukla yer bulmaktadır.

Bu tezde, yapay sinir ağları(YSA) tabanlı reaktif güç kompanzasyonu gerçekleştirilmesiyle beraber tesiste bulunan Endüstri 4.0 uyumlu cihazlardan da alınacak verilere göre öngörülü ve daha verimli sistemler tasarlanmıştır. Kompanzasyonu YSA ile yapmak için oluşturulan ağda öğrenme algoritmaları kullanılmıştır.

ANAHTAR KELİMELER: Endüstri 4.0, yapay sinir ağları, YSA, reaktif güç kompanzasyonu.

ABSTRACT

MSc Thesis

Artificial Neural Networks Controlled Reactive Power Compensation in the Perspective of Industry4.0

Mehmet Bedri DOĞRUYOL

**Harran University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronic Engineer**

**Supervisor: Assist. Prof. Dr. Nurettin BEŞLİ
Year: 2020, Page: 76**

Efficiency in electrical energy is the ability to do the same work by using less energy without compromising the service quality and the current production. Reactive Power Compensation is the most important way to increase efficiency and save energy in energy plants. As a result of the three major industrial revolutions in the past, today's technology is in the fourth industrial revolution, which requires less manpower, provides data flows via communication networks, and integrates information and automation into production process. Industry 4.0 enables the use of data or information obtained from applications with computer software and hardware. Artificial intelligence methods are mostly involved in the creation of intelligent systems to be designed.

In this thesis, reactive power compensation based on Artificial Neural Networks (ANN) has been realized and more efficient systems has been designed according to the data obtained from Industry 4.0 compatible devices in the facility. In order to make the compensation with ANN, training algorithms were used in the network.

Keywords: Industry 4.0, artificial neural network, ANN, reactive power compensation.

TEŞEKKÜR

Bu tez çalışmamın başında beri benden desteğini esirgemeyen saygıdeğer danışman hocam Sayın Dr. Öğretim Üyesi Nurettin BEŞLİ'ye, bölüm hocalarıma, her konuda desteğini gördüğüm çocukluk arkadaşım Sayın Fahrettin AY'a teşekkür ederim.

Hayatım boyunca desteklerini hiçbir zaman esirgemeyen ve beni bu günlere getiren aileme sonsuz şükranlarımı sunarım.



ŞEKİLLER DİZİNİ

Sayfa No

Şekil 3.1. Endüstrinin Tarihsel Gelişimi Süreci	9
Şekil 3.2. Endüstri 4.0'ın Genel Yapısı	10
Şekil 3.3. Biyolojik Sinir Öbeği	18
Şekil 3.4. Sinir Sistemi Öbek Şeması	18
Şekil 3.5. Yapay Sinir Hücresi.	19
Şekil 3.6. Yapay Sinir Ağ Modeli.	22
Şekil 3.7. Tek Katmanlı Yapay Sinir Ağı.....	24
Şekil 3.8. İleri Beslemeli Yapay Sinir Ağı	25
Şekil 3.9. Geri Beslemeli Yapay Sinir Ağı.....	25
Şekil 3.10. Gözetimli Öğrenme	28
Şekil 3.11. Gözetimsiz Öğrenme.....	28
Şekil 3.12. Destekleyici Öğrenme Yapısı.	29
Şekil 3.13. Akım, Gerilim ve Güçlerin Fazör Diyagram Gösterimi.....	30
Şekil 3.14. Aktif Güç Sabitken Reaktif Güç Tespiti.	35
Şekil 3.15. Görünür Güç Sabitken Reaktif Güç Tespiti.	36
Şekil 3.16. Bireysel Kompanzasyon Tasarımı.	37
Şekil 3.17. Grup Kompanzasyon Tasarımı.....	38
Şekil 3.18. Merkezi Kompanzasyon Tasarımı.	39
Şekil 3.19. Kapasitif ve Endüktif Güçlerin Vektörel Gösterilmesi.	41
Şekil 3.20. Kondansatörlerin Üçgen ve Yıldız Bağlantı Şeması	42
Şekil 3.21. Üç Fazlı Reaktör Bağlantı Şeması	43
Şekil 3.22. Kompanzasyon Panosu Genel Hat Şeması	47
Şekil 3.23. Kompanzasyon Panosu Ön Kapağı	48
Şekil 3.24. Kompanzasyon Panosu İç Görüntüsü.	49
Şekil 3.25. İş Akış Şeması.....	50
Şekil 3.26. MATLAB NNtool Genel Görüntüsü.....	51
Şekil 3.27. Oluşturulan Yapay Sinir Ağı.....	52
Şekil 3.28. Yapay Sinir Ağı Performans Grafiği.....	52
Şekil 3.29. Yapay Sinir Ağı Modelinin Regresyon Grafikleri.	53
Şekil 4.1. Kullanıcı Arayüzü.	54
Şekil 4.2. Gerçek Veriler Kullanıcı Arayüzü.	55
Şekil 4.3. Kompanzasyon Yokken Kullanıcı Arayüzü.....	56
Şekil 4.4. Statik Kompanzasyon Kullanıcı Arayüzü	57
Şekil 4.5. Yapay Zeka Kullanıcı Arayüzü.....	58

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 3.1. Siber-Fiziksel Sistemlerin Tarihsel Gelişimi Süreci.	11
Çizelge 3.2. Toplam Kondansatör Gücünü Tahmin Etmek İçin Kullanılan Giriş-Çıkış Verileri.	45
Çizelge 3.2. (devam).	46
Çizelge 3.3. Şönt Reaktör Akımlarını Tahmin Etmek İçin Kullanılan Giriş-Çıkış Verileri.	46
Çizelge 5.1. Gerçek zamanlı sistem parametreleri ile tahmin edilen değerlerin karşılaştırılması	60



SİMGELER DİZİNİ

ω	Açısal Hız
I_a	Aktif Akım
P	Aktif Güç
$\cos$	Cosinüs Fonksiyonu
π	Çarpım Fonksiyonu
ε	Dielektrik Katsayısı
e	Euler sayısı
θ	Eşik Değeri
F	Frekans
G	Girdiler
X_i	Giriş Verileri
I	Görünür Akım
S	Görünür Güç
$\cos \varphi$	Güç Katsayısı
X_c	Kapasitif Reaktans
W_i	Kazanç Ağırlığı
C	Kondansatör Birimi
R	Korelasyon Değeri
P	Kutup Sayısı
d	Levhalar Arası Uzaklık
Ψ	Nöron ve Aktivasyon Fonksiyonu
I_r	Reaktif Akım
Q	Reaktif Güç
$\sin$	Sinüs Fonksiyonu
$\tan$	Tanjat Fonksiyonu
n	Toplam Girdi Sayısı
Σ	Toplama Fonksiyonu
U	Şebeke Gerilimi
F	Yüzey Alanı

Kısaltmalar

PIC	Mikrochip Firmasına Ait Mikrodenetleyici
MATLAB NNtool	MATLAB Yapay Sinir Ağları Araç Çubuğu
YSA	Yapay Sinir Ağları

1. GİRİŞ

Dünya nüfusu hızlı bir şekilde artıyor. Elektrik enerjisine duyulan ihtiyaç sanayileşme ve teknolojik gelişmelere bağlı olarak her gün artmaktadır. Gelişmişlik göstergesinin en önemli ölçütlerinden biri kişi başına düşen enerji miktarı olarak kabul edilmektedir. Sanayileşme artıkça teknolojinin gelişimine bağlı olarak yaşam kalitesinde artış yaşanmıştır. Bu artışa paralel olarak kullanılan elektrik enerjisinde kalite aranır olmuştur. Elektrik enerjisinde kesintinin olmaması, gerilim ve frekansın sabit olması, güç faktörünün istenen değerde olması enerji kalitesi olarak tanımlanmıştır. Uzun zamandır üzerinde çalışılan enerji kalitesi yeni kavram olarak hayatımıza girmiştir (Keçecioglu, 2012).

Elektrik enerjisi alternatif akım olarak üretilmekte olup aktif ve reaktif bileşenden oluşmaktadır. Aktif güç tüketiciler için faydalı olup aktif akımdan meydana gelmektedir. Trafo motor gibi makineler manyetik alanla çalışır. Bu makinelerin çektiği akım reaktif akımdır.

Enerji sistemlerindeki elektrik kalitesini artırmak için gerilimdeki dalgalanmaları, çökmeleri, kesintileri ve geçici olayları gidermekle beraber, en önemli unsur reaktif güç kompanzasyonu yapmaktır. Reaktif güç kompanzasyonu sayesinde enerji sistemlerinde verimi artırmak, tasarruf sağlamak mümkündür.

Reaktif güç kompanzasyonlarında çeşitli yöntemler kullanılmaktadır. Statik faz kaydırıcı olarak kademeli kondansatör grupları, dinamik faz kaydırıcı olarak senkron makineler kullanılmaktadır.

Statik kompanzasyonda, kondansatör ve şönt reaktör grupları kademeli olarak devreye alınırlar. Bu yöntemde, kademeler arasında kalan değerler arasında tam kompanzasyon yapılamayabilir ve kademelerin devreye girmesinde zaman gecikmesi oluşabilmektedir. Bundan dolayı aşırı veya düşük kompanzasyon yapılması durumunda enerji nakil hatlarında aşırı yüklenmelere sebep olur.

Dinamik kompanzasyonda senkron makine kullanılmakta olup makinenin uyartım akımı değiştirilerek motorun kapasitif ya da endüktif çalıştırılması sağlanabildiğinden dolayı senkron motorlar dinamik güç kompanzatörü olarak kullanılabilirler.

Tesis çalışırken devreye isteğe bağlı olarak alınabilen ve çıkartılabilen bir senkron motor, örneğin havalandırma fanı, dinamik kompanzasyonu daha etkin kullanmayı sağlayabilir. Endüstri 4.0'ın yaygınlaşmasıyla üretimdeki verim artışı enerji alanında da sağlanabilir. Tesisin güç talebi, güç katsayısının bilinmesiyle daha etkili kompanzasyon yapılabilir. Akıllı tesislerde, sistem seçici anahtar yardımıyla istenilmesi durumunda ya kompanzasyon devre dışı bırakılacaktır, ya statik kompanzasyon ya da dinamik kompanzasyon aktif edilecektir.

Bu tezde, yapay sinir ağları(YSA)kullanılarak reaktif güç kompanzasyonu gerçekleştirilmiştir. Tesiste bulunan Endüstri 4.0 uyumlu cihazlar yardımıyla alınacak verilere göre öngörülmesi ve daha sağlıklı sistem tasarlanması hedeflenmiştir. YSA tasarımı için gerekli olan kondansatör güçleri ve şönt reaktör akımları tahmin edilmiştir. YSA modeli MATLAB® &SIMULINK® programında tasarlanmıştır. YSA araçlarından Levenberg – Marquart (LM) modeli tercih edilmiştir. Gerçek sistemden alınan 350 adet veri toplanarak veri tabanına kaydedilmiştir. Ağın eğitilmesinde verilerin %70'i, ağın test edilmesinde verilerin %15'i ve ağın doğrulanmasında ise verilerin %15'u kullanılmıştır.

Yapılan yazılımda YSA'nın oldukça yüksek tahminler yaptığı gözlemlendiğinden YSA yapısının gerçek zamanlı sistemlerde kullanılabileceği görülmüştür. Bu bilgiler ışığında daha hassas kompanzasyon sistemlerinin tasarlanması sağlanmıştır.

2. ÖNCEKİ ÇALIŞMALAR

Kaplan (2005), kendi çalışmasında birden çok kullanıcı bir sistemde mikrodenetleyicili güç rölesi tasarlayarak reaktif gücü devamlı kontrol etmiştir. Sistem verileri PIC yardımıyla takip edilmektedir. Bu bilgiler ışığında yazılım ile reaktif güç hesaplanmış ve gerekli kondansatörler devreye tek seferde alınmıştır.

Şekelli ve Tarkan (2005), yeni röle tasarımını mikroişlemci tabanlı olarak tasarlamışlardır. Kondansatör gruplarını tek tek devreye almayarak tekli, ikili veya üçlü gruplar şeklinde almışlardır. Bunu yaparak kondansatörlerin anahtarlama sayısını ile endüktif akımı en aza indirmişler ve kayıpları azaltmışlardır.

Kocabaş (2006), bilgisayar ortamında reaktif güç kompanzasyonun gerekliliğini, matematik esaslarını, modern röleler işlevlerini gerçekleştirmiştir. Simülasyon sonuçlarına göre sistemde kullanılması gereken röle, kondansatör, akım trafosu vb. cihazların seçimini daha kolay olacağını öngörerek hatalı seçimlerin önüne geçilebileceğini düşünmüştür.

Sesveren (2008), çalışmasında dinamik kompanzasyon denetiminde yapay sinir ağları modeli tasarlamıştır. Yaptığı tasarımı eğitim amaçlı kullanabileceğini, YSA yapı ve algoritmalarının test edebileceği uygulamalarda da kullanabileceğini görmüştür.

Bayındır ve Görgün (2009), kendi çalışmalarında PIC 18F452 mikrodenetleyici kullanarak senkron motor yardımıyla kompanzatör uygulaması yapmışlardır. PI denetimli senkron motoru gerçekleştirmişlerdir. Uyartım akımlarını değiştirilerek motorun çalışma karakteristiklerini incelemişlerdir. LCD ekrandan kullanıcıya güç katsayısı, gerilim, akım, frekans bilgileri gösterilmiştir.

Görgün (2009), YSA denetimli senkron motorla reaktif güç kompanzasyon uygulamasını gerçekleştirmiştir. Yaptığı uygulamada reaktif güç kompanzasyonu

denetiminde yapay sinir ağlarının eğitim ve test verilerini kullanılmıştır. Elde ettiği ağırlık değerlerini mikrodenetleyici yardımıyla gerçek zamanlı sistemlere uygulamıştır.

Baran (2010), bilgisayar kontrollü reaktif güç kompanzasyonu için eğitim seti oluşturmuş ve kondansatör gruplarını kullanmıştır. PLC kullanarak sistemin bilgisayar ile izlenmesini sağlamıştır. PLC sayesinde güç katsayısı, akım, gerilim, reaktif güç, aktif güç gibi parametrelerin takibini yapmıştır. Hazırlanan kullanıcı arayüzü yardımıyla kullanıcı yük değişimlerini izlemekte ve devreye yük alıp çıkarabilmektedir.

Hatay (2010), ev tipi taşınabilir kompanzasyon sistemini bulanık mantık kontrollü tasarlamıştır. Tasarlanan sistemde akım ile gerilim arasındaki faz farkını tespit etmiş, çekilen reaktif güç miktarına göre çıkışları bulanıklaştırmıştır. Uygulama sonucunda tasarlanan sistemin güç katsayısını optimum seviyeye getirerek tasarruf sağlandığını, hataların en aza indirgendiğini ve maliyeti düşürdüğünü tespit etmiştir.

Baş (2014), çalışmasında MATLAB® & SIMULINK® programında endüstriyel bir tesiste nonlinear yüklerden dolayı oluşan harmonik bozulmaları pasif filtre yardımıyla harmoniklerden arındırmak için uygulama gerçekleştirmiştir. İlk önce sistemin reaktif güç ihtiyacını ve harmonik özelliklerini incelemiştir. Sonrasında tek ayarlı filtre için lazım olan bağıntıları vermiştir. Belirlenen harmonikler için pasif filtre tasarlamıştır. Her seferinde tasarlanan filtrelerin harmonik bozulmaları ve sistemin güç katsayısına olan tepkileri incelemiştir. Tasarladığı filtre baskın harmoniklere uygulandığı zaman diğer harmoniklerin genliklerinin azaldığını görmüştür.

Vadi (2014), sistemde bulunan reaktif gücü, web arayüzü ile internet üstünden kontrol edilebilen senkron motorla yapmıştır. Bu işlemi PIC, PLC gibi denetleyicilerden üstün olan veri toplama kartıyla yapmıştır. Çalışmasında yüklerin şebekeden çektikleri akım, gerilim ve güç katsayısı web arayüzü üzerinden gerçek

zamanlı olarak takip edilmiştir. Tasarladığı sistem sayesinde sık kullanılmayan internet tabanlı bilgisayar kontrollü ve izlemeli reaktif güç kompanzasyonu için yeni bir yaklaşım sunmuştur.

Aydın (2015), çalışmasında farklı yük koşulları için 3 fazlı sistemde hem kondansatör hem de senkron motor ile kompanzasyon yaparak sonuçları analiz etmiştir. Kompanzasyon uygulamaları için hem manuel hem de otomatik kumanda şeklinde tekrarlamıştır. Manuel kompanzasyonu gerekli veri tabanı için kullanmıştır. Kondansatör ile yaptığı otomatik kumanda kompanzasyonda, belirli aralıklarda kondansatör veya kondansatör gruplarını anahtarlamıştır. Senkron motor ile yaptığı kompanzasyonu ise YSA denetimli gerçekleştirmiştir. U, I, Cos ϕ , P, F gibi sistem parametrelerini tasarladığı devreden tek faz olarak ölçmüştür. Yaptığı arayüz yardımıyla bu parametreleri bilgisayardan izleyerek sistemi kontrol etmiştir.

Gökcan (2016), çalışmasında reaktif güç kompanzasyonu için MATLAB tabanlı arayüz(GUI) kullanarak sanal bir laboratuvar oluşturmuştur. Bunu yapmasındaki amaç kompanzasyon ile ilgili sistemlerinin öğrenilip test edilmesi, yazılımlarının yapılması ve uygulanması hem zaman alıcı hem de asıl hedeften uzaklaşılmasıdır. Bu amaçla MATLAB GUI ile hazırlanan arayüz sayesinde kullanıcıların çok iyi matematik, güç elektroniği ve programlama bilmeden kompanzasyon eğitimi sağlamıştır.

Doğrul (2018), PLC yardımıyla, harmonikleri ihmal ederek reaktif güç kompanzasyonun yapmak için tristör kontrollü reaktör ve sabit kapasite kullanmıştır. Çalışmasında PID, bulanık mantıklı PID ve ANFIS kontrollü reaktif güç kompanzasyonu gerçekleştirmiştir. Uyguladığı yöntemler arasında kıyaslama yapmıştır. Sistemde PID performansını etkileyen aşmayı önemli ölçüde azaltmıştır. Aynı zamanda yamuk ve üçgen üyeli BPID'in yerleşme zamanlarını azaltmıştır.

Şerbetçi (2019), çalışmasında harmonik filtreli kompanzasyon sistemlerini ele almıştır. Harmonik problemi olan bir çelik tel üretim fabrikasında ölçümler yaparak filtre tasarlamıştır. Harmonik filtreli kompanzasyon sistemi sayesinde tesisteki

harmonikler yönetmelikte belirlenen limitlerin altına düşürülmüştür. Harmoniklerin yüksek olduğu tesislerde reaktif güç kondansatörlerinin şebekeden çektiği akım harmoniklerini azaltmada etkili olduğunu tespit etmiştir. Aynı zamanda MATLAB® & SIMULINK® programında filtre yönteminin simülasyonunu yapmıştır.



3. MATERYAL ve YÖNTEM

Bu bölümün ilk kısmında Endüstri 4.0 tanıtılmış, endüstrinin tarihsel süreci ve yapısı, Endüstri 4.0'ın amacı, avantajları ve karşılaşılan zorluklar hakkında bilgi verilmiştir.

İkinci kısımda yapay sinir ağlarına ait genel tanımlar, ağların tarihçesi, genel yapısı ve elemanları tanıtılmıştır. Ayrıca yapay sinir ağları sınıflandırılarak temel öğrenme kuralları anlatılmıştır.

Üçüncü kısımda temel güç kavramları, reaktif güç kompanzasyonu, kompanzasyonun faydaları, avantajları ile dezavantajları, sistem tasarımları, kompanzasyon için gerekli olan hesaplamalar, kompanzasyon çeşitleri ve şönt reaktörler hakkında bilgi verilmiştir.

Dördüncü kısımda tasarımı yapılan Yapay Sinir Ağları tabanlı Kompanzasyon uygulaması tanıtılmıştır. Veri toplama sistemi ile gerekli parametrelere ait değerler toplanmıştır. Gerçek sistem parametreleri yardımıyla MATLAB® & SIMULINK® programında yapay sinir ağları programlanarak sonuçlar bulunulmuştur.

3.1. Endüstri 4.0

Sanayi denilince, bir yerde, bölge ya da ülkede üretim yapabilecek kuruluşların tüzel olarak oluşturulan işletmeleri akla gelir (Walsh, Möhring, Koot, Schaarschmidt, 2014). Çimento sanayi, demir – çelik sanayi vb. sanayiye örnek verilebilir.

Yapılan bu sınıflandırmaya göre genellikle, üç sektör teorisi çerçevesinde olmaktadır. Bunlar: birincil ya da özütleyici, ikincil ya da imalat, üçüncül ya da hizmet sektörleri olmaktadır (De Treville, Shapiro, Hameri, 2004). Bazı bilim

adamlarına göre de dördüncü (bilgi) ya da beşinci (kültür ve araştırma) sektörleri bulunmaktadır. Zaman içerisinde uygulanan sektörel ayrışmalar bulunduğu toplum ve ekonomilerine göre değişim göstermekte ve sanayi türlerinin sınıflandırılmalarında ürettikleri ürünler önem kazanmaktadır (Wigand, Picot, Reichwald, 1997).

Birçok şekilde sanayi sınıflandırılması yapılabilir.

Birincil sanayi sınıflandırılmasına, genel olarak doğa temelli yapılan üretimler gösterilebilir. Burada doğa temelli üretim olmasından dolayı insan gücüne az gereksinim duyulmaktadır. Örnek verecek olursak; tarım, ormancılık, balıkçılık çiftçilik.

Ağır ve hafif sanayi olmak üzere ikincil sanayi ikiye ayrılır. Fabrika ve makinelerin kullanıldığı ağır sanayilere yoğun sermaye gerekmektedir. Bu sanayi kolunda mamul üretimi fazladır. Sanayinin yapısı karmaşık olduğundan işinde uzman olan kişilere ihtiyaç duyulur. Ağır sanayiye çimento, maden, petrol arıtımı, demir-çelik endüstrisi örnek verilebilir.

Dayanıksız tüketim mallarının imalatı hafif sanayidir. Ağır sanayiye göre daha az sermaye gerektirir. Fabrikasyon mobilya, dokumacılık, gıda işleme, konfeksiyon, gibi sanayi örnek verilebilir.

Üçüncül sanayi ise hizmet sektörü olarak bilinir ve elle tutulmaz yararlar sağlar. Somut olarak mal üretmeyen sanayiye ifade eder. Turizm, polis, güvenlik, savunma, bankacılık, emlak hizmetleri, eğitim ve öğretim, sağlık, sigortacılık, ulaşım, haberleşme hizmetleri üçüncül sektöre örnek verilebilir.

3.1.1. Endüstrinin Tarihsel Gelişimi

Buhar makinesinin 1712 yılında bulunması birinci sanayi devrimi (1.0) kabul edilmektedir. Mekanik üretim sistemleri buhar ve su gücünden faydalanarak ortaya

çıkıştır.

Henry Ford araba imalatı için kitlesel üretime geçtiği dönem ikinci sanayi devrimin (2.0) başlangıcı kabul edilmektedir. Telgraf 1840 yılında ve telefon 1880 yılında icat edilmiştir. Ayrıca Taylorizm fikri 1920 yılında ise ortaya çıkmıştır. Elektrik gücüyle seri üretime geçilmiştir.

Dijital devrim üçüncü sanayi devriminde (3.0) gerçekleşmiştir. Elektroniklerin kullanım alanı artmış, bilgi teknolojilerinin gelişimiyle programlanabilen imalat sistemi gelişme göstermiştir. Altair 8800 ilk mikro bilgisayar 1971 yılında, 1976 senesinde ise S. Jobs ve S. Wozniak tarafından ise Apple I üretilmeye başlanmıştır. 2000’li yıllarda Nesnelerin İnterneti ön plana çıkmaya başlamıştır. Endüstrinin tarihsel gelişim süreci Şekil 3.1.’de verilmiştir.



Şekil 3.1. Endüstrinin tarihsel gelişim süreci (Özkurt, 2016)

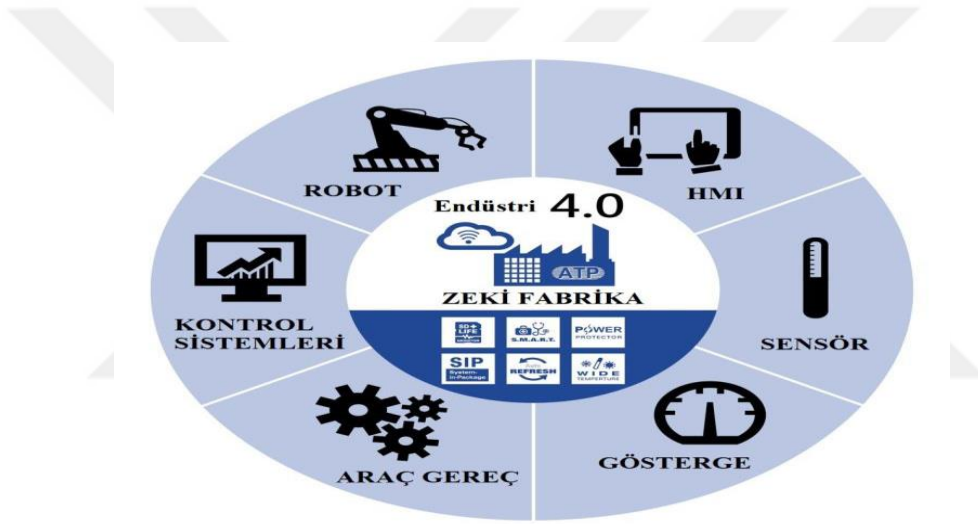
3.1.2. Endüstri 4.0 Kavramı

➤ Endüstri 4.0'ın Yapısı

Endüstri 4.0 üç yapıdan meydana gelmektedir.

- Siber – Fiziksel Sistemler
- Nesnelerin İnterneti
- Büyük Veri ve Bulut Sistemi

Endüstri 4.0, üretim teknolojileri içerisinde otomasyonlu ve veri alışverişi yapabilen “zeki fabrika” olarak da nitelendirilebilecek tarzda yapılar oluşturmak için, Siber - Fiziksel Dizgeleri, nesnelerin internetini ve bulut bilişim sistemlerini günümüz şartlarına uyarlanması olarak değerlendirilebilir. (Kagermann, Wahlsterand Helbig, diğ., 2013). Zeki fabrikalar, Nesnelerin İnterneti yardımıyla Siber - Fiziksel Dizgeler kendileri ve insanlar ile eş zamanlı olmak üzere iletişim kurarak uyumlu çalışabilecektir. İmalat süreçlerinin yenileşme ve çözüm sunma temelinde bilgi ve haberleşme teknolojileri, gömülü sistemler, internet ve veri hizmetleri yer almaktadır (Blanchet, Rinn, Thaden, Thieulloy, 2014). Şekil 3.2.’de Sanayi 4.0’ın genel yapısı verilmiştir.



Şekil 3.2. Endüstri 4.0'ın genel yapısı (Özkurt, 2016)

➤ Siber – Fiziksel Sistemler

Siber -Fiziksel Sistemler, siber ortam ile fiziki dünyayı, internet yardımıyla bağlayan sistemlere denir. Bu sistemler fiziksel dünyadaki değişimler sensör vasıtası internet aracılığıyla toplayarak nesnelerin etkileşimini gerçekleştirir (Geisberger ve Broy, 2012).

1940'lı yıllarda bilgisayarlar, haberleşme teknolojileri ve internet temelli kontrol süreçlerini ifade etmek için siber kavramı kullanılmıştır. Canlılar ile

makinelerin iletişimini konu edinen sibernetik kuramı gelişmiştir. (Bradley ve Atkins, 2015).

Siber – Fiziksel Sistemler kavramı, ilk olarak 2006 yılında ABD’de Lee tarafından kullanılmıştır.

2015 yılında Bradley ve Atkins tarafından Siber – Fiziksel Sistemlerin gelişimini gösteren tarihsel süreç aşağıdaki çizelge 3.1. verilmiştir.

Çizelge 3.1. Siber - Fiziksel Sistemlerin tarihsel gelişim süreci (Bradley, J. M. ve Atkins, E. M., 2015)

Tarih	Olay/Olgu
1932	Frekans teknikleri Nyquist tarafından geliştirmiştir.
1940 - 1945	Örnekleilmiş Data Dizgeleri Kuramı çıkmıştır.
1946	İlk taşınır hücreli telefon çalışmaları devam etmiştir.
1946	İlk bilgisayar olan ENIAC yapılmıştır.
1950	Root Locus yöntemi üzerinde çalışmalar devam etmiştir.
1954	Sayısal Kontrol Dizgeleri geliştirilmiştir.
1969	İnternetin ilk hali ARPANET bulunmuştur.
1973	Gerçek zamanlı veri işleme dizgeleri geliştirilmiştir.
1973	Adaptif, stokastik sistemler, optimal, non-lineer kontrol sistemleri ile geliştirilmiştir.
1990	Hibrit sistemleri çalışmaları devam etmiştir.
1997	IEEE 802.11Wifi standardı bulunmuştur.
2000	Ağ önceliği sistemi olan QoS başlatılmıştır.
2006	Siber - Fiziksel Sistem terimi kullanılmaya başlanmıştır.

Siber - Fiziksel Sistemler akıllı bilgisayar olarak adlandırılabilirler. Fiziksel sistemler için gerekli olan iş ve eylemler ile ilgili dataları sensörler aracılığıyla elde ederek senkron bir şekilde sanal bulut sistemine bağlayan mekatronik bileşenlerdendir. Siber dizgeler sosyoteknik düzenin bir parçasıdır, imalat sürecinde insan benzeri makine arayüzü kullanmaktadır (Hirsch - Kreinsen ve Weyer, 2014).

Hizmet, lojistik, tasarım ve imalat biçimleri, enformasyon ve iletişim teknolojilerinde yaşanan hızlı gelişmeler sonucunda büyük ölçüde etkilenmiştir.

Endüstri 4.0 sürecinde mikrodenetleyici ve fiziksel sürücüler yardımıyla, tüm araç ve makineler kendi kendine kontrolü yapacak ve otomasyonunu olanaklı hale getirecektir. Bilgisayar, haberleşme ve kontrol teknolojileri, senkron algılama, sistemlerin dinamik kontrolü gelişmesine rağmen bu teknolojiler hala ihtiyaçlarımıza tam cevap verememektedir. Siber - Fiziksel Sistemler “akıllı izleme” ve “akıllı kontrol” gerçekleştirme amacı gütmektedir (Yue ve diğerleri, 2015).

İmalatçı firmalar tarafından 2000 - 2010 yılları arasında motorlu taşıt üreticileri ağırlıklı olmak üzere “Akıllı Ortamlar” ve “Kablosuz Sensör Ağları” olarak değerlendirilebilecek radyo frekanslı kimlik sistemini kullanmaya başladılar. Volvo firması RFID sistemini kamyon üretiminin sürekliliği için kullanmıştır. Sonrasında RFID sistemini Toyota, otomobil parçalarının üretiminde kullanmıştır (Sánchez ve diğerleri, 2015).

➤ Nesnelerin İnterneti

2011 yılında Nesnelerin İnterneti (IoT – Internet of Things) kavramı Kopetz tarafından kullanılmıştır. Bir tesisteki farklı kaynakların datalarının elde edilebilmesi, bunların çoğaltılabilmesi ve uyumlu çalışabilmesini ifade eder. Nesnelerin İnterneti büyük veriden elde ettiği verileri kullanarak Siber - Fiziksel Dizgeleri çalıştıracak bilgiye dönüştürmede etkilidir (Lee ve diğerleri, 2015).

Günümüzde üretim sanayisi üzerinde Sanayi 4.0 gelişmeleri etkili olmaktadır. Nesnelerin İnterneti, akıllı fabrikalar, ürünler ve servislerin temelini oluşturan Endüstriyel İnternet olarak da tanımlanabilir (Kagermann ve diğerleri, 2015).

Siber alanda bulunan cihazlar arasında iletişim kurulması, nesneler arası işbirliğinin sağlanması ve verimin artması Nesnelerin İnterneti kavramıyla birlikte olmuştur. Benzer şekilde cihazlar ve insanlar arasında eş zamanlı bağ oluşmaya başlamış ve büyük oranda gelişmiştir.

➤ Büyük Veri ve Bulut Sistemi

İnternet ya da intranet sayesinde endüstriyel cihazların birbirleriyle olan iletişimini gerçekleştirebilmek için büyük serverlara ihtiyaç duymaktadır. Bu bağlamda sistemlerin kontrol ve yönlendirilmesi için big data teknolojisindeki araştırmalar önem kazanmaktadır (Pan ve diğerleri, 2015).

Bulut teknolojilerinde yaşanan gelişmeler, bilgi teknolojilerinde üreticilerin ve bunları tüketicilerinin düşüncesinde değişimler meydana getirmiştir. Bulut teknolojileri, işlemlerin temel yapısında, platformunda, yazılımında ve internet servislerinde köklü değişimlere neden olmuştur (Wang ve diğerleri, 2015).

Firmaların sunucu gereksinimleri büyük veri sistemleriyle azaltmakta, maliyetleri düşmekte ve tüketicilerin de düşük fiyatlarla hizmet almasına olanak sağlamaktadır. Aynı zamanda imalatla kullanılan verilere ulaşmasını kolaylaştırmaktadır. Diğer taraftan büyük veri sistemleri internet erişimine açıktırlar. Bu durumda siber güvenliği mühim bir hal almıştır. İfşa edilen ya da kaybolan datalar işletmeler açısından sakıncalı olmaktadır (Alçın, 2016).

➤ Endüstri 4.0'ın Amacı

Endüstri 4.0'ın temelinde bilgi teknolojileriyle endüstriyi aynı ortama getirmek bulunmaktadır. Sanayi 4.0'ın ana bileşenlerinden ilki, yeni nesil yazılım ve donanım kabul edilir. Bu yazılım ve donanımlar az ısıyan, az yer kaplayan, düşük maliyetli olmaları, yüksek güvenilirlikte çalışmaları sebebiyle geleneksel donanımlarından farklıdırlar.

En mühim ikinci bileşen cihaz tabanlı internettir. Siber – Fiziksel Sistemler, yeryüzünde bulunan bütün makinelerin birbirleri ile bilgi alışverişinde kullanıldığı, her türlü cihaza entegre edilerek, sensör ve işleticilerle donatılmış, internet tabanlı zeki elektronik sistemlerdir.

İmalat yapan fabrikalarda bulunan makinelerde Siber - Fiziksel dizgelerin bulunması nerdeyse tamamen insansız şekilde eş güdümlü, kendi kendini organize ederek üretim yapabilecek “zeki fabrikalar” demektir. Sanayi 4.0 ile beraber giderler, imalat süresi, imalat için tüketilen enerji miktarı azalacaktır. Buna bağlı olarak imalatın miktar ve kalitesinde artış yaşanacaktır.

➤ **Endüstri 4.0’ın Avantajları**

Endüstri 4.0’ ait avantajlardan bazıları aşağıda verilmiştir.

- Sistemin takip edilmesi ve sorun bulmanın kolay olması,
- Dizgeler çevre dostu olacak ve kaynaklarda tasarruf sağlanacak olması,
- Verim artışının olması,
- Üretime esneklik kazandırması,
- Giderlerin azalması,

➤ **Karşılaşılan Zorluklar**

Yeni nesil donanım ve yazılım için Endüstri 4.0 ile enerjinin az olacağı tahmin edildiğinden, bu aşamasında sorun yaşanmayacağı öngörülmektedir. İnternetin 6. sürümü olan IPV6 protokolüyle birlikte milyarlarca cihazın internete bağlanması sağlanacaktır. Burada dikkat edilmesi gereken husus üretimde kullanılan cihazların Sanayi 4.0’ın standartlarına uyumlu olması için yapılacak programlamanın zorluğudur.

Firmalar ve ortakları risk almak istemediklerinde aralarındaki anlaşamamazlık, yapılan değişikliklerin maliyeti arttırması, imalat süreçlerinde bütünlüğün bozulmamasının zorunluluğu gibi sorunların çözümü kolay olmayacak gibi durmaktadır.

3.2. Yapay Sinir Ağları

3.2.1. Yapay Sinir Ağlarına Giriş

Yapay sinir ağları hakkında genel bilgiler bu bölümde anlatılmıştır. Burada ilk önce yapay sinir ağlarına ait genel tanımlamalar verilmiştir. Daha sonra YSA'nın özelliklerinden bahsedildikten sonra yapay sinir ağlarının tarihsel gelişimine değinilmiştir. Yapay sinir ağlarının genel yapısı ve ağların ana unsurları anlatıldıktan sonra yapay sinir ağı modellerinden bahsedilmiştir.

➤ Yapay Sinir Ağlarının Genel Tanımları ve Genel Özellikleri

Yapay sinir ağları (YSA), insanların öğrenme özelliğini kullanarak bilgiyi yorumlayarak çıkarım yapabilme, bilgi türetebilme ve keşfedebilme gibi yetenekleri yardım almaksızın, kendi kendine yapmak için yapılan bilgisayar programlarıdır. Bundan ötürü yapay sinir ağları, çok zor programlanabilen ya da programlanması olanaksız problemler için geliştirilmiş bilgisayar dalıdır (Öztemel, 2003). Aynı zamanda YSA; insan beyninin fonksiyonlardan yola çıkarak geliştirilmiş, birbirleriyle paralel bağlantılı çalışan, aralarında bilgi alışverişi bulunan organizasyonlardır (Elmas, 2003).

Yapay sinir ağları, önceki verilerden faydalanarak hadiseleri öğrenebilen ve benzer hadiselere karşı nasıl tepki vereceğini belirleyen bilgisayar algoritmalarıdır. YSA, öğrenme, taklit etme, gruplandırma, ilişkilendirme, genelleme yapma gibi konularda insan beyninin özelliklerine benzer şekilde model oluşturan bilgisayar programlarıdır. Olaylardan yaptıkları çıkarımlar ile kendi modellemelerini yaparak benzer konularda benzer kararları alırlar. Bu nedenle günümüzdeki birçok problemlere çözüm üretebilirler. Yapay sinir ağlarının en önemli görevi girdi verilerine karşı uygun çıktı verilerini belirlemektir (Öztemel, 2003).

Yapay sinir ağları kısaca, mevcut olayları öğrenerek model olaylar karşısında önceki olaylara benzeri kararlar almaya çalışmaktadır. Yapay sinir ağları dağıtık belleğe sahiptirler. Bilgiler diğer programlardan farklı olarak veri tabanında yer almazlar. Bilgiler ağın bağlantılarıyla ölçülerek bu bağlantılarda saklanırlar. Mevcut durumdaki örnekleri kullanarak öğrenirler ve olay hakkında genelleme yaparlar. Ağın güvenli sonuçlar alması için önce eğitilirler ve sonra da test edilirler. Ağ kendisine verilen örneklerden faydalanıp daha önce bilgi sahibi olmadığı durumlar için de bilgiler üretebilir (Karademir, 2014).

Yapay sinir ağları bilgi dayalı çözümlerden ziyade algılamaya yönelik bilgileri işlemede kullanılırlar. Olaylar karşısında sınıflandırma ve ilişkilendirme yapabilirler, kendiliğinde organize olma ve öğrenebilme kabiliyetleri vardır. Klasik sistemler veri eksikliği olması durumunda görevini yapamazlar ancak yapay sinir ağları eğitimden sonra eksik bilgiler olması durumunda da çalışarak sonuç üretebilirler. Buna rağmen eksik bilgilerle çalışabilmeleri sayesinde hatalara karşı toleranslıdırlar (Karademir, 2014).

Yapay sinir ağları belirsizlik durumunda bile eğitildikten sonra öğrendikleri ile karar verebilirler. Yapay sinir ağları dijital veriler ile çalıştıklarından sembolik ifadeler gösterilen veriler dijital hale dönüştürülmelidir (Öztemel, 2003).

Yapay sinir ağların en önemli dezavantajı ağın nasıl oluşturabileceği ile ilgili kurallarının olmamasıdır. Uygun ağ algoritması deneme yanılma yöntemiyle bulunur. Mevcut probleme ağ uygun olarak oluşturulmazsa tahlili mümkün olan problemin çözümüne ilişkin sonuçlara ulaşamayabilir. Yapay sinir ağlarının ürettikleri sonuçlar kabul edilebilir olmalarına rağmen hiçbir zaman optimum çözüm garantisi vermezler. Katman sayısı, öğrenme katsayısı gibi parametrelerin belirlenmesinde herhangi bir kuram olmadığından bunlar kullanıcının tecrübesi ile belirlenmektedir. Ağın eğitiminin bitim zamanına dair de bir metot yoktur. Burada mühim olan ağın veriler üzerindeki hatanın belirlenen değerin altında olmasıdır. Diğer önemli bir sorun ise ağın olaylar karşısındaki davranışdır. Ağ eğitildikten

sonra bunun nasıl olduğu konusunda bilgiye ulaşmak mümkün değildir (Öztemel, 2003).

➤ Yapay Sinir Ağlarının Tarihçesi

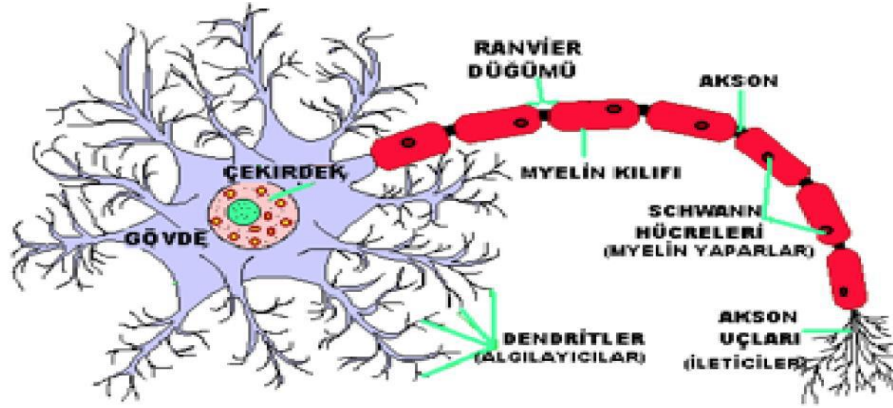
1943 yılında yapay sinir ağlarının ilk modeli, insan beyninin hesaplama yeteneğinde yola çıkarak, Warren McCulloch ile matematikçi Walter Pitts aracılığıyla “Sinir Aktivitesinde Düşüncelere Ait Mantıksal Hesap” kuramıyla ortaya çıkmıştır (Öztemel, 2012, Adıyaman, 2007).

3.2.2. Yapay Sinir Ağlarının Yapısı ve Temel Elemanları

➤ Yapay Sinir Hücresi ve Çalışma İlkesi

İnsan beyninden esinlenerek geliştirilen YSA ile biyolojik sinir hücresi paralellik gösterir. Bu nedenle biyolojik sinir hücresinden bahsedecek olursak;

Beynimizde bulunan çok sayıda sinir hücresinin birleşimi biyolojik sinir ağlarını oluşturur. Sinir hücreleri birbirlerine bağlanarak işlevlerini yerine getirebilmek için milyarlarca sinir ağı bir araya gelmektedir. İnsan beyninde bulunan sinir hücresi sayısı 10^{10} adettir. Bu sinirlerin 6×10^3 'ten daha fazla bağlantısı olduğu bilinmektedir. Biyolojik sinir ağlarının yüksek olan performansları sayesinde karmaşık durumları işleyecek güçte olmaları baz alınarak yapay sinir ağları yardımıyla bu performansın bilgisayara aktarımı amaçlanmıştır (Öztemel, 2003). Biyolojik sinir hücresine ait öbek yapısı Şekil 3.3.'te gösterilmiştir.



Şekil 3.3. Biyolojik sinir öbeği (Aydın, 2019)

Biyolojik sinir hücresi, hücre gövdesi, dendritler ve aksondan oluşur. Bir sinir hücresi, dendrit adı verilen kılcal yollar yardımıyla diğer sinir hücresinden iletilen uyarımları, toplayarak aksonlar yardımıyla beyne iletirler. Sinaps denilen düğüm noktaları akson dallarının sonunda yer alır. Bu düğüm noktaları aksondan aldıkları bilgi sinyallerini beyne iletetek öğrenmeyi gerçekleştirmiş olurlar (Stergiou ve Siganos, 2010).

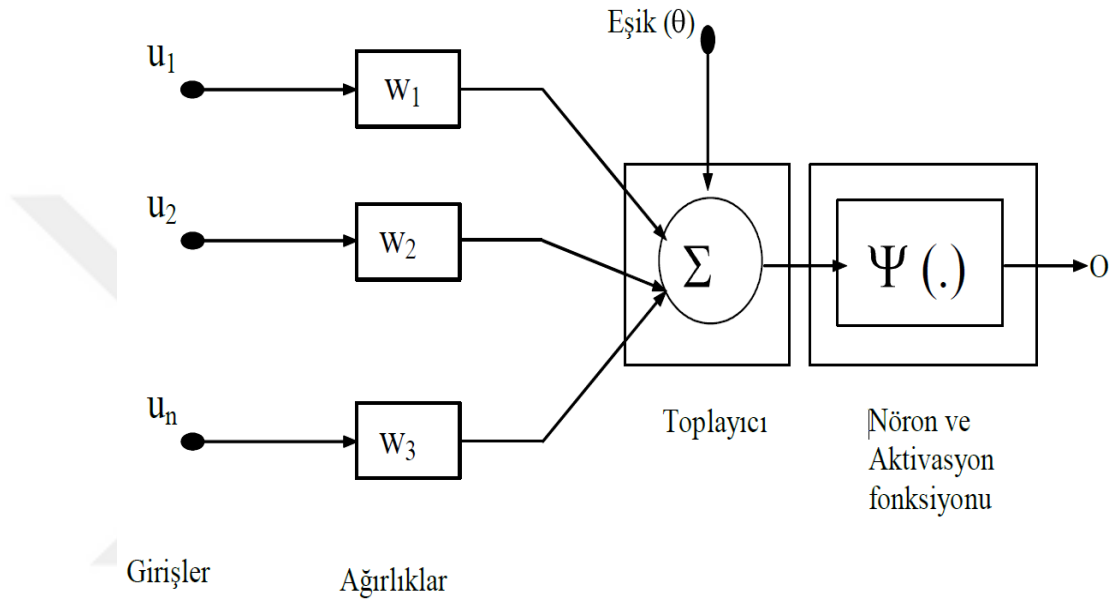
Nöron adı verilen sinir hücreleri, biyolojik sistemden gelen uyarıları alarak bunları yorumlar ve uygun çıktılara dönüştürürler. Bu özellikleri sayesinde çevreden aldıkları hücresel bilgileri işleme ve iletme yetenekleriyle donatılmıştır. Nöronlar yeterince uyarılırsa, aksondan gelen elektriksel uyarıya hemen tepki verirler. Aşağıda biyolojik sinir sisteminin işleyişine ait öbek diyagram Şekil 3.4.'te verilmiştir.



Şekil 3.4. Sinir sistemi öbek şeması (Haykin, 1999).

Burada insana özgü olan sinir uçları yardımıyla biyolojik sinir sistemine veri girişi yapılır. Merkezi sinir sisteminde bu bilgi işlenir. İşlenen bu bilgi sonucunda verinin tepkiye dönüşümü gerçekleşir.

Yapay sinir hücrelerinin diğer bir deyişle nöronların bir araya gelmesiyle yapay sinir ağları oluşur. Buradaki temel fark biyolojik sinir hücresi, yapay sinir ağı hücresine göre daha karmaşıktır. Yapay sinir ağı hücresinde temelde dışardan veya nöronlardan alınan ve giriş olarak kabul edilen veriler, toplama fonksiyonu, aktivasyon fonksiyonu, ağırlıklar ve çıkış bulunmaktadır (Öztemel, 2003). Şekil 3.5.'te yapay sinir hücresi verilmiştir.



Şekil 3.5. Yapay sinir hücresi (Karademir, 2014).

Yapay sinir hücresinin Şekil 3.5.'te katmanlar halinde matematiksel fonksiyonları gösterilmektedir. Nöron çıkışında değişim olması için girdilerde değişimin olması gerekmektedir. Girdinin ağırlık ölçüsünde değişim büyüklüğü değişmektedir. Burada kazanç ağırlıklarını W_i sembolü, giriş verilerini u_i sembolü, eşik değerini θ sembolü, toplama fonksiyonunu Σ sembolü, nöron ve aktivasyon fonksiyonu ise Ψ sembolü ile gösterilir.

Girdiler: Dışarıdan yapay sinir hücresine gelen verilerdir. Bunlar hücreye direkt gelebileceği gibi başka hücreden de gelebilir. Girdilerin verileri kendinden sonraki aşamaya iletmekten başka işlevi bulunmamaktadır. Bir nöronun çevreyi algılamasını sağlayan sınırsız sayıda girişi olmasına rağmen sadece tek çıkışı vardır.

Ağırlıklar: Girdilerden gelen verileri çekirdeğe göndermeden önce geldikleri bağlantıların ağırlıklarıyla çarpıldıktan sonra çekirdeğe iletilir. Başka bir deyişle nörona girecek olan verilerin matematiksel katsayısı olmalarıdır. Bu katsayılar sıfır, negatif ve pozitif olabilir.

Toplama Fonksiyonu: Biyolojik nörondaki dendritlerin yaptığı işi yapan fonksiyondur. Bir hücreye gelen net girdiyi hesaplayan katmandır. Burada girişlerin ağırlıkları ile çarpıldıktan sonra toplanır. Bir olaya uyan toplama fonksiyonu belirlemek için belirli bir yöntem olmayıp, deneme yanılma yoluyla belirlenir. Bir hücreye ağırlıklarıyla çarpılarak gelen girdiler toplanarak o hücrenin net girdi (3.1) eşitliği yardımıyla hesaplanır.

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

Burada ağırlıklar A, girdiler G, hücreye gelen toplam girdi sayısı ise n ile ifade edilir (Öztemel, 2003).

Diğer toplama fonksiyonlarına örnek verecek olursak;

Çarpım: $NET = \pi(G_i \cdot A_i) \quad (3.2)$

Maksimumu: $NET = Max(G_i \cdot A_i) \quad (3.3)$

Minimum: $NET = Min(G_i \cdot A_i) \quad (3.4)$

Çoğunluk: $NET = \sum_{sgn}(G_i \cdot A_i) \quad (3.5)$

Kümülatif Toplam: $NET = NET_{(eski)} + \sum(G_i \cdot A_i) \quad (3.6)$

Aktivasyon Fonksiyonu: Hücreye gelen net girdiyi işleyerek hücrenin üreteceği çıktıyı belirler. Toplama fonksiyonunda olduğu gibi çıktı değerini bulmak için değişik algoritmalar kullanılabilir. Kimi tasarımlarda fonksiyon seçimi yapılırken türevin alınabilmesi şartı koşulmuştur. Probleme en uygun fonksiyonu kullanıcı deneme yanılma yoluyla belirleyebilir. Genellikle kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur(Öztemel, 2003). Bu fonksiyonun formülü aşağıda

verilmiştir. Formülde yer alan NET proses elemanına gelen NET girdi değerini gösterir.

$$\text{Sigmoid fonksiyonu: } F(NE\text{T}) = \frac{1}{1+e^{-NE\text{T}}} \quad (3.7)$$

Bazı aktivasyon fonksiyonları aşağıda verilmiştir.

$$\text{Lineer fonksiyonu: } F(NE\text{T}) = NE\text{T} \quad (3.8)$$

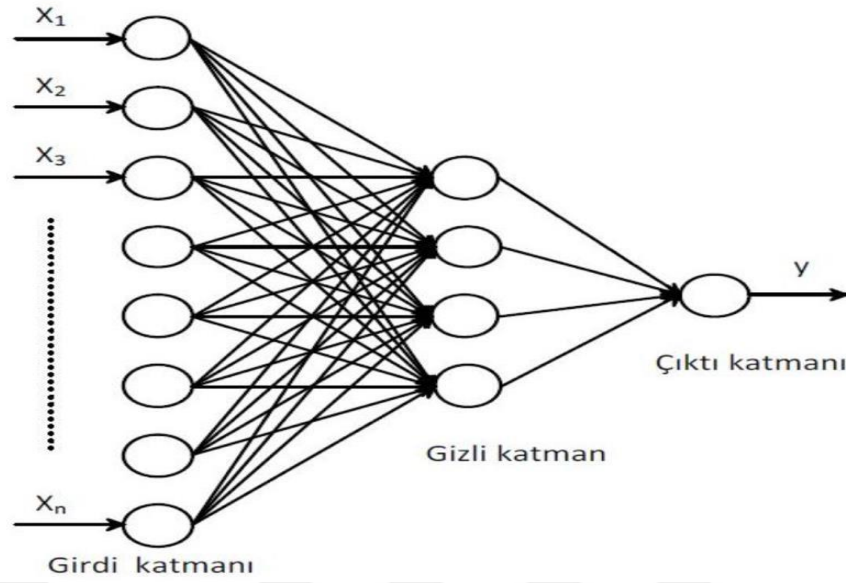
$$\begin{aligned} \text{Step fonksiyonu: } F(NE\text{T}) &= 1 \text{ } NE\text{T} > \text{eşik değeri} \\ F(NE\text{T}) &= 0 \text{ } NE\text{T} \leq \text{eşik değeri} \end{aligned} \quad (3.9)$$

$$\text{Sinüs fonksiyonu: } F(NE\text{T}) = \sin(NE\text{T}) \quad (3.10)$$

Hücre çıkışı: Aktivasyon fonksiyonunun belirlediği hücrenin çıkışıdır. Hücre çıkışı dış dünyaya iletebileceği gibi farklı bir hücreye de iletebilir veya hücre kendi ürettiği çıktıyı kendine girdi verebilir. Bir hücrenin birden çok girdisi olabilir ancak tek çıkışı olmaktadır. Bu çıktı birden fazla hücreye bağlanabilir.

➤ Yapay Sinir Ağları ve Çalışma İlkesi

Yapay sinir hücreleri olan nöronlar farklı şekillerde bir araya gelerek yapay sinir ağını oluştururlar. Yapay sinir ağları Şekil 3.6.'dan anlaşılacağı gibi katmanlardan meydana gelmektedir. Yapay sinir ağları genellikle 3 katmadan oluşmakta ve her katman kendi içinde paralel halde ağı oluşturmaktadır.



Şekil 3.6. Yapay sinir ağ modeli (Aydın, 2019).

Yapay sinir ağları girdi katmanı, gizli katman ve çıktı katmanı olmak üzere 3 katmandan oluşmaktadır.

Girdi katmanı: En az bir girdi unsurunun bulunduğu ve yapay sinir ağına dış ortamdan gelen verilerin ara katmanlara iletiildiği katmandır. Bu katmandaki datalar işlenmeden, olduğu gibi giriş ile aynı değerde çıktılar üretmektedir. Bu yüzden pasif bir katmandır. Bu katmandaki hücreler giriş bilgilerinin gizli katmana ulaştırmakla görevlidir.

Gizli (Ara) Katman: Girdi katmanından gelen verilerin belirli işlemlere tabi tutularak çıkış katmanına gönderildiği bölümdür. Problemin yapısına uygun olarak seçilen ağ yapısına göre katmanın yapısı, fonksiyonu ve sayısı değişebilir.

Çıktı Katmanı: Gizli katmanın ilettiği verileri işleyerek ağın çıkış verilerinin üretildiği katmandır. Bu katmandaki hücreler üretilen çıkış verileri problemin çözümünü içermekte olup dış dünyaya üretilen outputlar gönderilir.

YSA, işlem sürecince girdi verilerini aldıktan sonra nöron ve aksonlar yardımıyla işlem sürecini çıkışa doğru sürdürürler. Dış ortamdan veya nöronlardan

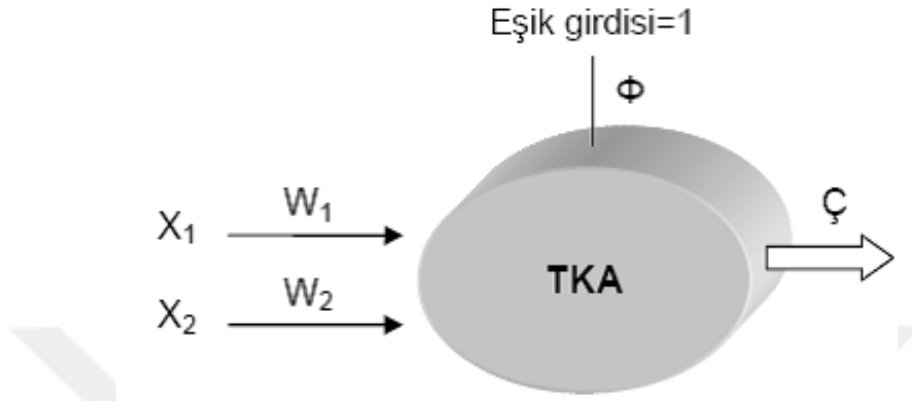
alınan veri girişlerini bir takım işlemten geçirdikten sonra çıktı setine çevirmektedir. Doğru sonuçların alınabilmesi için giriş verilerin sayısal verilere dönüştürüldükten sonra, ağı eğitilmesi gerekmektedir. Ağırlıkların, yapay nöronlar arasında en uygun şekilde ayarlanmasıyla öğrenme gerçekleşebilir. Yapay sinir ağları klasik programlama benzer şekilde belirli bir algoritmayla öğrenme yapmadıklarından “Kara Kutu” olarak adlandırılırlar. Bu da yapay sinir ağlarına olan güveni sarsmaktadır. Yapay nöronlar mevcut veriler üzerindeki girdi ile çıktılar arasındaki bağlantıların ağırlıklarını değiştirmek suretiyle eğitilirler. İlk önce bu ağırlık değerleri rastgele belirlenir. Bu değerler ağı tanıtıldıkça ağırlıklarında değişim olur. Bu değişim doğru sonuca ulaşana kadar devam eder. Böylece ağ, örneklerin temsil ettiği değerler hakkında genelleme yapmaya başlar. Bu duruma öğrenme kuralları denir.

İki aşamalı öğrenme süreci vardır. Birinci aşamada ağ örnek veriler kullanılarak eğitilir. Bu aşamada ise ağı bağlantı ağırlıkları değiştirilir. İkinci aşamada ağı öğrenip öğrenmediğini test etmek için denemeler yapılır. Deneme aşamasında ağı daha önce kullanmadığı veriler kullanılarak doğruluğu test edilir. Test sonuçlarına göre çıktıların doğruluğu hakkında bilgi verir. Sonuçların iyi olması demek, ağı eğitimi iyi olması demektir. Eğitim sırasında kullanılan verilere “eğitim seti”, test yapılırkenki verilere “test seti” denilmektedir (Öztemel, 2003).

3.2.3. Yapay Sinir Ağları Sınıflandırılması

Yapay sinir ağları, sinir hücrelerinin bağlantı modellerine göre farklı mimarilerden oluşmaktadırlar. Hücrelerin bağlantı modellerine, öğrenme yordamlarına ve aktivasyon işlevlerine göre ağı model belirlenmektedir. Yapay sinir ağları, tek katmanlı veya çok katmanlı olarak birbirinden ayrılabilirler.

Tek katmanlı algılayıcılar, yalnızca bir girdi ve bir çıktı katmanından oluşur. Bütün bağlantıların ağırlığı (W) vardır, çıktı üniteleri bütün girdilere (X) bağlantılanmıştır. Şekil 3.7.'de kek katmanlı bir yapay sinir ağı gösterilmiştir.

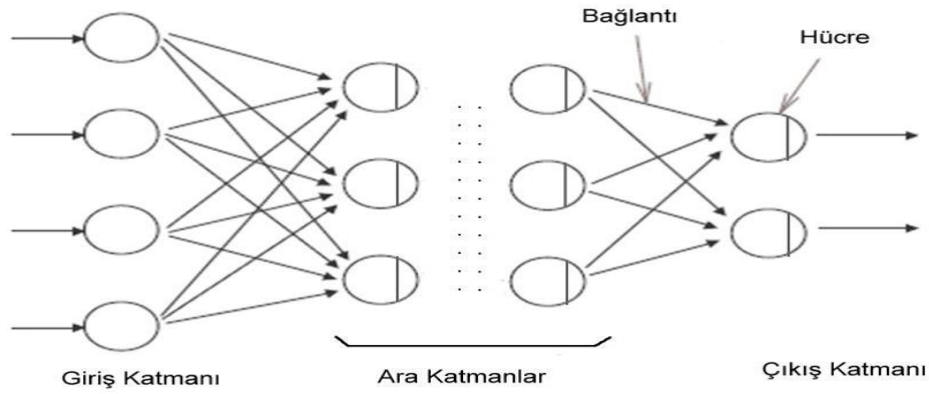


Şekil 3.7. Tek katmanlı yapay sinir ağı (Öztemel,2003)

Çok katmanlı algılayıcılar, doğrusal olmayan fonksiyonlar için kullanılmaktadır. Bu ağı kullanılış amacı ağdan beklenen çıktı ile ağdan üretilmiş olan çıktı arasındaki farkı en aza indirmektir. Bu ağlarda girdi katmanına veriler ara katmanlardan geçerek çıkış katmanına ulaşır ve çıktılar dış ortama verilir.

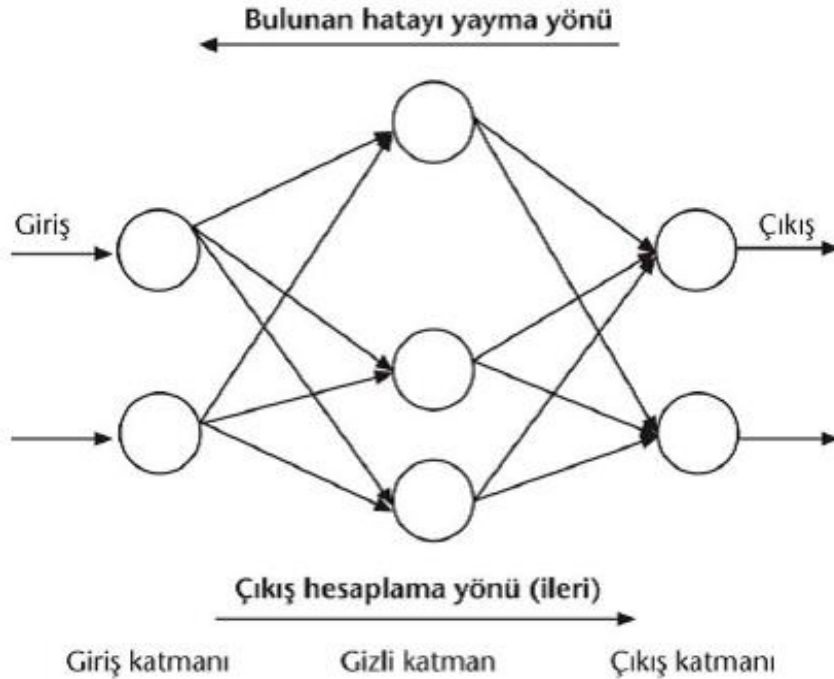
YSA yapılarına göre ise ileri (feed - forward) ve geri (feed - back) beslemeli olmak üzere iki farklı yapıya ayrılabilir.

İleri beslemeli yapay sinir ağlarında, nöronlar girişten çıkışa doğru katmanlar şeklinde düzenlenmiştir. Bir katman kendinden sonra gelen katmanla arasında bağ vardır. YSA'ya giriş katmanına gelen bilgiler daha sonra sırasıyla ara katmanlarda ağırlıkları üzerinden işlendikten sonra çıkış katmanına iletilerek dış ortama aktarılır. Şekil 3.8.'de ileri beslemeli yapay sinir ağına örnek verilmiştir.



Şekil 3.8. İleri beslemeli yapay sinir ağı (Öztemel, 2003)

Geri beslemeli yapay sinir ağlarında, ağda bulunan en az bir hücre çıktısı ya kendisine ya da diğer hücrelere giriş olmaktadır. Geri besleme yapılması için bir geciktirme elemanının kullanılması gerekmektedir. Söz konusu geri besleme, katmanlarda olabileceği gibi katmanlar arasında da olabilir. Geri beslemeli yapay sinir ağına örnek şekil 3.9.'da verilmiştir.



Şekil 3.9. Geri beslemeli yapay sinir ağı (Öztemel, 2003).

3.2.4. Yapay Sinir Ağlarının Temel Öğrenme Kuralları

Yapay sinir ağlarının öğrenme yeteneğine sahip olmaları en önemli özellikleridir. Bilgiler ağdaki sinirlerin bağlantıları ağırlıklarında tutulur. İyi bir öğrenme yapılabilmesi için ağırlıkların belirlenmesi önemli bir husustur. Bilgiler tüm ağda tutulduğundan bir düğümün ağırlık değeri tek başına bir anlam ifade etmemektedir. Ağda bulunan ağırlıklar en uygun değerde olmalıdır. Bu ağırlıklara ulaşma işlemine ağırlık eğitimi denilmektedir. Başka bir ifadeyle, öğrenme işlemi ağırlıkların en iyi değerini bulma işlemidir denebilir. Günümüzde en yaygın kullanılan öğrenme modelleri şunlardır.

➤ Hebb Öğrenme Kuralı

1949 yılında Donald Hebb tarafından geliştirilmiştir. Donald Hebb'in "The Organization of Behaviour" kitabında bahsedilmiştir. Bu kurala göre şayet sinyali alan ile sinyali gönderen nöronlar arasındaki ağırlıkların her ikisi matematiksel aynı işaretli ise söz konusu nöronların bağı güçlendirilmelidir. Farklı işaretli ise bağların kuvvetleri düşürülmelidir (Çelik, 2008).

Öğrenme kurallarından en eskisi olup kendinden sonraki kuralların temelini oluşturmaktadır. Bu kurala göre öğrenme sinyali ile hücre çıkışı aynıdır. Eğer hücre aktif durumdaysa bağlandığı hücreyi aktif yapmaya, pasif ise bağlandığı hücreyi pasif etmeye yeltenir.

➤ Hopfield Öğrenme Kuralı

Hebb kuralına benzemesinin yanında bağ kuvvetlerindeki değişimlerin ne kadar olacağını ifade eder. Bu kurama göre öğrenme hızı ile bağ kuvveti bağlantılıdır. Hız artırılır veya azaltılır. Öğrenme hızı negatif ya da pozitif sayısal değerdedir ve 0 ile 1 arasında seçilir (Efendigil, 2008).

➤ Delta Öğrenme Kuralı

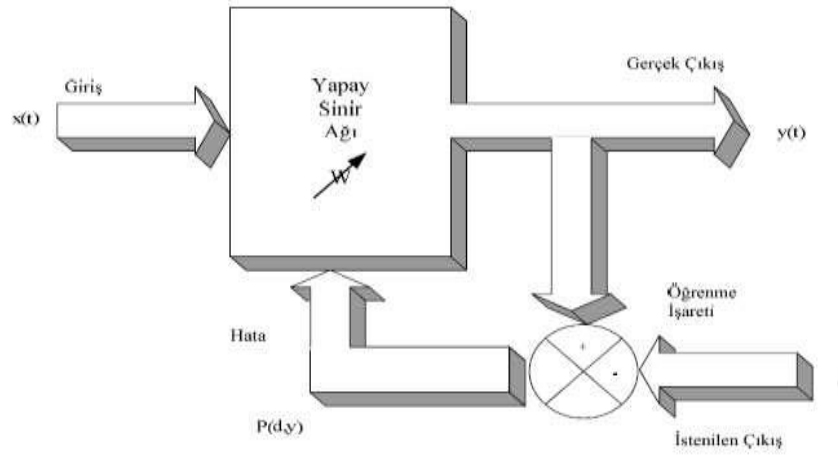
Windrow ve Hoff tarafında geliştirilmiştir olup günümüzde kullanılan en yaygın kuraldır. Delta kuralında, yapay sinir hücresindeki nöronların ağırlık değerlerinin sürekli değiştirilmesiyle sinir hücresinin gerçek çıktı değeri ile beklenen değeri arasındaki farkı azaltmaya çalışılmaktadır (Saygılı, 2008).

➤ Kohonen Öğrenme Kuralı

Kohonen tarafından biyolojik sistemdeki öğrenmeden yola çıkarak ortaya atılmıştır. Giriş ve çıkış tabakası olarak iki tabakadan oluşan bir ağıdır. Bu kuralda hücreler birbirleri ile yarış içerisindeyler. Kazanan nöronun ağırlıkları güncellenmekte olup buna “kazanan tamamını alır” adı verilmektedir. Çıkışı en büyük olan nöron kazanır. Kazanan nöron komşularını ikaz etme ve men etme kapasitesine sahiptir. Bu kuralda hedef çıkışa ihtiyaç bulunmamaktadır. Bundan dolayı danışmansız öğrenme algoritmasıdır.

➤ Gözetimli(Danışmanlı) Öğrenme Yöntemi

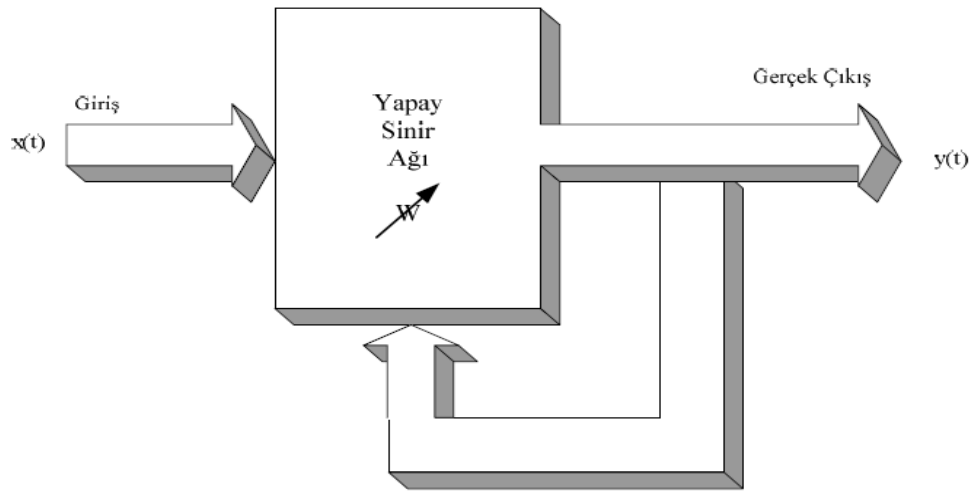
Gözetimli öğrenme sırasında ağa hem giriş değerlerini hem de bu giriş değerlerine karşılık gelen çıkış değerleri verilmektedir. Yapay sinir ağı verilen bu girdilere karşılık istenen çıkış değerlerini elde etmek için kendi ağırlıklarını güncellerler. Ağa ait çıktı değerleri ile beklenen değerler arasındaki hatayı hesaplayarak ağı yeni ağırlıklarını bu hata payına istinaden yeniden düzenler. Şekil 3.10.’da gözetimli öğrenme algoritması verilmiştir.



Şekil 3.10. Gözetimli öğrenme (Adıyaman, 2007).

➤ Gözetimsiz(Danışmansız) Öğrenme Yöntemi

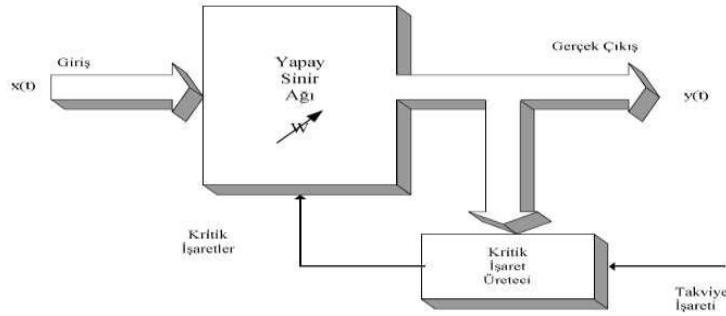
Ağın eğitilmesi sırasında ağa yalnızca örnek girdilerin verildiği öğrenme türü danışmansız öğrenmedir. Ağdan beklenen çıktı verisi girilmez. Ağ girişine verilen bilgiler ışığında her bir örneği kendi arasında gruplandırarak kendi yasalarını belirler. Ağ aynı ağırlıktaki dokuları ayırabilecek şekilde düzenledikten sonra öğrenme işlemini bitirir. Şekil 3.11.'de gözetimsiz öğrenme algoritması gösterilmiştir.



Şekil 3.11. Gözetimsiz öğrenme (Adıyaman, 2007).

➤ Destekleyici Öğrenme

Bu öğrenme çeşidinde ağa bir danışman yardımcı olur. Bu algorithmada ağın her iterasyon sonucunun iyi ya da kötü olup olmadığına ilişkin bilgi verilir. Bu bilgiler ışığında ağ yeniden kendini eğitir. Böylelikle ağ girdi dizisiyle hem öğrenme yaparak hem de sonuç çıkararak işlemlerini devam ettirir. Şekil 3.12.'de destekleyici öğrenme algoritması gösterilmiştir.



Şekil 3.12. Destekleyici öğrenme yapısı (Hopfield,1982).

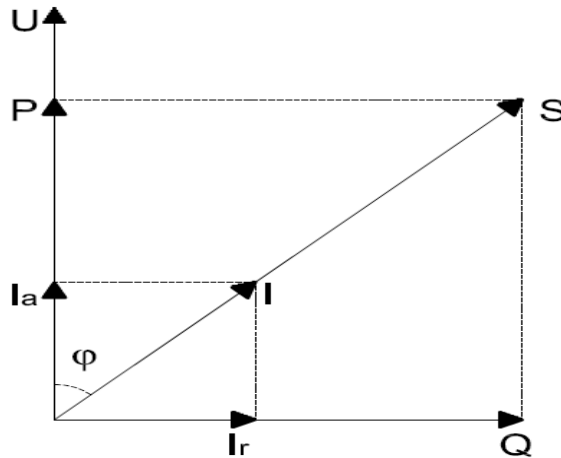
3.3. Reaktif Güç Kompanzasyonu

3.3.1. Reaktif Güç ve Güç Katsayısı

Alternatif akım olarak üretilen elektrik enerjisi aktif ve reaktif olarak iki bileşenden oluşmaktadır. Tüketiciler için faydalı olan aktif güç, aktif akımdan meydana gelmektedir. Örnek verecek olursak bu güç bileşeni, motor millerinde mekanik güce, aydınlatma sistemlerinde ışık akışına, ısı merkezlerinde termik güce dönüşmektedir (Bayram, 2000).

Jeneratör, trafo, bobin ve motor gibi elektrik makineleri elektrodinamik prensibe göre çalışmaktadırlar. Bu makinelerin çalışması için lazım olan manyetik alanı meydana getiren mıknatıslanma akımına, reaktif akım denilmektedir. Reaktif akımın çektiği güce reaktif güç denilmektedir. Reaktif güç, aktif güç gibi faydalı güce çevrilmez (Bayram, 2000).

Aşağıdaki akım, gerilim ve güçlerin fazör diyagramı Şekil 3.13.'te gösterimi verilmiştir.



Şekil 3.13. Akım, gerilim ve güçlerin fazör diyagram gösterimi (Aydın, 2015)

Şekil 3.13.'te göre bir tüketicilerin şebekeden çektiği güçler ve akımlar aşağıdaki gibidir.

$$\text{Görünür Güç} \quad S = U \cdot I \quad (3.11)$$

$$\text{Aktif Akım} \quad I_a = I \cdot \cos \varphi \quad (3.12)$$

$$\text{Aktif Güç} \quad P = S \cdot \cos \varphi = U \cdot I \cdot \cos \varphi \quad (3.13)$$

$$\text{Reaktif Akım} \quad I_r = I \cdot \sin \varphi \quad (3.14)$$

$$\text{Reaktif Güç} \quad Q = S \cdot \sin \varphi = U \cdot I \cdot \sin \varphi \quad (3.15)$$

Denklemlerde gerekli düzenlemeler yapılırsa aşağıdaki denklemler elde edilebilir.

$$\text{Görünür Akım} \quad I^2 = I_a^2 + I_r^2 \quad (3.16)$$

$$\text{Görünür Güç} \quad S^2 = P^2 + Q^2 \quad (3.17)$$

Yukarıdaki denklemler incelenecek olursa aktif güç, görünür güç ile $\cos \varphi$ 'nin çarpımına eşittir. Burada bulunan gerilim ile akım arasındaki faz açısı φ açısıdır. φ açısının kosinüs değeri güç katsayısıdır (Bayram, 2000).

İki karakterli güç sistemleri bulunmaktadır. Birincisi endüktif karakterli olup burada akım gerilimden geridedir. İkincisi ise kapasitif karakterlidir ve akım gerilimden öndedir.

3.3.2. Reaktif Güç Kompanzasyonu

Elektrik tesislerinin ve yüklerin reaktif gücünü gerektiği durumlarda belirli tekniklerle karşılanmasına “Reaktif Güç Kompanzasyonu” denir. Diğer bir ifadeyle; düşünsel olarak gerilim ile akım arasında fark olmaz. Kapasitif ya da endüktif yüklerin oluşturduğu etki sonucunda akım ile gerilim arasında $\pm 90^\circ$ kayma olur. Gerilim ile akım arasındaki faz farkını düzeltip, ideale yakın tutmak için yapılan işleme “reaktif güç kompanzasyonu” denilmektedir.

EPDK 10/11/2004 tarihli ve 25639 sayılı Resmi Gazetede yayımladığı Elektrik İletim Sistemi Arz Güvenilirliği ve Kalitesi Yönetmeliğinin 11. maddesine göre sisteme verilen kapasitif reaktif enerjinin sistemden çekilen aktif enerjiye oranı ise %15'i ve sistemden çekilen endüktif reaktif enerjinin sistemden çekilen aktif enerjiye oranı %20'yi geçemez.

Reaktif güç kompanzasyonu sayesinde tüketicinin güç faktörü düzeltilerek tüketicilerin reaktif güçten kaynaklı cezaların oluşması önlenmiş olur. Dağıtım şebekesindeki kompanze edilmemiş yüklerden dolayı meydana gelen ani reaktif güç değişimleri şebeke gerilimin de değişmesine neden olur. Gerilimde yaşanan bu dalgalanma aynı baraya bağlı diğer tüketicileri de olumsuz etkileyerek tüketicilerin arıza yaşamasına neden olur. Kompanzasyon yapılarak bu gerilim değişimleri en aza indirilmeye çalışılmaktadır. Böylece tüketicilerin yaşayacağı sıkıntılar önlenmiş olur.

3.3.3. Reaktif Güç Tüketicileri ve Sistemin Tasarlanması

Tüketiciler reaktif güç sarfiyatı bakımında iki ana gruba ayrılır. Akkor flamanlı lambalar gibi ısı üreten tüketiciler birinci grupta yer alır. Bu gruptaki tüketiciler sadece aktif güç tüketirler. İkinci grupta yer alan tüketiciler ise manyetik ve statik alan ile çalışan tüketicilerdir. Bu gruptaki tüketiciler aktif güçle beraber reaktif güç de çekmektedirler.

Bazı reaktif güç tüketicileri aşağıda verilmiştir.

- Bobinler,
- Elektrik makineleri,
- Havai hatlar,
- Endüksiyon fırınları,
- Ark ocakları,
- Kaynak makineleri,
- Balastlar,
- Doğrultucular.

Yukarıda örnek verilen yükler işletmede harmonik üretimine ve akım darbelerine neden olurlar. Burada reaktif değişimler çok hızlı olup çektikleri reaktif güç değerleri de büyük olmaktadır. Bundan dolayı reaktif güç kompanzasyonu yapılırken çok kısa zamanda büyük değişimleri karşılayacak şekilde tasarlanmalıdır (Özaslan, 1997 ve Ersamut, 2009).

Bir reaktif güç kompanzasyon sisteminin tasarlanması sırasında aşağıdaki hususlar dikkate alınmalıdır.

- Maksimum reaktif güç gereksinimi,
- Tesisin aşırı yüklenme sınırı ve süresi,
- Aşılmaması gereken gerilim sınırları,
- Frekans ve değişimi,
- Gerilim regülasyonu,
- Reaktif güç kompanzasyon sisteminin cevap verme süresi,
- Maksimum harmonik bozulmaları,
- Dengesiz gerilimde ve yükte çalışma verimi,
- Koruma donanımı ve koordinasyonu,
- İşletmenin ileriye dönük genişleme olanakları,
- Güvenilirlik,
- Bakım,
- Çevresel faktörler, gürültü seviyesi, sıcaklık, nem, soğutma sistemi, kirlenme.

Reaktif tüketici yükleri işletmede harmonik üretimine ve akım darbelerine neden olurlar. Burada reaktif değişimler çok hızlı olup çektikleri reaktif güç değerleri de büyük olmaktadır. Bundan dolayı reaktif güç kompanzasyonu yapılırken çok kısa zamanda büyük değişimleri karşılayacak şekilde tasarlanmalıdır (Özaslan, 1997 ve Ersamut, 2009).

Reaktif güç kompanzasyonu sayesinde reaktif akım, kompanzasyon sistemi aracılığı ile karşılanacağı için, şebekeden daha düşük değerde akım çekilecektir. Buna bağlı olarak transformatörler ve hatlar daha az yüklenecek ve aşırı yüklenmeler

önlenmiş olacaktır. Tesiste aşırı yüklenme olmaması durumunda tesisin kapasitesi artacaktır (Ersamut, 2009).

3.3.4. Kompanzasyon Sistemlerinin Avantajları ve Dezavantajları

Şebeke açısından reaktif güç kompanzasyonun faydaları;

- Güç kalitesini artırır,
- Isı kayıplarını azaltır,
- Gerilim düşümünü azaltır,
- İletim hatlarının geçici durum kararlılığını iyileştirir,
- Gerilim dengesizliklerini azaltır,
- Gerilim dalgalanmalarını (flikler olayları) azaltır,

Reaktif güç kompanzasyonun tüketici açısından faydaları ise; (Bostancı, 2008).

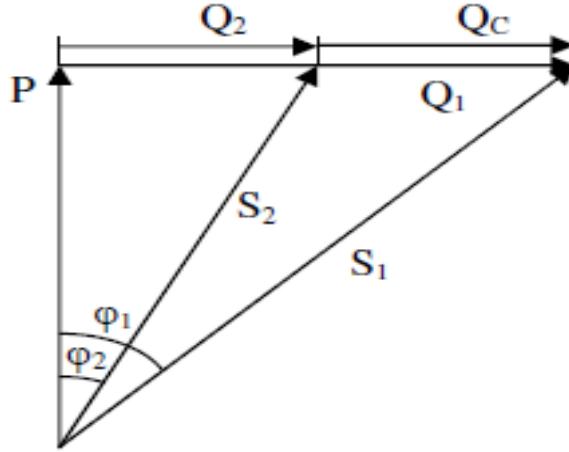
- İşletme maliyetlerini azaltır,
- Müşteri güç katsayısının düzeltilmesiyle şebekeden düşük miktarlarda reaktif güç çekerek daha az para öder.
- Kalitesiz enerjiden meydana gelecek arıza riskini indirgemiş olur.
- Enerji kalitesi artışı için imalat kalitesi de artar.

3.3.5. Reaktif Güç Kompanzasyonunda Kapasite Hesabı

Sistemin reaktif güç hesabının yapılabilmesi için şebekeden çekilen S_1 gücüyle, bu güce ait $\cos \varphi_1$ ve hedeflenen güç faktörü $\cos \varphi_2$ değerleri gereklidir. Reaktif güç kompanzasyonu iki yöntemle hesaplanır. Bunlardan ilki aktif gücün (P_1) sabit tutularak yapılmasıdır. İkinci hesaplama yöntemi ise görünür gücün (S_1) sabit tutularak yapılmasıdır. Genel anlamda tesiste kurulacak kompanzasyon sistemi tasarlanırken yüklerin dağılımına, reaktif güçteki değişimlere ve tesisteki harmoniklere göre yapılır (Keçecioğlu, 2012).

➤ Aktif Güç Sabitken Yapılan Kapasite Hesabı

Aktif Güç P sabit iken reaktif güç hesabında şekil 3.14.'teki fazör diyagramı kullanılarak aşağıdaki hesaplamalar yapılır.



Şekil 3.14. Aktif güç sabitken reaktif güç tespiti (Aydın, 2015)

Buna göre;

Sistemin ilk durumdaki reaktif gücü;

$$Q_1 = P \times \tan \varphi_1 \quad (3.18)$$

olarak hesaplanır. Sistemin istenilen reaktif gücü,

$$Q_2 = P \times \tan \varphi_2 \quad (3.19)$$

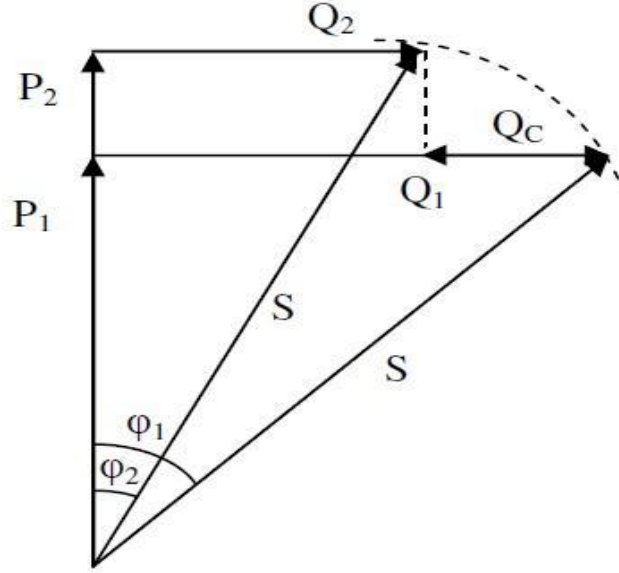
olarak ifade edilir. Denklem (3.18) ve (3.19) göre gerekli olan kondansatör gücü,

$$Q_c = Q_1 - Q_2 = P \times (\tan \varphi_1 - \tan \varphi_2) \quad (3.20)$$

olarak bulunur (Ersamut, 2009).

➤ Görünür Güç Sabitken Yapılan Kapasite Hesabı

Bu yöntemde görünür güç S sabit tutularak şebekeden çekilen aktif güç artırılır. Reaktif güç hesabında Şekil 3.15.'teki fazör diyagramı kullanılarak aşağıdaki hesaplamalar yapılır.



Şekil 3.15. Görünür güç sabitken reaktif güç tespiti (Aydın, 2015)

Buna göre;

Sistemin ilk durumdaki reaktif gücü;

$$Q_1 = S \times \sin \varphi_1 \quad (3.21)$$

olarak hesaplanır. Sistemin istenilen reaktif gücü,

$$Q_2 = S \times \sin \varphi_2 \quad (3.22)$$

olarak ifade edilir. Denklem (3.21) ve (3.22) göre gerekli olan kondansatör gücü,

$$Q_c = Q_1 - Q_2 = S \times (\sin \varphi_1 - \sin \varphi_2) \quad (3.23)$$

olarak bulunur.

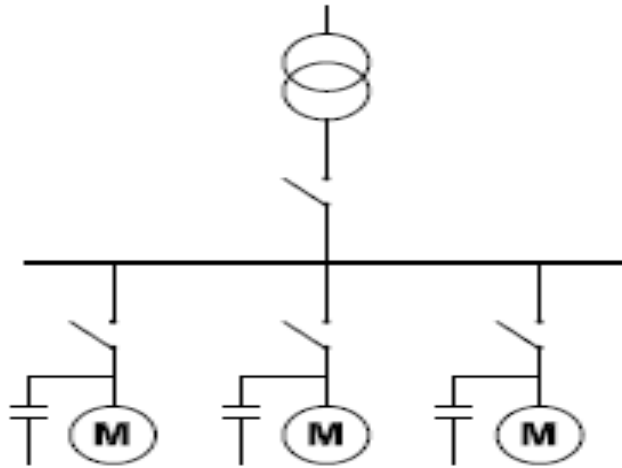
3.3.6. Reaktif Güç Kompanzasyon Çeşitleri

➤ Bireysel Kompanzasyon

Bir tesiste sürekli çalışan sabit güçlü tüketicilerin reaktif güç ihtiyacını karşılamak için söz konusu sabit tüketicinin uçlarına paralel kondansatör bağlanarak bireysel kompanzasyon yapılır. Bu tasarımda her sabit tüketicinin reaktif gücü için kendine paralel uygun güçteki kondansatör bağlanır. Bu kondansatör, anahtarlama elemanları yardımıyla tüketiciyle beraber devreye girip çıkartılır (Aydın, 2015).

Bu tasarımda dikkat edilmesi gereken en önemli husus seçilecek olan kondansatör gücü sabit tüketicinin gücüne uygun olmalıdır. Bireysel kompanzasyonda her tüketici için ayrı ayrı kondansatör ve anahtarlama elemanları gerektiğinden pahalı bir tasarımdır. Ancak sürekli çalışan sabit tüketiciler için ekonomiktir. Bu tasarım sürekli devreye girip çıkmayan sabit yüklü motorlarda balastlı lambalarda kullanılır.

Bireysel kompanzasyon tasarımı Şekil 3.16.'da gösterilmiştir.

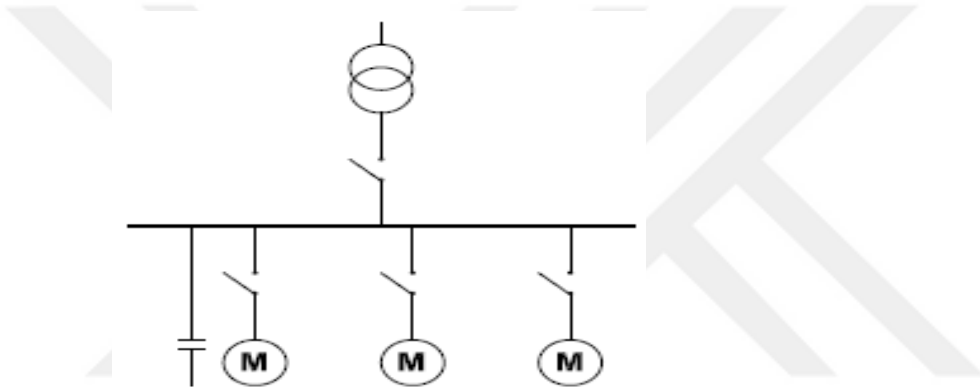


Şekil 3.16. Bireysel kompanzasyon tasarımı (Aydın, 2015)

➤ Grup Kompanzasyonu

Bir tesiste tek kontaktör yada ortak şalter üzerinden devreye alınıp çıkarılan yük kompanzasyonu tasarımlarında tercih edilir. Tesisteki yükleri bireysel kompanze etmektense birlikte sisteme girip çıkan yükleri kompanze etmekte kullanılır. Bu tasarımda tesisin kompanzasyon masrafları daha az olur ve bireysel kompanzasyon tasarımından ekonomiktir.

Şekil 3.17.'de grup kompanzasyon tasarımı verilmiştir.

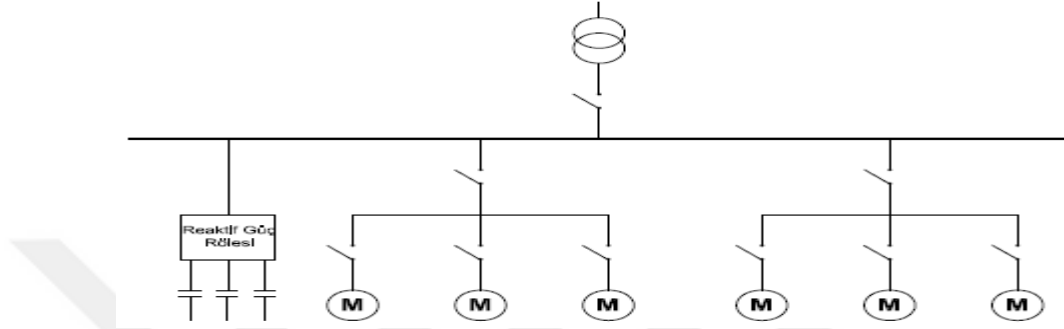


Şekil 3.17. Grup kompanzasyon tasarımı (Aydın, 2015)

➤ Merkezi Kompanzasyon

Bir tesiste ana panoya bağlı çok tüketici varsa ve bu tüketicilerin ne zaman devreye girip çıkacakları belli değilse çekilen yüke göre ayarlı bir kompanzasyon yapılır. Merkezi kompanzasyon elle kumanda edilecek veya otomatik çalışacak şekilde yapılabilir. İşletmelerde daha çok merkezi kompanzasyon tercih edilir. Merkezi kompanzasyon sistemi, farklı yüklerde otomatik çalışan, grup kompanzasyonuna göre gelişmiş bir tasarımdır. Merkezi kompanzasyon sisteminde, tüketici miktarı fazladır. Tüketicilerin her biri farklı güçlerde ve zamanlarda devrede bulunur. Bu sebeple gereken kondansatör gücü, talebe göre ayarlanmalıdır. Bunu yapabilmek için merkezi kompanzasyonda ayar sistemi kullanılır. Merkezi

kompanzasyon işletmelerinde güç katsayısını sürekli kontrol eden bir cihazla, duruma binaen kondansatör grupları devreye girerler veya çıkarlar. Bu cihazdan gelen ölçüm değerleri ile gerekli sayıda kondansatör grubu devreye alınarak güç faktörünün istenen seviyede kalması sağlanır. Şekil 3.18.'de merkezi kompanzasyon tasarımı verilmiştir.



Şekil 3.18. Merkezi kompanzasyon tasarımı (Aydın, 2015)

3.3.7. Kompanzasyon Sistem Tasarımında Kullanılan Geleneksel Yöntemler

➤ Sabit Kondansatörler ile Reaktif Güç Kompanzasyonu

Kondansatörler, iki iletken levha arasına bir yalıtkan konulmasıyla elde edilir. İki iletken arasında bulunan yalıtkan levhaya dielektrik adı verilmektedir. Kağıt, tantal mika, seramik, polyester vb. maddeler dielektrik madde olarak kullanılır. Kondansatör birimi olarak Farad kullanılır ve 'C' ile gösterilir.

Kondansatörlerin kapasitesi;

$$C = \varepsilon \cdot \frac{F}{d} \quad (3.24)$$

formülüyle hesaplanır.

Burada F levhanın yüzey alanı (m^2), d levhalar arası uzaklık (m), ϵ dielektrik katsayısı olarak ifade edilir.

➤ Kondansatörlerde Güç Hesabı

Alternatif akım şebekesinde kondansatörler reaktans özelliği gösterirler. Kapasitif reaktans aşağıdaki formüle göre hesaplanır.

$$X_c = \frac{1}{\omega \cdot C} \quad (3.25)$$

Burada ω (derece/saniye) şebeke sinyalinin açısal hızı;

$$\omega = 2\pi f \quad (3.26)$$

tir. f şebeke frekansıdır ve birimi Hz.'dir.

Bir kondansatörün şebekeden çektiği kapasitif akım;

$$I_c = \frac{U}{X_c} = U \cdot \omega \cdot C \quad (3.27)$$

olarak hesaplanır.

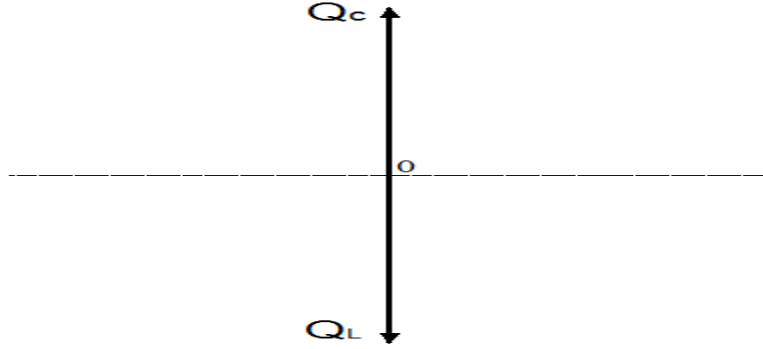
Kondansatörlerde, kapasitif akım, gerilime göre 90° ileride gider. Bir fazlı AC sistemlerde kondansatörün gücü;

$$Q_c = U \cdot I_c \cdot 10^{-3} \quad (3.28)$$

veya

$$Q_c = U^2 \cdot \omega \cdot C \cdot 10^{-3} = \frac{I_c^2}{\omega \cdot C} \cdot 10^{-3} . \quad (3.29)$$

olarak bulunur.



Şekil 3.19. Kapasitif ve endüktif güçlerin vektörel gösterilmesi (Aydın, 2015)

Şekil 3.19.'da kapasitif (Q_c) ve endüktif (Q_L) reaktif güçler vektörel olarak gösterilmiştir. Burada endüktif reaktif güç, kapasitif reaktif gücün 180° gerisindedir. Söz konusu güçler aynı doğrultuda ama birbirilerine zıttırlar. İki gücün vektörel toplamında kapasitif gücün endüktif gücü kompanze ettiği anlaşılmaktadır. Bu bağlamda kondansatörlerin reaktif güç kompanzasyonunda kullanılacağını göstermektedir.

Üç fazlı sistemlerde kondansatörler şekil 3.20.'deki gibi sisteme üçgen ya da yıldız olarak bağlanabilirler.

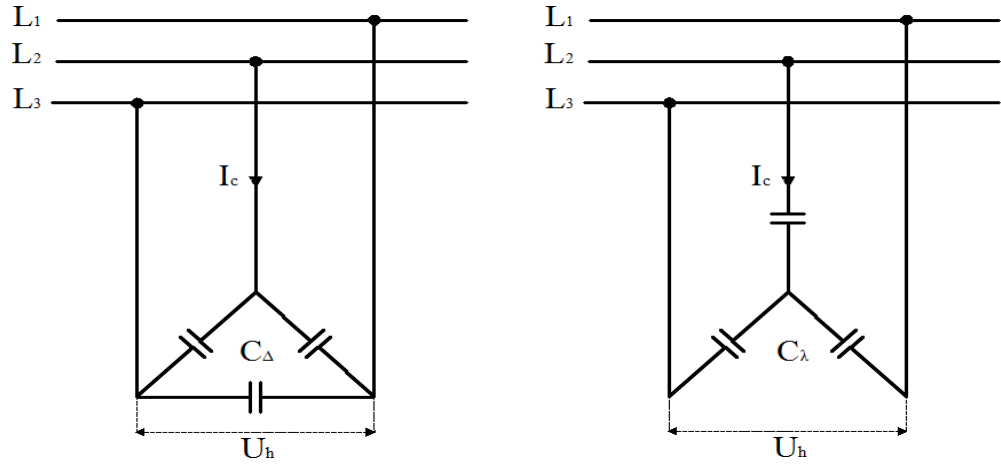
Üçgen bağlama için;

$$Q_c = 3 \cdot U_h^2 \cdot \omega \cdot C_\Delta \cdot 10^{-3} = \sqrt{3} \cdot U_h \cdot I_c \cdot 10^{-3} = \frac{I_c^2}{\omega \cdot C_\Delta} \cdot 10^{-3} \quad (3.30)$$

Yıldız bağlama için;

$$Q_c = U_h^2 \cdot \omega \cdot C_\lambda \cdot 10^{-3} = \sqrt{3} \cdot U_h \cdot I_c \cdot 10^{-3} = 3 \cdot \frac{I_c^2}{\omega \cdot C_\lambda} \cdot 10^{-3} \quad (3.31)$$

Burada U_h hatlar arası gerilimi, I_c kapasitif akımı ifade etmektedir.



Şekil 3.20. Kondansatörlerin üçgen ve yıldız bağlantı şeması (Aydın, 2015)

Her iki sistem tasarımında reaktif güç eşit kabul edilirse;

$$C_{\lambda} = 3 \cdot C_{\Delta} \quad (3.32)$$

olur. Buna göre kondansatörler yıldız bağlanırsa, kondansatörlerin kapasitesi üçgen bağlamaya göre üç kat fazladır. Kondansatörler yıldız bağlanırsa uçlarına faz-nötr gerilimi uygulanır. Üçgen bağlanması durumunda uçlarına $\sqrt{3}$ kadar daha fazla gerilim uygulanması gerekir. Ekonomik olarak üçgen bağlantı, yıldız bağlantıya göre daha 1/3 oranında daha ucuz olduğundan alçak gerilim tesislerinde kondansatörlerin üçgen bağlanmaları tercih edilir (Aydın, 2015).

3.3.8. Şönt Reaktörler

Şönt reaktörler kapasitif yüklerin fazla olduğu işletmelerde, endüktif etki yaratarak, kapasitif enerjinin karşılanmasını sağlayan statik kompanzasyon elemanlarıdır. Elektronik devre elemanların yaygınlaşmasıyla işletmelerde kapasitif yüklerde artış meydana gelmiştir. Şönt reaktörler kapasitif reaktif enerjinin fazla olduğu sistemlerde, sisteme endüktif yük basarak kompanze yaparlar.

Güç elektroniği cihazlarında kullanımın artmasıyla birlikte sadece kondansatörlerle yapılan statik kompanzasyon yetersiz kalmaktadır. Burada kapasitif reaktif enerjiyi kompanze etmek için reaktörlerin devreye girmesi gerekmektedir.

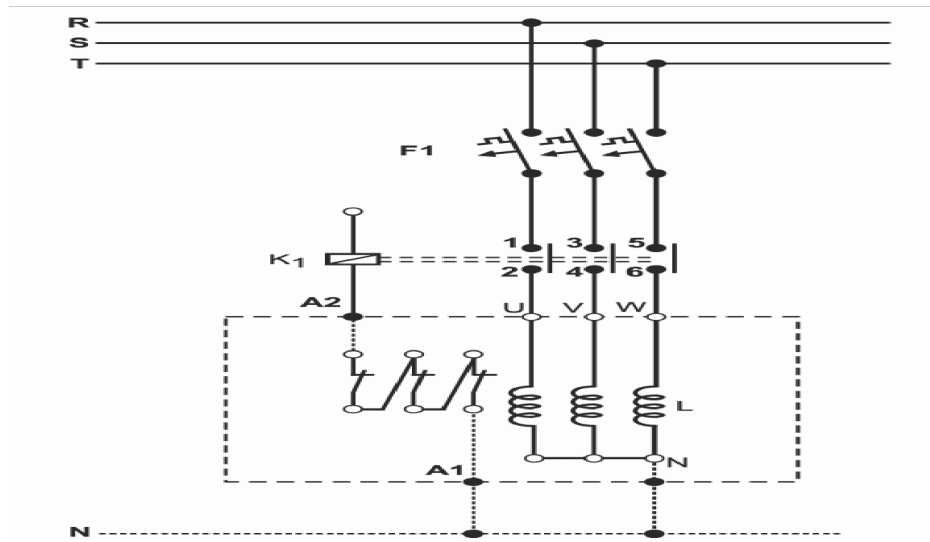
Şönt reaktörler, sistemdeki kapasitif reaktif enerjiyi karşılayabilmek için endüktif yük olarak sisteme bağlanırlar. Böylelikle sistem limitleri EPDK tarafından belirlenen sınırlar içerisinde tutulabilmektedir (Entes, 2019)

Şönt reaktörler;

- Bankalar,
- Radyolink istasyonları,
- DC sürücülerin fazla olduğu işletmeler,
- Uzun yeraltı kablo hatları,
- Demiryolu tesisleri,
- Yarı iletken bulunduran güç elektroniği cihazları,
- Florasan ve LED ürünlerinin kullanımın fazla olduğu yerler,
- Hastaneler,
- Mağazalar vb,

yerlerde kullanılırlar.

Şekil 3.21.'de 3 fazlı şönt reaktörün bağlantı şeması verilmiştir.



Şekil 3.21. Üç fazlı şönt reaktör bağlantı şeması (Entes, 2019)

Şönt reaktörler sisteme herhangi bir noktadan bağlanabilirler fakat sistemde ihtiyaç olmadığı zaman devrede çıkmaları gerektiğinden reaktif güç kontrol rölesiyle beraber bağlanmaları daha doğru bir bağlantı türüdür.



3.4. YSA'nın Tasarımı

Bu tez çalışmasında bir spor merkezine ait sistemde tesisin toplam kurulu gücü 60 kW'dır. Tesiste 12 kademeli SVC röle, 12 kademe konsansatör grubu, her fazda ayrı ayrı olmak üzere 3 kVAR monofaze şönt reaktör ve toplam 51.50 kVAR kondansatör bulunmaktadır.

Sisteme ait 350 adet veri 27.12.2018 ile 10.02.2019 tarihleri arasında gerçek sistemden Endüstri 4.0 tabanlı cihazlar yardımıyla toplanmıştır. Sistemden her faz için aktif güç, endüktif güç, kapasitif güç, akım, gerilim, sistemin o anki $\cos \varphi$ değeri, kompanzasyon sonrası $\cos \varphi$ değeri, yüzde harmonik oranları, sistemin o anki endüktif yüzde oranı ile kapasitif yüzde oranı, kondansatör miktarı ile şönt reaktör kullanım oranları alınmıştır. Toplanan veriler veritabanına kaydedilmiştir. Yapay zekanın eğitimi sırasında veriler veritabanından alınarak yapay zeka eğitilmiştir. Tasarlanan arayüzdeki veriler veritabanından alınmıştır.

Şönt reaktör akımları, sistemdeki reaktör kullanım oranları baz alınarak reaktörün çektiği akımlar o anki sistem değerleri üzerinden hesaplanmıştır.

Her fazda ayrı ayrı olmak üzere toplam kondansatör gücünü tahmin etmek için YSA eğitilirken Çizelge 3.2.'de verilen giriş çıkış verileri kullanılmıştır.

Çizelge 3.2. Toplam kondansatör gücünü tahmin etmek için kullanılan giriş - çıkış verileri

Giriş Verileri	Çıkış Verisi
Akım Değeri	Toplam Kondansatör Gücü
Gerilim Değeri	
Aktif Güç Değeri	
Endüktif Güç Değeri	
Kapasitif Güç Değeri	
Sistemin O Anki Cos φ Değeri	

Çizelge 3.2.(devam)

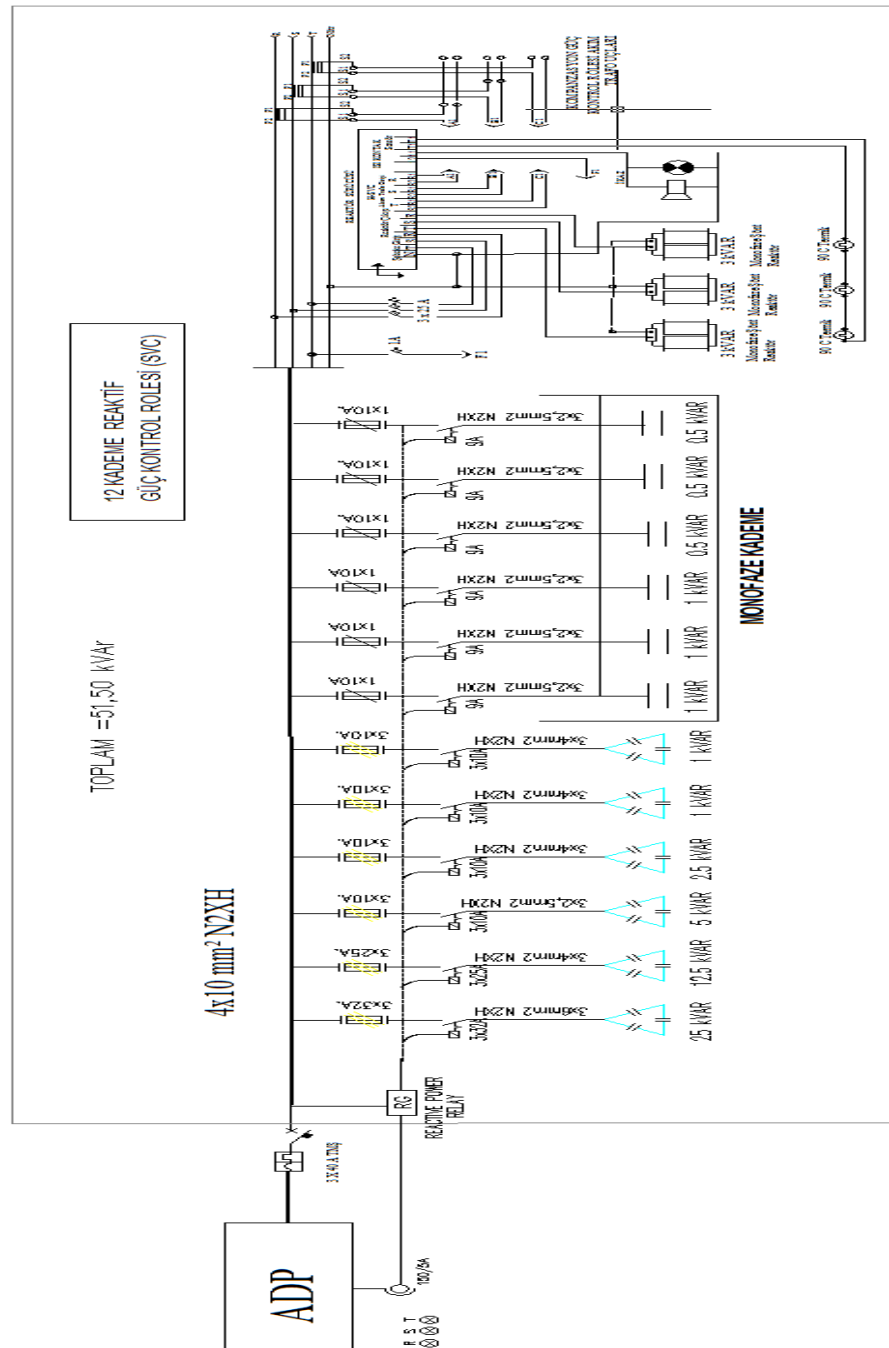
Sistemin O Anki Güç Açısı	Toplam Kondansatör Gücü
Sistemin Kompanzasyon Sonrası $\cos\phi$ Değeri	
Sistemin Kompanzasyon Sonrası Güç Açısı	
Yüzde Harmonik Oranı	
Endüktif Yüzde Oranı	
Kapasitif Yüzde Oranı	
Reaktör Kullanım Oranı	
Reaktör Akımı	

Her fazda ayrı ayrı olmak üzere şönt reaktör akımlarını tahmin etmek için YSA eğitilirken Çizelge 3.3.'te verilen giriş çıkış verileri kullanılmıştır.

Çizelge 3.3. Şönt reaktör akımlarını tahmin etmek için kullanılan giriş - çıkış verileri

Giriş Verileri	Çıkış Verisi
Akım Değeri	Şönt Reaktör Akımı
Gerilim Değeri	
Aktif Güç Değeri	
Endüktif Güç Değeri	
Kapasitif Güç Değeri	
Sistemin O Anki $\cos\phi$ Değeri	
Sistemin o Anki Güç Açısı	
Sistemin Kompanzasyon Sonrası $\cos\phi$ Değeri	
Sistemin Kompanzasyon Sonrası Güç Açısı	
Yüzde Harmonik Oranı	
Endüktif Yüzde Oranı	
Kapasitif Yüzde Oranı	
Reaktör Kullanım Oranı	
Sistemde O An Devrede Bulunan Kondansatör Gücü	

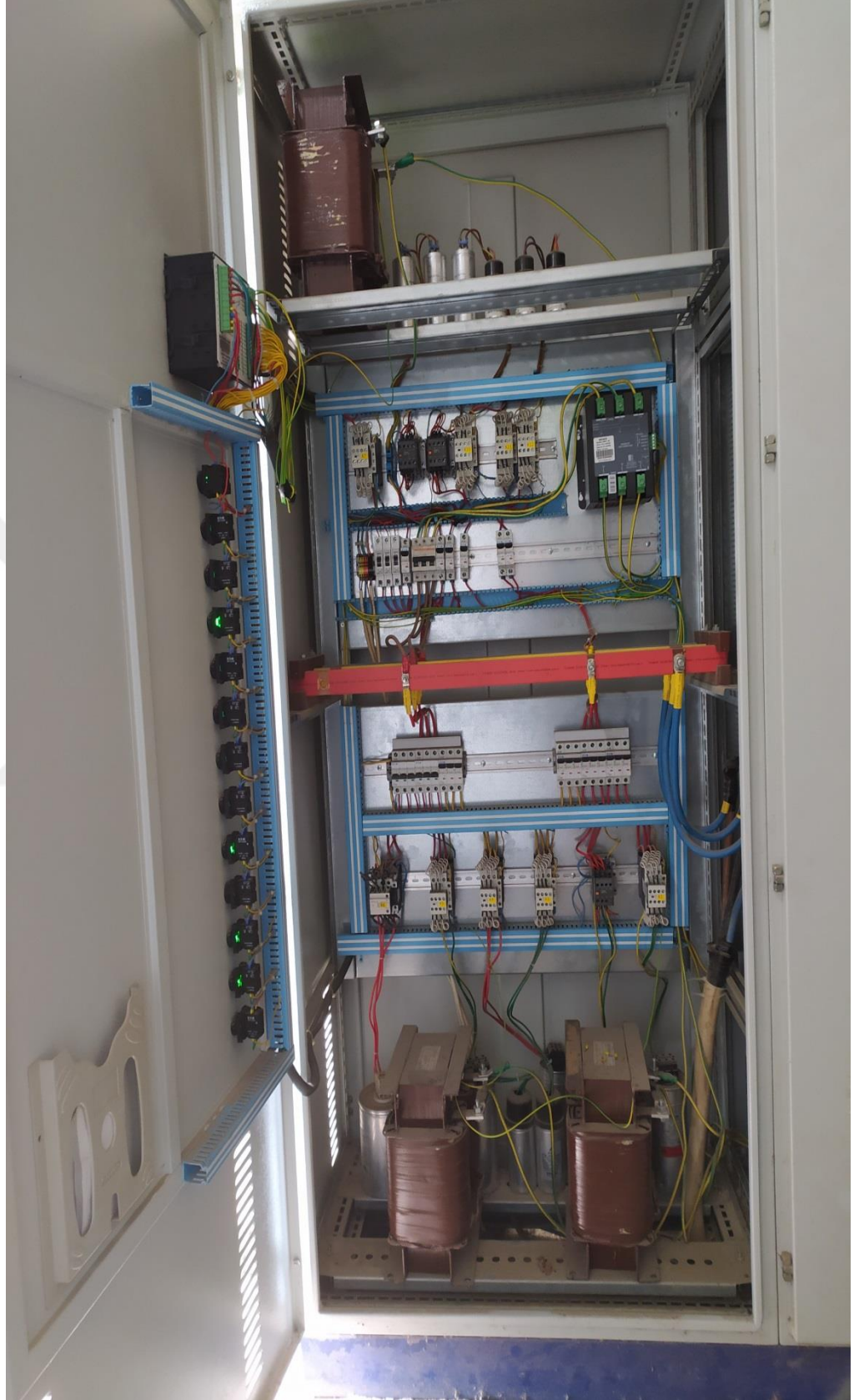
Kompanzasyon panosunun genel hat şeması Şekil 3.22.'de verilmiştir. Kompanzasyon panosunun genel görüntüsü Şekil 3.23 ile Şekil 3.24.'te verilmiştir.



Şekil 3.22. Kompanzasyon panosu genel hat şeması



Şekil 3.23. Kompanzasyon panosu ön kapağı



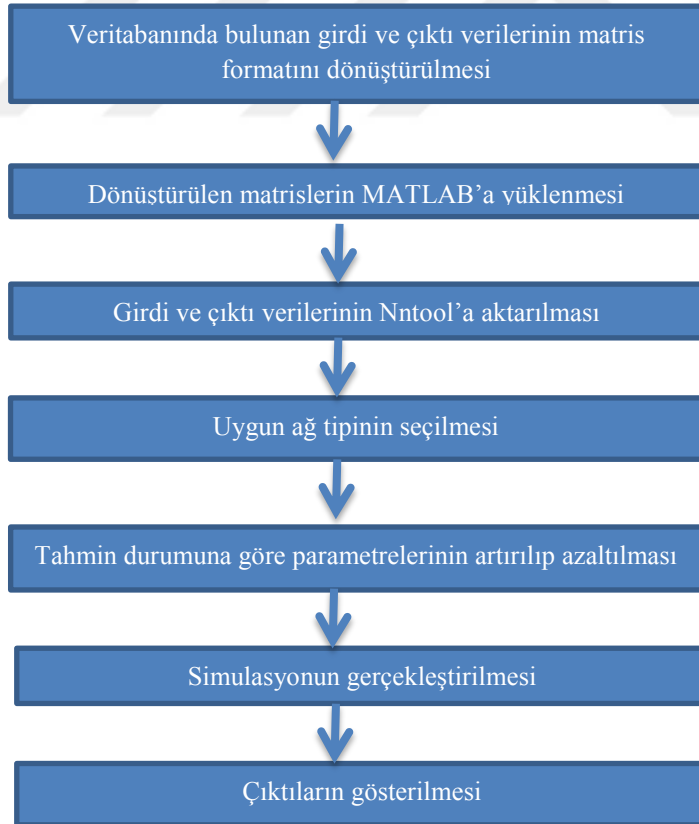
Şekil 3.24. Kompanzasyon panosu iç görüntüsü

YSA'nın başarılı olması kullanılacak olan yöntem ve verilerle yakından ilgilidir.

YSA'nın geliştirilmesi aşamasında aşağıdaki faktörler önemlidir.

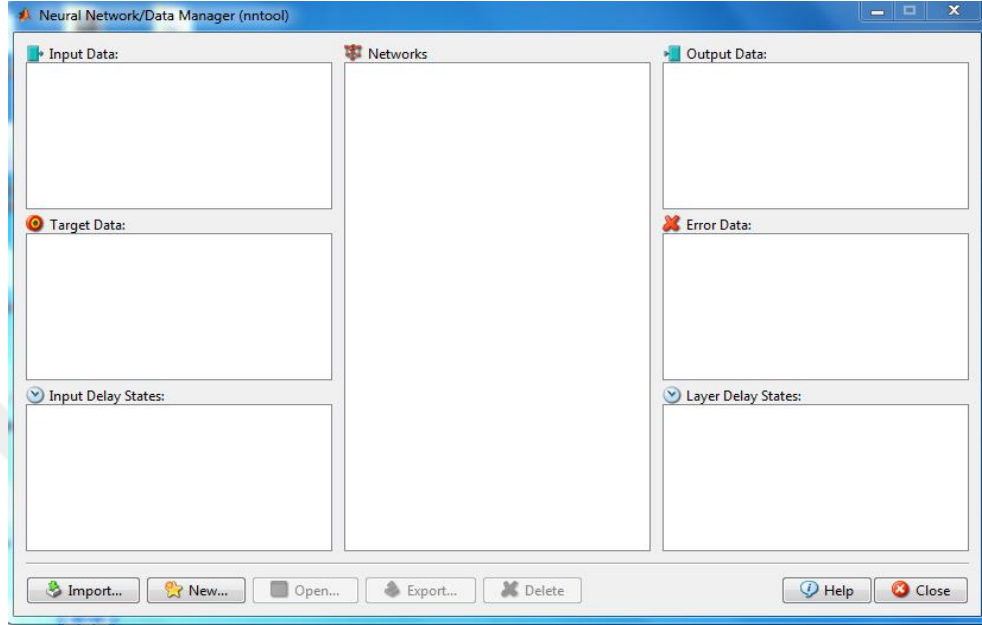
- Ağ mimarisi seçilerek yapı özelliklerinin belirlenmesi,
- Gizli katman sayısının belirlenmesi,
- Gizli katman ile çıktı katmanı arasında bulunan aktivasyon formülünün belirlenmesi,
- Veri dönüşümleri ve normalizasyon yöntemleri,
- Nöron fonksiyon karakteristiklerinin ve sayısının belirlenmesi,
- Öğrenme algoritmaları ile parametrelerinin belirlenmesi,
- Eğitim ve test verilerinin oluşturulması.

Yapılan çalışmaya ait iş akışı Şekil 3.25.'te verilmiştir.



Şekil 3.25. İş akış şeması

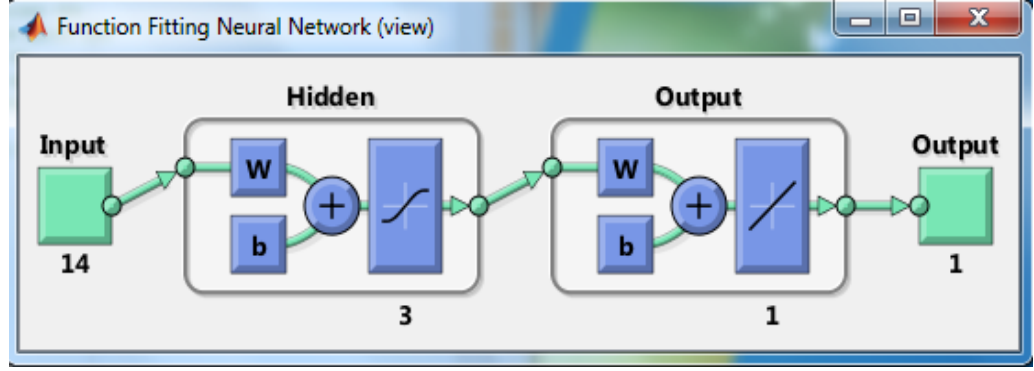
YSA tarafından sistemdeki her fazdaki kondansatör güçleri ve şönt reaktör akımlarını ayrı ayrı tahmin etmek için MATLAB NNtool kullanılmıştır. MATLAB NNtool genel görüntüsü Şekil 3.26.'da gösterilmiştir.



Şekil 3.26. MATLAB NNtool genel görüntüsü.

Yapay sinir ağlarının tasarımında modeller arası performans ve karakteristik farklılıklar olduğundan uygulamaya en elverişli olan seçilir. Bu tez çalışmasında tek katmanlı yapay sinir ağı modeli kullanılmıştır.

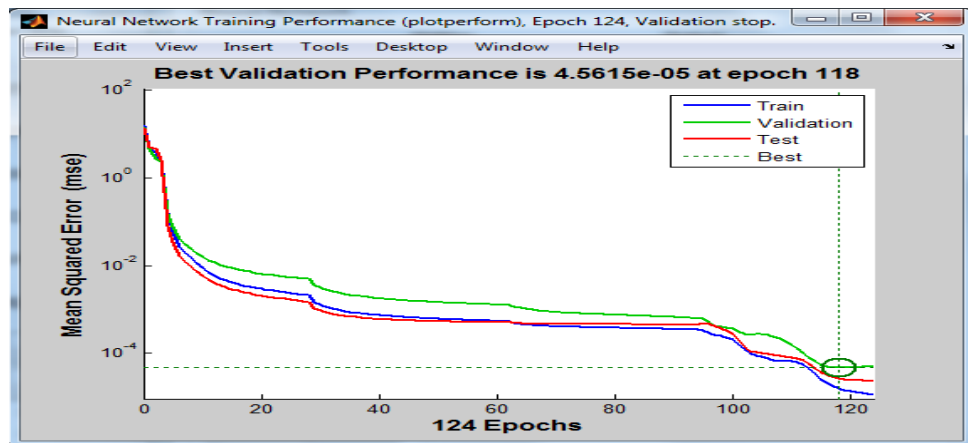
YSA'nın tasarımının diğer bir adımı ise ağıdaki katman sayısı ile nöron sayısını belirlemektir. Bu çalışmada giriş, gizli ve çıktı katmanları olmak üzere 3 katman kullanılmıştır. 3 adet nöron ve logaritmik sigmoid fonksiyonu gizli katmanda kullanılmıştır. Çıkış katmanında ise 1 adet nöron ile doğrusal aktivasyon fonksiyonu tercih edilmiştir. Az sayıda nöron kullanılmasının sebebi YSA'nın öğrenme yaparken ezber yapmasının önleyip genelleme yapmasını sağlamaktır (Şekil 3.27.).



Şekil 3.27. Oluşturulan yapay sinir ağı

Bu çalışmada Levenberg - Marquardt öğrenme algoritması ile ileri beslemeli danışmanlı ağ yapısı ile tercih edilmiştir.

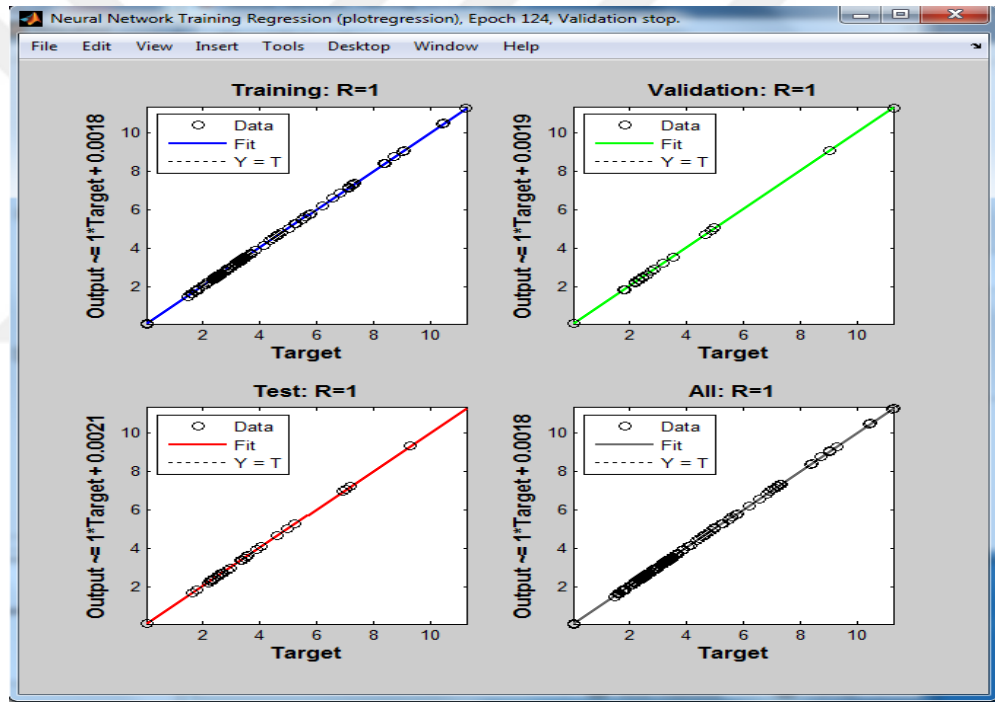
Yapay sinir ağlarının eğitilmesi, nöronların bağlantı ağırlıklarının belirlenmesi anlamına gelmektedir. Yapay sinir ağlarının öğrenmesi için eğitim ve test verileri gerekmektedir. Eldeki verilerin %70 eğitim verisi, %15 test verisi ve %15 doğrulama verisi olarak kullanılmıştır. Eğitimi tamamlanan yapay sinir ağlarının hata oranının kabul edilebilir bir seviyede olması beklenir. Test verileriyle yapay sinir ağlarının çıktıları karşılaştırılarak YSA'nın yeterli öğrenme yapıp yapmadığı gözlenmiştir. İstenilen performans elde edildiğinde YSA modeli kullanılabilir. Yapılan modele ait eğitim ve test aşamalarına ait performans Şekil 3.28'de gösterilmiştir.



Şekil 3.28. Yapay sinir ağı performans grafiği

Şekil 3.28.'deki grafik incelendiğinde mavi renkle eğitim seti, yeşil renkle doğrulama seti ve kırmızı renkle test seti çizilmiştir. Grafikten de anlaşılabacağı gibi 124. iterasyon sonucunda sıfıra yakın hata değerine inmiştir. MSE değeri 0.01'in altına indiği anda eğitimin sona erdiği ve sonuçların kabul edilebilecek seviyede olduğunu göstermektedir.

Yapılan tahmin sonuçları ile gerçek değerler arasındaki ilişkini regresyon sonuçlarına bakarak değerlendirilir. R korelasyon değerinin 1'e yaklaşması girdi ve çıktılar arasındaki uyumu göstermektedir. Şekil 3.29.'da çıktı varyasyonları arasındaki uyum gösterilmiştir.



Şekil 3.29. Yapay sinir ağı modelinin regresyon grafikleri

YSA'nın iyi tahmin etmesi için modelleme, veriler gibi birçok kritere bakılır. YSA'da bir olay için kesin kullanılacak olan eğitim algoritması olmadığı için deneme yanılma yoluyla algoritma geliştirilir. YSA'da doğruluk kriteri, gerçek değer ile öngörülen değer arasındaki farktır. Bu farka tahmin hatası denir. Tahmin hatasına göre sistem modellenir.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

MATLAB® & SIMULINK® programında 4 bölümlü tasarlanan kullanıcı arayüzü Şekil 4.1.'de verilmiştir.

The screenshot shows a MATLAB/Simulink user interface titled 'eem_mbd'. It is divided into four main panels:

- Gerçek Veriler (Real Data):** Contains input fields for U(V), I(A), P(kW), COS, kVAR, and Şönt(A) for three phases (R, S, T). Below these are buttons for 'Oku' (Read), 'Temizle' (Clear), 'Hesapla' (Calculate), and 'Otomatik' (Automatic). At the bottom, there are radio buttons for 'Kompanzasyon Yokken' (No Compensation), 'Statik Kompanzasyon' (Static Compensation), and 'YSA' (YSA).
- Kompanzasyon Yokken (No Compensation):** Contains input fields for 'Mevcut COS' (Existing COS), 'Aktif Güç(W)' (Active Power), 'Reaktif Güç(VAR)' (Reactive Power), and 'Görünür Güç(VA)' (Apparent Power).
- Statik Kompanzasyon (Static Compensation):** Contains input fields for 'Hedef Cos' (Target COS) and 'kVAR' for three phases (R, S, T).
- YSA (YSA):** Contains input fields for 'kVAR' and 'Şönt(A)' for three phases (R, S, T). Above these fields are colored boxes labeled 1 through 12, with R, S, and T phases indicated.

Şekil 4.1. Kullanıcı arayüzü

4.1. Gerçek Veriler Kullanıcı Arayüzü

Kullanıcı arayüzünün gerçek veriler kısmında, mevcut sistemden elde edilen verilen veritabanından okunmaktadır. Oku butonuna basıldığı anda verilerin bulunduğu veritabanının ilk veri satırı okunmaktadır. Temizle butonu arayüzdeki tüm verileri silmektedir. Hesapla butonuyla veritabanından alınan veriler yardımıyla gerekli hesaplamalar yapılmaktadır. Otomatik butonunda ise sistem gerçek zamanlı çalışıyormuş gibi üç saniye aralıklarla veritabanından bir alttaki satırı okuyarak gerekli hesaplamalar ve tahminler yapılmaktadır. Sekmeler yardımıyla hesaplanmasını istenilen bölüm seçilerek gerekli hesaplar yapılmaktadır. Gerçek veriler arayüzü Şekil 4.2.'de verilmiştir.

Şekil 4.2. Gerçek veriler kullanıcı arayüzü

4.2. Kompanzasyon Yokken Kullanıcı Arayüzü

Bu arayüze geçiş yapmak için gerçek veriler arayüzünden “Kompanzasyon Yokken” sekmesi seçilir. Verileri okumak için oku butonuna basılarak veritabanından mevcut sisteme ait gerçek veri değerleri alınır.

Tesiste kompanzasyon sistemi olmadığı varsayılarak tüketilen aktif, reaktif ve görünür güçler bu arayüzde hesaplanır. Tesisin mevcut $\cos \varphi = 0.7$ olduğu varsayılarak gerekli hesaplamalar bu bölümde yapılmıştır. Tesisin $\cos \varphi$ değeri manuel olarak girilerek farklı değerler için hesaplama yapılabilir. Hesaplamalar trifaze, 380 V gerilim düzeyinden hesaplanmıştır. Kompanzasyon yokken arayüzü Şekil 4.3.’te verilmiştir.

	R	S	T
U(V)	210	212	214
I(A)	43.2	40.4	39.2
P(kW)	9.72	9.19	9
COS	0.99	-1	1
kVAR	5.52	4.42	4.49
Şönt(A)	0.07452	0.0752	0.07056

Mevcut COS	0.7
Aktif Güç(W)	56577.1
Reaktif Güç(VAR)	57720.2
Görünür Güç(VA)	80824.4

Şekil 4.3. Kompanzasyon yokken kullanıcı arayüzü

Şekildeki gerçek verilere göre sistemde aktif güç tüketimi 56 577.1 W, reaktif güç tüketimi 57 720.2 VAR ve görünür güç tüketimi ise 80 824.4 VA'dır. Reaktif tüketim aktif tüketimin %102 oranında olduğu görülmektedir. Dağıtım firmaları endüktif oranı en fazla %20, kapasitif tüketimi ise en fazla %15 oranında belirlemiştir. Bu durumda tesis reaktif tüketim bedeli ödeyecektir.

4.3. Statik Kompanzasyon Kullanıcı Arayüzü

Statik kompanzasyon arayüzüne geçiş yapmak için gerçek veriler arayüzünden "Statik Kompanzasyon" sekmesi seçilir. Verileri okumak için oku butonuna basılarak veritabanından mevcut sisteme ait gerçek veri değerleri alınır.

Klasik kompanzasyon sistemlerinde reaktif güç rölesi, kontaktör, sigorta, ve kondansatör grupları bulunmaktadır. Bu bölümde klasik kompanzasyon yapılması durumunda gerekli olan kondansatör gücü hesaplanmıştır. İdealde olması gereken kondansatör güçleri, mevcut sistemlerde tam anlamıyla kompanze yapmamaktadır. Reaktif güç kontrol rölesi alınan kademenin büyük olduğuna karar verirse sürekli anahtarlama yaparak kondansatör gruplarını devreye alıp çıkartır. Bu durum sistemin

kararsız olmasına neden olur. Bunu engellemek için kompanzasyon sistemine şönt reaktörler bağlanır. Röle büyük gelen kondansatör grubunu devreden çıkarmak yerine sisteme şönt reaktör sürerek dengeli kompanzasyon yapılmasını sağlar. Statik kompanzasyon arayüzü Şekil 4.4.'te verilmiştir.

	R	S	T
U(V)	210	210	217
I(A)	46.8	55.6	40.4
P(kW)	9.93	10.32	9.45
COS	0.99	1	0.99
KVAR	5.75	4.26	4.78
Şönt(A)	0.07452	0.07594	0.06958

Oku Temizle

Hesapla Otomatik

☐ Kompanzasyon Yokken
☒ Statik Kompanzasyon
☐ YSA

	R	S	T
Hedef Cos	0.99		
kVAR	8.71568	9.05799	8.29438

eem_mbd

Şekil 4.4. Statik kompanzasyon kullanıcı arayüzü

Şekil 4.4. incelendiğinde mevcut kondansatör güçleri ile hesaplanan kondansatör güçleri arasında fark olduğu görülmektedir. İdealde R fazı için 210 V ve 46.8 A akım değeri için 8,715 kVAR, S fazı için 210 V ve 55.6 A akım değeri için 9.057 kVAR ve T fazı için 217 v ve 40.4 A akım değeri için 8.294 kVAR kondansatör gücü hesaplanmıştır. Bu durumda sisteme 30 kVAR kondansatör bağlanması gerekmektedir. 30 kVAR kondansatör her faza 10 kVAR kondansatör olarak gücü bağlanmış olacaktır. Böyle olunca reaktif güç kontrol rölesi bu kondansatörün büyük olduğuna karar verecek ve bu kademeyi devreden çıkaracaktır. Sürekli olarak reaktif güç kontrol rölesi bu işlemi tekrarlayacaktır. Ancak gerçek zamanlı sistemde bize gerekli olan kondansatör gücü R fazı için 5,75 kVAR, S fazı için 4,26 kVAR ve T fazı 4,78 kVAR'dır. Sisteme 15 kVAR'lık kondansatör bağlanması durumunda her fazda 5 kVAR kondansatör gücü olur. R fazı için eksik olan 0,75 kVAR kondansatör gücü monofaze kondansatörler tarafından karşılanır. S

fazı için fazla olan 0,74 kVAR kondansatör gücü ile T fazı için fazla olan 0,22 kVAR kondansatör gücü şönt reaktörler tarafından egale edilerek sağlıklı bir kompanzasyon yapılmış olacaktır.

4.4. Yapay Zeka Kullanıcı Arayüzü

Yapay zeka arayüzüne geçiş yapmak için gerçek veriler arayüzünden “Yapay Zeka” sekmesi seçilir. Verileri okumak için oku butonuna basılarak veri tabanından mevcut sisteme ait gerçek veri değerleri alınır.

Şekil 4.5.'te yapay zekaya ait kullanıcı arayüzü verilmiştir.

	R	S	T
U(V)	210	211	215
I(A)	43.6	41.2	40.4
P(kW)	9.9	10.02	7.12
COS	0.99	-0.99	-1
kVAR	5.83	4.73	4.84
Şönt(A)	0.07452	0.07556	0.07023

	R	S	T
kVAR	5.83833	4.73252	4.85055
Şönt(A)	0.074596	0.07602	0.070191

Şekil 4.5. Yapay zeka kullanıcı arayüzü

Yapay zeka kullanıcı arayüzü incelendiğinde kompanzasyon için gerekli olan kondansatör güçleri ile şönt reaktör akımları tahmin edilmiştir. R fazı için gerçek zamanlı sistemde 5.83 kVAR kondansatör gücü ile 0.07452 A şönt reaktör akımı kullanılmıştır. Yapay zeka tarafından aynı faz için 5.83833 kVAR kondansatör gücü ile 0,074596 A şönt reaktör akımı tahmin edilmiştir. %100'e yakın tahmin oranıyla gayet sağlıklı bir tahmin yapıldığı görülmüştür. Diğer fazlar için de aynı durum

geçerlidir. Led ışıkları reaktif güç kontrol rölesinde olduğu gibi çekilen kademe görseli olarak konulmuştur. Devrede olan kademeler yeşil renkte göstermiştir.



5. SONUÇLAR ve ÖNERİLER

Bu tez çalışmasında gerçek sistemden alınmış veriler yardımıyla statik ve yapay zekalı reaktif güç kompanzasyonu için kararlı çalışabilecek bir simülasyon gerçekleştirilerek başarıyla uygulanmıştır. Çizelge 5.1.'de gerçek zamanlı sistem parametreleri ile yapay zeka tarafından tahmin edilen değerler verilmiştir. Tahmin edilen değerler %100'e yakındır.

Çizelge 5.1. Gerçek zamanlı sistem parametreleri ile tahmin edilen değerlerin karşılaştırılması.

	Gerçek Değer	Sinir Ağı Çıktısı	Mutlak Hata
R fazı İçin Kondansatör Gücü	5.9100	5.9140	0.0040
R fazı İçin Şönt Reaktör Akımı	0.073474	0.073410	0.000064
S fazı İçin Kondansatör Gücü	4.7700	4.7725	0.0025
S fazı İçin Şönt Reaktör Akımı	1.407790	1.407740	0.000005
T fazı İçin Kondansatör Gücü	4.9100	4.9171	0.0071
T fazı İçin Şönt Reaktör Akımı	3.686390	3.686090	0.000300

Çizelge 5.1. incelendiğinde yapılan tahminler ile gerçek değerler arasındaki mutlak hata incelendiğinde, tasarlanan sistemin güvenilir olduğunu ve kullanılabileceğini göstermektedir.

Bu tezde uygulamalar elle ya da otomatik olarak gerçekleştirilmiştir. Otomatik için sistem gerçek zamanlı gibi davranarak her 3 saniyede veriler, veritabanından okunarak kompanzasyon yapılmıştır. Elle yapılan kompanzasyon için; farklı yükler altında sistemin nasıl cevap vereceği karşılaştırmalı olarak gözlenmiştir. Veriler gerçek sistemden alındığı için yazılımın doğru çalışıp çalışmadığı da tespit edilmiştir.

Sistemin farklı güçler için nasıl yüklendiğini karşılaştırmalı olarak görmek için genel amaçlı bir yazılım oluşturulmuştur. Konu ilgili karşılaştırmalı bir deney ortamı

olmadığından simülasyonun önemini bize göstermiştir. Hazırlanan arayüz sayesinde tercih edilen kompanzasyon türüne göre farklı yükler altında sistemin nasıl tepki vereceği etkin bir şekilde karşılaştırmalı olarak gerçekleştirilmiştir.

Tasarlanan arayüzde yazılım için herhangi bir donanıma ihtiyaç duymaması bir avantajdır. Herhangi bir maliyet getirisi yoktur. Herhangi bir yazılım maliyetinin olmaması da diğer bir avantajdır. Yapılan yazılım istenilen her bilgisayarda kullanılabilir. Kişi sınırlaması yoktur. Okullarda verilen teorik bilginin uygulamalı olarak gerçekleştirilmesi için bu yazılımın kullanılmasının uygun olacağı öngörülmektedir.

Senkron motor veya kondansatör kullanılarak reaktif güç kompanzasyonu da yapılabilmektedir. Mevcut sistemde hazır çalışan senkron motor varsa senkron motorla kompanzasyon yapılabilir. Mevcut senkron motorun anma uyartım akımı yettiği ölçüde senkron motor ile güç katsayısı düzeltilebilir. Çalışan mevcut motor yoksa senkron motorla kompanzasyon yapmak ek maliyet getireceği için ekonomik olmayacaktır. Bu durumda statik kompanzasyon yapılmalıdır. Statik kompanzasyonda küçük güçlü kondansatörler ve şönt reaktörler tercih edilerek daha hassas kompanzasyon yapılabilir. Statik kompanzasyonda kondansatör kayıpları, bakımları ve tesis maliyetlerin azlığı nedeniyle senkron motora göre daha avantajlıdır.

KAYNAKLAR

- ADİYAMAN, F., 2007. Talep Tahmininde Yapay Sinir Ağlarının Kullanılması. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 109s.
- ALÇIN, S., 2016. Üretim İçin Yeni Bir İzlek: Sanayi 4.0. *Journal of Life Economics*, 3(2):19-30.
- AYDIN, O., 2015. Senkron Motor ile Güç Katsayısı Kontrolünde Klasik Sistemler ile Uzman Sistemlerin Karşılaştırılması. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 138s.
- AYDIN, Ö. F., 2019. Yapay Sinir Ağları ile Enflasyon Tahmini. Beykent Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 70s.
- BARAN, L., 2010. Üç Tabanlı Asenkron Motorun Güç Katsayısının PLC ile Kontrolü ve İzlenmesi. Gazi Üniversitesi, Fen Bilimler Enstitüsü, Yüksek Lisans Tezi, Ankara, 76s.
- BAŞ, Z., 2014. Güç Sistemlerinde Reaktif Güç Kompanzasyonu ve Harmonik Analizi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya, 86s.
- BAYINDIR, R., GÖRGÜN, A., 2009. PIC Tabanlı Bir PI Denetleyici ile Senkron Motor Kullanılarak Bir Kompanzator Uygulaması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 15(1):87-99.
- BAYRAM, M., 2000. Kuvvetli Akım Tesislerinde Reaktif Güç Kompanzasyonu. Birsan Yayınevi, İstanbul, 254s.
- BLANCHET, M., RINN, T., THADEN, G., THIEULLOY, G., 2014. Industry 4.0. The new industrial revolution. How Europe will succeed. Hg V Roland Berg. Strategy Consult. GmbH Münch. Abgerufen Am 1105 2014 Unter. <http://www.Rolandberger.Com/med.0403/Pdf>.
- BOSTANCI, A., 2008. Statik Var Kompanzasyon Sistemi Tasarımı ve Uygulaması, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 67s.
- BRADLEY, J. M. ve ATKINS, E. M., 2015. Optimization and Control of Cyber-Physical Vehicle Systems, *Sensors*, Number:15, ss. 23020-23049.
- ÇELİK, B., 2008. Yapay Sinir Ağları Metodolojisi ile Zaman Serisi Analizi: Teori ve Uygulama, Marmara Üniversitesi Sosyal Bilimler Enstitüsü ,Yüksek Lisans Tezi, İstanbul, 241s.
- DE TREVILLE, S., SHAPIRO, R.D., HAMERI, A.-P., 2004. From Supply Chain to Demand Chain: The Role of Lead Time Reduction in Improving Demand Chain Performance. *J. Oper. Manag.* 21, 613–627.
- DOĞRUL, Y., 2018. Bulanık Mantık Denetimli Alçak Gerilim Statik VAR Kompanzasyon Sisteminin Modellenmesi ve PLC ile Gerçeklenmesi. Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kütahya, 112s.
- EFENDİGİL, T., ÖNÜT, S., ve KAHRAMAN, C., 2009. A Decision Support System For Demand Forecasting with Artificial Neural Networks And Neuro-

- Fuzzy Models: A Comparative Analysis, Expert Systems with Applications, 36: 6697-6707.
- ELMAS, Ç., 2003. Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama). Seçkin Yayıncılık, Ankara, 192s.
- ERSAMUT, R., 2009. Statik VAr Kompanzasyonu Sistemlerinin İncelenmesi ve Karşılaştırılması. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 102s.
- GEISBERGER, E., BROY, M., 2012, Agenda CPS, Integrierte Forschungsagenda Cyber-Physical Systems (acatech Studie). acatech –Deutsche Akademie der Technikwissenschaften, Munich.
- GÖKCAN, A. O., 2016. Reaktif Güç Kompanzasyonu Eğitimi İçin Sanal Bir Laboratuvarın Oluşturulması. Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Elazığ, 64s.
- GÖRGÜN, A., 2009. Yapay Sinir Ağları Denetimli Senkron Motor ile Reaktif Güç Kompanzasyonu. Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 120s.
- HATAY, Ö. F., 2010. Bulanık Mantık Kontrollü Ev Tipi Taşınabilir Kompanzasyon Sistemi Tasarımı, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya, 93s
- HOPFIED, J.J., 1982. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proceedings of National Academy of Science, 79: 2554-2558.
- HAYKIN, S., 1999. Neural Networks: A Comprehensive Foundation, Prentice-Hall, Second Edition, New Jersey.
- HIRSCH-KREINSEN, H. ve WEYER J., 2014. Wandel von Produktionsarbeit – Industrie 4.0. Soziologisches Arbeitspapier 38, TU Dortmund.
- https://entes.com.tr/blog_detay.asp?blogID=34 Erişim Tarihi: 10 Aralık 2019.
- HU, F. ve diğerleri., 2016. Robust Cyber-Physical Systems: Concept, Models and Implementation, Future Generation Computer Systems, sayı: 56, ss. 449-475.
- KAGERMANN, H., W. WAHLSTER ve J. HELBIG, diğ., 2008. Recommendations for Implementing the Strategic Initiative Industrie 4.0: Final Report of the Industrie 4.0 Working Group Zukunftsprojekt Industrie 4.0.
- KAGERMANN, H ve diğerleri, 2015. Abschotten ist keine Alternative. In: VDI Nachrichten, Sayı 16.
- KAPLAN, O., 2005. PIC Denetimli Kondansatör Grupları ile Güç Katsayısının Düzeltilmesi. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, , Ankara, 58s.
- KARADEMİR, Ö., 2014. Transformatör Üretiminde Üretim Zamanlarının Yapay Sinir Ağları ile Tahmini İçin Bir Çalışma. Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Balıkesir, 63s.
- KEÇECİOĞLU, Ö. F., 2012. Tristör Kontrollü Reaktif Güç Kompanzasyon Sistemi Denetimi ve Uygulaması. Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kahramanmaraş, 98s.
- KOCABAŞ, E., 2006. Reaktif Güç Kompanzasyonu ve Simülasyonu. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 186s.
- LEE, J ve diğerleri, 2015. Industrial Big Data Analytics and Cyber-Physical Systems for Future Maintenance & Service Innovation, Procedia CIRP, Sayı: 38, ss. 3-7.

- ÖZASLAN, S., 1997. Statik VAr Sistemleri ile Reaktif Güç Kompanzasyonu. Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kocaeli, 97s.
- ÖZKURT, C., 2016. Endüstri 4.0 Perspektifinden Türkiye’de İmalat Sanayinin Durumu: Sakarya İmalat Sanayi Üzerine Bir Anket Çalışması. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya, 137s.
- ÖZTEMEL, E. 2003. Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul, 44s.
- PAN, M., 2015. Applying Industry 4.0 to the Jurong Island Eco-industrial Park, EnergyProcedia, 75, ss. 1536-1541.
- SANCHEZ, B. B. ve diğerleri, 2015. A Framework for Developing Traceability Solutions in Small Manufacturing Companies. Sensors, 15, ss. 29478-29510.
- SAYGILI, Y. S., 2008. İstatistiksel Yöntemlerle Yapay Sinir Ağları Uygulamalarının Karşılaştırılması: Milli Savunma Bakanlığı Bütçesinin Öngörülenmesi, Kara Harp Okulu Savunma Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 122s.
- SESVEREN, Ö., 2008. Yapay Sinir Ağları Temelli Reaktif Güç Kompanzasyonu Eğitim Seti Tasarımı. Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 123s.
- STERGIOU, C. ve SIGANOS, D. 2010. Neural Networks, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html, 11 Kasım 2013.
- ŞEKELLİ, M., TARKAN N., 2005. Reaktif Güç Kontrol Rölesinde Minimum Anahtarlama Sayısı ve Optimal Reaktif Güç Seçimi. İTÜ Mühendislik dergisi, 4(6), 15-22.
- ŞERBETÇİ, M. B., 2019. Harmonik Problemi Olan Tesislerde Harmonik Filtreli Kompanzasyon Sistemlerinin Tasarımı ve Analizi. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 121s.
- VADİ, S., 2014. Senkron Motor İle Reaktif Güç Kompanzasyonu Benzetimi İçin Web Tabanlı Bir Eğitim Aracı. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 81s.
- WALSH, G., MOHRİNG, M., KOOT, C., SCHAARSCHMIDT, M., 2014. Preventive Product Returns Management Systems-a Review and Model. In: Proceedings of the 21th European Conference on Information Systems (ECIS), Tel Aviv, Israel.
- WANG, L., TÖRNGREN, M. ve ONORI, M., 2015. Current Status and Advancement of Cyber-Physical Systems in Manufacturing. Journal of Manufacturing Systems, sayı:37, ss. 517-527.
- WIGAND, R.T., PICOT, A., REICHWALD, R., 1997. Information, Organization and Management: Expanding Markets and Corporate Boundaries. Wiley, Chichester.
- YUE, X. ve diğerleri, 2015. Cloud,assisted industrial cyber-physical systems: An insight, Microprocessors and Microsystems. sayı: 39, ss. 1262-1270.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı soyadı : Mehmet Bedri DOĞRUYOL
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Savur - 01.09.1987
Telefon : (0) 555 640 3969
e-mail : bedri_dogruiol47@hotmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	Savur Lisesi	2005
Üniversite	Samsun Ondokuz Mayıs Üniversitesi/ Elektrik - Elektronik Mühendisliği	2011
	Anadolu Üniversitesi/ Adalet	2016
Yüksek Lisans	Harran Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Elektronik ABD.	2020
Doktora	:	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2013 -	Toprak Mahsulleri Ofisi Şanlıurfa Şube Müdürlüğü	Elektrik Mühendisi

YABANCI DİLLER

İngilizce, Arapça

HOBİLER

Sinema, balık tutmak, dağ sporları.

YAYINLAR

1. Artificial Neural Networks Controlled Reactive Power Compensation in the Perspective of Industry 4.0, 1. International Conference On Virtual Reality, 04 - 05 April 2019, Harran University, Şanlıurfa, p.217 - 222.
2. Yapay Sinir Ağları Kontrollü Dinamik Reaktif Güç Kompanzasyonu. 3. Uluslararası GAP Matematik - Mühendislik - Fen ve Sağlık Bilimleri Kongresi, 29 Kasım - 1 Aralık 2019, Harran Üniversitesi, Şanlıurfa, s.6 - 15.

EKLER

EK 1 Veri Okuma Kaynak Kodu

```
function [X, T] = read_data(file_name)
veri = readtable(file_name);
X = veri(:,1:end-1);
X = table2array(X)';
T = veri(:, end);
T = table2array(T)';
```



EK 2 Yapay Sinir Ağı Modeli Oluşturma Kaynak Kodları

```
function create_model(input_file, model_name)
% Solve an Input-Output Fitting problem with a Neural Network
% X - input data.
% T - target data.

[X, T] = read_data(input_file);
x = X;
t = T;
% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.
% Create a Fitting Network
hiddenLayerSize = 3;
net = fitnet(hiddenLayerSize,trainFcn);
% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
% Train the Network
[net,tr] = train(net,x,t);
% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
% View the Network
view(net)
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
% save net
save(['./models/', model_name], 'net');
```

EK 3 Program Kaynak Kodları

```
function varargout = eem_mbd(varargin)
% EEM_MBD MATLAB code for eem_mbd.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @eem_mbd_OpeningFcn, ...
    'gui_OutputFcn', @eem_mbd_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function eem_mbd_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
handles.rowCounter = 1;
handles.auto_run = 0;
handles.default_facility_cos = '0.7'
handles.default_target_cos = '0.99'

function varargout = eem_mbd_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function txt_kvar_r_2_Callback(hObject, eventdata, handles)
function txt_kvar_r_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_s_2_Callback(hObject, eventdata, handles)
function txt_kvar_s_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_t_2_Callback(hObject, eventdata, handles)
function txt_kvar_t_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_shunt_r_2_Callback(hObject, eventdata, handles)
function txt_shunt_r_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_shunt_s_2_Callback(hObject, eventdata, handles)
function txt_shunt_s_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_shunt_t_2_Callback(hObject, eventdata, handles)
```

```

function txt_shunt_t_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_r_Callback(hObject, eventdata, handles)
function txt_kvar_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_s_Callback(hObject, eventdata, handles)
function txt_kvar_s_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_t_Callback(hObject, eventdata, handles)
function txt_kvar_t_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit25_Callback(hObject, eventdata, handles)
function edit25_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_target_cos_Callback(hObject, eventdata, handles)
function txt_target_cos_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit27_Callback(hObject, eventdata, handles)
function edit27_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colA_Callback(hObject, eventdata, handles)
function txt_colA_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colP_Callback(hObject, eventdata, handles)
function txt_colP_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAE_Callback(hObject, eventdata, handles)
function txt_colAE_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function txt_colE_Callback(hObject, eventdata, handles)
function txt_colE_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colT_Callback(hObject, eventdata, handles)
function txt_colT_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAL_Callback(hObject, eventdata, handles)
function txt_colAL_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colF_Callback(hObject, eventdata, handles)
function txt_colF_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colU_Callback(hObject, eventdata, handles)
function txt_colU_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAJ_Callback(hObject, eventdata, handles)
function txt_colAJ_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colB_Callback(hObject, eventdata, handles)
function txt_colB_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

function txt_colQ_Callback(hObject, eventdata, handles)
function txt_colQ_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAF_Callback(hObject, eventdata, handles)
function txt_colAF_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function readBtn_Callback(hObject, eventdata, handles)
% veri dosyasini oku
veri = readtable('./data/veri.csv');
[nRow, nCol] = size(veri);
satir = veri(handles.rowCounter, :);
handles.rowCounter = handles.rowCounter + 1;
if handles.rowCounter > nRow
    handles.rowCounter = 1;
end
disp(handles.rowCounter);
handles.satir = satir;
guidata(hObject, handles);
% datalari doldur
set(handles.txt_colD,'string', satir.d);
set(handles.txt_colS,'string', satir.s);
set(handles.txt_colAH,'string', satir.ah);
set(handles.txt_colA,'string', satir.a);
set(handles.txt_colP,'string', satir.p);
set(handles.txt_colAE,'string', satir.ae);
set(handles.txt_colF,'string', satir.f);
set(handles.txt_colU,'string', satir.u);
set(handles.txt_colAJ,'string', satir.aj);
set(handles.txt_colE,'string', satir.e);
set(handles.txt_colT,'string', satir.t);
set(handles.txt_colAL,'string', satir.ai);
set(handles.txt_colB,'string', satir.b);
set(handles.txt_colQ,'string', satir.q);
set(handles.txt_colAF,'string', satir.af);
set(handles.txt_colO,'string', satir.o);
set(handles.txt_colAD,'string', satir.ad);
set(handles.txt_colAS,'string', satir.as);

function clrBtn_Callback(hObject, eventdata, handles)
handles = guidata(hObject);
if (handles.auto_run == 1)
    handles.auto_run = 0;
    guidata(hObject, handles);
    return
end
% datalari temizle
set(handles.txt_colD,'string', ' ');
set(handles.txt_colS,'string', ' ');
set(handles.txt_colAH,'string', ' ');
set(handles.txt_colA,'string', ' ');

```

```

set(handles.txt_colP,'string',' ');
set(handles.txt_colAE,'string',' ');
set(handles.txt_colF,'string',' ');
set(handles.txt_colU,'string',' ');
set(handles.txt_colAJ,'string',' ');
set(handles.txt_colE,'string',' ');
set(handles.txt_colT,'string',' ');
set(handles.txt_colAL,'string',' ');
set(handles.txt_colB,'string',' ');
set(handles.txt_colQ,'string',' ');
set(handles.txt_colAF,'string',' ');
set(handles.txt_colO,'string',' ');
set(handles.txt_colAD,'string',' ');
set(handles.txt_colAS,'string',' ');
set(handles.txt_active_power,'string',' ');
set(handles.txt_reakt_power,'string',' ');
set(handles.txt_apparent_power,'string',' ');
set(handles.txt_kvar_r,'string',' ');
set(handles.txt_kvar_s,'string',' ');
set(handles.txt_kvar_t,'string',' ');
set(handles.txt_kvar_r_2,'string',' ');
set(handles.txt_kvar_s_2,'string',' ');
set(handles.txt_kvar_t_2,'string',' ');
set(handles.txt_shunt_r_2,'string',' ');
set(handles.txt_shunt_s_2,'string',' ');
set(handles.txt_shunt_t_2,'string',' ');
% --- ledleri yakmak icin
function blink(handles, bin)
for i = 1:12
    color = 'red';
    if bin(i:i) == '1'
        color = 'green';
    end
    set(eval(['handles.',num2str(i)]),'BackgroundColor', color);
end

function calcBtn_Callback(hObject, eventdata, handles)
colA = str2double(get(handles.txt_colA, 'string'));
colP = str2double(get(handles.txt_colP, 'string'));
colAE = str2double(get(handles.txt_colAE, 'string'));
facility_cos = str2double(get(handles.txt_facility_cos, 'string'));
active_power = sqrt(3)*380*(colA+colP+colAE)*facility_cos;
set(handles.txt_active_power, 'string', active_power);
reactive_power = sqrt(3)*380*(colA+colP+colAE)*sind(acosd(facility_cos));
set(handles.txt_reakt_power, 'string', reactive_power);
apparent_power = sqrt(3)*380*(colA+colP+colAE);
set(handles.txt_apparent_power, 'string', apparent_power);
% statik hesaplamalari yap
target_cos = str2double(get(handles.txt_target_cos, 'string'));
colF = str2double(get(handles.txt_colF, 'string'));
colU = str2double(get(handles.txt_colU, 'string'));
colAJ = str2double(get(handles.txt_colAJ, 'string'));
kvar_r = colF*(-tand(acosd(target_cos)) + tand(acosd(facility_cos)));
set(handles.txt_kvar_r, 'string', kvar_r);
kvar_s = colU*(-tand(acosd(target_cos)) + tand(acosd(facility_cos)));
set(handles.txt_kvar_s, 'string', kvar_s);
kvar_t = colAJ*(-tand(acosd(target_cos)) + tand(acosd(facility_cos)));
set(handles.txt_kvar_t, 'string', kvar_t);

```

```

% NNA Hesaplamaları Yap
satir = handles.satir;
inp = satir(:, {'a','c','d','e','f','g','h','i','j','k','l','m','n','o'});
inp{1,'f'} = str2double(get(handles.txt_colF, 'string'));
inp{1,'a'} = str2double(get(handles.txt_colA, 'string'));
inp{1,'d'} = str2double(get(handles.txt_colD, 'string'));
load './models/net11.mat', 'net';
inp = table2array(inp);
predict = net(inp);
set(handles.txt_kvar_r_2, 'string', predict);
inp = satir(:, {'p','r','s','t','u','v','w','x','y','z','aa','ab','ac','ad'});
inp{1,'u'} = str2double(get(handles.txt_colU, 'string'));
inp{1,'p'} = str2double(get(handles.txt_colP, 'string'));
inp{1,'s'} = str2double(get(handles.txt_colS, 'string'));
load './models/net12.mat', 'net';
inp = table2array(inp);
predict = net(inp);
set(handles.txt_kvar_s_2, 'string', predict);
inp = satir(:, {'ae','ag','ah','al','aj','ak','al','am','an','ao','ap','aq','ar','as'});
inp{1,'ah'} = str2double(get(handles.txt_colAH, 'string'));
inp{1,'ae'} = str2double(get(handles.txt_colAE, 'string'));
inp{1,'aj'} = str2double(get(handles.txt_colAJ, 'string'));
load './models/net13.mat', 'net';
inp = table2array(inp);
predict = net(inp);
set(handles.txt_kvar_t_2, 'string', predict);
inp = satir(:, {'a','b','c','d','e','f','g','h','i','j','k','l','m','n'});
inp{1,'f'} = str2double(get(handles.txt_colF, 'string'));
inp{1,'a'} = str2double(get(handles.txt_colA, 'string'));
inp{1,'d'} = str2double(get(handles.txt_colD, 'string'));
load './models/net21.mat', 'net';
inp = table2array(inp);
predict = net(inp);
set(handles.txt_shunt_r_2, 'string', predict);
inp = satir(:, {'p','q','r','s','t','u','v','w','x','y','z','aa','ab','ac'});
inp{1,'u'} = str2double(get(handles.txt_colU, 'string'));
inp{1,'p'} = str2double(get(handles.txt_colP, 'string'));
inp{1,'s'} = str2double(get(handles.txt_colS, 'string'));
load './models/net22.mat', 'net';
inp = table2array(inp);
predict = net(inp);
set(handles.txt_shunt_s_2, 'string', predict);
inp = satir(:, {'ae','af','ag','ah','al','aj','ak','al','am','an','ao','ap','aq','ar'});
inp{1,'ah'} = str2double(get(handles.txt_colAH, 'string'));
inp{1,'ae'} = str2double(get(handles.txt_colAE, 'string'));
inp{1,'aj'} = str2double(get(handles.txt_colAJ, 'string'));
load './models/net13.mat', 'net';
load './models/net23.mat', 'net';
inp = table2array(inp);
predict = net(inp);
set(handles.txt_shunt_t_2, 'string', predict);
str = '000000000000000000';
for i = 1:12
    str(i:i) = num2str(randi([0,1]));
end
bin = str;
blink(handles, bin)

```

```

function autoBtn_Callback(hObject, eventdata, handles)
veri = readtable('./data/veri.csv');
[nRow, nCol] = size(veri);
n = 1;
handles.auto_run = 1;
guidata(hObject, handles);
while n < nRow;
    drawnow
    handles = guidata(hObject);
    readBtn_Callback(hObject, eventdata, handles);
    handles = guidata(hObject);
    calcBtn_Callback(hObject, eventdata, handles);
    drawnow
    n = n + 1;
    handles = guidata(hObject);
    if (handles.auto_run == 0)
        break;
    end
    java.lang.Thread.sleep(3000);
end
drawnow

function txt_colD_Callback(hObject, eventdata, handles)
function txt_colD_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colS_Callback(hObject, eventdata, handles)
function txt_colS_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAH_Callback(hObject, eventdata, handles)
function txt_colAH_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colO_Callback(hObject, eventdata, handles)
function txt_colO_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAD_Callback(hObject, eventdata, handles)
function txt_colAD_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_colAS_Callback(hObject, eventdata, handles)
function txt_colAS_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function txt_facility_cos_Callback(hObject, eventdata, handles)
function txt_facility_cos_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_active_power_Callback(hObject, eventdata, handles)
function txt_active_power_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_installed_power_Callback(hObject, eventdata, handles)
function txt_installed_power_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_reakt_power_Callback(hObject, eventdata, handles)
function txt_reakt_power_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_apperrant_power_Callback(hObject, eventdata, handles)
function txt_apperrant_power_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_r_3_Callback(hObject, eventdata, handles)
function txt_kvar_r_3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_s_3_Callback(hObject, eventdata, handles)
function txt_kvar_s_3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txt_kvar_t_3_Callback(hObject, eventdata, handles)
function txt_kvar_t_3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function radiobutton1_Callback(hObject, eventdata, handles)
if (get(handles.radiobutton1, 'Value')==1)
    set(handles.radiobutton2, 'Value', 0);
    set(handles.radiobutton3, 'Value', 0);
    set(handles.uipanel5, 'visible', 'on');
    set(handles.uipanel2, 'visible', 'off');
    set(handles.uipanel4, 'visible', 'off');
else
    if (get(handles.radiobutton2, 'Value')==0 && get(handles.radiobutton3, 'Value')==0 &&
        get(handles.radiobutton1, 'Value')==0)

```

```

        set(handles.uipanel5, 'visible', 'on');
        set(handles.uipanel2, 'visible', 'on');
        set(handles.uipanel4, 'visible', 'on');
    end
end

function radiobutton2_Callback(hObject, eventdata, handles)
if (get(handles.radiobutton2, 'Value')==1)
    set(handles.radiobutton1, 'Value', 0);
    set(handles.radiobutton3, 'Value', 0);
    set(handles.uipanel2, 'visible', 'on');
    set(handles.uipanel5, 'visible', 'off');
    set(handles.uipanel4, 'visible', 'off');
else
    if (get(handles.radiobutton1, 'Value')==0 && get(handles.radiobutton3, 'Value')==0 &&
        get(handles.radiobutton2, 'Value')==0)
        set(handles.uipanel5, 'visible', 'on');
        set(handles.uipanel2, 'visible', 'on');
        set(handles.uipanel4, 'visible', 'on');
    end
end

function radiobutton3_Callback(hObject, eventdata, handles)
if (get(handles.radiobutton3, 'Value')==1)
    set(handles.radiobutton2, 'Value', 0);
    set(handles.radiobutton1, 'Value', 0);
    set(handles.uipanel4, 'visible', 'on');
    set(handles.uipanel2, 'visible', 'off');
    set(handles.uipanel5, 'visible', 'off');
else
    if (get(handles.radiobutton2, 'Value')==0 && get(handles.radiobutton1, 'Value')==0 &&
        get(handles.radiobutton3, 'Value')==0)
        set(handles.uipanel5, 'visible', 'on');
        set(handles.uipanel2, 'visible', 'on');
        set(handles.uipanel4, 'visible', 'on');
    end
end

function setMyValue(var, val)
eval(['global ', var]);
if isnumeric(val)
    eval([var, ' = ', num2str(val), ';']);
else
    eval([var, ' = ', char(39), val, char(39), ';']);
end

function val = getMyValue(var)
eval(['global ', var]);
val = eval(var);

function txt_I_Callback(hObject, eventdata, handles)
function txt_I_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```