

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**TEK FAZLI ASENKRON MOTORUN HIZ KONTROLÜ İÇİN YSA  
AYARLAMALI BİR BULANIK MANTIK DENETLEYİCİ**

**YÜKSEK LİSANS TEZİ**

**Elektrik-Elektronik Müh. Yeliz YILDIRIM**

**EYLÜL 2014**

**TRABZON**

**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**TEK FAZLI ASENKRON MOTORUN HIZ KONTROLÜ İÇİN YSA**  
**AYARLAMALI BİR BULANIK MANTIK DENETLEYİCİ**

**Elektrik-Elektronik Müh. Yeliz YILDIRIM**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde**  
**“ELEKTRİK YÜKSEK MÜHENDİSİ”**  
**Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 08.09.2014**  
**Tezin Savunma Tarihi : 23.09.2014**

**Tez Danışmanı : Prof. Dr. Cemil GÜRÜNLÜ**

**Trabzon 2014**

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalında  
Yeliz YILDIRIM tarafından hazırlanan

TEK FAZLI ASENKRON MOTORUN HIZ KONTROLÜ İÇİN YSA  
AYARLAMALI BİR BULANIK MANTIK DENETLEYİCİ

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 27/05/2014 gün ve 155 sayılı  
kararıyla oluşturulan jüri tarafından yapılan sınavda  
YÜKSEK LİSANS TEZİ  
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Cemil GÜRÜNLÜ

Üye : Prof. Dr. İsmail Hakkı ALTAŞ -

Üye : Yrd. Doç. Dr. Tuğrul ÇAVDAR

.....  
.....  
.....  
1

Prof. Dr. Sadettin KORKMAZ

Enstitü Müdürü

## ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesinde Fen bilimler Enstitüsü Elektrik-Elektronik Mühendislik Anabilim Dalı, Elektrik mühendisliği Yüksek Lisans Programından hazırlanmıştır. Bu tezde ADALINE yardımıyla ağırlıkları ayarlanan Yapay sinir Ağı kullanarak Bulanık Mantık Denetleyici üzerinde üyelik fonksiyonlarının parametrelerinin ayarlanmasıyla elde edilen denetleyici ile tek fazlı Asenkron motor kontrolü Matlab/SIMULINK ortamında çalışılmıştır.

Öncelikle eğitimin herşeyden daha önemli olduğunu benimseten, bu konuda ve hiçbir konuda desteğini esirgemeyen aileme şükranlarımı sunarım.

Tezim süresince yönlendirmeleri ve yardımlarından dolayı danışmanım Prof. Dr. Cemil GÜRÜNLÜ'ye, daha iyi şartlarda çalışma imkanı sağlamak için çalışan bölüm başkanım Prof. Dr. İsmail Hakkı ALTAŞ'a ve GESLAB (Güç Elektroniği ve Sürücü Sistem Laboratuvarı) kapılarını bana açarak her türlü desteği ve imkanı veren değerli hocam Doç. Dr. Halil İbrahim OKUMUŞ'a, bu laboratuvardaki sorunlarıma koşan EMPEDSCON (Electrical Machines, Power Electronics Drive System and Control Research Group) elemanlarına ve tezim boyunca bana yardım eden kıymetli arkadaşım Reza MIRZAPOUR'a teşekkürü borç bilirim.

Yaptığım çalışmanın sonraki çalışmalara faydalı olmasını dilerim.

Yeliz YILDIRIM  
Trabzon, 2014

## **TEZ BEYANNAMESİ**

Yüksek Lisans tezi olarak sunduđum “Tek Fazlı Asenkron Motorun Hız Kontrolü İçin YSA Ayarlamalı Bir Bulanık Mantık Denetleyici” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Cemil GÜRÜNLÜ’nün sorumluluğunda tamamladıđımı, örnekleri kendim topladıđımı, başka kaynaklardan aldıđım bilgileri metinde ve kaynakçada eksiz olarak gösterdiđimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandıđımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiđimi beyan ederim. 08.09.2014

Yeliz YILDIRIM

## İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET .....	VII
SUMMARY .....	VIII
ŞEKİLLER DİZİNİ .....	IX
TABLolar DİZİNİ.....	XI
SEMBOLLER DİZİNİ .....	XII
1. GENEL BİLGİLER.....	1
1.1. Giriş .....	1
1.2. Asenkron Motor.....	5
1.2.1. Tek Fazlı Asenkron Motorun Çalışma Prensibi .....	5
1.2.2.2. Kalkış Kondansatörlü Tek Fazlı ASM .....	10
1.2.3.3. Sürekli (Daimi) Çalışma Kondansatörlü Tek Fazlı ASM.....	10
1.2.3.4. Hem Kalkış Hem Daimi (Çift) Kondansatörlü Tek Fazlı ASM.....	11
1.2.3. Tek Fazlı Asenkron Motorun Matematiksel Modellemesi.....	12
1.3. Denetleyiciler.....	17
1.3.1. Bulanık Mantık Denetleyiciler .....	17
1.3.1.1. Bulanık Mantık Denetleyicilerin Yapısı.....	18
1.3.1.1.1. Bulanıklaştırma.....	18
1.3.1.1.2. Kural Tabanı .....	19
1.3.1.1.3. Durulaştırma .....	22
1.3.2. Yapay Sinir Ağları.....	23
1.3.2.1. Yapay Sinir Hücresi.....	24
1.3.2.2. Yapay Sinir Ağı Yapısı.....	26
1.3.2.2.1. İleri Besleme Yapay Sinir Ağı.....	26
1.3.2.2.2. Geri Beslemeli Yapay Sinir Ağı .....	27
1.3.2.3. Yapay Sinir Ağlarında Öğrenme .....	27
1.3.2.3.1. Denetimli Öğrenme .....	27
1.3.2.3.2. Denetimsiz Öğrenme .....	28
1.3.2.3.3. Takviyeli Öğrenme .....	29

1.3.2.3.	ADALINE (Adaptive Linear Element ) .....	30
1.3.3.	Neuro-Fuzzy Denetleyiciler.....	31
1.3.3.1.	ANFIS.....	32
1.3.3.2.	ANFIS Yapısı .....	34
2.	YAPILAN ÇALIŞMALAR.....	38
2.1.	Adaline Optimizeli Yapay Sinir Ağlarıyla Ayarlanan Bulanık Arayüz Denetleyici.....	38
2.2.	Tek Fazlı Asenkron Motor Hız Kontrolünün Benzetim Uygulamaları .....	48
2.2.1.	Tek Fazlı Asenkron Motorun PID ile Kontrolü.....	49
2.2.2.	Tek Fazlı Asenkron Motorun Bulanık Mantık ile Kontrolü.....	51
2.2.3.	Tek Fazlı Asenkron Motorun Yapay Sinir Ağları ile Kontrolü.....	53
2.2.5.	Tek Fazlı Asenkron Motorun Önerilen Denetleyici ile Kontrolü.....	55
3.	TARTIŞMA VE SONUÇLAR .....	57
4.	ÖNERİLER.....	59
5.	KAYNAKLAR .....	60
ÖZGEÇMİŞ		

Yüksek Lisans Tezi

ÖZET

TEK FAZLI ASENKRON MOTORUN HIZ KONTROLÜ İÇİN YSA AYARLAMALI  
BİR BULANIK MANTIK DENETLEYİCİ

Yeliz YILDIRIM

Karadeniz Teknik Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Danışman: Prof. Dr. Cemil GÜRÜNLÜ  
2014, 64 Sayfa

Tek fazlı Asenkron Motorlar, endüstri uygulamalarına olan uygunluğu ve ticari sebeplerden dolayı günümüzde pek çok alanda yaygın olarak kullanılmaktadır. Asenkron Motora verilecek gerilimin kontrolü için ise yaygın olarak PWM (Pulse Width Modulation) tercih edilir. Çünkü PWM asenkron motorun kontrol uygulamalarını kolaylaştıran bir tekniktir.

Bu çalışmada, Matlab/SIMULINK ortamında PWM Eviriciyle besli tek fazlı Asenkron Motorun hızı yeni bir tür kontrolör ile denetlenmiştir. Önerilen bu denetleyicide, Bulanık Mantık üyelik fonksiyonlarının aralıkları Yapay Sinir Ağlarıyla ayarlanmaktadır. Bu ayarlamayı yapan Yapay Sinir Ağının en iyi ağırlıkları ise ADALINE (Adaptif Lineer Element) yardımıyla belirlenmektedir. Bu amaçla oluşturulan tez, üç ana kısma ayrılmıştır. İlk kısım olan giriş kısmında tezin uygulama kısmının anlaşılabilmesi için bilinmesi gereken teknik kısımların ve kontrol yöntemlerinin açıklaması yapılmıştır. İkinci kısımda kontrol amaçlı kullanılan yöntemlere alternatif olarak tezde önerilen Yapay Sinir Ağlarıyla ayarlanan Bulanık Mantık yöntemi ile ilgili açıklama yapılmış ve son kısımda ise Matlab/SIMULINK uygulamalarına yer verilmiştir. Bu son kısımda geleneksel yöntemlerle akıllı yöntemlerin farkını ortaya koyabilmek için öncelikle PWM ile besli tek fazlı Asenkron Motorun PID ile kontrol benzetimi yapıp sonrasında Bulanık Mantık ve Yapay Sinir Ağları gibi akıllı sistemlerle kontrolü yapılmıştır. Uygulama kısmının son adımında da önerilen denetleyicinin Matlab/SIMULINKte tek fazlı Asenkron Motor hız kontrolü yapılmış ve çıktıları elde edilmiştir. Bu çıktılar ise tezin sonuç kısmında kıyaslanmıştır.

**Anahtar Kelimeler:** Tek fazlı Asenkron Motor, PWM (Pulse Width Modulation), Bulanık Mantık, Yapay Sinir Ağları, ADALINE (Adaptif Liner Element), ANFIS (Adaptive Network based Fuzzy Inference System).



Master Thesis

SUMMARY

AN ARTIFICIAL NEURAL NETWORK TUNED-FLC FOR SINGLE PHASE  
ASYNCHRONOUS MOTOR SPEED CONTROL

Yeliz YILDIRIM

Karadeniz Technical University  
The Graduate School of Natural and Applied Science  
Electrical and Electronics Graduate Program  
Supervisor: Prof. Dr. Cemil GÜRÜNLÜ  
2014, 64 Pages

Asynchronous Motors, due to its suitability to industrial application and commercial reasons, are commonly used today in many fields. To control the feeding voltage of Asynchronous motor, PWM(Pulse Width Modulation) is widely preferred. Because PWM is a technic which ease the speed control of Asynchronous Motor.

In this study, the speed of PWM Inverter fed- single phase Asynchronous Motor is controlled in Matlab/SIMULINK environment with a newkind of controller. In this proposed controller, the intervals of membership function of Fuzzy Logic Controller is tuned with Artificial Neural Network. The weights of this Artificial Neural Network is adjusted with the help of ADALINE (Adaptive Linear Element). The thesis, which is written in this purpose, is examined into three main parts. In introduction part, in order to make understandable the simulation part of the thesis, technical information and some control technics are explained. In second part, Artificial Neural Network tuned Fuzzy Logic Controller, which is suggested as an alternative to controllers used before studies with the aim of controlling single phase Asynchronous Motor, is introduced and in the last part Matlab/SIMULINK speed control applications are done and outputs of them are given. In the last part, to clarify the difference between conventional controllers and intelligent controllers, PWM-fed single phase Asynchronous Motor is controlled with PID, then with intelligent controllers such as Fuzzy Logic Controller and Artificial Neural Network. Last part of the control application, controlling single phase Asynchronous Motor with proposed method is simulated in Matlab/SIMULINK. Outputs of those simulations are compared in the conclusion part of thesis.

**Key Words:** Single Phase Asynchronous Motor, PWM (Pulse Width Modulation), Fuzzy Logic, Artificial Neural Network, ADALINE (Adaptif Liner Element), ANFIS (Adaptive Network based Fuzzy Inference System).

## ŞEKİLLER DİZİNİ

### Sayfa No

Şekil 1.	Ana sargısı uyarılmış olan tek fazlı Asenkron Motorun akımının ve manyetik akısının ani değişimi.....	6
Şekil 2.	Hava aralığı toplam manyetik akısı zamanla değişim.....	7
Şekil 3.	Yardımcı sargılı tek fazlı asenkron motor yapısı .....	8
Şekil 4.	Direnç yol vermeli ASM .....	9
Şekil 5.	Kalkış kondansatörlü tek fazlı ASM .....	10
Şekil 6.	Sürekli (daimi) çalışma kondansatörlü tek fazlı ASM.....	11
Şekil 7.	Hem kalkış hem daimi (çift) kondansatörlü tek fazlı ASM .....	11
Şekil 8.	Yardımcı sargı eşdeğer devresi .....	12
Şekil 9.	Ana sargı eşdeğer devresi.....	12
Şekil 10.	Tek fazlı ASM Matlab/SIMULINK modeli.....	16
Şekil 11.	Bulanık mantık denetleyicinin genel yapısı .....	18
Şekil 12.	Üçgen üyelik fonksiyonu .....	19
Şekil 13.	Faz yörüngesi kullanılarak kural tablosu ayarlaması (a) Faz düzlem yörüngesi (b) Sistem adım fonksiyonuna yanıtı.....	20
Şekil 14.	İki kesin girişi olan iki girişli iki kurallı mamdani bulanık mantık sonuç sistemi .....	22
Şekil 15.	Ağırlık merkezi yöntemiyle durulaştırma .....	22
Şekil 16.	Bir yapay sinir hücresinin çalışma şekli.....	24
Şekil 17.	Sinir hücresi yapısı (a) Biyolojik sinir hücresi (b)Yapay sinir hücresi .....	25
Şekil 18.	İleri beslemeli yapay sinir ağı yapısı.....	26
Şekil 19.	Geribeslemeli yapay sinir ağı yapısı .....	27
Şekil 20.	Denetimli öğrenme .....	28
Şekil 21.	Takviyeli öğrenme yöntemi .....	29
Şekil 22.	Adaline yapısı.....	30
Şekil 23.	ANFIS genel taslağı .....	33
Şekil 24.	ANFIS yapısı.....	34
Şekil 25.	ANFIS'te çıkışın belirlenmesi.....	36
Şekil 26.	Bulanık Mantık arayüzü .....	39
Şekil 27.	Bulanık Mantık arayüzünde kullanılan. fis dosyası .....	40
Şekil 28.	Önerilen yöntemde kullanılan yapay sinir ağı.....	41
Şekil 29.	Tansig fonksiyon grafiği .....	41

Şekil 30.	Purelin fonksiyon grafiği.....	42
Şekil 31.	Sistemde kullanılan yapay sinir ağı benzetim modeli.....	43
Şekil 32.	Yapay Sinir Ağının öğrenme.....	44
Şekil 33.	Öğrenme algoritması.....	45
Şekil 34.	Yapay Sinir Ağlarıyla üyelik fonksiyonlarının ayarlanmasının genel yapısı.....	45
Şekil 35.	Sistemde kullanılan ADALINE'in yapısı.....	46
Şekil 36.	Önerilen denetleyicide kullanılan yapay sinir ağının modeli.....	47
Şekil 37.	Tek fazlı asenkron motorun önerilen denetleyici ile kontrolünün genel şeması.....	48
Şekil 38.	Tek fazlı asenkron motor genel kontrol şeması.....	49
Şekil 39.	Tek fazlı asenkron motorun PID ile kontrolünün genel şeması.....	50
Şekil 40.	Tek fazlı asenkron motorun PID ile kontrolünün benzetim modeli.....	50
Şekil 41.	Değişken yükte PID ile kontrol edilen tek fazlı asenkron motor çıktısı.....	51
Şekil 42.	Tek fazlı asenkron motorun bulanık mantık ile kontrolünün genel şeması.....	52
Şekil 43.	Tek fazlı asenkron motorun bulanık mantık ile kontrolünün benzetim modeli.....	52
Şekil 44.	Değişken yükte bulanık mantık denetleyici ile kontrol edilen tek fazlı asenkron motor çıktısı.....	53
Şekil 45.	Tek fazlı asenkron motorun yapay sinir ağları ile kontrolünün genel şeması.....	54
Şekil 46.	Tek fazlı asenkron motorun yapay sinir ağlarıyla ile kontrolünün benzetim modeli.....	54
Şekil 47.	Değişken yükte yapay sinir ağları ile kontrol edilen tek fazlı asenkron motor çıktısı.....	55
Şekil 48.	Tek fazlı asenkron motorun önerilen denetleyici ile kontrolünün benzetim modeli.....	56
Şekil 49.	Önerilen denetleyici ile denetlenen tek fazlı ASM'nin çıktısı.....	56
Şekil 50.	Elde edilen Matlab çıktılarının kıyaslanması.....	57

## TABLULAR DİZİNİ

	<b><u>Sayfa No</u></b>
Tablo 1. Bölge ve noktalara göre kural ayarlaması .....	21
Tablo 2. Bulanık mantık dokuz kurallı tablosu .....	21
Tablo 3. Bulanık Mantık kural tablosu .....	40

## SEMBOLLER DİZİNİ

$R_{sd}$	: Yardımcı sargı stator direnci
$L_{Isd}$	: Yardımcı sargı kaçak endüktansı
$R_{sq}$	: Ana sargı stator direnci
$L_{Isq}$	: Ana sargı kaçak endüktansı
$R'_{rd}$	: Yardımcı sargı rotor direnci
$L'_{Ird}$	: Yardımcı sargı kaçak endüktansı
$R'_{rq}$	: Ana sargı rotor direnci
$L'_{rq}$	: Ana sargı kaçak endüktansı
$L_{md}$	: Yardımcı sargı magnetik endüktansı
$L_{mq}$	: Ana sargı magnetik endüktansı
$L_{sd}$	: Toplam yardımcı sargı stator endüktansı
$L_{sq}$	: Toplam ana sargı stator endüktansı
$L'_{rd}$	: Toplam yardımcı sargı rotor endüktansı
$L'_{rq}$	: Toplam ana sargı rotor endüktansı
$v_{sd}$	: Yardımcı sargı stator gerilimi
$i_{sd}$	: Yardımcı sargı stator akımı
$v_{sq}$	: Ana sargı stator gerilimi
$i_{sq}$	: Ana sargı stator akımı
$v'_{rd}$	: Yardımcı sargı rotor gerilimi
$i'_{rd}$	: Yardımcı sargı rotor akımı
$v'_{rq}$	: Ana sargı rotor gerilimi
$i'_{rq}$	: Ana sargı rotor akımı
$\lambda_{sd}$	: Stator d ekseni akısı
$\lambda_{sq}$	: Stator q ekseni akısı
$\lambda'_{rd}$	: Rotor d ekseni akısı
$\lambda'_{rq}$	: Rotor q ekseni akısı
$\omega_m$	: Rotor açısal hızı
$\theta_m$	: Rotor açısal konumu

<b>P</b>	: Kutup çifti sayısı
<b><math>\omega_r</math></b>	: Elektriksel açısal hızı ( $p \times \omega_m$ )
<b><math>\theta_r</math></b>	: Elektriksel rotor açısal konumu ( $p \times \theta_m$ )
<b><math>T_e</math></b>	: Elektromagnetik tork
<b><math>T_m</math></b>	: Mekanik tork
<b>J</b>	: Toplam eylemsizlik momenti
<b><math>N_q</math></b>	: Ana sargının etkin dönüştürme sayısı
<b><math>N_d</math></b>	: Yardımcı sargının etkin dönüştürme sayısı
<b><math>\phi</math></b>	: Manyetik akı (Wb)
<b><math>\lambda</math></b>	: Sargı akısı (Wb)
<b>ASM</b>	: Asenkron Motor
<b>PWM</b>	: Pulse Width Modulation
<b>FLC</b>	: Fuzzy Logic Controller
<b>BMD</b>	: Bulanık Mantık Denetleyici
<b>YSA</b>	: Yapay Sinir Ağları
<b>ADALINE</b>	: Adaptive Linear Element
<b>ANFIS</b>	: Adaptive Network Based Fuzzy Inference Systems
<b>LMS</b>	: Least Mean Square

## 1. GENEL BİLGİLER

### 1.1. Giriş

Tek fazlı Asenkron Motorlar, sürekli bakım gerektirmemesi, diğer elektrik motorlarına nispeten daha sağlam yapıya sahip olması, bu sayede de işletmesinin daha kolay olması, daha ucuz olması ve çalışırken ark yapmaması gibi birçok sebeple uygulamalarda tercih edilen bir türdür. Tek fazlı Asenkron Motorların kontrolü karmaşık olsa da geliştirilen kontrol yöntemleri bu sorunu ortadan kaldırmış. Tek fazlı asenkron motorun kontrolünün kolaylaşması, kontrol uygulamalarında da yaygınlaşmasını sağlamıştır.

Tek fazlı Asenkron Motor, bu avantajlarına rağmen doğrusal olmayan, beşinci dereceden, modellemesi karmaşık bir sistemdir [1]. Geleneksel hız kontrol yöntemleri doğrusal uygulamalar için yeterli olsalar da non-lineer bir yapıya sahip olan tek fazlı Asenkron Motor için yeterli genellikle yeterli değildir. Bu sorunların üstesinden gelebilmek için matematiksel modele ihtiyaç duymayan akıllı sistemler geliştirilmiştir. Tek fazlı asenkron motorun akıllı bir denetleyici olan Bulanık Mantıkla denetiminde, hem matematiksel model gerektirmemesi hem de çevresel değişikliklere daha dayanıklı olması, Bulanık denetleyicinin geleneksel yöntemlere göre daha çok tercih edilmesini sağlamıştır. Ancak Bulanık Mantıktaki üyelik fonksiyonlarının ve kuralların belirlenmesindeki zorluklar başka akıllı sistemleri daha cazip kılabilir. Yapay Sinir Ağları (YSA), matematiksel model gerektirmeyen akıllı sistemlerden bir diğeridir. Ancak bazı uygulamalar için yeterli olmaması bu iki denetleyicinin beraber kullanılarak olumsuz yanlarının giderilebilmesi için fikir vermiştir.

Bu tezin amacı da bu iki akıllı sistemi beraber kullanarak Bulanık Mantığın ve Yapay Sinir Ağlarının sorunlarını elemektir. Bu amaçla tezde matematiksel model gerektirmeyen Bulanık Mantık denetleyici kullanılmıştır. Bulanık Mantığın üyelik fonksiyonlarının ayarlanması sorununa çözüm getirebilmek için, üyelik fonksiyonlarının aralıkları Yapay Sinir Ağları kullanılarak belirlenmiştir. Sistemin değişikliklere karşı dayanıklı olabilmesi için ise Yapay Sinir Ağlarının bir başka çeşidi olan ADALINE (Adaptive Linear Element) yardımı ile Yapay Sinir Ağlarının ağırlıkları ayarlanmış böylece adaptif bir sistem elde edilmiştir.

Tezin yapısında, öncelikle tek fazlı Asenkron Motor ve Bulanık Mantık, Yapay Sinir Ağları, ADALINE, ANFIS gibi bazı denetleyiciler hakkında genel bilgiler verilmiştir. Sonraki kısımda yukarıda bahsedilen amaçla önerilen denetleyicinin yapısı açıklanmıştır. Son kısmında ise tek fazlı Asenkron Motor hız denetimi

- geleneksel yöntemlerle kıyaslama yapılabilmesi için PID denetleyici ile
- akıllı sistemlerin farkını ortaya koyabilmek için Bulanık Mantık ve Yapay Sinir Ağlarıyla
- önerilen sistem ile Matlab/SIMULINK ortamında değişken yük için benzetimleri yapılmıştır.

Tezin sonucunda, Bulanık Mantıktaki parametre ayarlama sorununu ve Yapay Sinir Ağlarındaki yeterli eğitim verisi elde edemeyebilme sorunu çözebilen, hızlı yanıt verebilen ve değişiklikler karşısında daha dayanıklı bir denetleyici elde edilmiştir.

Tek fazlı Asenkron Motorun kontrol uygulamaları tartışmasız üzerinde çokça çalışılan bir konudur. Bununla ilgili sayısız çalışmalar mevcuttur. Bu çalışmaların herbirini inceleyebilmek mümkün olmasa da son zamanlarda yapılan ve daha çok akıllı sistemlerle olan bazı hız kontrol uygulamalarına göz atmak gerekir.

Kuo-Kai Shyu, Hsin-Jang Shieh ve Sheng-Shang Fu Asenkron Motor üzerinde Yapay Sinir Ağlarıyla adaptif bir hız kontrol uygulaması yaptılar. Eğitim örüntüleri oluşturmak için de gözlemci tabanlı bir referans izleme tekniği kullandılar. Daha sonra bu eğitilmiş Yapay Sinir Ağını, Asenkron Motorun bir referans modeli izlemesi için adaptif bir denetleyici olarak kullandılar [2].

Jaime Fonseca, João L. Afonso, Julio S. Martins ve Carlos Couto Bulanık Mantık tekniğiyle asenkron motor hızının kontrolünün uygulamasını yaptılar. Bunun dizaynı için de Matlab/SIMULINK ve TECH MCU96 yazılımlarını kullandılar. Donanım olarak da ek bir aygıt gerektirmeden 16/32 bitlik bir işlemciyle Bulanık Mantığı uyguladılar. Çalışmayla ilgili çıktılarını ve sonuçları sundular [3].

Mouloud Azzedine Denai ve Sid Ahmed Attia geleneksel denetleyiciler, Bulanık Mantık ve Yapay Sinir Ağlarını birleştirerek oluşturdukları bir denetleyiciyle asenkron motor kontrolü yaptılar. Böylelikle sistemin performansını arttırdılar ve daha kararlı bir denetleyici elde ettiler. Geleneksel denetleyiciler kapalı çevrimin kararlı durum ve geçici durum karakteristiklerinde farklı pek çok duruma izin verdiklerinden Bulanık Mantık ve Yapay Sinir Ağlarını geleneksel denetleyicilere ekleyerek sistem parametrelerindeki bu



belirsizliklerin üstesinden gelmeyi amaçladılar. Bunun için uygun sonuçlar elde ettiklerini ifade ettiler [4].

A. Miloudi, E. A. AI Radadi, A. Draou ve Y. Miloud Doğrudan Torklu Neuro-Fuzzy kontrollü Asenkron Motor sürücüsü için değişken kazançlı bir PI denetleyici tasarladılar. Bunun için öncelikle değişken kazançlı bir PI tasarladılar. Daha sonra PWM ile besli Asenkron Motor sürücüsü için Doğrudan Torklu Neuro-Fuzzy kontrolörünü tanıttılar. Bu kontrol şeması, eviricinin anahtarlamasını yapacak durum vektör modülasyonunun kullanacağı gerilim vektör modülasyonunu Adaptif bir ANFIS aracılığıyla üretmek için stator akı genliğini ve elektromagnetik tork hatalarını kullanıyordu. Umut verici benzetim sonuçları aldıklarını, motorun referans hıza aşma yapmadan hızlı bir şekilde ulaştığını, yük değişikliklerinin hızlı bir şekilde reddeden ve parametre değişikliklerine aldırılmayan bir denetleyici elde ettiklerini belirttiler [5].

Hasan A. Yousef ve Manal A. Wahba asenkron motor hız kontrolü için yeni bir adaptif Bulanık Mantık tekniğini tanıttılar. Bulanık Mantık, sistem önbilgisi gerektirmeden, geri beslemedeki fonksiyonu tahmin etmesi için kullanıldı. Bilinmeyen yük değerlerinde ve değişen parametrelerde de iyi ve etkili sonuç verdiğini çıktılarıyla sundular [6].

S. M. Gadoue, D. Giaouris ve J. W. Finch doğrudan tork kontrolüyle PI hız denetleyicisini birleştirerek Asenkron Motor kontrolü yapmayı amaçladılar. Ancak bunun için sürekli ayar yapabilen adaptif denetleyici gerektiğini düşündükleri için dört farklı model tasarladılar. Bu modellerin ikisinde ayarlama yapması için geleneksel PI denetleyici kullandılar, üçüncüsünde Bulanık Mantık ve sonuncusunda da Hibrit Kayan Mod kontrol teorisine dayanıyordu. Her bir modelin dayanıklılığını ve bozucu etkilere karşı direnme yetisini test ettiler [7].

Ding Wang ve Ying Shi de hız tahmincisi, takip ayıredici ve Bulanık Mantık içeren bir kontrol şeması oluşturdular. Rotor hız tahmincisini ve matematiksel model sonucu elde edilen rotor hızının ayırt edicilerden sonraki çıkışlarının farkları alınarak Bulanık Mantık girişleri olarak sunulmasıyla Hibrit Bulanık Ayırtedici Denetleyiciyi ortaya koymuş oldular. Sonuçların iyi olduğunu belirttiler [8].

R. A. Gupta, Rajesh Kumar ve Rajesh S. Surjuse yüksek performans Asenkron Motor uygulamaları için Model Referans Adaptif Sistem(MRAS)e dayanan yeni bir algoritmayı gözlemleyici olarak önerdiler. Bunun amaçla stator hız hatasını, gözlemlemek için kullanıyordu. MRAS'a dayalı stator akım hatasının referans modeli, stator akım

aygıtlarla ölçülüyor ve adaptif model, Neuro-Fuzzy'ye dayalı stator akım gözlemleyicisiydi. Adaptif Model aynı zamanda rotor akımını kullanmaya da ihtiyaç duyuyordu. Bu rotor akımı, rotor akım gözlemleyicisine dayanan Yapay Sinir Ağları tarafından tahmin ediliyordu. Önerilen bu sistemin verimliliğini göstermek için 5 tane performans testi yapıp sonuçları sundular [9].

Yine R. A. Gupta, Rajesh Kumar, Rajesh S. Surjuse tek fazlı Asenkron Motoru sürmek için Adaptive Neuro-Fuzzy Inference System (ANFIS)'e dayalı akıllı bir denetleyici önerdiler. Önerilen bu Neuro-Fuzzy denetleyici, Bulanık Algoritma ile beş katmanlı bir Yapay Sinir Ağını beraber çalıştırdılar. ANFIS' in, hibrit öğrenme algoritması kullanarak Bulanık Arayüzün ayarlamasını yapmasını sağladılar. Böylece Bulanık Mantığın öğrenmesini sağlamış oldular. Çıkan sonuçların, yüksek performanslı sürücü uygulamaları için uygun olduğunu iddia ettiler [10].

D. D. Neema, R. N. Patel ve A. S. Thoke çalışmalarında Asenkron Motorun Bulanık Mantık ile kontrol tasarımı yaptılar. Bunun için Asenkron Motor hızının hatasını ve hata değişimini, Bulanık Mantığın girişleri olarak kullandılar. Yaptıkları çalışmanın ne derece etkili olduğunu gösterebilmek için farklı durumlarda elde ettikleri sonuçları DSP 2407' de demostre ettiler [11].

T. D. Dongale ve arkadaşları Bulanık Mantık Denetleyici ve Neuro-Fuzzy ile hız kontrol uygulaması yaptılar. Bunun için Bulanık Mantık Denetleyicinin kural belirleme kısmında Mamdani yaklaşımını kullanarak tasarım yaptılar ve aldıkları sonuçlarla geleneksel yöntemlerden daha iyi bir uygulama olduğunu göstermeye çalıştılar. Bundan daha iyi bir hız kontrol uygulaması için de Asenkron Motorun hızını Yapay Sinir Ağlarıyla denetlediler ve sonuçları kıyasladılar [12].

B. Karthikeyan ve D.Sri Vidhya Matlab ortamında Neuro-Fuzzy ile Asenkron Motor hız uygulaması yaptılar. Neuro-Fuzzy ile Bulanık Mantıkla kontrolünün çıktılarını kıyasladılar. Neuro-Fuzzynin üstünlüğünü savundular[13].

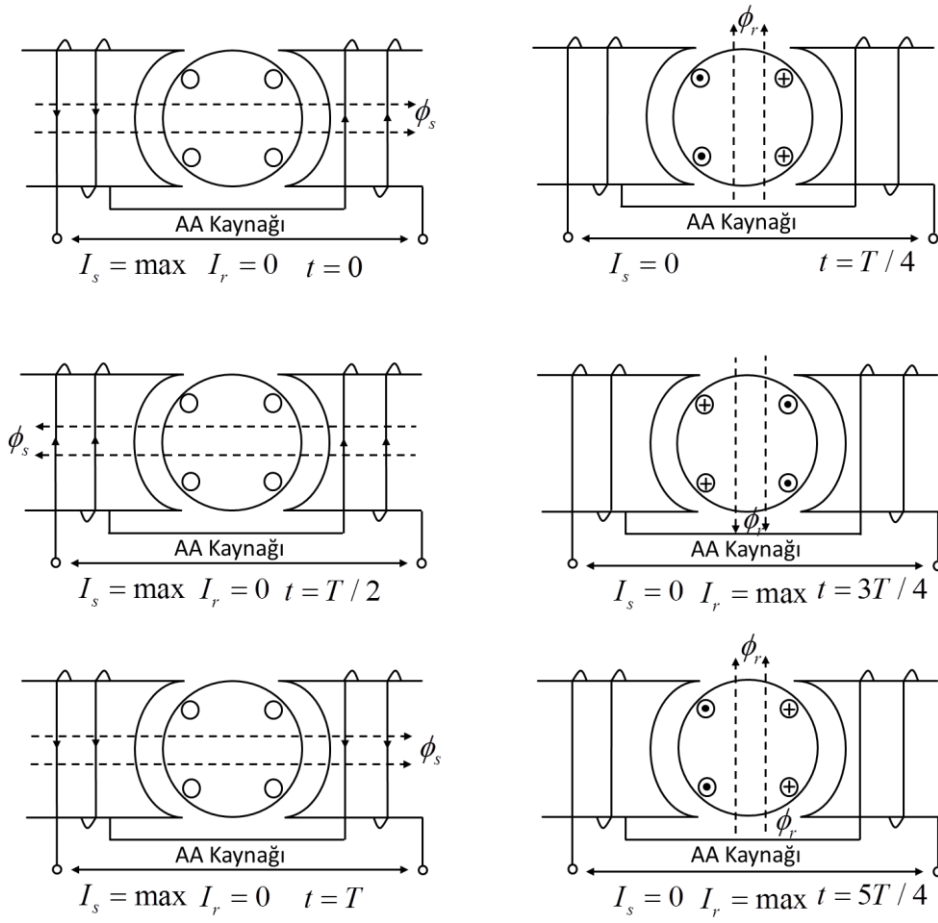
Menghal P. M. ve Dr. A. Jaya Laxmi geleneksel yöntemlerin asenkron motorları kontrol etmekte yetersiz kaldığı ve klasik Bulanık Mantığın parametrelerinin ayarlanmasının zor olduğu gerekçesiyle denetleyici olarak Adaptif Neuro-Fuzzy kontrolörünü önerdiler. Çalışmalarında geleneksel PI denetleyici, Bulanık Denetleyici ve Adaptif Neuro-Fuzzy denetleyicilerin, Asenkron Motor hız uygulamasındaki performanslarını sundular ve Adaptif Neuro-Fuzzy denetleyicinin hız kontrolünde daha güvenilir olduğunu ifade ettiler [14].

Kenza BOUHOUNE, Krim YAZID ve Mohamed Seghir BOUCHERIT çalışmalarında Asenkron Motor hız kontrolü için Bulanık Mantığa dayalı bir vektör kontrol önerdi ve çalışmalarının geleneksel yöntemlerle yapılan tek fazlı Asenkron Motor hız uygulamalarına göre daha verimli olduğunu ve daha güçlü kararlılık gösterdiğini belirttiler [15].

## **1.2. Asenkron Motor**

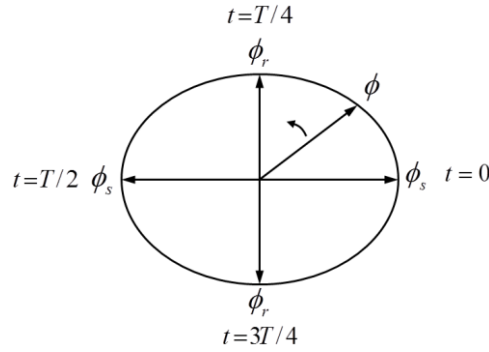
### **1.2.1. Tek Fazlı Asenkron Motorun Çalışma Prensibi**

Tek fazlı asenkron motorun stator sargılarına ilk hareket verilirse stator akısı rotor sargılarını keser ve bu da rotor sargılarında bir elektromotor kuvvet oluşmasına yani gerilim endüklenmesine sebep olur. Hızla ile doğru orantılı olan bu gerilim rotor içinde bir  $I_r$  akımını oluşturacaktır. Bu da rotor manyetik akısının oluşmasını sağlar. Stator manyetik akısı  $\phi_r$  ile stator manyetik açısı  $\phi_s$  arasında  $90^\circ$  olması bu nedenle aynı anda maksimum değere ulaşmazlar. Bu süreç Şekil 1’de görülmektedir.



Şekil 1. Ana sargısı uyarılmış olan tek fazlı Asenkron Motorun akımının ve manyetik akısının ani değişimi

Bu sayede Şekil 2'de görüldüğü gibi stator ve rotor manyetik akılarının bileşkesi döner bir manyetik alan meydana getirecektir.



Şekil 2. Hava aralığı toplam manyetik akısı zamanla değişim

Stator sargısının, eksen üzerinde oluşturduğu alternatif alanın, sargı eksenine  $\theta$  açısı yapan bir doğrultu üzerinde ve hava alanında akı  $\phi$  olmak üzere bu durumun bağıntılarla gösterimi şu şekilde olabilir (harmonikler ihmal edilmiştir).

$$\phi(\theta) = \phi(t) \cos \theta \quad (1)$$

Stator sargısı akımı kosinüs fonksiyonuna göre değiştiğini ve demir kayıplarının ihmal edildiği düşünülürse denklem (2) elde edilir.

$$\phi(t) = \phi_{\max} \cos \omega t \quad (2)$$

(1) ve (2) denklemlerinden (3) denklemi oluşur.

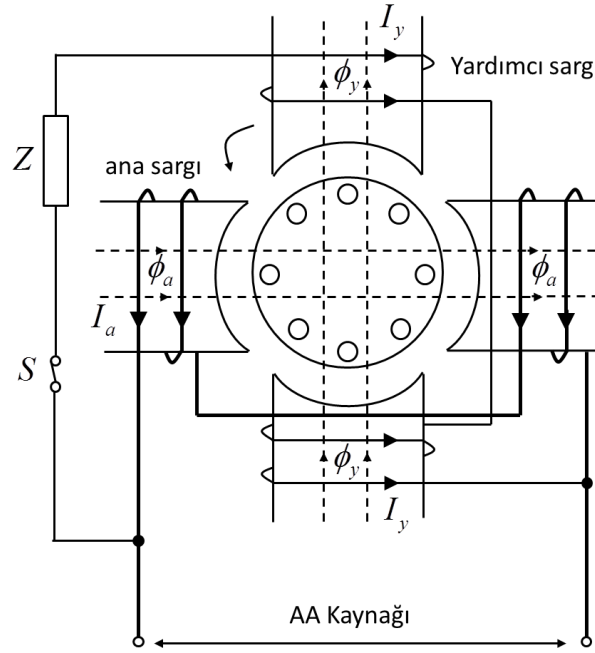
$$\phi(\theta, t) = \phi_{\max} \cos(\omega t) \cos \theta \quad (3)$$

(3)'e kosinüs fonksiyonu uygulanırsa (4) denklemi elde edilmiş olur

$$\phi(\theta, t) = \frac{1}{2} \phi_{\max} \cos(\theta - \omega t) + \frac{1}{2} \phi_{\max} \cos(\theta + \omega t) \quad (4)$$

Statorda genliđi yarıya bölen ama eşit büyüklükte birbirlerine zıt yönde iki alan yani indüklenen iki moment demektir. Motorun ürettiđi moment her iki moment karakteristiđinin cebirsel toplamı olduđuna göre toplam moment sıfır olacaktır [5].

Bu nedenle motorun kalkınabilmesi için bir müdahale gerekmektedir. Bu müdahale için de döner alanı sağlayacak ve ilk hareketi oluşturabilecek bir yardımcı sargı gerekmektedir. Bunun için de tek fazlı asenkron motorlarda Şekil 3' deki gibi ana sargının yanında bir de yardımcı sargı bulunur ve bu iki sargı birbirlerine paralel bağlanırlar ve aralarında 90° elektriksel açı bulunacak şekilde yerleştirilirler [16].



Şekil 3. Yardımcı sargılı tek fazlı asenkron motor yapısı

Manyetik alanın döner alan oluşturabilmesi için:

1. Ana sargının kesit olarak daha kalın ve daha spiralli, yardımcı sargının daha ince ve daha az spiralli yapılmasıyla ana sargının omik direncini düşürüp endüktifliđi artırılarak akımın gerilimden 90° geri kalması sağlanır.

2. Yardımcı sargıya bağlanan kondansatör yardımıyla ana sargı ve yardımcı sargı akımları arasında 90° açı farkı oluşturulur [16-18].

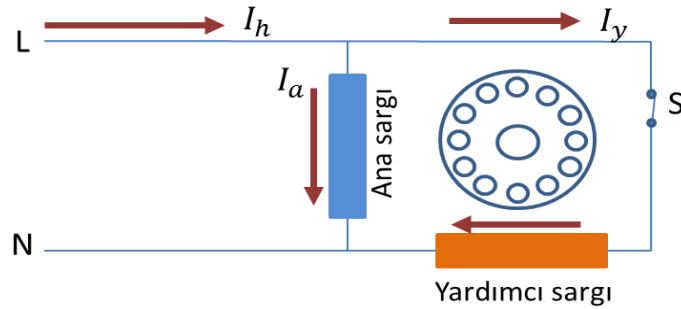
### 1.2.2. Yardımcı Sargılı Tek Fazlı Asenkron Motor Çeşitleri

Bu farklı yöntemlere dayanarak bir fazlı yardımcı sargılı tek fazlı asenkron motorlar 4 çeşide ayrılır:

1. Direnç Yol Vermeli ASM
2. Kalkış Kondansatörlü ASM
3. Sürekli (Daimi) Çalışma Kondansatörlü ASM
4. Hem Kalkış hem Daimi (Çift) Kondansatörlü ASM

#### 1.2.2.1. Direnç Yol Vermeli Tek Fazlı ASM

Bu tür tek fazlı ASM lerde yardımcı sargıyla ana sargı arasında faz farkı oluşturma yöntemi olarak  $X_y/R_y$  oranı küçültülür ve bunun için de yardımcı sargı ince telden yapılır ve aynı zamanda yardımcı sargılar olukların ağzına yakın yerleştirilerek reaktansı da küçük yapılır. Motor senkron hızının %60-%75' inde yardımcı sargı üzerinde bulunan merkezkaç anahtarı yardımıyla devre dışı bırakılır [16,17,20]. Direnç yol vermeli tek fazlı asenkron motor yapısı Şekil 4'deki gibidir.

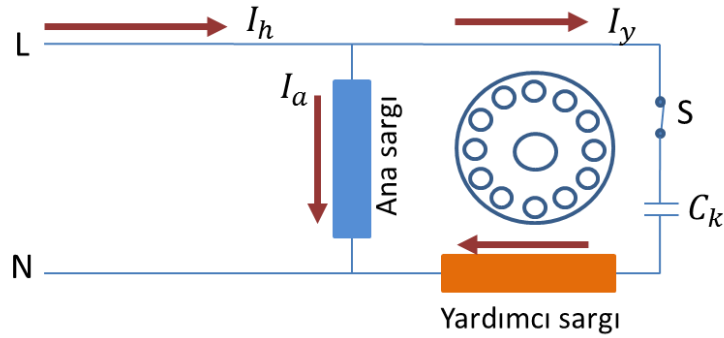


Şekil 4. Direnç yol vermeli ASM

Genellikle sürekli durma-başlatma gibi küçük uygulamalarda ve büyük moment gerektirmeyen çamaşır makinesi, buzdolabı, kurutucu, aspiratör kullanılır.

### 1.2.2.2. Kalkış Kondansatörlü Tek Fazlı ASM

Yardımcı sargı akımı ve ana sargı akımı arasındaki faz farkının yardımcı sargıya seri bir kondansatör bağlanmasıyla oluşturulan tek fazlı asenkron motor çeşididir. Motor yol alma hızının %80' ine ulaştığında kapasite merkezkaç anahtar yardımıyla kondansatör devre dışı bırakılır. Yardımcı sargı çıkarıldığında ana sargı tek başına moment üretmeye devam eder. Moment bir miktar düşer ama bir süre sonra kararlı çalışmaya devam eder. Kalkış kondansatörlü motorların bir avantajı da kondansatörün, kalkış anında fazla çekilen akımı kısıtlamasıdır [16,17-21]. Şekil 5'de Kalkış Kondansatörlü tek fazlı Asenkron Motor yapısı görülebilir.



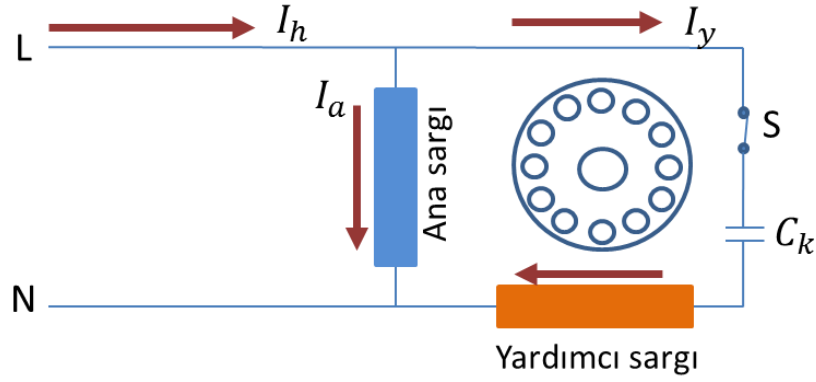
Şekil 5. Kalkış kondansatörlü tek fazlı ASM

Kalkış Kondansatörlü Tek fazlı asenkron motor kompresörler, pompalar, büyük vantilatörler başlıca kullanım yerleridir.

### 1.2.3.3. Sürekli (Daimi) Çalışma Kondansatörlü Tek Fazlı ASM

Bu tür yardımcı sargılı motorlarda C kondansatörü kalkıştan sonra da devrede kalır ve merkezkaç anahtarı gerektirmediğinden maliyeti daha ucuz olur ancak kapasite ve yardımcı sargı sürekli çalışmaya uygun üretilmesi gerektiği için bu da maliyeti arttırır. Bu şekilde bir uygulama hem çalışma momentini yükseltir hem de güç katsayısını 1'e yaklaştırır. Sürekli (Daimi) Çalışma Kondansatörlü tek fazlı ASM Şekil 6'da görülmektedir.



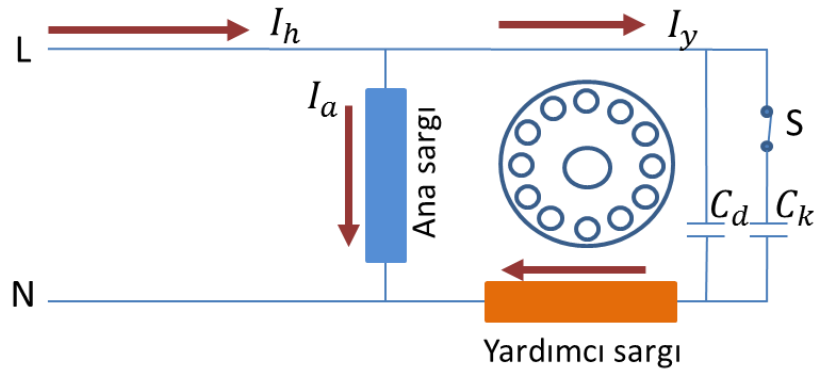


Şekil 6. Sürekli (daimi) çalışma kondansatörlü tek fazlı ASM

Bu tür motorlar genellikle yüksek başlama momenti gerektiren yerlerde kullanılır.

#### 1.2.3.4. Hem Kalkış Hem Daimi (Çift) Kondansatörlü Tek Fazlı ASM

Bu tür tek fazlı Asenkron Motorlar biri kalkış sırasında kullanılacak ve daha sonra merkez kaç anahtarıyla devre dışı bırakılacak diğeri ise sürekli çalışmada da yardımcı sargıya seri bağlı şekilde devrede kalacak paralel iki kapasite bulunur. Bu motorlarda en iyi kalkınma ve normal çalışma şartları görülür. Yapısı Şekil 7'deki gibidir.

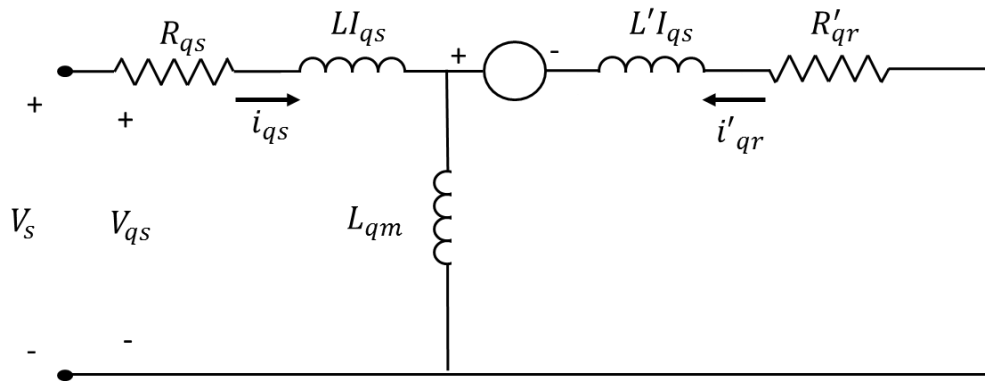


Şekil 7. Hem kalkış hem daimi (çift) kondansatörlü tek fazlı ASM

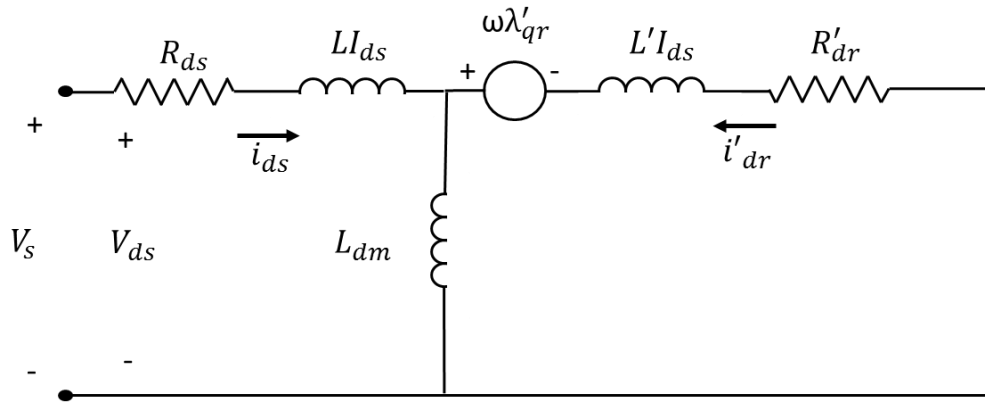
Hem Kalkış hem Daimi(Çift) Kondansatörlü ASM nin kalkınma momentleri düşüktür ve güç katsayısı yüksek, sessiz ve yüksek momentli çalışırlar. Bu nedenle hastane, stüdyo gibi sessiz çalışma ve kompresör, pompa gibi yüksek moment gerektiren yerlerde kullanılır.

### 1.2.3. Tek Fazlı Asenkron Motorun Matematiksel Modellemesi

Daha önce bahsedildiği gibi tek fazlı Asenkron Motorlar, birbirlerine  $90^\circ$  lik açı yapacak şekilde konumlandırılan ana sargı ve yardımcı sargı olmak üzere iki sargı grubundan oluşur ve bunları matematiksel modelde ayırt edebilmek için ana sargının bulunduğu eksen q eksenini olarak, yardımcı sargının bulunduğu eksen de d eksenini olarak adlandırılmıştır. Yardımcı sargı ve ana sargı eşdeğer devreleri sırasıyla Şekil 8' de ve Şekil 9' da çizilmiştir.



Şekil 8. Yardımcı sargı eşdeğer devresi



Şekil 9. Ana sargı eşdeğer devresi

Tek fazlı tek fazlı asenkron motorun elektriksel matematiksel modeli aşağıda verilen denklemlerle tanımlanabilir.

Elektriksel model denklemleri (5)-(25)' de verilmiştir.

$$v_{sd} = R_{sd} i_{sd} + \frac{d\lambda_{sd}}{dt} \quad (5)$$

$$v_{sq} = R_{sq} i_{sq} + \frac{d\lambda_{sq}}{dt} \quad (6)$$

$$v'_{rd} = 0 = R'_{rd} i'_{rd} + \frac{d\lambda'_{rd}}{dt} - \frac{N_d}{N_q} \omega_r \lambda'_{rq} \quad (7)$$

$$v'_{rq} = 0 = R'_{rq} i'_{rq} + \frac{d\lambda'_{rq}}{dt} - \frac{N_q}{N_d} \omega_r \lambda'_{rd} \quad (8)$$

Denklem (5)-(8)' deki akılar, denklem(9)-(12)' de verilmiştir.

$$\lambda_{sd} = L_{sd} i_{sd} + L_{msd} i'_{rd} \quad (9)$$

$$\lambda_{sq} = L_{sq} i_{sq} + L_{msq} i'_{rq} \quad (10)$$

$$\lambda'_{rd} = L_{msd} i_{sd} + L'_{rd} i'_{rd} \quad (11)$$

$$\lambda'_{rq} = L_{msq} i_{sq} + L'_{rq} i'_{rq} \quad (12)$$

Modellemenin kolaylığı açısından yukarıdaki denklemlerden akımlar tek tek elde (13)-(16)'da elde edilmiştir.

$$(L_{sq} L'_{rq} - L_{msq} L_{msq}) i'_{rq} = L_{sq} \lambda'_{rq} - L_{msq} \lambda_{sq} \quad (13)$$

$$(L_{sd} L'_{rd} - L_{msd} L_{msd}) i'_{rd} = L'_{rd} \lambda_{sd} - L_{msd} \lambda'_{rd} \quad (14)$$

$$(L_{sd} L'_{rd} - L_{msq} L_{msq}) i_{sq} = L'_{rd} \lambda_{sq} - L_{msq} \lambda'_{rq} \quad (15)$$

$$\left( L_{sd} \dot{L}_{rd} - L_{msd} L_{msd} \right) \dot{i}_{sd} = \dot{L}_{rd} \lambda_{sd} - L_{msd} \dot{\lambda}_{rd} \quad (16)$$

Bu denklemlerdeki endüktanslar (17)-(20) denklemlerinde verilmiştir.

$$L_{sd} = L_{Isd} + L_{msd} \quad (17)$$

$$L_{sq} = L_{Isq} + L_{msq} \quad (18)$$

$$\dot{L}_{rd} = \dot{L}_{Ird} + L_{msd} \quad (19)$$

$$\dot{L}_{rd} = \dot{L}_{Ird} + L_{msq} \quad (20)$$

Kaçak endüktansları sıfır ve dönüşüm oranını 1 kabul edilip d/dt oranına p denirse denklemler (21)-(24)' deki gibi değişecektir.

$$v_{sd} = \left( R_{sd} + \frac{p}{\omega} L_{msd} \right) \dot{i}_{sd} + \frac{p}{\omega} \dot{i}_{rd} \quad (21)$$

$$v_{sq} = \left( \frac{p}{\omega} + R_{sq} \right) \dot{i}_{sq} + \frac{p}{\omega} \dot{i}_{rq} \quad (22)$$

$$0 = \frac{p}{\omega} L_{msd} \dot{i}_{sd} + \frac{\omega_r}{\omega} L_{msq} \dot{i}_{sq} - \frac{\omega_r}{\omega} L_{msq} \dot{i}_{rd} + \frac{p}{\omega} R_{rd} L_{msd} \dot{i}_{rd} \quad (23)$$

$$0 = -\frac{\omega_r}{\omega} L_{msd} \dot{i}_{sd} + \frac{p}{\omega} L_{msq} \dot{i}_{sq} \left( R_{rd} + \frac{p}{\omega} L_{msd} \right) \dot{i}_{rd} - \frac{\omega_r}{\omega} L_{msq} \dot{i}_{rd} \quad (24)$$

Bunları matris olarak ifade edilirse denklem (25)' teki gibi olur.

$$\begin{bmatrix} v_{sd} & v_{sq} & 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{sd} + \frac{p}{\omega} L_{msd} & 0 & \frac{p}{\omega} & 0 \\ 0 & \frac{p}{\omega} + R_{sq} & 0 & \frac{p}{\omega} \\ \frac{p}{\omega} L_{msd} & -\frac{\omega_r}{\omega} L_{msq} & R'_{rd} + \frac{p}{\omega} L_{msd} & -\frac{\omega_r}{\omega} L_{msq} \\ -\frac{\omega_r}{\omega} L_{msd} & \frac{p}{\omega} L_{msq} & -\frac{\omega_r}{\omega} L_{msq} & R'_{rd} + \frac{p}{\omega} L_{msq} \end{bmatrix} \begin{bmatrix} i'_{sd} \\ i'_{sq} \\ i'_{rd} \\ i'_{rq} \end{bmatrix} \quad (25)$$

Tek fazlı Asenkron Motorun mekanik matematiksel denklemleri ise (26) ve (27)'deki gibidir.

$$T_e = \frac{P N_d}{2 N_q} (i'_{sq} i'_{rd} - i'_{sd} i'_{rq}) \quad (26)$$

$$T_L = T_e - J \frac{d\omega_r}{dt} - B_m \omega_r \quad (27)$$

Bütün bu denklemlerle tek fazlı Asenkron Motorun modeli Matlab/SIMULINK ortamındaki modeli Şekil 10'daki gibi oluşturulabilir.



### 1.3. Denetleyiciler

Sistemin tasarlanmasında kullanılan bazı denetleyiciler ve bunlar hakkındaki genel bilgiler aşağıdaki gibidir.

#### 1.3.1. Bulanık Mantık Denetleyiciler

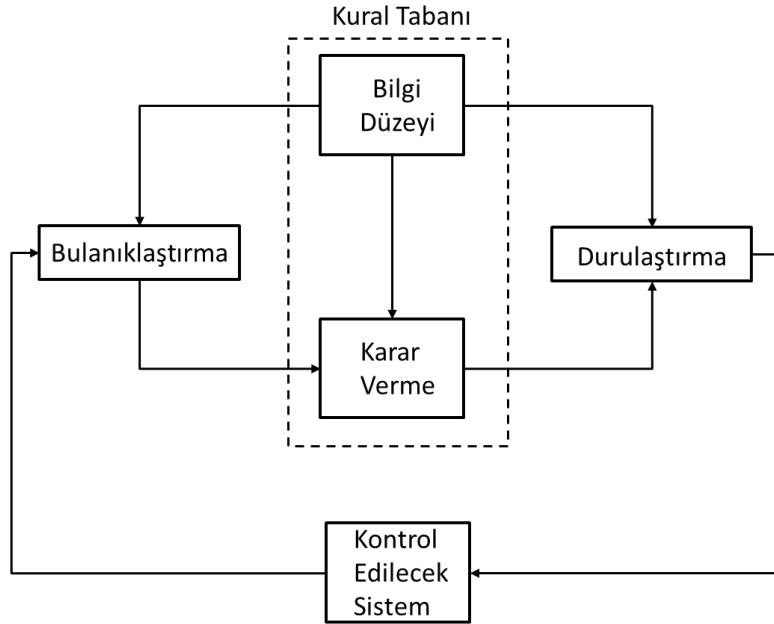
Günlük konuşmalara bakıldığında kabul edilebilir bir derecede konuşmanın temel anlamı anlaşılabilir ancak genellikle herhangi bir şeyi miktarlandırma konusunda tam olarak hemfikir olunamayabilir. Yani doğal dil belirsizliklerden oluşmaktadır. Gerçek hayatta bazı ifadeler vardır ki bunların belli sınırları yoktur mesela çok, uzun, çok kısa, genç gibi. Bu tür nitelik belirten ifadeler bazı durumlar için doğru olabildiği gibi yanlış da olabilirler yani bulanıktırlar. Bu tür ifadeler insanlar tarafından kullanılabilir ve bir nebze anlaşılabilirken bilgisayar dili 1'ler ve 0'lardan oluştuğu için ona ifade etmek zordur. Ancak 1985'te Masaki Togai ve Hiroyuki Watanabe tarafından ilk bulanık çipin oluşturulmasıyla, bu ifadelerin bilgisayarla da kullanılabilmesine bir ışık oldu [22].

Bulanık mantık denetleyici ilk olarak 1965'te Lotfi Zadeh [23] tarafından tanıtıldı ve bu, belirsizliklere dayanan matematiksel bir denetleyiciydi. Yazılımsal hesaplamalarla dilsel hesaplamaların işbirliği içinde çalışabilmesine olanak sunuyordu [24]. Daha sonra Mamdani and Assilian [25] bulanık dizileri, oyuncak bir buhar motorunun adaptif geri beslemeli kontrol sisteminde kullandı ve bu, Bulanık Denetleyicinin ilk pratik uygulanmasıydı ve daha sonra da pek çok endüstriyel uygulamalarda denendi ve günümüzde piyasada Bulanık Mantığın kullanıldığı pek çok ürün bulunmaktadır.

Bulanık Mantık, tecrübeler sonucu oluşan harfsel kontrol yöntemini otomatik kontrol yöntemine dönüştüren bir algoritma oluşturur. Eğer bir sistem geleneksel çok fazla nicelik gerektiren geleneksel yöntemlerle kontrol etmek için çok karmaşıksa veya sistemin bilgisi tam olarak belli değilse, kararsızlıklar, belirsizlikler varsa Bulanık Mantık tam da böyle bir durum için uygundur [26]. Çünkü Bulanık Denetleyici giriş işaretindeki değişikliklerin çıkışa etkisinin gözlemlenmesiyle oluşturulurlar ve bu nedenle de sistemin matematiksel modeline ihtiyaç duymazlar ve bu özellikleriyle de karmaşık, lineer olmayan sistemlere ilaç olmuşlardır [27].

### 1.3.1.1. Bulanık Mantık Denetleyicilerin Yapısı

Bulanık Mantık Şekil 11’de gösterildiği gibi genelde üç kısımdan oluşur: *Bulanıklaştırma*, *Kural Tabanı* ve *Durulaştırma*. *Bulanıklaştırma* olarak adlandırılan ilk kısımda denetleyiciye uygulanan giriş kesin değerleri bulanıklaştırılır. Yani bu toplanan veriler üyelik fonksiyonlarıyla toplanarak bulanık veriler oluşturulmuş olur. Daha sonra bu bulanık veriler *kural tabanında* kullanılarak çıkış bulanık kümelerini oluştururlar ve en son *durulaştırma* da oluşturulan bu bulanık çıkış kullanılabilir kesin değerlere dönüştürülür.



Şekil 11. Bulanık mantık denetleyicinin genel yapısı

#### 1.3.1.1.1. Bulanıklaştırma

Bulanıklaştırmanın yapıldığı kısımda alınan veriler bulanık kümelere dönüştürülürler. Bir bulanık küme gelen verilerle oluşturulmuş  $u$  adı verilen bir evrende ,  $[0,1]$  arası değer alan belli bir şekli olan üyelik fonksiyonuyla ifade edilir. Bulanık set  $F$  şeklinde gösterilirse ve  $u$  sürekli değilse (28)' deki gibi ifade edilir.

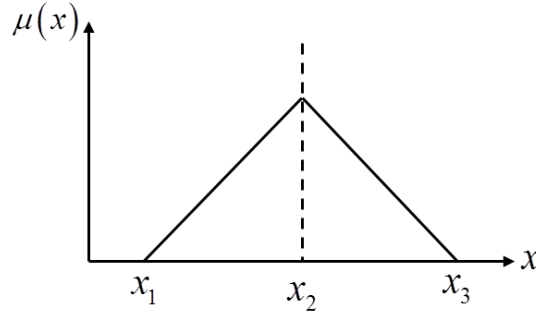


$$F = \sum_{i=1}^n \mu_F(u_i)/u_i \quad (28)$$

Bu üyelik fonksiyonları grafiksel olarak ifade edilebilirler ve bu fonksiyonlar farklı şekiller alabilir. Bu şekiller gauss, ikizkenar yamuk, çan, üçgen gibi geometrik şekillerde olabilirler. Ancak üçgen kullanımı daha yaygındır ve şekli denklem (29) ile oluşturmak mümkündür.

$$\mu(x) = \max \left[ \min \left( \frac{x-x_1}{x_2-x_1}, \frac{x_3-x}{x_3-x_2} \right), 0 \right] \quad (29)$$

Bu denklemdeki  $x$ ; tanımlı uzaydaki değişkeni,  $x_1$ ,  $x_2$  ve  $x_3$  ise Şekil 12' deki üçgen üyelik fonksiyonuna ait parametreleri ifade etmektedir.

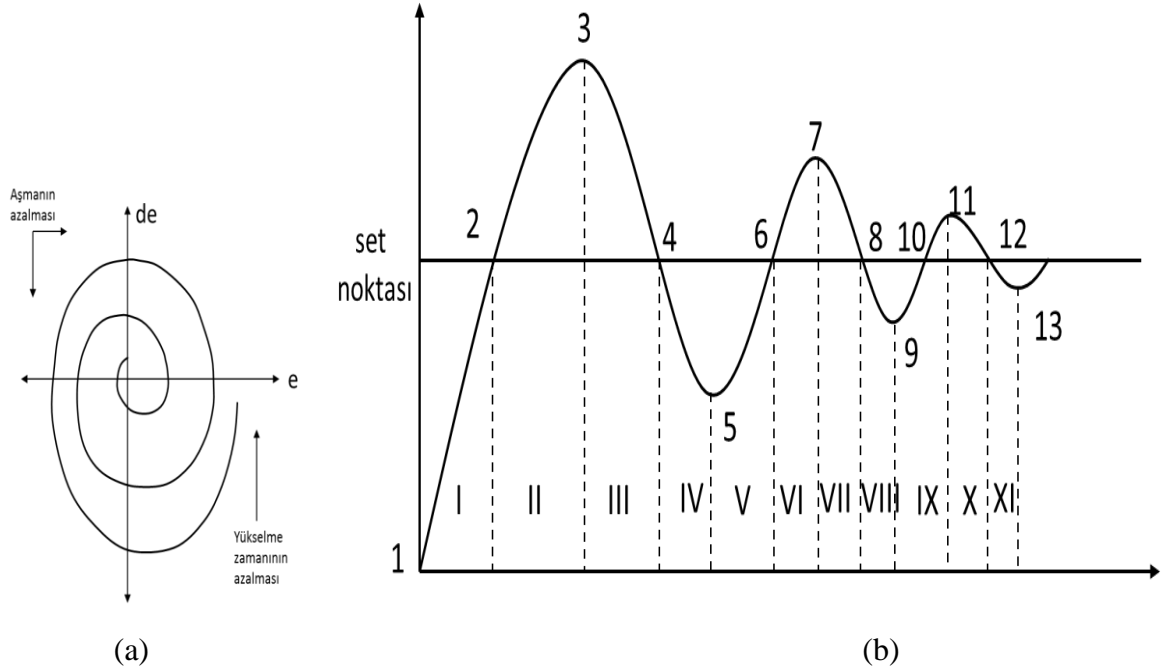


Şekil 12. Üçgen üyelik fonksiyonu

### 1.3.1.1.2. Kural Tabanı

Kural tablosunu belirlerken de farklı yöntemler kullanılabilir [28,34] ancak King ve Mamdani'nin önerdiği yöntem olan 'ölçekli harita[34]' yönteminde önce ölçekleme ayarlanmalı ve ona göre de uçları belirlenebilir. Faz düzlemine göre parametre analizi ve kapalı çevrimdeki sistemin davranışı üstüne sezgi gerektirir. Örneğin, sistem hatası  $e$  ve hata değişimi  $de$  olmak üzere iki girişi bulunan bir bulanık mantık düşünülürse ve çıkışı da giriş işaretindeki değişimi  $du$  olursa ve giriş/çıkış için kullanılan harfsel ifadeler de aynı

gibi düşünülüp Negati (N), Sıfır (S) ve Pozitif (P) alınacak olursa Şekil 13' teki gibi bölgelere ayrılacak ve kural tablosu Tablo1'de verildiği şekilde olacaktır [26].



Şekil 13. Faz yörüngesi kullanılarak kural tablosu ayarlaması (a) Faz düzlem yörüngesi (b) Sistem adım fonksiyonuna yanıtı

Şekil 13'deki her bir durum için üç kurallı girişleri ve çıkışları şöyle olacaktır:

*a durumu*(1,5,9):  $e=r-y$ =Pozitif,  $de$  Negatif öyleyse  $du$  Pozitif olmalıdır.

*b durumu*(2,6,10):  $e=r-y$ =Sıfır,  $de$  Negatif öyleyse  $du$  Negatif olmalıdır.

*c durumu*(3,7,11):  $e=r-y$ =Negatif,  $de$  Sıfır öyleyse  $du$  Negatif olmalıdır.

*d durumu*(4,8,13):  $e=r-y$ =Sıfır,  $de$  Pozitif öyleyse  $du$  Pozitif olmalıdır.

*e durumu*(set nok.):  $e=r-y$ =Sıfır,  $de$  Sıfır öyleyse  $du$  Sıfır olmalıdır.

*f durumu*(I, V): (Şekil10.a.dan)  $e$  Pozitif,  $de$  Negatif öyleyse  $du$  Pozitif olmalıdır.

*g durumu*(II, VI): (Şekil10.a.dan)  $e$  Negatif,  $de$  Negatif öyleyse  $du$  Negatif olmalıdır.

*h durumu*(III, VII): (Şekil10.a.dan)  $e$  Negatif,  $de$  Pozitif öyleyse  $du$  Negatif olmalıdır.

*i durumu*(IV, VIII): (Şekil10.a.dan)  $e$  Pozitif,  $de$  Pozitif öyleyse  $du$  Pozitif olmalıdır.

$j$  durumu(IX): (Şekil10.a.dan) e Pozitif, de Negatif öyleyse du Pozitif olmalıdır.

$k$  durumu(XI): (Şekil10.a.dan) e Negatif, de Pozitif öyleyse du Negatif olmalıdır.

Her bir duruma göre kural ayarlaması Tablo 1' deki gibi tek bir tablo üzerinden gösterilebilir.

Tablo 1. Bölge ve noktalara göre kural ayarlaması

Pozitif, Negatif, Sıfır ifadeleriyle Kural Ayarlaması											
	a	b	c	d	e	f	g	h	i	j	k
	1,5,9	2,4,10	3,7,11	4,8,12	set nok.	I	II	III,VII	IV,VIII	IX	XI
e	+	0	-	-	0	+	-	-	+	+	-
de	-	-	0	+	0	-	-	+	+	-	+
du	+	-	-	+	0	+	-	-	+	0	0

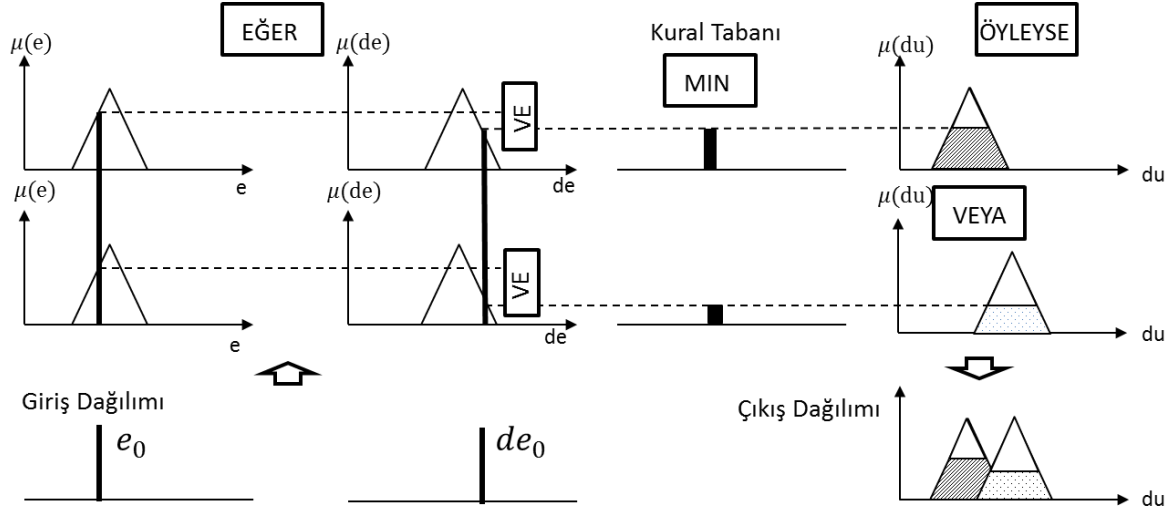
Bu ayarlamalar sonucunda oluşturulan 9 kurallı kural tablosu Tablo 2' deki gibidir.

Tablo 2. Bulanık mantık dokuz kurallı tablosu

DE \ E	N	Z	P
E	P	P	P
P	P	P	P
Z	N	Z	P
N	N	N	N

Özetle, öncelikle kesin değerler Bulanık Mantığa giriş olarak verilmiştir. Daha sonra bu girişler, Bulanık Mantık kümeleri şeklinde harfsel ifadeler kullanılarak üyelik fonksiyonları şekline dönüştürülmüştür. Bu bulanıklaştırma adımından sonra bulanık girişler AND (VE) operatörü kullanılarak çarpılır ve her bir çarpıma karşılık gelen kurallar belirlenmiştir. Bunun belirlenmesi için Mamdani yönteminden faydalanılarak değeri

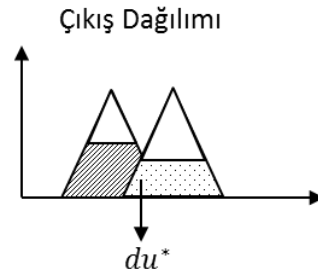
küçük olanı çıkışa aktarılmıştır. Sonuçta, aktarılan her bir çıkış da OR(VEYA) toplama operatörüyle toplanmış ve çıkış dağılımı  $\mu(du)$  belirlenmiştir.



Şekil 14. İki kesin girişi olan iki girişli iki kurallı mamdani bulanık mantık sonuç sistemi

### 1.3.1.1.3. Durulaştırma

Kural tabanının sonucunda elde edilen çıkış dağılımı bulanık bir veri olduğundan dolayı son olarak yapılması gereken adım elde edilen çıkışın durulaştırılmasıdır. Bunun için kullanılan pek çok farklı yöntemler vardır ancak yaygın olarak kullanılanı ağırlık merkezi methodudur.



Şekil 15. Ağırlık merkezi yöntemiyle durulaştırma

Bu yöntem, çıkış dağılımını alır ve ağırlığın merkezini bularak kesin değer olarak elde edilmesini sağlar. Ağırlık metodu yöntemiyle bu değer (30)'daki formülle elde edilir.

$$du = \frac{\sum_{i=1}^q du_i u_c (du_i)}{\sum_{i=1}^q u_c (du_i)} \quad (30)$$

Bu formülde  $du$ , Şekil 15' deki kütle ağırlık merkezi;  $i$ , çıkış için belirlenen kural numaraları;  $u_c$ , o kural numarasındaki çıkış değerinde  $c$  sınıfındaki üyeliğidir.

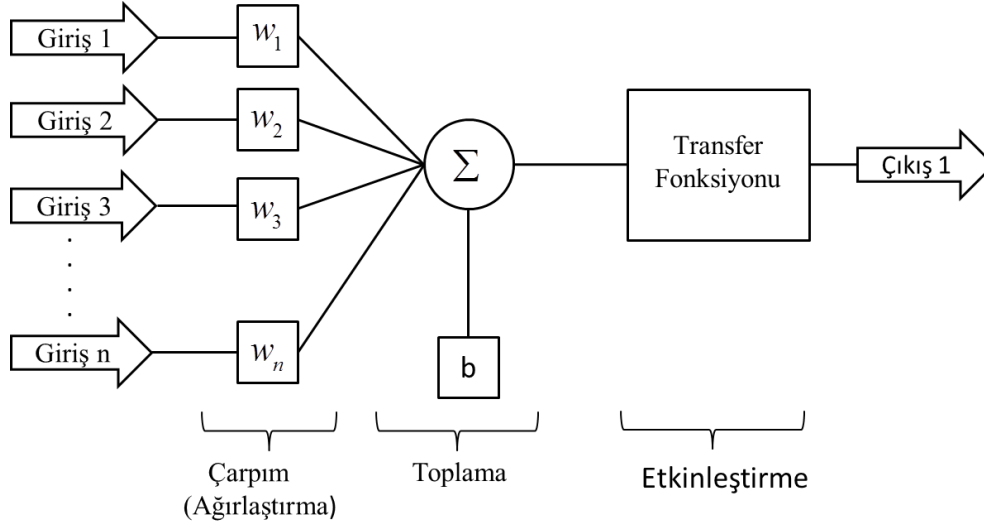
### 1.3.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA), biyolojik sinir ağlarının yapısı ve işlevinin benzetimini yapmaya çalışan bir matematiksel modeldir. Beyinde bulunan sinirlerin algılaması, işlemesi, diğer ucuna aktarması ve diğer sinirlerle bağlantılı olarak çalışarak sinir ağı oluşturmasından ilham alınarak oluşturulan bir sistemdir. YSA'nda her bir Yapay Sinir Ağının temel yapı taşı basit matematiksel fonksiyon olan yapay sinirlerdir ve bu basit fonksiyonlu sinirler diğer sinirlerle de iletişim kurarak Yapay Sinir Ağını oluştururlar. Yani biyolojik ağlarda sinaps uçlarıyla yapılan sinirler arası bağlantı olduğu gibi yapay sinirler arasında da bağlantı vardır[30-32].

Yapay Sinir Ağlarının ilk tasarlayıcısı olarak bilinen McCulloch ve Pitts' in 1943' teki çalışmalarıyla birlikte, YSA' nın hızlı gelişiminin kökleri atılmış olur. Daha sonra Hebb(1949) tarafından ilk öğrenme kuralı geliştirildi. 1950-60 larda ise Block, Minsky, Popert ve Rosenblatt gibi pek çok araştırmacı tarafından geliştirilen ve ağırlık ayarlaması yapılarak öğrenme algoritması oluşturan algılayıcı(perceptron)nun Hebb tarafından oluşturulan öğrenme kuralından çok daha güçlü olduğu farkedildi. Minsky ve Papert (1969) perceptronun lineer olmayan sistemlerin ayırımını yapmakta yeterli olmadığını gösterdi. Bu nedenle 1970' lerde duraksamaya giren yapay sinir ağları, 1985-86' da Parker ve LeCun tarafından geri yayılım (back propogation) olarak adlandırılan çok katlı YSA' nın için yeni bir öğrenme metodu geliştirmesiyle yeniden hız kazandı [31-34].

Yapay Sinir Ağları Şekil 16' da gösterildiği gibi çarpma, toplama ve etkinleştirme şeklinde ifade edilebilecek üç basit kısımdan oluşur. Çarpım kısmında YSA nın girişleri ağırlaştırılır yani her bir giriş değeri bireysel ağırlıklarla çarpılır. Toplama kısmında ise ağırlıklarla çarpılmış girişler ve eşik (bias) toplanır. Yapay sinir ağlarının son kısmı olan

etkinleştirme kısmında ise ağırlaştırılmış girişler ve biaslar toplamı Transfer Fonksiyonu olarak da adlandırılan bir etkinleştirme fonksiyonundan geçirilir.



Şekil 16. Bir yapay sinir hücresinin çalışma şekli

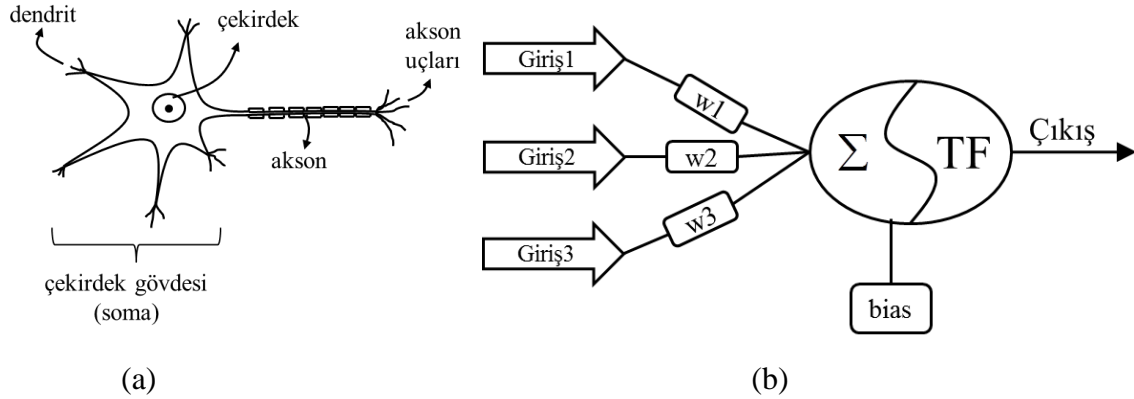
Bir sinirin çalışma şekli tek başına çok bir fonksiyonu yokmuş gibi görünse de birçok sinir hücresinin işbirliği içinde çalışması halinde çok sorunu halledebilirler. Bu sinir hücrelerinin birbirlerine bağlanma şekilleri de farklılık gösterebilir. Problemin çeşidine göre birbirlerine bağlanma şekilleri belirlenebilir. Bu belirlendikten sonra ise kendisine ve parametrelerine ince ayar çekilmesi gerekir. Yani bu demek oluyor ki sinir ağı kurulup hemen kullanılması yeterince verimli olmaz. Öncelikle verilen problemin nasıl çözülmesi gerektiği öğretilmesi gerekir. Bu da farklı çeşitlerde Yapay Sinir Ağlarının oluşmasına sebep olur: denetimli öğretim, denimsiz öğretim ve takviyeli öğretim.

Yapay Sinir Ağının topolojisini belirlenip, topolojinin ince ayarının yapılarak tam olarak nasıl davranması gerektiği öğretildikten sonra Yapay Sinir Ağı kullanıma hazır duruma gelmiş demektir.

### 1.3.2.1. Yapay Sinir Hücresi

Yapay sinir hücresi, yapay sinir ağını oluşturan temel taşlardan biridir. Temel yapısı ve işleyiş şekli biyolojik sinir sistemini oluşturan hücrelerin gözlemlenmesi sonucu oluşturulmuştur. Çalışma şekli ve yapısındaki bu benzerlikler Şekil 17' de görülebilir.

Biyolojik sinir hücrelerini gösteren Şekil 17 (a)' da dendritler uyarıtı alan kısım, soma işleyen kısım akson ise alınan uyarıtı dışarı veren gerekirse başka sinir hücrelerinin dendritiyle sinaps oluşturarak bağlantı kuran ve ona bilgiyi iletebilen kısımdır. Şekil 17 (b)' de YSA, biyolojik sinir ağının bir modeli şeklinde girişlerden, ağırlıklarından, eşikten, transfer fonksiyonundan ve çıkışlardan oluşmuştur.



Şekil 17. Sinir hücresi yapısı (a) Biyolojik sinir hücresi (b)Yapay sinir hücresi

Biyolojik sinir hücresinde dendritlerle gelen bilgi somada işlenip aksonla iletilirken, yapay sinir hücresinde ağırlıklarla çarpılıp ağırlaştırılmış girişlerle gelen bilgiler eşikle toplanarak bir transfer fonksiyonunda işlenir ve çıkış yoluyla işlenen bilgiyi aktarır. Matematiksel olarak aşağıdaki gibi ifade edilebilir.

$$y_i(k) = f\left(\sum_{i=0}^n w_i(k) \cdot x_i(k) + b\right) \quad (31)$$

Burada  $x_i(k)$ , k ayrık zamanındaki giriş değeri;  $w_i(k)$ , k ayrık zamanındaki ağırlık değeri; b, bias yani eşik; f, transfer fonksiyonu;  $y_i(k)$  ise k ayrık zamanındaki çıkış değeridir.

Denklemden asıl bilinmeyen transfer fonksiyonudur. Transfer fonksiyonu herhangi bir matematiksel fonksiyon olabilir ve bu da yapay sinir hücresinin türünü belirler. Yapay Sinir Ağının çözmesi gereken soruna göre fonksiyon belirlenir ve genel olarak basamak fonksiyonu, lineer fonksiyon ve non-lineer fonksiyon yani sigmoid seçilebilir.

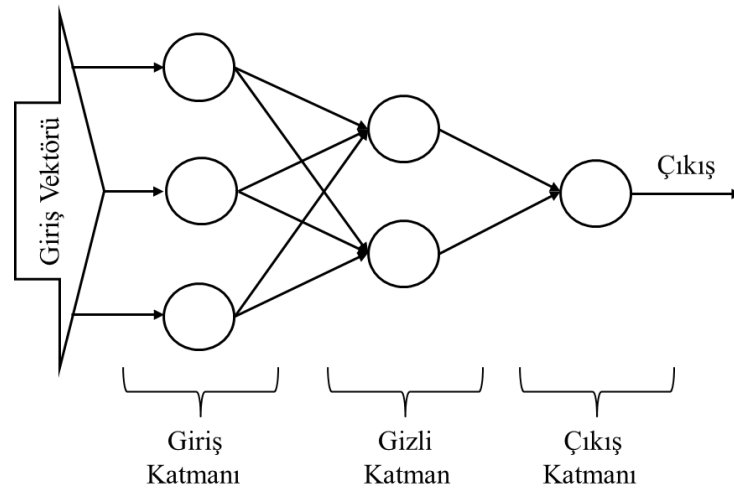
### 1.3.2.2. Yapay Sinir Ağı Yapısı

İki veya daha fazla Yapay Sinir Hücresinin biraraya gelmesiyle Yapay Sinir Ağı oluşur. Tek başına bir yapay sinir hücresi çok fazla kullanışlı olmamaktadır bu nedenle yapay sinir ağı oluşturulur.

Yapay sinir ağı oluşturabilmek için sayısız yöntem vardır. Ancak bu yöntemlerle oluşturulan Yapay sinir Ağlarını iki temel sınıfa ayırmak mümkündür. Bunlar, bilginin girişten çıkışa yalnızca bir yönde aktığı İleri Besleme Yapay Sinir Ağı ve her zaman sadece ileri yönde değil bazı bilgilerin geri yönde de aktığı Geribeslemeli Yapay Sinir Ağıdır.

#### 1.3.2.2.1. İleri Besleme Yapay Sinir Ağı

İleri Besleme Yapay Sinir Ağında bulunması gereken tek şart Şekil 18'deki gibi bilginin çıkıştan girişe kadar sadece tek bir yönde akması yani geri beslemenin olmamasıdır. Giriş sayısı, katmanların sayısı, her bir yapay hücresinde kullanılan transfer fonksiyonunun türünde ya da bu hücreleri birbirlerine bağlayacak olan transfer fonksiyonunun türünde, çıkış sayısında hiçbir sınırlama yoktur.

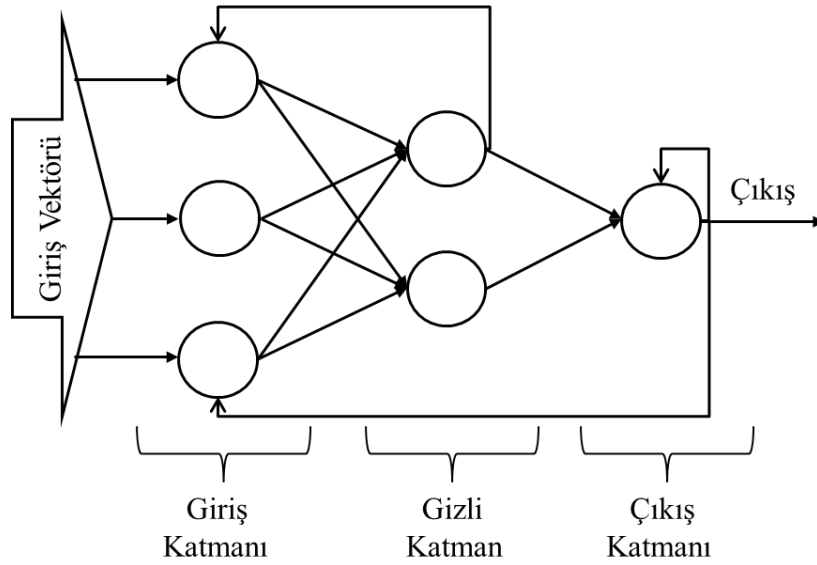


Şekil 18. İleri beslemeli yapay sinir ağı yapısı



### 1.3.2.2.2. Geri Beslemeli Yapay Sinir Ağı

Geri Beslemeli Yapay Sinir Ağı, İleri Beslemeli Yapay Sinir Ağının aynısıdır ancak bunda geri besleme olmaması şartı yoktur. Yani bu durumda bilgi sadece ileri yönde değil aynı zamanda geri yönde de aktarılabilir. Çıktılarından herhangi birini giriş olarak kullanabilmek için kendi dahili hafızaları bulunmaktadır. Tekrarlı Yapay Sinir Ağının yapısı Şekil 19' daki gibi gösterilebilir.



Şekil 19. Geribeslemeli yapay sinir ağı yapısı

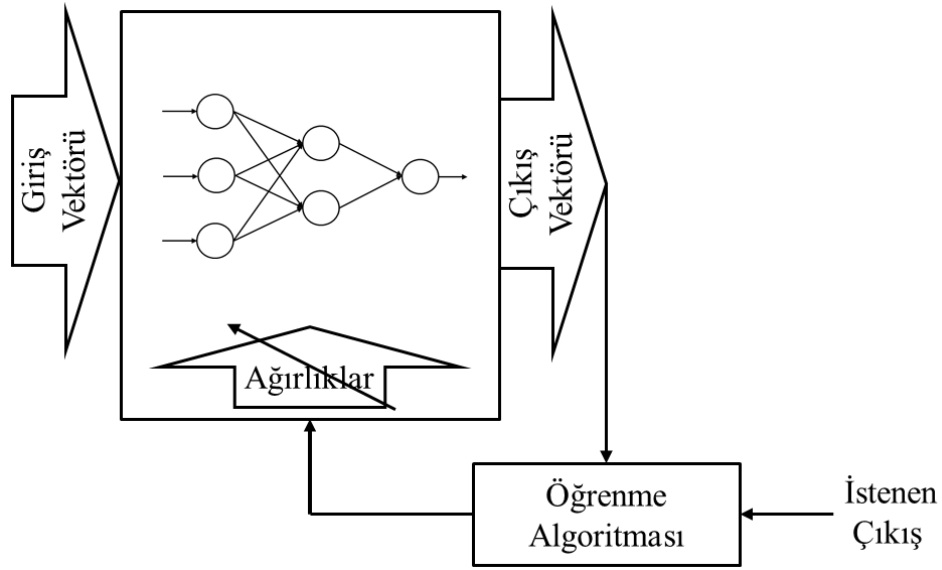
### 1.3.2.3. Yapay Sinir Ağlarında Öğrenme

Yapay Sinir Ağları üç ana öğrenme yöntemi kullanırlar. Bunlar denetimli, denetimsiz ve takviyeli öğrenmedir. Genellikle her tür Yapay Sinir Ağlarıyla herbiri kullanılabilir. Her öğrenme çeşidinin de pek çok eğitim algoritması vardır.

#### 1.3.2.3.1. Denetimli Öğrenme

Denetimli Öğrenme (Supervised Learning), Yapay Sinir Ağının parametrelerini eğitim verisinden (training data) alan bir öğrenme türüdür. Herhangi bir girişin çıkış değerini gördükten sonra parametrelerini ayarlar. Eğitim verisinde ise giriş ve istenen çıkış çiftleri

bulunur. Verilen soruna göre uygun sınıflandırıcı (Çok Katlı Perceptron, Gaussian, Radyal Temelli Sınıflandırıcı,..) [30] seçilebilir. Verilen bir problemin denetimli öğrenme yöntemiyle çözebilmek için öncelikle eğitim örnek türleri belirlenir ve bu örnekler verilen sorunu tam olarak uygun olacak şekilde bir eğitim verisinde toplanır. Öğrenme bu şekilde yapıldıktan sonra veriminin anlaşılabilmesi için geçerlilik testi yapılır. Testin verisi ise öğrenme sırasında oluşturulur. Şekil 20’de denetimli öğrenme yapısı görülmektedir.



Şekil 20. Denetimli öğrenme

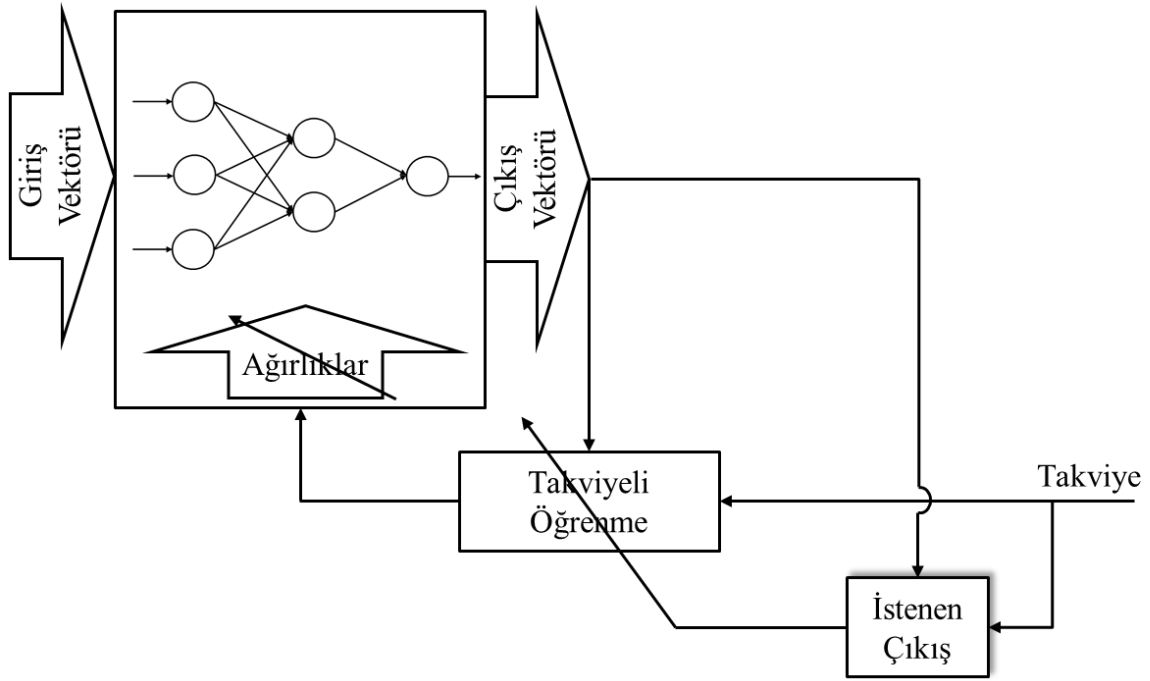
Denetimli öğrenme yöntemleri olarak Gradient Descent, Widrow-hoff (LMS), Generalized delta, Error-correction yöntemleri kullanılabilir.

### 1.3.2.3.2. Denetimsiz Öğrenme

Denetimsiz Öğrenmede (Unsupervised Learning), alınan çıkışın istenen çıkış olup olmadığını kontrol etmek için bir geri besleme yoktur. Bütün ayarlamalar girişten yapılır. Denetimsiz öğrenme girişlere ve en aza indirilmesi gereken herhangi bir fonksiyona göre parametreleri ayarlar. Daha çok istatistiksel modelleme, filtreleme, sınıflandırma gibi tahmin gerektiren yerlerde kullanılır.

### 1.3.2.3.3. Takviyeli Öğrenme

Takviyeli öğrenme, genellikle bilginin verilmediği zaman Yapay Sinir Ağının çevreyle etkileşiminden bilginin oluşturulduğu ve ona göre parametrelerinin ayarlandığı bir öğrenme türüdür. Yani aslında bu tür öğrenmede geri besleme çevreden alınır. Şekil 21’de takviyeli öğrenme yapısı görülmektedir.



Şekil 21. Takviyeli öğrenme yöntemi

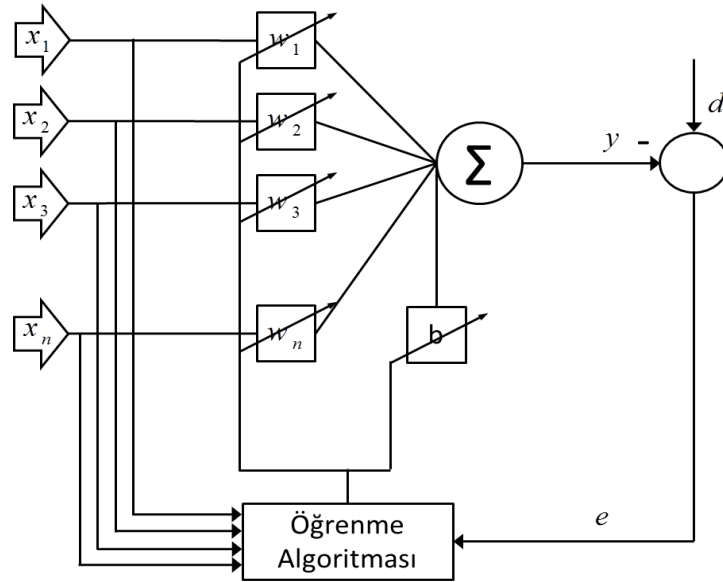
Takviyeli Öğrenme metodu robot kontrolü, haberleşme, satranç gibi oyunlarda ve diğer ardışık karar verme gerektiren yerlerde kullanılabilir.

Yapay Sinir Ağları, kimya, oyun sektörü, radar sistemleri, otomotiv endüstrisi, uzay endüstrisi, astronomi, genetik, bankacılık gibi pek çok alanda Yapay Sinir Ağlarının fonksiyon yaklaşımı, sınıflandırma, örüntü tanıma, karar verme, fonksiyon yaklaşımı gibi özelliklerinden faydalanılmaktadır. Sorunun türüne göre Yapay Sinir Ağlarının özelliklerinden ve türlerinden faydalanılabilir. Yapay Sinir Ağlarının pek çok türünden biri de ADALINE (Adaptive Linear Element). Bu çalışmada karşılaştırma yapılacak kontrol yöntemlerinden biri olarak ADALINE’ de kullanılacağı için daha ayrıntılı anlatılmasında fayda vardır.

### 1.3.2.3. ADALINE (Adaptive Linear Element )

ADALINE (Adaptive Linear Element) 1959 yılında Standord Üniversitesinden Widrow ve Hoff [35] tarafından keşfedilmiştir. Çok girişli ve tek çıkışlı yapısı vardır. Girişler ağırlıklarla çarpılıp eşikle de toplandıktan sonra aktivasyon fonksiyonu olarak lineer fonksiyon (purelin) kullanılır. Ağırlıklar ise çıkış istenen çıkışa eşit olana kadar Least Mean Square (LMS) ile öğrenme metodu ile güncellenir. Basit yapısı ve sinyalde oluşacak herhangi bir değişikliği, hataya da yansıtıp eğitim sürecinde güncellemelerle iyileştirerek, hemen takip edebilir olması ADALINE i tercih edilmesine imkan vermektedir [36].

ADALIN in yapısı Şekil 22' de görülebilir.



Şekil 22. Adaline yapısı

Burada giriş vektörü  $X = [x_1, x_2, x_3, \dots, x_n]$ , ağırlık vektörü  $W = [w_1, w_2, w_3, \dots, w_n]$ ,

ağ çıkışı  $y = \sum_{i=1}^n w_i x_i = WX^T$  ve hata  $e = d - y$ , dır.

d: istenen çıkış

ADALINE i eğitim algoritması içi genellikle Widrow-Hoff Yöntemi kullanılır ve bu kısımda ağırlık ayarlaması yapılarak istenen çıkış elde edilmeye çalışılır. Bu, denklemlerle ifade edilecek olursa hata (32)' deki gibidir.

$$E = \left[ d_k - \sum_{i=1}^n w_{i,k} x_{i,k} \right]^2 \quad (32)$$

Burda da hatanın ağırlıklara bağlı olduğu görülebilmektedir bu da demek oluyor ki hatayı minimize edebilmek için ağırlık ayarı yapılabilir. Hatanın karesi denklem (33)' deki gibidir.

$$E = (d_k)^2 - 2d_k \sum_{i=1}^n w_{i,k} x_{i,k} + \left[ \sum_{i=1}^n w_{i,k} x_{i,k} \right]^2 \quad (33)$$

Karesi alınmış bu hataların kısmı türevi alınırsa denklem (34) elde edilir.

$$\frac{\partial E}{\partial w_{i,k}} = -2 \left[ d_k - 2 \sum_{i=1}^n w_{i,k} x_{i,k} \right] x_{i,k} \quad (34)$$

Bunu ağırlık değişimi formülünde yerine konulduğunda denklem (35) oluşacaktır.

$$\Delta w_i^t = -\eta \left( \frac{\partial E}{\partial w_{i,k}} \right) = \frac{\partial E}{\partial w_{i,k}} = 2\eta \left[ d_k - 2 \sum_{i=1}^n w_{i,k} x_{i,k} \right] w_i^{k+1} = w_i^k + 2\eta (d_k - y_k) x_{i,k} \quad (35)$$

Güncellenmiş yeni ağırlık  $w_{i,k}^{k+1} = w_{i,k}^k + 2\eta (d_k - y_k) x_{i,k}$  olarak elde edilmiş olur.

### 1.3.3. Neuro-Fuzzy Denetleyiciler

Neuro-Fuzzy Denetleyiciler, Bulanık Mantık ve Yapay Sinir Ağlarının çeşitli şekillerde bir arada işlev yaptığı sistemlerdir. Bulanık Mantık, daha kesin girişler ve hızlı işlemci gerektirmeden, üyelik fonksiyonları ve kurallarıyla daha az yer kaplayarak ve parametre değişimlerine karşı diğer sistemlere kıyasla daha kararlı durarak pek çok uygulamada tercih edilmesini sağlamıştır. Ancak Bulanık Mantıkta üyelik fonksiyolarının şekline karar vermek ve aralıklarını belirlemek ona dezavantaj olmuştur. Yapay Sinir Ağları ise insan beyni, yüksek hesaplama oranına sahip olduğundan ve nonlineer

fonksiyonların üstesinden rahatlıkla gelebildiğinden ilgi görmektedir. Ancak bazı durumlarda yetersiz olması ve Bulanık Mantıkla birlikte kullanıldığı uygulamalarda başarılı sonuç vermesi birçok uygulamada Neuro-Fuzzynin tercih edilmesine sebep olmuştur.

Neuro-Fuzzy çalışmaları 1974' te S. C. Lee ve E. T. Lee' nin Bulanık Yapay Sinir Hücresi [38] ni tanıtmayla başladı. 1990' da is Lea, Jani ve Berenji tarafından uygulaması yapıldı ve uzay mekiği kontrolünde kullanıldı [39]. Daha sonra Archer ve Wang, 1991 yılında üyelik fonksiyonlarının Yapay Sinir Ağlarıyla belirlenebileceği bir yöntem önerdi [40]. Altrock and Krause, 93'te Neuro-Fuzzy yöntemini gömülü otomotivde kullandı [41]. Günümüze kadar pek çok uygulayıcının araştırmasına konu olan Neuro-Fuzzy pek çok alanda kullanılmıştır ve gitgide kullanım şekli ve alanı artmaktadır.

Bulanık Mantık ve Yapay Sinir Ağlarının beraber kullanımı pek çok farklı şekilde olabilir. Bu nedenle Neuro-Fuzzy; Yardımlaşmalı, Eş zamanlı ve Hibrit Neuro-Fuzzy olmak üzere üç genel şekle ayrılabilir. Bunlardan Yardımlaşmalı Neuro-Fuzzyde, Yapay Sinir Ağlarından sadece Bulanık Mantığın girişlerini ayarlarken faydalanılır; Eş zamanlı Neuro-Fuzzyde, Bulanık Mantık ve Yapay Sinir Ağı sistemin çalışmasının başından sonuna kadar beraberdirler; Hibrit Neuro Fuzzynin ise pek çok Yapay Sinir Ağı ve Bulanık Mantığın etkileşimi çok farklı şekillerde olabilir. Genellikle Neuro-Fuzzy Hibrit çeşitleriyle bilinir [42-46].

Hibrit Neuro Fuzzynin çeşitlerinden bazıları ise şunlardır: Adaptive Network based Fuzzy Inference System(ANFIS) by R. R. Jang [47], Generalized Approximate Reasoning based Intelligence Control (GARIC)by H. Berenji [48], Fuzzy Adaptive Learning Control Network (FALCON) by C. T. Lin and C. S. Lee [46], Self Constructing Neural Fuzzy Inference Network (SONFIN) by Juang and Lin [49].

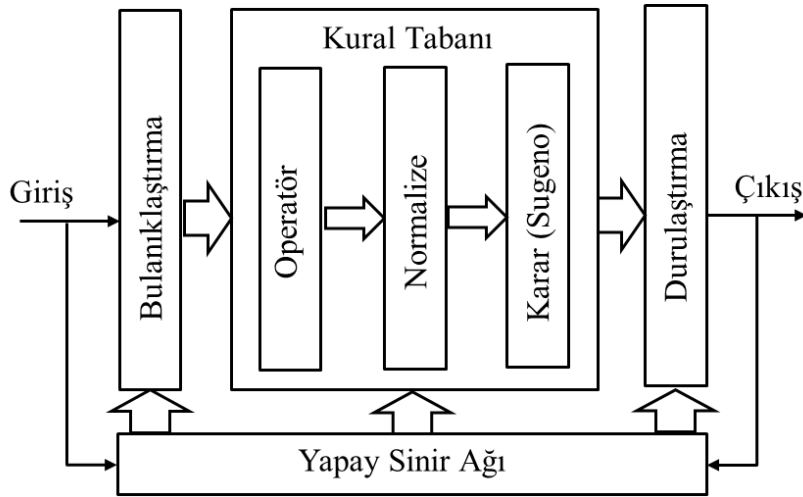
Bunlar arasında Adaptive Network based Fuzzy Inference System (ANFIS) çok ilgi gördü ve üstünde küçük değişiklikler yapılarak ANFIS kaynaklı pek çok farklı model oluşturuldu [50]. Neuro-Fuzzy sisteminin en yaygını olan ANFIS'ten daha ayrıntılı bahsedilmesinde fayda olacaktır.

### **1.3.3.1. ANFIS**

R.R. Jang [57] tarafından ortaya atılan ANFIS (Adaptive Network Based Fuzzy Inference Systems), Yapay Sinir Ağı ile Bulanık Mantık Bloğunun parametrelerini ayarlar.

Birçok uygulama için çok başarılı sonuç vermiş ve üstünde yapılan küçük değişikliklerle oluşturulan pek çok ANFIS türü farklı sorunlar için iyi çözüm olmuştur. Uygulamasının kolay olması, hızlı ve doğru öğrenebilmesi, hızlı genelleme yeteneği, Fuzzy kurallarından dolayı çok iyi açıklayabilme özelliği ve hem sayısal hem sözel verileri kullanarak problem çözümünü hızlandırması [46, 47] gibi yönlerinden dolayı kullanımı hızla artmıştır.

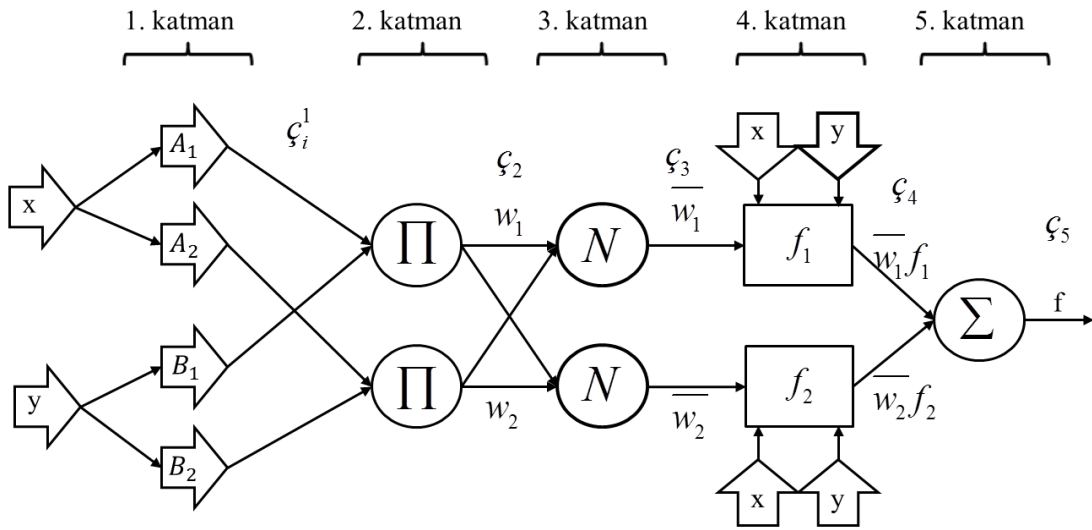
ANFIS temel olarak Yapay Sinir Ağları gibi olsa da Şekil 23’ de görüldüğü gibi işlev olarak Bulanık Mantık gibidir. Bu nedenle Bulanık Mantıktaki gibi öncelikle kesin girişler üyelik fonksiyonlarına dönüştürülür ve bulanıklaştırılır, sonra bu üyelik fonksiyonlarının hangi tür operatöre tabi tutulacağı belirlenir ki burada kesişim operatörü kullanılmıştır. Bu kurallar bir karar mekanizmasıyla çıkışı oluşturur ve burada çıkışın ne olacağına karar veren yöntem Takagi-Sugeno [48] yöntemidir. Daha sonra bu çıkış daha önce Bulanık Mantık kısmında anlatıldığı şekilde durulaştırılır. Bu Bulanık Mantık süreci, üyelik fonksiyonlarının zaten keyfi olarak seçilip belirlendiği sistemler için kullanılabilir. Ancak bazı durumlarda hangi üyelik fonksiyonunun daha iyi sonuç vereceğini belirlemek mümkün olmayabilir. İşte böyle durumlar için Yapay Sinir Ağlarıyla Bulanık Mantık birleştirilmiş ve kullanıcının uygun üyelik fonksiyonunu belirlemesi için deneme-yanılma yöntemi kullanıp zaman ve enerji kaybetmesi önlenmiştir.



Şekil 23. ANFIS genel taslağı

### 1.3.3.2. ANFIS Yapısı

ANFIS giriş, çıkış düğümlerinden ve içerisinde üyelik fonksiyonları ve kuralları oluşturan gizli düğümlerden oluşmaktadır. Böylelikle ANFIS' in bu yönüyle, çok katlı ileri beslemeli Yapay Sinir Ağının incelenmesi, değiştirilmesi veya anlaşılmasının zorluğu giderilmiş olur. Kolaylık olması için incelenecek olan FIS (Fuzzy Inference System) in iki girişi ve bir çıkışı olduğu farzedilir [44]. Sistemin tam olarak nasıl oldu ve ne şekilde işlev yaptığı Şekil 23' den takip edilerek ifade edilebilir.



Şekil 24. ANFIS yapısı, [35]

Görüldüğü gibi ANFIS 5 katmandan oluşmaktadır.

1. Katman: Giriş katmanıdır. Bu katmanda gelen veriler belirli bir fonksiyon kullanılarak bu fonksiyona göre üyelik fonksiyonuna dönüştürüldüğü ve 'Küçük', 'Büyük' gibi sözel etiketler aldığı katmandır.

Bu katmanda giriş olarak gelen  $x$ ,  $A_i$  ye  $y$  de  $B_{i-2}$  dönüştürülür. Yani bu katın çıkışı şu şekilde olacaktır:

$$i = 1, 2 \text{ için } \zeta_i^1 = \mu_{A_i}(x) \quad (36)$$

$$i = 3, 4 \text{ için } \zeta_i^1 = \mu_{B_{i-2}}(y) \quad (37)$$



Genellikle bu dönüşümün yapılması için kullanılacak olan fonksiyon maksimumu 1 ve minimumu 0 olan (38)'de eşitliği verilen çan fonksiyonudur.

$$\mu_{A_i} = \left[ 1 / \left( 1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i} \right) \right] \quad (38)$$

Buradaki  $a_i$ ,  $b_i$  ve  $c_i$  fonksiyon parametreleridir ve bunlar değiştirilerek farklı şekillerde çan elde edilebilir ve böylelikle üyelik fonksiyonunun şekli belirlenebilir. Bunlar, öğrenme aşamasında geri yayılım (backpropagation) [45] yöntemi yardımıyla ayarlanan adaptif değerlerdir. Parametreler değiştikçe de üyelik fonksiyonlarının sözel ifadesi  $A_i$  değişir. Bu parametrelere de öncül parametreler denilmektedir [44].

2. Katman: Bu katman kural katmanıdır ve bu kısımda ilk katmandan gelen sinyallere bulanık bir operatör uygulanır. Burada kullanılan operatör kesişim operatörüdür. Yani oluşturulan bulanık alt kümeler (39)'daki şekilde çarpılır.

$$\zeta_i^2 = w_i = \mu_{A_i} \wedge \mu_{B_i} \quad (39)$$

Burada  $\mu_{A_i}$  ve  $\mu_{B_i}$ , x ve y girişlerinin bulanık kümeleri olan  $A_i$  ve  $B_i$ 'nin üyelik fonksiyonlarıdır. ' $\wedge$ ' ise minimum ve cebirsel çarpımı gösteren kesişim(AND) operatörüdür.

3. Katman: Ortalama katmanıdır. Bu katmandaki her bir düğüm N olarak etiketlenir ve her bir düğümün bir önceki düğümde hesaplanan ağırlığı normalize edilir. Yani aşağıdaki formüldeki gibi i. kuralın ağırlığı, i. kurala kadarki bütün düğümlerin ağırlıklarına bölünür.

$$i = 1, 2 \text{ iken } \zeta_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2} \quad (40)$$

4. Katman: Sonuç katmanıdır. Bu katmanda bulunan düğümler adaptif düğümlerdir. Her bir düğümün çıkışı, önceki katmanda bulunan normalize ağırlık ile birinci dereceden bir polinomun çarpımıdır. Buna göre katmanın çıkışı (41)'de gösterilmiştir.

$$\zeta_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad (41)$$

Burada  $P_i, Q_i$  ve  $r_i$  bu düğümün ayarını yapabilen parametrelerdir ve sonuç parametreleri olarak da isimlendirilebilir.

Yani burada bulanık çarpımın sonucu belirlenir ve bunu yapabilmek için daha önce de belirtilen Takagi-Sugeno yöntemine kullanılmıştır. Bu yöntemde göre sonuçlar şu şekilde belirlenir:

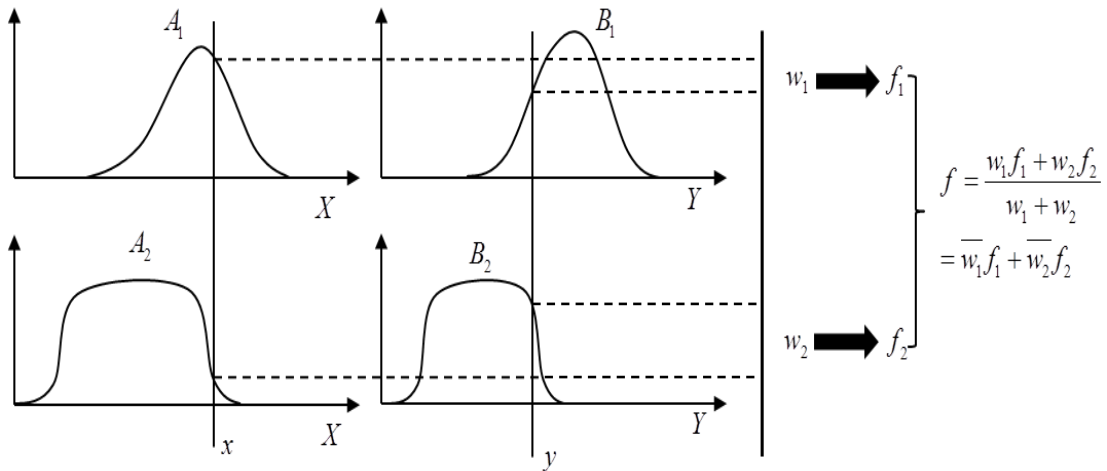
Kural 1: If x is  $A_1$  and y is  $B_1$ , then  $f_1 = p_1 x + q_1 y + r_1$

Kural 2: If x is  $A_2$  and y is  $B_2$ , then  $f_2 = p_2 x + q_2 y + r_2$

5. Katman: Çıkış katmanıdır. Tek bir düğüm olan bu katmanda, önceki katmandan gelen bütün sinyaller toplanır.

$$\zeta_i^5 = f = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (42)$$

Bu da  $i = 1, 2$  için Şekil 25'te ve denklemlerde belirtildiği şekilde olacaktır.



Şekil 25. ANFIS'te çıkışın belirlenmesi

Parametre ayarı yapabilmek için öncül parametreler belirlendikten sonra sonuç parametrelerine LMS (Least Mean Square) (hata karesinin ortalaması) uygulanabilir. Ya da

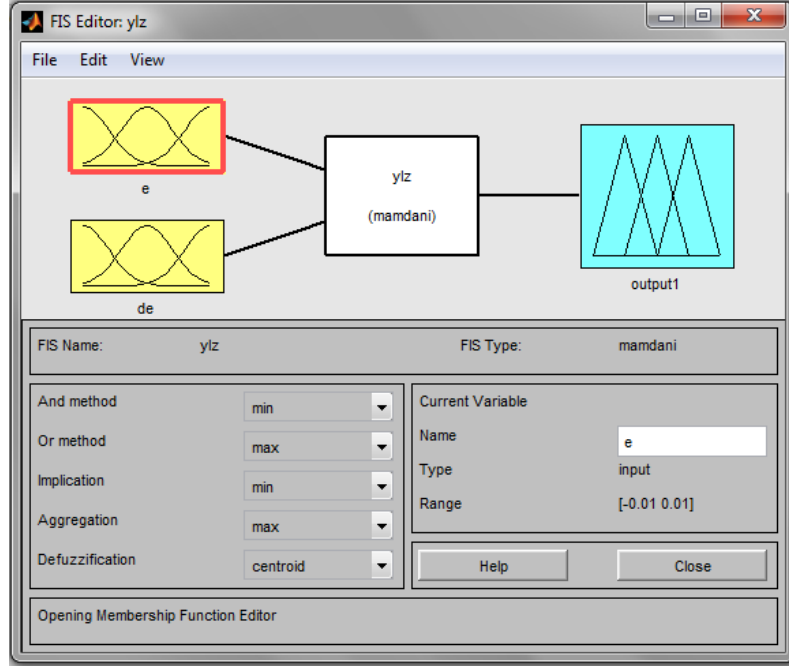
öncül parametrelerin Gradient Descent, sonuç parametrelerin LMS ile ayarlandığı Hibrit Öğrenme metodu kullanılabilir [46]. Böylece daha hızlı sonuç alınmış olur.

## **2. YAPILAN ÇALIŞMALAR**

### **2.1. Adaline Optimizeli Yapay Sinir Ağlarıyla Ayarlanan Bulanık Arayüz Denetleyici**

Yukarıda Literatür taramasından görülmüştür ki akıllı sistemler geleneksel sistemlere kıyasla pek çok yönden daha avantajlıdır ve iki akıllı denetleyici aynı sistemde kombine edilip kullanıldığında ise daha kararlı denetleyiciler elde edilebilmektedir. Bu nedenle önerilen denetleyicide de akıllı sistemler tercih edilmiştir. Daha güçlü bir sistem için Bulanık Mantık ve Yapay Sinir Ağı bir arada kullanılmıştır. Kullanılan Bulanık Mantık denetleyicinin, hata (e) ve hata değişimi (de) olmak üzere iki girişi ve çıkış değişimi (du) olmak üzere bir çıkışı bulunmaktadır. Bu giriş ve çıkışlarının üyelik fonksiyon aralıkları Yapay Sinir Ağı kullanılarak ayarlanmıştır. Böylece hem Bulanık Mantıktaki üyelik fonksiyonlarının belirlenmesi gibi büyük bir sorunun üstesinden gelinmiş hem de Yapay Sinir Ağlarının tekbaşına yapacağından daha iyi bir sonuç elde edilmiştir. Yapay Sinir Ağlarında da en iyi ağırlık ve biasları bulabilmek hem zaman hem de enerji kaybıdır. Bunun için de daha geniş alanda araştırma yaparak en iyi sonucu bulabilen ve parametre değişikliklerinde çok güçlü tavır sergileyebilen ADALINE yönteminde yardım alınmıştır. Böylece istenen çıkışı alabilmek için Yapay Sinir Ağlarında en uygun ağırlık ayarlama problemi kolaylaştırılmış olur.

Önerilen bu yöntemde Bulanık Mantık denetleyici olarak FIS (Fuzzy Inference System) dosyası kullanılmıştır. Yani gerekli kuralların kaydedildiği fis uzantılı bir dosya Matlab Fuzzy bloğundan çağırılmıştır. Fuzzy bloğunun yapısı Şekil 26' da görülmektedir.



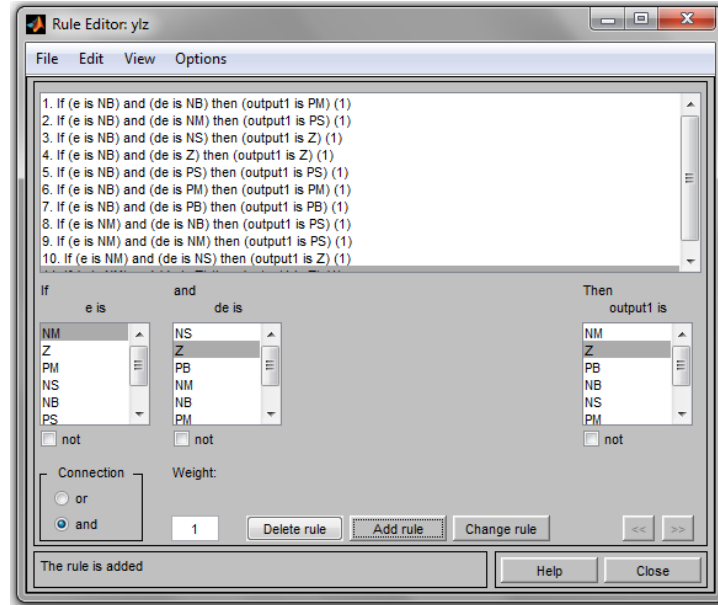
Şekil 26. Bulanık Mantık arayüzü

Fuzzy arayüzü oluşturulurken hata (e) ve hata değişimi (de) olmak üzere iki bulanık giriş belirlenir. Bu girişlerde kullanılacak üyelik fonksiyonlarının şeklinin nasıl olacağı seçilir. Bu çalışma için üçgen üyelik fonksiyonu seçilmiştir. Üçgenin fonksiyonlarının kenar açılarının ne kadar olacağını bulabilmek için üçgenlerin aralıkları ayarlanır. Bu şekilde en uygun üçgen şekli belirlenmiş olur. En uygun aralıkların belirlenmesi önceki çalışmalarda deneme yanılma yoluyla belirlenmekteydi ki bu da çok zaman almakta ve enerji kaybına sebep olmaktadır. Bu çalışmada ise aralıkların belirlenmesi konusunda Yapay Sinir Ağlarından faydalanılmıştır. Bu şekilde kesin değerler bulanık veriye dönüştürülmüştür ve herbir fonksiyona Negatif Büyük (NS), Negatif Orta (NM), Negatif Pozitif Küçük (PB), Sıfır (Z), Pozitif Küçük (PS), Pozitif Orta (PM), Pozitif Büyük (PB) gibi harfsel isimler verilmiştir. Bu dönüşümden sonra kural tabanı oluşturulması gerekir. Bunun için de “ ylz. fis ” dosyası oluşturulmuştur. Bu dosyada hız ve hız değişiminin daha önce (bkz. 1.3.1.1.2. Kural Tabanı) bahsedilen yöntemle kullanı belirlenmiştir. Bu yöntemle göre oluşturulan kural tablosu Tablo 3’ de görülmektedir.

Tablo 3. Bulanık Mantık kural tablosu

de e	NB	NM	NS	ZE	PS	PM	PB
NB	PM	PS	ZE	ZE	PS	PM	PB
NM	PS	PS	ZE	ZE	PS	PM	PM
NS	PS	PS	ZE	ZE	PS	PS	PS
ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE
PS	NS	NS	NS	ZE	ZE	NS	NS
PM	NM	NM	NS	ZE	ZE	NS	NS
PB	NB	NM	NS	ZE	ZE	NS	NM

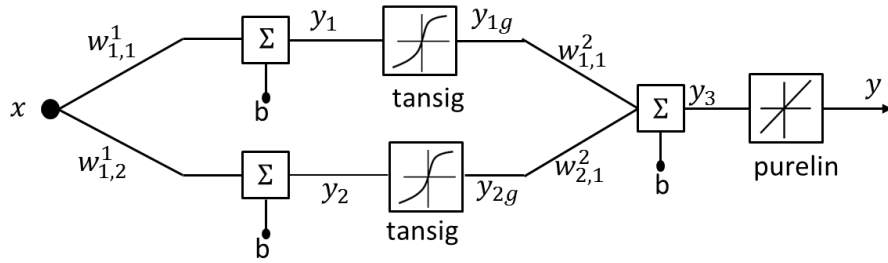
Belirlenen bu kural tablosu Şekil 27'deki görüldüğü gibi 'ylz. fis' dosyasına aktarılmıştır. Bu dosya da Fuzzy arayüzü tarafından çağırılarak çalıştırılır.



Şekil 27. Bulanık Mantık arayüzünde kullanılan. fis dosyası

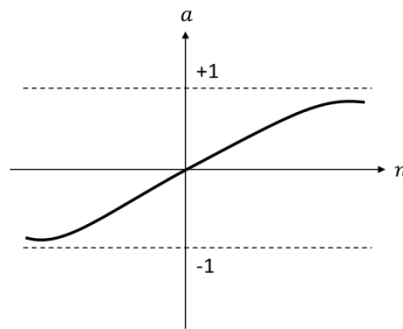
Kuralları ve sonuçları da bu şekilde oluşturulduktan sonra elde edilen bulanık verilerin kullanılabilir keskin değerlere dönüştürülmesi gerekir. Bunun için de açılan bulanık mantık arayüzünden durulaştırma yöntemi seçilir. Bu çalışmada alanların merkezi yöntemi kullanılmıştır. Bu yüzden de durulaştırma seçeneklerinden alanların merkezi yöntemine karşılık gelen 'centroid' seçilir.

Bu dosyadaki üyelik fonksiyonlarını ayarlamak için kullanılan Yapay Sinir Ağı Şekil 28' deki gibidir. Bu Yapay Sinir Ağı'nın ilk katmanında nonlinearluklerin üstesinden gelebilmesi için sigmoid öğrenme fonksiyonu, ikinci katmanında ise işlevi daha basitleştirmesi ve harmoniklerle mücadele edebilmesi için lineer öğrenme fonksiyonu kullanılmıştır. Sigmoid fonksiyonu kullanan her iki katman da aynı girişi kullanmıştır. Bu giriş her iki katmanda ayrı ayrı ağırlaştırılıp eşik değeriyle toplanmıştır ve tansig fonksiyonu uygulanmıştır. Tek giriş tek çıkışlı olan bu katmanların çıkışları ise lineer fonksiyon kullanan katmanın girişleri olarak kullanılmıştır. Bu çıkışlar da ağırlaştırılıp bir eşik değeriyle beraber toplandıktan sonra purelin fonksiyonu uygulanmıştır.



Şekil 28. Önerilen yöntemde kullanılan yapay sinir ağı

Bu ağda kullanılan tansig fonksiyonunun grafiği Şekil 29' da denklemleri ise (44) ve (45) denklemlerinde verilmiştir.

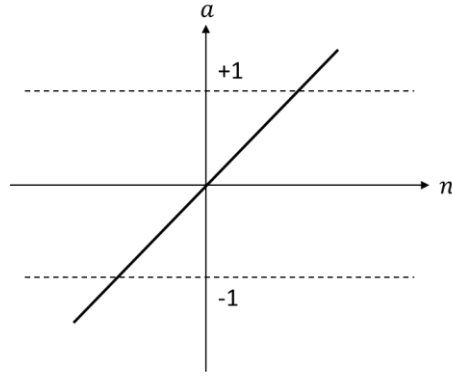


Şekil 29. Tansig fonksiyon grafiği

$$a = \tan \text{sig}(n) \quad (44)$$

$$a = \frac{2}{1 + e^{(-2n)}} - 1 \quad (45)$$

Çıkış katmanında kullanılan purelin fonksiyonunun grafiği ise Şekil 30'da gösterilmiş denklemleri (45) ve (46)' da verilmiştir.



Şekil 30. Purelin fonksiyon grafiği

$$a = \text{purelin}(n) \quad (45)$$

$$a = n \quad (46)$$

Bütün bu aşamalar aşağıdaki (47)–(54) denklemlerleriyle ifade edilmiştir.

$$y_1 = xw_{1,1}^1 + b_1 \quad (47)$$

$$y_2 = xw_{1,2}^1 + b_2 \quad (48)$$

Sigmoid fonksiyonu uygulandıktan sonraki çıkışlar (49) ve (50)' daki denklemler gibidir.

$$y_{1g} = f_{\text{tan sig}}(y_1) = f_{\text{tan sig}}(xw_{1,1}^1 + b_1) \quad (49)$$

$$y_{2g} = f_{\text{tan sig}}(y_2) = f_{\text{tan sig}}(xw_{1,1}^1 + b_2) \quad (50)$$



Birinci katmanın çıkışları ikinci katmanın girişleri olduğuna göre ikinci katmanda toplam işleminin çıkışı denklem (51), diğer bir şekliyle (52) olarak elde edilir.

$$y_3 = y_{1g} w_{1,1}^2 + y_{1g} w_{2,1}^2 + b_3 \quad (51)$$

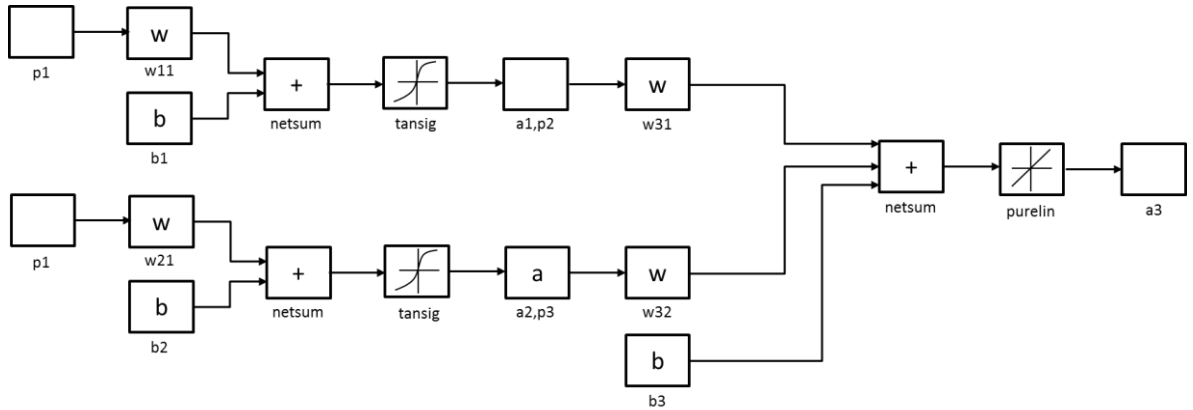
$$y_3 = f_{\tan sig}(xw_{1,1}^1 + b_1)w_{1,1}^2 + f_{\tan sig}(xw_{1,1}^1 + b_2) + b_3 \quad (52)$$

İkinci katmana fonksiyon uygulandıktan sonra nihai çıkış, (53) ve (54) denklemleri ile elde edilir.

$$y = f_{purelin}(y_3) \quad (53)$$

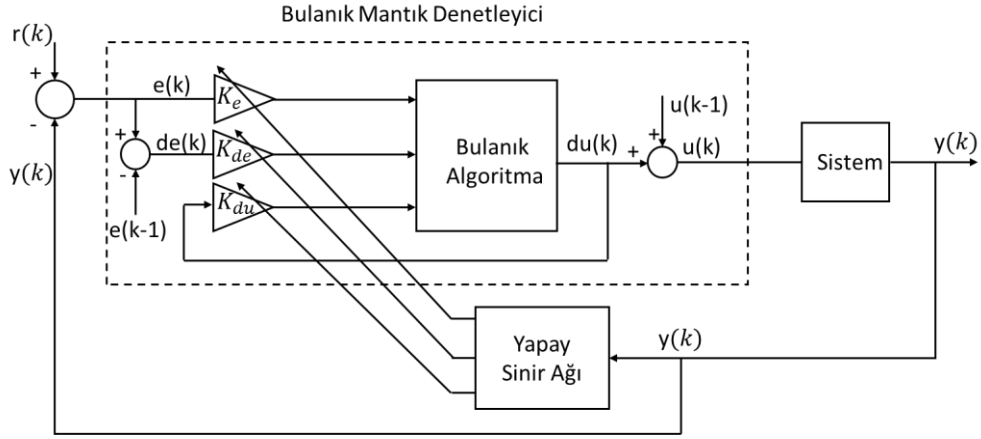
$$y = f_{purelin}(f_{\tan sig}(xw_{1,1}^1 + b_1)w_{1,1}^2 + f_{\tan sig}(xw_{1,1}^1 + b_2) + b_3) \quad (54)$$

Bu Yapay Sinir Ağının Matlab/SIMULINK ortamında oluşturulmuş hali ise Şekil 31' de görülmektedir.



Şekil 31. Sistemde kullanılan yapay sinir ağı benzetim modeli

Kullanılan bu Yapay Sinir Ağı denetimli öğrenme yapısına sahip olduğu için öncelikle çıkışın Yapay Sinir Ağına öğretilmesi gerekir. Bu amaçla da sistem öncelikle Yapay Sinir Ağına sistemin çıkışı uygulanırken çalıştırılır.



Şekil 32. Yapay Sinir Ağı'nın öğrenme

Bu çalışmada öğrenme yöntemi olarak Least Mean Square (LMS) yöntemi kullanılmıştır. Bu yöntemde kullanılan denklemler X sistem girişi, Y sistem çıkışı, D istenen çıkış,  $\eta$  öğrenme katsayısı, E hata ve w ağırlıklar olmak üzere (55)-(59) gibidir.

$$E \cong \frac{1}{2}(D_i - Y_i)^2 \quad (55)$$

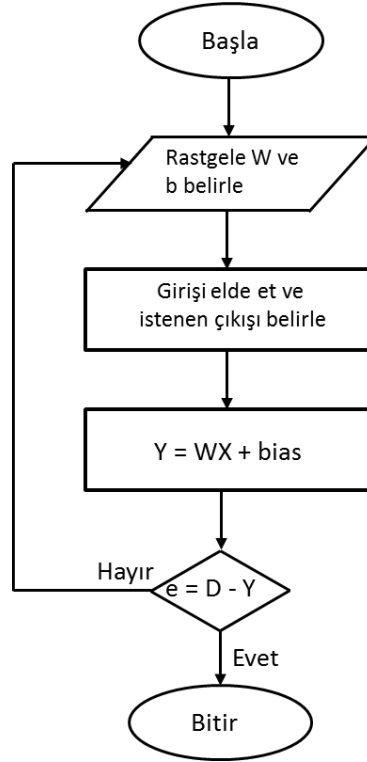
$$\Delta w_i^k = -\eta \left( \frac{\partial E}{\partial w_i} \right) \quad (56)$$

$$\frac{\partial E}{\partial w_i} = (D_i - Y_i) \quad (57)$$

$$\Delta w_i^k = -\eta (D_i - Y_i) X_i \quad (58)$$

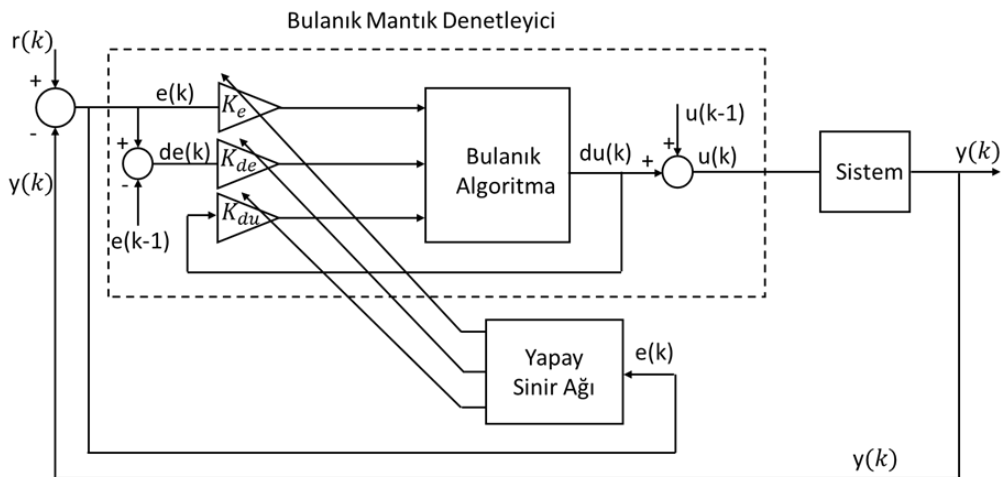
$$w_i^{k+1} = w_i^k + 2\eta (D_i - Y_i) X_i \quad (59)$$

Burada kullanılan YSA'nın öğrenmesi algoritma şeklinde de ifade edilirse Şekli 33' deki algoritma elde edilmiş olacaktır.



Şekil 33. Öğrenme algoritması

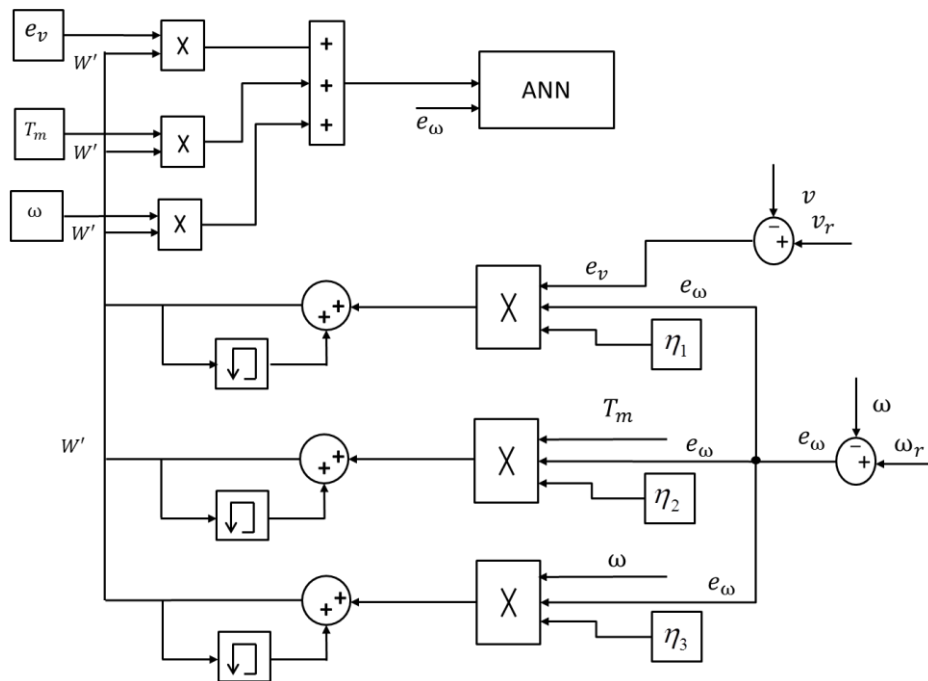
Bu öğretim işleminden sonra ise önerilen metod, sistemde denetleyici olarak kullanılabilir. Şekil 34'de üyelik fonksiyonları YSA yardımıyla ayarlanan BMD' nin denetleyici olarak kullanılması görülmektedir.



Şekil 34. Yapay Sinir Ağlarıyla üyelik fonksiyonlarının ayarlanmasının genel yapısı

Kullanılan bu yöntemde ki Yapay Sinir Ağındaki nöron sayısı arttırıldıkça sistemin bozulmalara karşı daha dayanıklı olduğu görülmüştür. Ancak nöron sayısının bu denli arttırılması işlem süresini çok arttırmaktadır. Bunun yerine sistemde kullanılan ağırlıkların belirlenmesinde Yapay Sinir Ağlarının basit bir şekli olan ADLINE (Adaptive Linear Element) 'den yardım alındığında hem işlem süresinin çok nöronlu YSA' na göre daha az olduğu hem de motorun çalışmaya başladığı anda oluşan dalgalanmaları azalttığı görülmüştür.

Şekil 35'te gösterilen sistemde kullanılan ADALINE' in yapısına bakılacak olursa giriş değerleri, değişim ihtimali çok olan ve bu şekilde harmoniğe sebep olma ihtimali çok olan gerilim, tork ve hız olarak alınmıştır.



Şekil 35. Sistemde kullanılan ADALINE'in yapısı

ADALINE'de de ağırlık güncelleme yöntemi olarak LMS kullanılmıştır ve bu sayede  $w'$  güncel ağırlıkları elde edilmiştir. Daha sonra yeniden bu güncel ağırlıklar girişlerle çarpılmış ve girişler ağırlaştırılmıştır. Bunun denklemlerle ifade edilmesi anlaşılmasını kolaylaştıracaktır. Bu çalışmada kullanılan ADALINE'in ağırlık güncelleme denklemleri (60)-(63)' de verilmiştir.

$$w_{i,k}^{k+1} = w_{i,k}^k + 2\eta(d_k - y_k)x_{i,k} \quad (60)$$

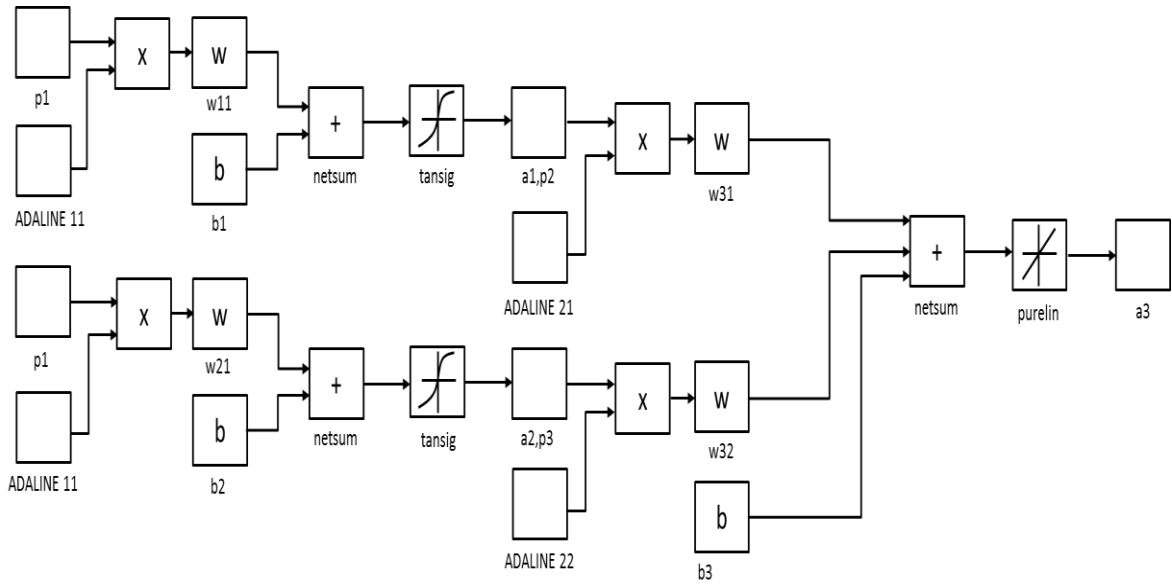
Genel ağırlık güncelleme denklemi, denklem (60)' tır. Bu denklemde  $x$  yerine sırasıyla gerilim hatası, hız ve tork konulursa (61)-(63) denklemlerinde edilir.

$$w_{e_v}^{k+1} = w_{e_v}^k + 2\eta e_v \quad (61)$$

$$w_{T_m}^{k+1} = w_{T_m}^k + 2\eta T_m \quad (62)$$

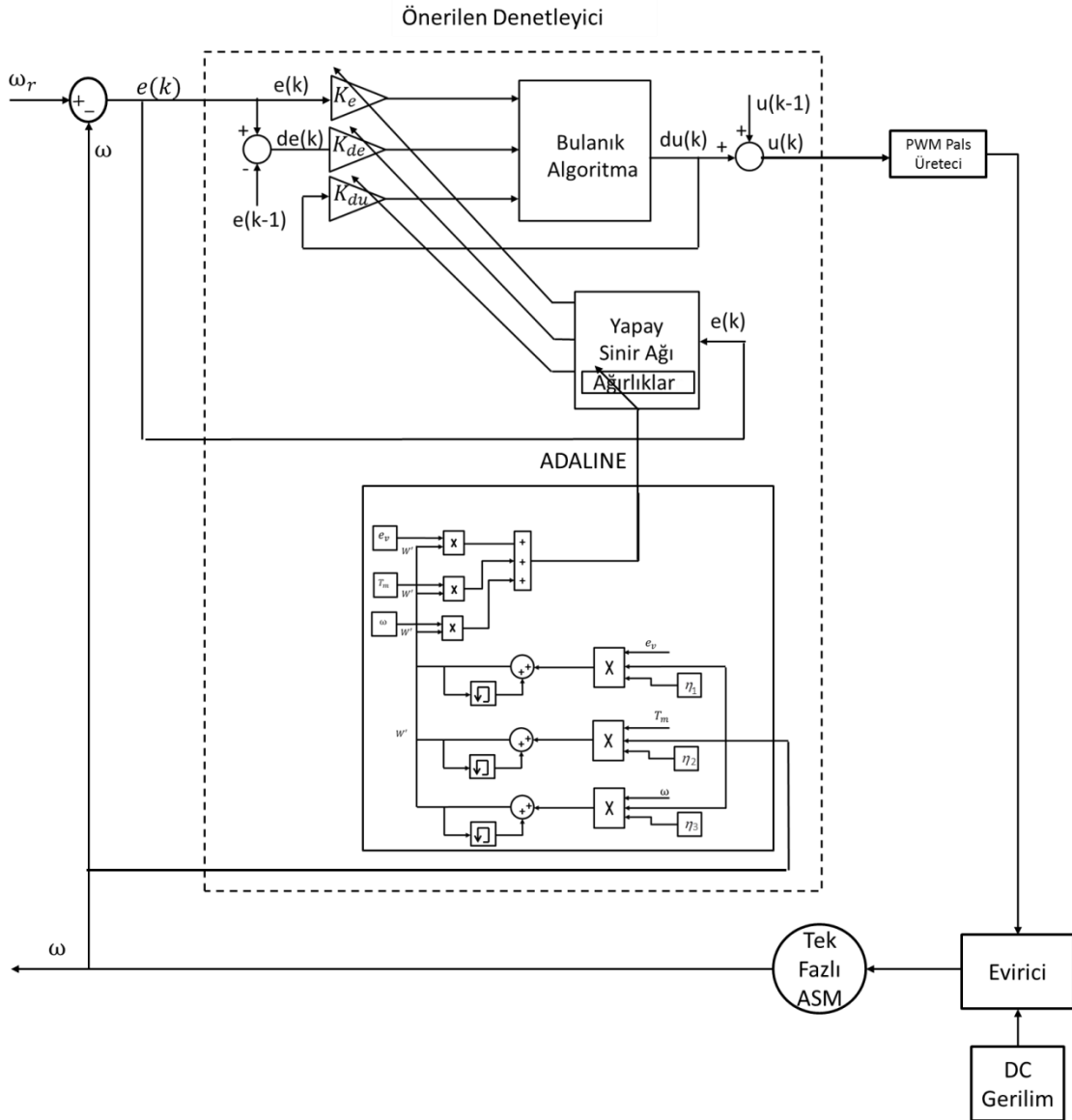
$$w_{\omega}^{k+1} = w_{\omega}^k + 2\eta \omega \quad (63)$$

Bu denklemlere göre ağırlaştırılan girişler toplanarak ağırlık ayarlamak üzere YSA bloğuna uygulanır. Bu şekilde oluşacak Yağay Sinir Ağı Şekil 36' da gösterilmiştir.



Şekil 36. Önerilen denetleyicide kullanılan yapay sinir ağının modeli

Bütün bu aşamalardan sonra önerilen denetleyicinin tek fazlı Asenkron Motorun hızını kontrol etmesinin genel yapısı Şekil 37' de görüldüğü gibi oluşturulabilir.



Şekil 37. Tek fazlı asenkron motorun önerilen denetleyici ile kontrolünün genel şeması

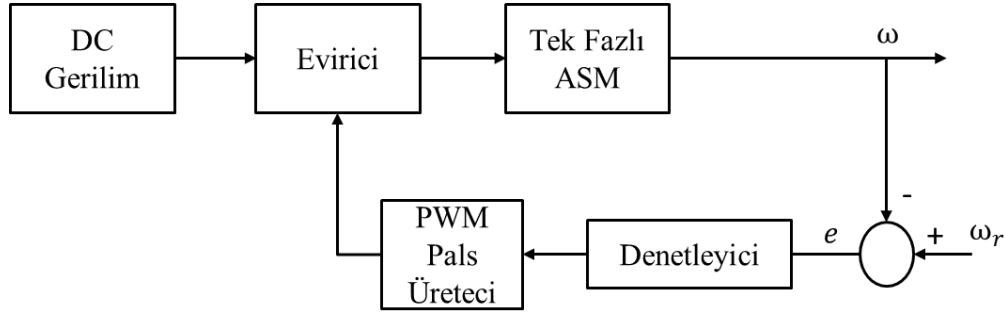
Oluşturulan sistemin genel yapısından da görüldüğü gibi ADALINE yardımıyla ağırlıkları belirlenen bir YSA kullanılarak Bulanık Mantığın üyelik fonksiyonları ayarlanmıştır. Böylece değişikliklere ve dalgalanmalara karşı daha dayanıklı bir sistem elde edilmiştir.

## 2.2. Tek Fazlı Asenkron Motor Hız Kontrolünün Benzetim Uygulamaları

Tek fazlı Asenkron Motorların lineer olmayan yapısından dolayı matematiksel olarak modellenmesi zordur. Bu yapısı, ASM' nin kontrol uygulamalarında çok tercih

edilmemesine sebep olmuştur. Ancak son zamanlarda geliştirilen Bulanık Mantık, Yapay Sinir Ağları gibi matematiksel modelleme gerektirmeyen ve non-linear sistemlerin üstesinden gelebilen akıllı denetleyiciler, tek fazlı asenkron motorun kontrolünü kolaylaştırmış ve zaten kullanımı geniş olan tek fazlı ASM' lerin uygulama alanını gitgide arttırmıştır.

Tek fazlı Asenkron Motorun genel kontrol şeması Şekil 38' de gösterilmiştir.

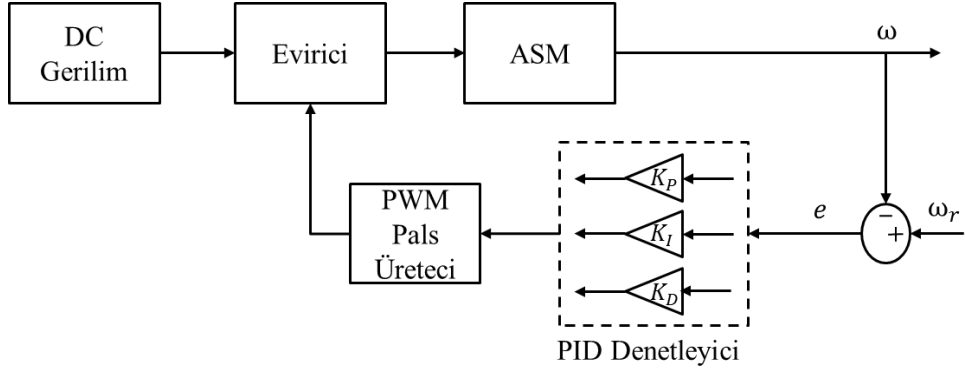


Şekil 38. Tek fazlı asenkron motor genel kontrol şeması

Tek fazlı Asenkron Motoru farklı pek çok yöntemle kontrol edebilmek mümkündür. Bu kısımda, sadece önerilen denetleyiciyle değil diğer yöntemlerle de Matlab/SIMULINK ortamında kontrolü yapılmıştır. Değişken yükte çıktıları elde edilmiştir.

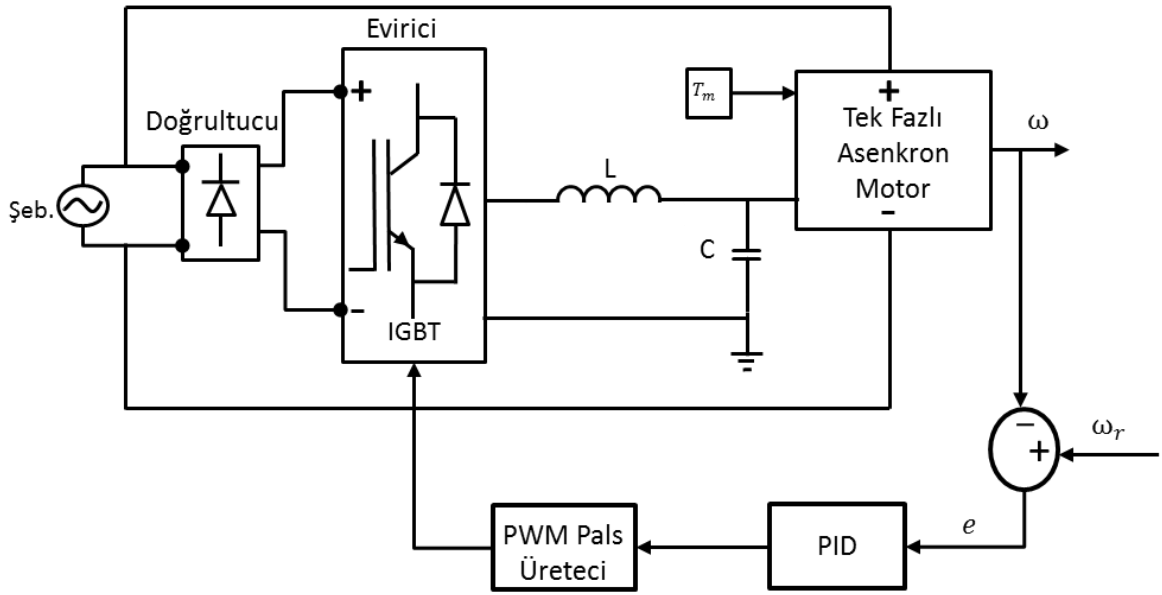
### 2.2.1. Tek Fazlı Asenkron Motorun PID ile Kontrolü

Akıllı denetleyicilerle geleneksel denetleyicilerin farkını ortaya koyabilmek için tek fazlı Asenkron Motor hızı, PID (Proportional-Integral-Derivative) denetleyici ile kontrol edilmiştir. PID ile tek fazlı ASM kontrolünün genel şeması Şekil 39'da görülmektedir.



Şekil 39. Tek fazlı asenkron motorun PID ile kontrolünün genel şeması

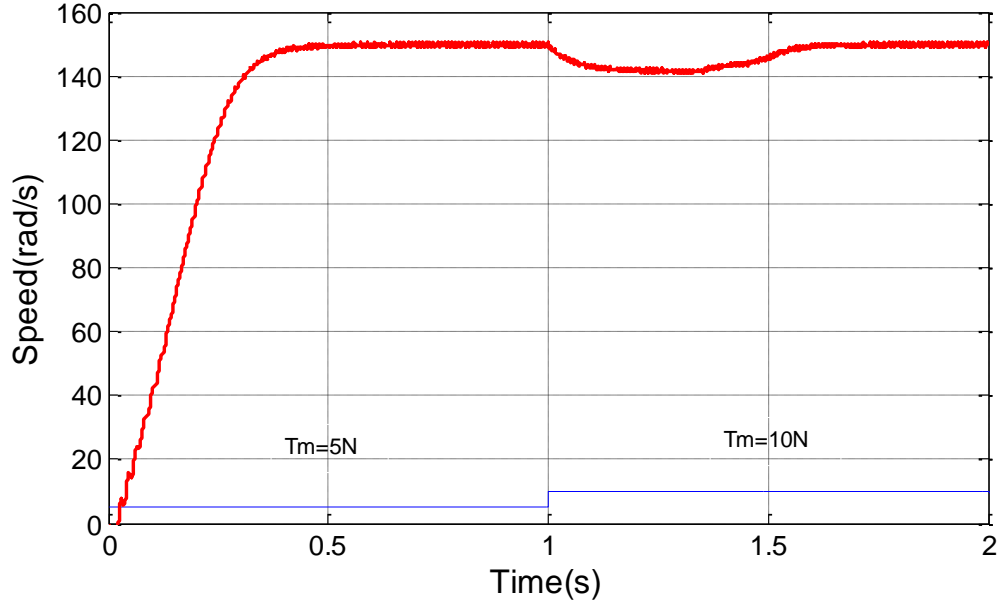
Tek fazlı Asenkron Motorun PID ile denetlenmesinin Matlab/SIMULINK'te uygulaması ise Şekil 40'daki gibi olacaktır.



Şekil 40. Tek fazlı asenkron motorun PID ile kontrolünün benzetim modeli

Bu uygulama sonucu değişken yük için elde edilen çıktı ise Şekil 41'deki elde edilmiştir.



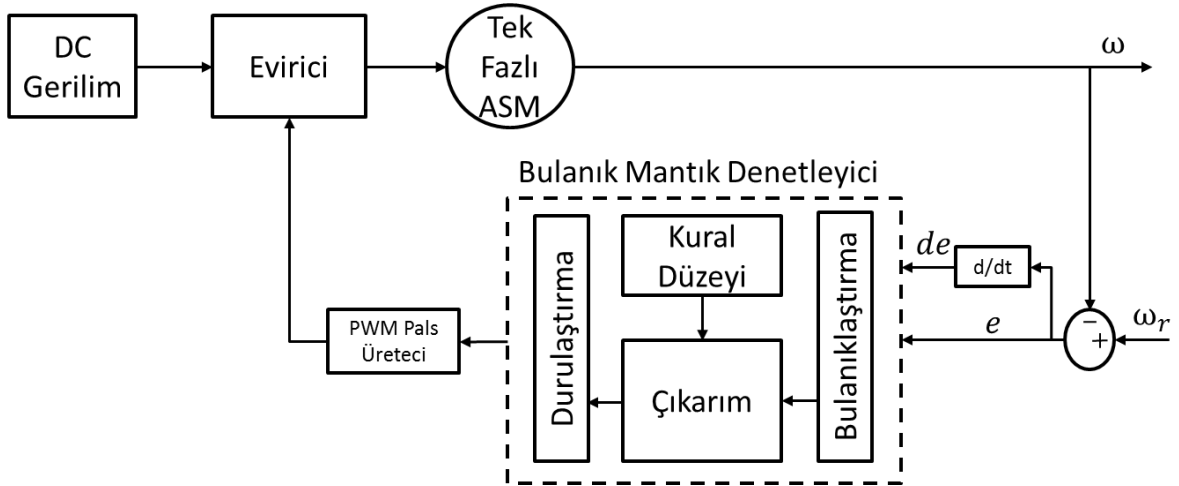


Şekil 41. Değişken yükte PID ile kontrol edilen tek fazlı asenkron motor çıktısı

### 2.2.2. Tek Fazlı Asenkron Motorun Bulanık Mantık ile Kontrolü

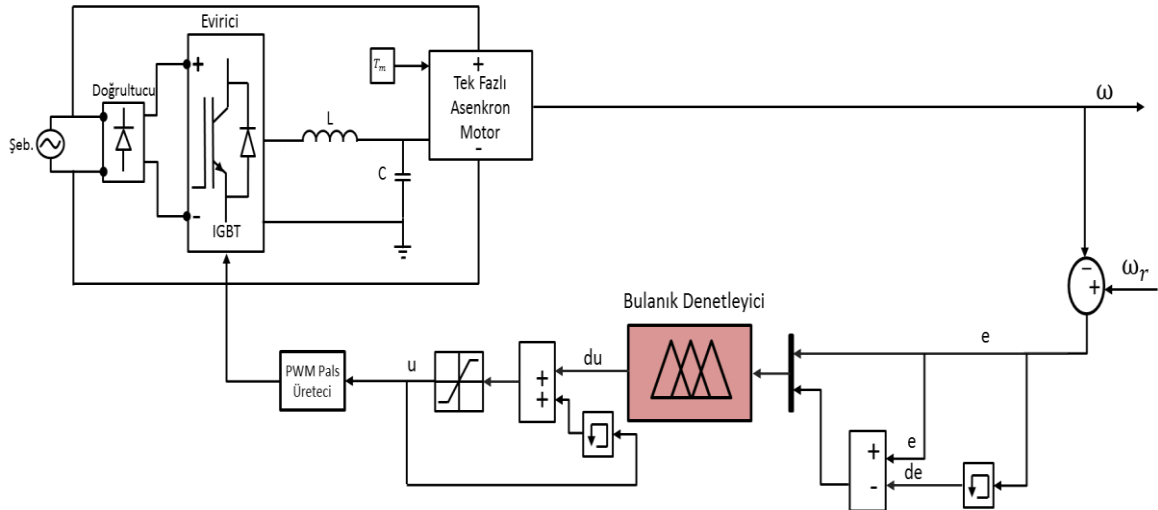
Geleneksel kontrol yöntemlerinin, tek fazlı Asenkron Motor gibi karmaşık sistemler ve lineer olmayan sistemler için yeterli olmaması yeni yöntemler arayışına yöneltmiştir. Kesin bir matematiksel model gerektirmeyen, karmaşık ve lineer olmayan sistemlerde de yanıt verebilen akıllı bir sistem olan Bulanık Mantık, bu tür sistemlerin üstesinden gelebilecek iyi bir alternatif olmuştur.

Tek fazlı ASM' nin Bulanık Mantıkla kontrol edilmesini gösteren genel kontrol diyagramı Şekil 42' de çizilmiştir.



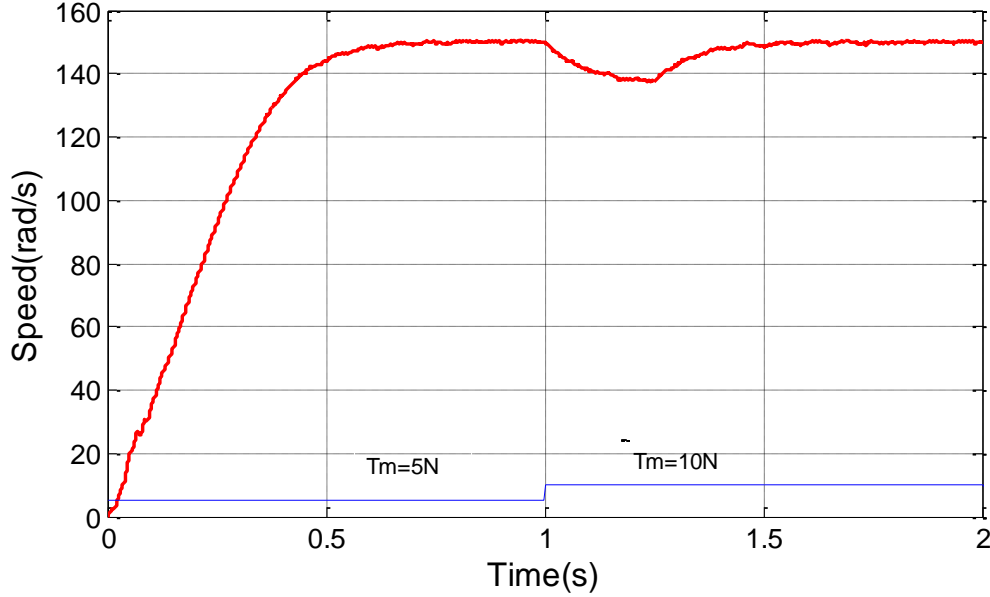
Şekil 42. Tek fazlı asenkron motorun bulanık mantık ile kontrolünün genel şeması

Tek fazlı Asenkron Motorun, Bulanık Mantıkla kontrolünün simulink uygulamasının genel şeması Şekil 43’ de görüldüğü gibi oluşturulmuştur. Hata ve hata değişimi Bulanık Denetleyici bloğunun girişi olarak alınmıştır. Uygulaması sırasında kullanılan kurallar Tablo 3’ de belirtilen kurallarla aynıdır. Bu kurallarla daha önce bahsedildiği gibi ylz. fis dosyasında oluşturulmuş ve Bulanık Denetleyici Bloğu tarafından çağırılmıştır. Sonrasında da durulaştırılan çıkış PWM pals üreticine gönderilmiş ve eviricinin anahtarları bu şekilde ayarlanarak motor hızı kontrol edilmiştir.



Şekil 43. Tek fazlı asenkron motorun bulanık mantık ile kontrolünün benzetim modeli

Matlab/SIMULINK ortamında elde edilen çıktısı değişken yük momenti için ise Şekil 44' deki gibidir.



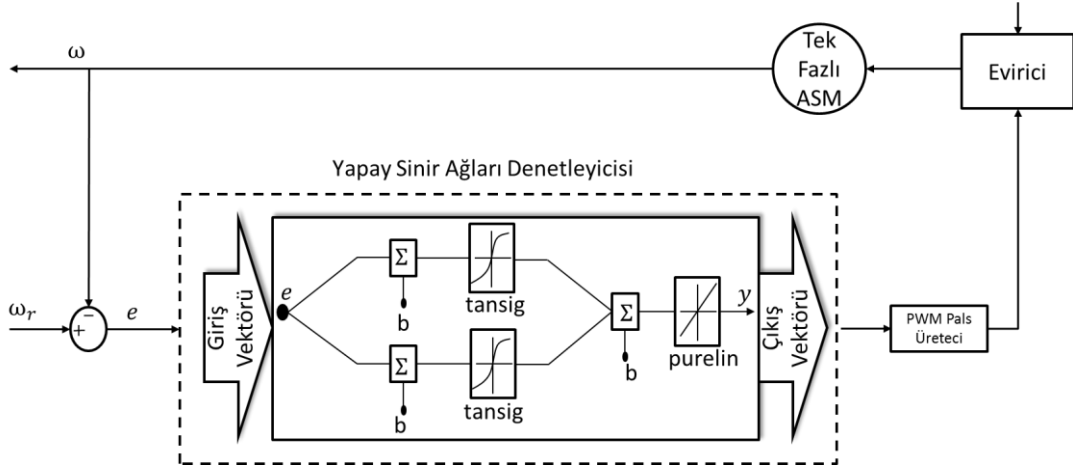
Şekil 44. Değişken yükte bulanık mantık denetleyici ile kontrol edilen tek fazlı asenkron motor çıktısı

Yük momenti 1. saniyeye kadar 5 N iken, 1. saniyeden sonra 10 N' a çıkarılmıştır. Görüldüğü gibi bu değişim sırasında bozulma olmasına rağmen 0,5 s gibi kısa bir sürede motor, 150 rad/s olan referans hızı yakalayabilmiştir.

### 2.2.3. Tek Fazlı Asenkron Motorun Yapay Sinir Ağları ile Kontrolü

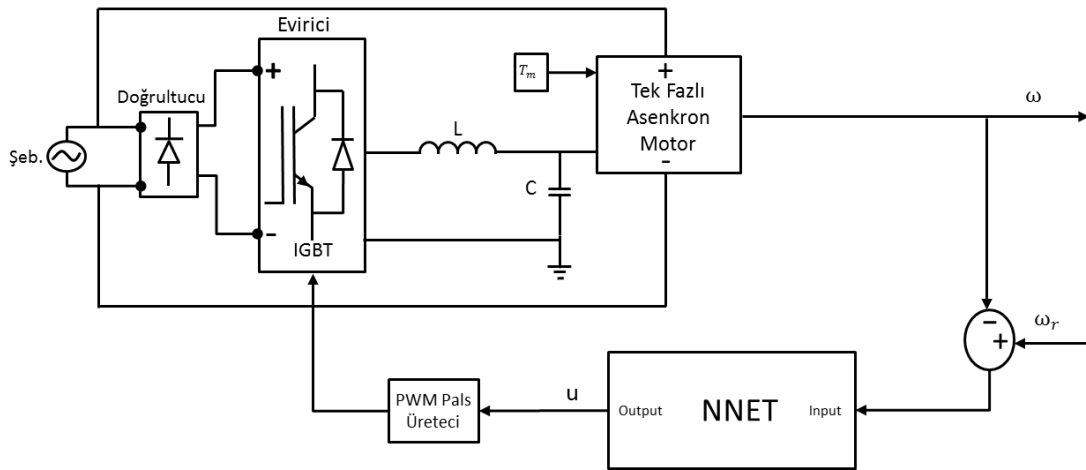
Yapay Sinir Ağları, değişken parametrelere dayanıklı olması istenen sistemler ya da matematiksel modeli kolayca çıkarılamayan sistemler için uygulanabilecek akıllı kontrol yöntemlerinden biridir. Tek fazlı Asenkron Motorun non-lineer yapısından dolayı YSA tek fazlı Asenkron Motor hız kontrol uygulamalarında çokça kullanılan kontrol sistemlerinden biri haline gelmiştir. Bulanık Denetleyiciler de akıllı sistem olmalarına rağmen üyelik fonksiyonlarının belirlenmesi, aralıklarının ayarlanması, kurallarına karar verilmesi gibi dezavantajlarından dolayı Yapay Sinir Ağları ile kontrolü gündeme gelmiştir.

Tek fazlı Asenkron Motorun, Yapay Sinir Ağlarıyla kontrolünün genel şeması Şekil 45' deki gibi çizilebilir.



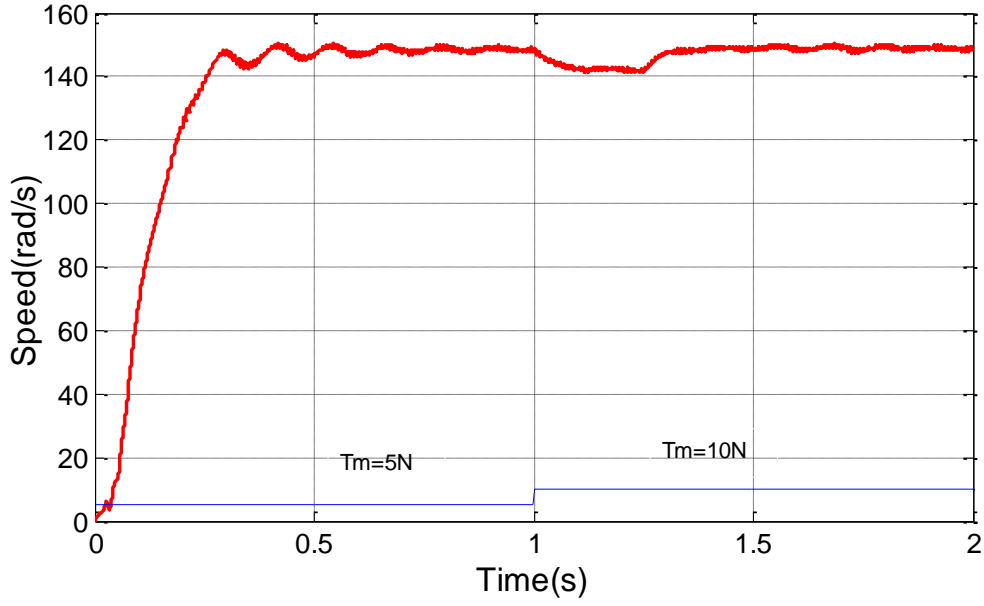
Şekil 45. Tek fazlı asenkron motorun yapay sinir ağları ile kontrolünün genel şeması

Bu sistemin Matlab/SIMULINK'te uygulanması ise Şekil 46' deki gibi oluşturulmuştur. Burada kullanılan NNET bloğunda Şekil 32' de görülen Yapay Sinir Ağı modeli bulunmaktadır. Yapay Sinir Ağı bloğu, alınan çıkış verileri istenen çıkış değerine eşit oluncaya kadar ağırlık güncellemesine devam eder. Bu şekilde motor hızını kontrol etmiş olur.



Şekil 46. Tek fazlı asenkron motorun yapay sinir ağlarıyla ile kontrolünün benzetim modeli

Yapay Sinir Ağıyla kontrol edilen tek fazlı Asenkron Motordan elde edilen değişken yük çıktısı Şekil 47'de görülmektedir.



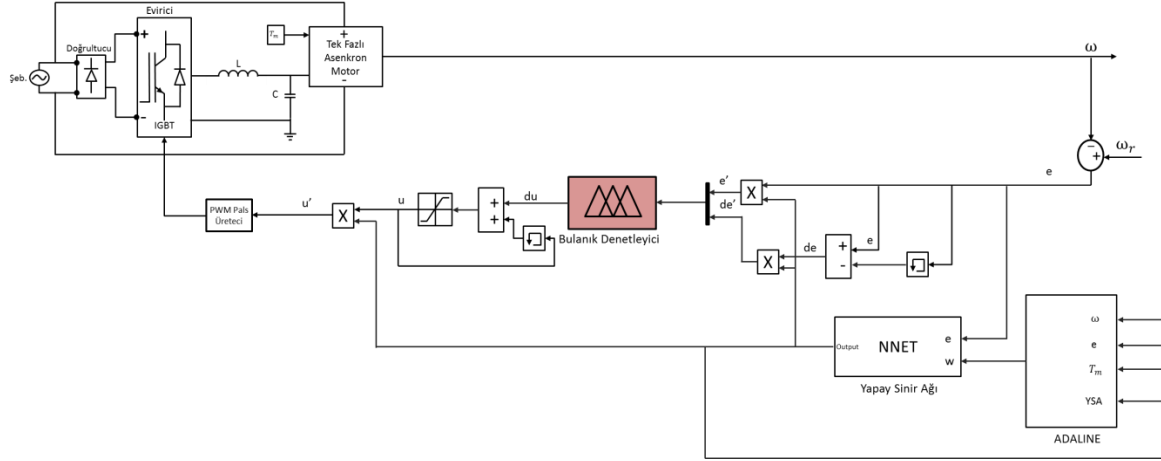
Şekil 47. Değişken yükte yapay sinir ağları ile kontrol edilen tek fazlı asenkron motor çıktısı

Motorun kalkışı ve referans hıza ulaşması 0,5 s'den az sürmüştür. Yük momenti değişimi sırasında hızı en çok 10 rad/s düşmüş ve 0,25 s gibi kısa bir sürede 150 rad/s referans hızını yakalayabilmiştir. Ancak YSA'nda dalgalanmalar çok olmaktadır. Bu da YSA'nın neden tek başına tercih edilmeyip daha çok başka bir denetleyiciyle beraber kullanılmasının yanıtıdır.

### 2.2.5. Tek Fazlı Asenkron Motorun Önerilen Denetleyici ile Kontrolü

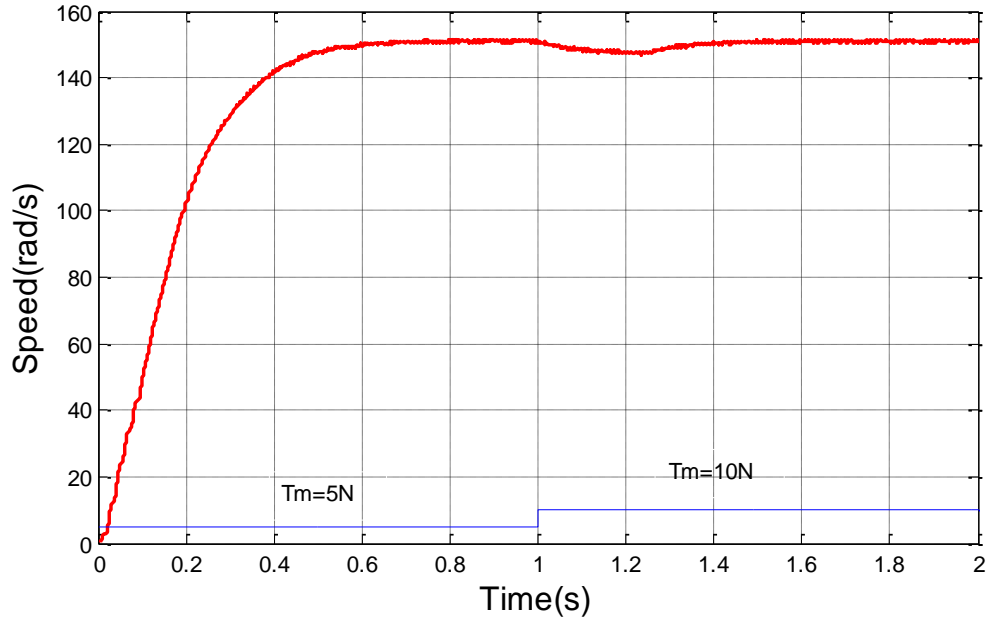
Bulanık Mantıktaki üyelik fonksiyonlarının belirlenme sorununa ortadan kaldırmak için tasarlanan bu denetleyici yöntemi hem daha hızlı işlem kabiliyeti, hızlı cevap vermesi hem de değişikliklere daha dayanıklı olup daha hızlı takip etme özellikleriyle pek çok uygulama için uygun bir kontrolör olacağı düşünülmüştür.

Önerilen denetleyici ile tek fazlı Asenkron Motor kontrolünün Matlab/SIMULINK' te oluşturulmuş modeli Şekil 48' dedir. Görüldüğü gibi Bulanık mantığın üyelik fonksiyon aralıkları YSA yardımıyla ayarlanmış ve YSA' nın ağırlıklarının ayarlanmasında da ADALINE' den yardım alınmıştır.



Şekil 48. Tek fazlı asenkron motorun önerilen denetleyici ile kontrolünün benzetim modeli

Önerilen denetleyici ile denetlenen tek fazlı ASM' nin değişken yük için elde edilen çıktı ise Şekil 49'da görülmektedir.

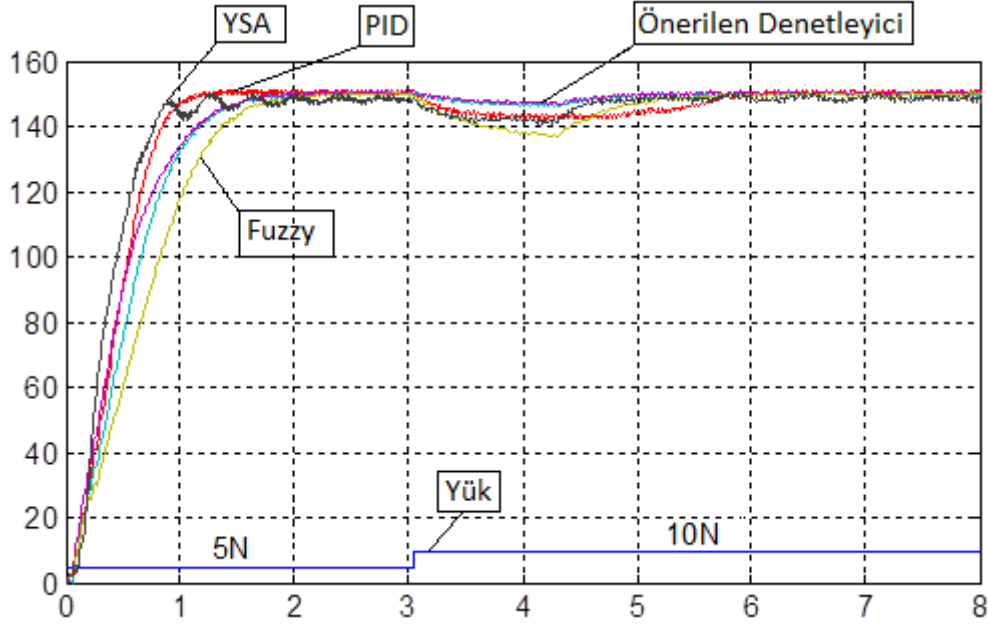


Şekil 49. Önerilen denetleyici ile denetlenen tek fazlı ASM'nin çıktısı

Çıktıda görüldüğü gibi kısa sürede 150 rad/s olan referans hız kısa sürede yakalanmıştır. Motor çalışmaya başladığında oluşan bozulmalar azalmıştır. Yük momenti değişimi sırasında hızı çok az düşmüştür ve çok kısa sürede kendini toparlamıştır. Önerilen bu denetleyicide oluşan salınımlar da çok azdır.

### 3. TARTIŞMA VE SONUÇLAR

Bu çalışmada, tek fazlı Asenkron Motorun hızı tasarlanan yeni bir denetleyici ile kontrol edilmiştir. Tasarlanan denetleyicide ana denetleyici Bulanık Mantıktır ve üyelik fonksiyon aralıklarının ayarlanmasını Yapay Sinir Ağları kullanarak yapmaktadır. Bu denetleyicide kullanılan Yapay Sinir Ağının ağırlıkları ise bir başka YSA çeşidi olan ADALINE yardımıyla ayarlanmıştır. Çalışmada öncelikle tezin anlaşılmasının kolaylığı açısından gerekli görülen bazı terimler hakkında genel bilgiler verilmiştir. İkinci kısmında ise yapılan çalışmalara yer verilmiştir. Öncelikle önerilen denetleyicinin nasıl oluşturulduğu hakkında genel bilgiler verilmiş daha sonra çeşitli denetleyicilerle yapılan tek fazlı ASM hız kontrol benzetim uygulamalarından ve bunların çıktılarından bahsedilmiştir. Bu kısımda ise elde edilen bu çıktıların kıyaslaması yapılacaktır.



Şekil 50. Elde edilen Matlab çıktılarının kıyaslanması

Şekil 50’de gösterilen tez sonucunda elde edilen çıktılara göre şu sonuçlar çıkarılabilir:

- PID denetleyici parametreleri ayarlandığında sabit yükte kabul edilir sonuç vermektedir. Ancak farklı her yük için yeniden parametre ayarı yapmak

gerekmektedir. En uygun  $K_P$ ,  $K_I$  ve  $K_D$  değerleri bulmak çok zor olduğundan zaman kaybına sebep olmaktadır. Bazen bulunan bu en uygun değerler, çok yüksek ve çok düşük olabilir. Bu da gerçeklikten uzaklaşmakta ve gerçek hayatta uygulanabilirliğini zorlaştırmaktadır. Değişken yükte ise yükün değiştiği anda çok fazla bozulmakta ve takipte zorluk çekmektedir.

- Bulanık Mantık denetleyici PID denetleyiciye göre daha hızlı oturmuştur. Değişken yükte ise akıllı bir sistem olduğundan bozulması daha az olmaktadır. Ancak Bulanık Mantık Denetleyicide parametrelerin ayarlanması sorun olduğu için tek başına kullanılması çok da uygun olmamaktadır.
- Yapay Sinir Ağlarında cevap verme konusunda diğerlerine nispeten daha hızlıdır. Kısa sürede referans hıza ulaşabilmiştir. Değişken yüke karşı daha dayanıklı olduğu görülmektedir. Daha az bozulmuş ve kendisini kısa sürede toparlamıştır. Ancak tek başına kullanımında küçük salınımlar oluşturabilir. Ayrıca yeterli öğrenme verisini bulabilmek de Yapay Sinir Ağlarının sorunu olduğu için bu da tek başına yeterli değildir.
- Önerilen denetleyici ise değişken yükte diğer denetleyicilere göre çok daha az bozulmuştur ve kısa sürede kendisini toparlamıştır. Motorun kalkışı sırasında oluşan bozulmalar da önerilen denetleyide diğer denetleyicilere nispeten daha azdır. Salınımlar Yapay Sinir Ağlarına nispeten çok daha azalmıştır.



#### **4. ÖNERİLER**

Tezde kullanılan, önerilen denetleyicideki Bulanık Mantıkta üyelik fonksiyonlarının aralıkları belirlenmiştir. Ancak sadece bu aralıkları belirlemenin yanında üyelik fonksiyonlarının şekillerini belirleyebilecek bir optimize metodu geliştirilebilir. Bulanık Mantıktaki kural tabelasını belirleyebilecek bir özellik eklenebilir.

Tezde kullanılan önerilen denetleyicideki Yapay Sinir Ağlarının en iyi neuron sayısı elle test edilerek belirlenmiştir. Bunun yerine en iyi neuron sayısını belirleyebilecek bir optimize yöntemi eklenebilir.

ADALINE yönteminde değişkenlerin ağırlıkları iki kez güncellendiğinde daha iyi sonuç verdiği görülmüştür. Kaç kez güncelleme yaptığında en iyi sonucu vereceğini belirleyen bir algoritma ile kullanılan ADALINE denetleyici bu yönden adaptif yapılabilir.

## 5. KAYNAKLAR

1. Holtz, J., Speed Estimation and Sensorless Control of AC Drives, Proc.IEEE Int. Conf. on Ind. Elec. Cont. and Intr. (IECON'93), Kasım 1993, Maui, Bildiriler Kitabı: 649-654.
2. Shyu, K., K., Shieh, H., J. ve Fu, S., S., Model Reference Adaptive Speed Control for Induction Motor Drive Using Neural Networks, IEEE Transactions on Industrial Electronics, 45, 1 (1998) 180-182.
3. Fonseca, J., Afonso, J., L., Martins, J., S. ve Couto, C., Fuzzy Logic Speed Control of an Induction Motor, Microprocessors and Microsystems, 22, 9 (1999) 523–534.
4. Denai, M. A. ve Attia, S. A., Fuzzy And Neural Control Of An Induction Motor, Int. J. Appl. Math. Comput. Sci., 12, 2 (2002) 221–233.
5. Miloudi, A., AI Radadi, E. A., Draou, A. ve Miloud, Y., Simulation and Modelling of a Variable Gain PI Controller For Speed Control of a Direct Torque Neuro Fuzzy: Controlled Induction Machine Drive, 35th Annual IEEE Power Electronics Specialists Conference, Haziran 2004, Achen , Bildiriler Kitabı: 3493-3498.
6. Yousef , H.A. ve Wahba, M.A., Adaptive Fuzzy Mimo Control of Induction Motors, Expert Systems with Applications, 36, 3 (2009) 4171–4175.
7. Gadoue, S.M., Giaouris, D. ve Finch J., W., Artificial Intelligence-Based Speed Control of DTC Induction Motor Drives - A Comparative Study, Electric Power Systems Research, 79, 1 (2009) 210–219.
8. Wang, D.ve Shi Y., Sensorless Fuzzy Speed Control For Single Phase Induction Motors, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Washington , Ağustos 2009, Bildiriler Kitabı: 163 -165.
9. Gupta, R., A., Kumar, R. ve Surjuse, R., S., An Adaptive Neuro-Fuzzy Based Speed Sensorless Induction Motor Drives, IEEE Nature & Biologically Inspired Computing, Coimbatore, Aralık 2009, Bildiriler Kitabı: 397-402.
10. Gupta, R. A., Kumar, R. ve Surjuse, R.S., ANFIS Based Intelligent Control of Vector Controlled Induction Motor Drive, Second International Conference on Emerging Trends in Engineering and Technology, Nagpur, Aralık 2009, Bildiriler Kitabı: 675-680.
11. Neema, D., D., Patel, R., N. ve Thoke, A., S., Speed Control of Induction Motor using Fuzzy Rule Base, International Journal of Computer Applications(0975-8887), 33, 5(2011).

12. Dongale, T., D., Kulkarni, T., G., Jadhav, S., R., Kulkarni, S., V. ve Mudholkar, R., R., AC Induction Motor Control - A Neuro-Fuzzy Approach, International Journal of Engineering Science & Advanced Technology[IJESAT], 2, 4 (2012) 863-870.
13. Karthikeyan, B. ve Vidhya, D., S., Performance Analysis of Neuro Fuzzy Based Speed Control of Three Phase Induction Motor, International Journal of Communications and Engineering, 5, 5 (2012) 8-14.
14. Menghal, P.M. ve Jaya Laxmi, Dr. A., Adaptive Neuro Fuzzy Based Dynamic Simulation of Induction Motor Drives, 2013 IEEE International Conference on Fuzzy Systems (FUZZ), Temmuz 2013, Hyderabad, Bildiriler Kitabı: 1-8.
15. Bouhoune, K., Yazıd, K. ve Boucherit ,M., S., Fuzzy Logic based control of the Single Phase Induction Motor without Start-Capacitor, 4th International Conference on Power Engineering, Energy and Electrical Drives, İstanbul, Mayıs 2013, İstanbul, Bildiriler Kitabı: 747-752.
16. Saçkan, A.H., Elektrik Makinaları III: Tek fazlı asenkron motorlar, Milli Eğitim Bakanlığı Mesleki ve Teknik Öğretim Kitapları Etüd ve Programlama Dairesi Yayınları, İstanbul, 1981.
17. Sarıoğlu, K., Elektrik Makinalarının Temelleri: Asenkron Makinalar, Çağlayan Kitabevi, İstanbul,1983.
18. Kaygısız, F., Asenkron Makinaların Matlab Tasarımı ve Analizi, Yüksek Lisans Tezi, Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü, Kütahya, 2008.
19. Bal, G., Özel Elektrik Makinaları, Seçkin Yayıncılık, Ankara, 2006.
20. Özdemir, M., Yardımcı Sargılı Asenkron Makinanın Doymalı Davranışının incelenmesi, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Trabzon, 1991.
21. Pala, O., Yılmaz O., Göçmen R., Tek Fazlı Asenkron Motorların Hız Kontrolü, Bitirme Tezi, Marmara Üniversitesi Teknik Eğitim Fakültesi, İstanbul, 2002
22. Sivanandam, S., N., Sumathi, S., Deepa, S., N., Introduction to Fuzzy Logic Using MATLAB, Springer-Verlag, Berlin, 2007.
23. Zadeh L., A., Outline of A New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. Systems, Man, and Cybernetics, SMC-3, 1 (1973) 28-44.
24. Sen, P., C., Principles of Electric Machines and Power Electronics, Wiley, New York, 1988.
25. Mamdani E., H. ve Assilian S., An Experiment in Linguistic Synthesis with A Fuzzy Logic Controller, Int. J. Man-Machine Studies, 7, (1975) 1-13.

26. Lee C., C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller I, IEEE Trans. Systems, Man, and Cybernetics, 20, 2 (1990) 404 –418.
27. Altas, I., H. ve Sharaf, A., M., A Generalized Direct Approach for Designing Fuzzy Logic Controllers in Matlab/Simulink GUI Environment, International Journal of Information Technology and Intelligent Computing (Int. J. IT&IC), 1, 4 (2007).
28. Baaklini N. ve Mamdani, E., H., Prescriptive Methods for Deriving Control Policy in A Fuzzy-Logic Controller, Electron. Lett., 11, (1975) 625-626.
29. King, P., J. ve Mamdani, E., H., The Application of Fuzzy Control Systems to Industrial Processes, Automatica, 13, 3 (1977) 235-242.
30. Rojas, R., Neural Networks A Systematic Introduction, Springer-Verlag, Berlin, 1996.
31. Chakraborty, R., C., Fundamental of Neural Network:Soft Computing Course Lecture Notes, Jaypee University of Engineering and Technology, 2007.
32. Krenker, A., Bešter J. ve Kos, A., Introduction to the Artificial Neural Networks, 2011.
33. Altaş, İ., H., Neural Fuzzy Systems Ders Notları, Karadeniz Teknik Üniversitesi, 2010.
34. Wilamowski, B., M. ve Irwin, J., D.,The Industrial Electronics Handbook: Intelligent Systems, CRC Press, USA, 2011.
35. Widrow, B. ve Lehr, M., A., 30 Years of Adaptive Neural Networks: Perceptrons, Madaline And Backpropagation, IEEE Proc., 78, 9 (1990) 1415-1442.
36. Zhou ,Y., AI, Q. ve XU,W., Adaline and Its Application in Power Quality Disturbances Detection, 5th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Eylül 2005, Malta, Bildiriler Kitabı: 41-46.
37. Dayhoff, J., Neural Network Architectures: An Introduction, Van Nostrand Reinhold, New York, 1990.
38. Lee, S., C. ve Lee, E., T., Fuzzy Sets and Neural Networks, Journal of Cybernetics, 4 (1974) 83 -103.
39. Lea, R., Jani, Y. ve Berenji, H., Fuzzy Logic Controller with Reinforcement Learning for Proximity Operations and Docking, Fifth IEEE International Symposium on Intelligent Control, Ağustos 1990, Dalian, Bildiriler Kitabı: 903-906.

40. Archer, K., P. ve Wang, S., Fuzzy Set Representation of Neural Network Classification Boundaries, IEEE Transactions on Systems, Man, and Cybernetics, 21, 4(1991) 735-742.
41. Altrock, C., ve Krause, B., Fuzzy Logic and Neurofuzzy Technologies in Embedded Automotive Applications, Third International Conference on Industrial Fuzzy Control and Intelligent Systems, 1993., Houston, Bildiriler Kitabı: A113- 1 - A113-9.
42. Vieira, J., Dias, F., M. ve Mota, A., Neuro-Fuzzy Systems: A Survey, 5th WSEAS NNA International Conference on Neural Networks and Applications, 2014, Udine, Bildiriler Kitabı: 101-104.
43. Azar, A.T., El-Said, S. A., Superior Neuro-Fuzzy Classification Systems, Springer-Verlag, London, 2012.
44. Azar, A.,T, Adaptive Neuro-Fuzzy Systems, Intech, India, 2010.
45. Lin, C., T., ve Lee, C., S., Neural-Network-Based Fuzzy Logic Control and Decision System, IEEE Transactions on Computers, 40, 12 (1991) 1320-1336.
46. Jang, R. , Neuro-Fuzzy Modelling: Architectures, Analysis and Applications, PhD Thesis, University of California, Berkley, 1992.
47. Berenji, H., R ve Khedkar, P., Learning and Tuning Fuzzy Logic Controllers through Reinforcements, IEEE Transactions on Neural Networks, 3, 5 (1992) 724-740.
48. Juang, F., C., Lin, C., T., An On-Line Self Constructing Neural Fuzzy Inference Network and Its Applications, IEEE Transactions on Fuzzy Systems, 6, 1 (1998) 12-32.
49. Jang, J., S., R., ANFIS: Adaptive-Network-Based Fuzzy Inference System, IEEE Transactions on Systems, Man and Cybernetics, 23, 3 (1993) 665–685.
50. Jang, J., S., R. ve Sun, C.,T., Neuro-Fuzzy Modeling and Control, Proceedings of the IEEE, 83, 3 (1995) 378-406.
51. Jang, J., S., R., Sun, C., T. ve Mizutani, E., Neuro-Fuzzy and Soft Computing - A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, USA, 1997.
52. Takagi T., ve Sugeno M., Fuzzy Identification of Systems and Its Applications to Modeling and Control, IEEE Trans. Syst. Man Cybern., SMC-15, 1 (1985) 116–132.

53. Poston, T., Lee, C., N., Choie, Y., J. ve Kwon, Y., Local Minima and Back Propagation, Seattle International Joint Conference on Neural Networks, Temmuz 1991, Seattle, Bildiriler Kitabı: 173–176.
54. Jang J., S., R., Neuro-Fuzzy Modeling: Architectures, Analyses, and Application, Doktora Tezi, University of California, Berkeley, 1992.
55. Chakraborty, R., C., Fundamental of Genetic Algorithm: Artificial Intelligence Lecture Notes, Dept. of Computer Science & Engineering, Jaypee University of Engineering and Technology (JUET), Guna, 2006-7-8-9-10.

## **ÖZGEÇMİŞ**

01.01.1989 tarihinde Artvin’de doğdu. İlköğrenimini Şavşat’ta liseyi Artvin merkezde tamamladı. 2006 yılında Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümünden 2011 yılında Elektrik-Elektronik Mühendisi ünvanıyla mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi Fen Bilimler Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’ndan yüksek lisans öğrenimine başladı. Yükseköğreniminin ilk yılına devam ederken aynı zamanda Elektrik Elektronik mühendisi olarak çalıştı. Yüksek lisansının ikinci yılı erasmus öğrenci değişimi programı kapsamında Kassel Üniversitesi’nde geçirdi. Yabancı dil olarak İngilizce, Almanca, Gürcüce bilmektedir.