

**T.C.  
SÜLEYMAN DEMİREL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BAĞLAŞIK İNDÜKTÖRLÜ ÜÇ-SEVİYELİ EVİRİCİ İLE ASENKRON  
MOTORLARIN HIZ KONTROLÜ İÇİN GENETİK TABANLI BULANIK  
MANTIK DENETLEYİCİ SİMÜLATÖR TASARIMI**

**Alphan MAT**

**Danışman  
Yrd. Doç. Dr. Okan BİNGÖL**

**YÜKSEK LİSANS TEZİ  
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI  
ISPARTA - 2013**

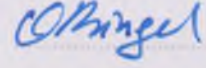
© 2013 [Alphan MAT]

## TEZ ONAYI

Alphan MAT tarafından hazırlanan "**Bağlaşık İndüktörlü Üç-Seviyeli Evirici ile Asenkron Motorların Hız Kontrolü için Genetik Tabanlı Bulanık Mantık Denetleyici Simülasyon Tasarımı**" adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Elektronik Bilgisayar Eğitimi Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak başarı ile savunulmuştur.

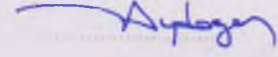
**Danışman**

**Yrd. Doç. Dr. Okan BİNGÖL**  
Süleyman Demirel Üniversitesi



**Jüri Üyesi**

**Doç. Dr. Tuncay AYDOĞAN**  
Süleyman Demirel Üniversitesi



**Jüri Üyesi**

**Yrd. Doç. Dr. Kubilay TAŞDELEN**  
Süleyman Demirel Üniversitesi

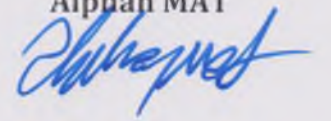


**Enstitü Müdürü**

**Prof. Dr. Mehmet Cengiz KAYACAN**

## TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Alphan MAT  


## İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET .....	iii
ABSTRACT .....	iv
TEŞEKKÜR.....	v
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ .....	viii
SİMGELER VE KISALTMALAR DİZİNİ.....	ix
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	7
3. ASENKRON MOTORLAR VE EVİRİCİLER .....	11
3.1. Asenkron Motorlar .....	11
3.1.1. Üç fazlı asenkron motorun matematiksel modeli.....	11
3.1.2. Üç fazdan iki faza dönüşüm .....	15
3.1.3. Asenkron motorun modelinde kullanılan referans düzlemleri..	16
3.1.4. Dolaylı vektör kontrol yöntemi.....	20
3.2. Üç-Seviyeli Evirici ve Modülasyon Teknikleri .....	23
3.2.1. İki seviyeli gerilim beslemeli evirici.....	24
3.2.2. Üç-seviyeli gerilim beslemeli eviriciler.....	25
3.2.3. Bağlaışık indüktörlü üç-seviyeli eviriciler .....	29
3.2.4. Çok seviyeli eviricilerde DGM teknikleri.....	30
3.2.5. Bağlaışık indüktörlü üç-seviyeli evirici ve uzay vektör DGM .....	31
3.3. Bulanık Mantık.....	41
3.3.1. Bulanık küme kuramı ve bulanık mantık.....	42
3.3.2. Bulanık mantık denetleyici.....	54
3.4. Geleneksel Tasarımda Karşılaşılan Temel Problemler .....	56
3.5. Genetik Algoritmalar ile Optimizasyon .....	58
3.5.1. Genetik algoritmalar .....	58
3.5.2. Gerçek kodlu genetik algoritmalar .....	60
3.5.3. Parametrelerin kodlanması ve amaç fonksiyonu .....	60
3.5.4. Başlangıç popülasyonu.....	62
3.5.5. Uygunluk değerlerinin hesaplanması.....	63
3.5.6. Değerlendirme ve doğal seçim .....	63
3.5.7. Eşleştirme.....	65
3.5.8. Çaprazlama .....	67
3.5.9. Mutasyon .....	69
3.5.10. Elitizm .....	69
3.5.11. Optimizasyon işlemi .....	70
3.6. Genetik Algoritma Tabanlı Bulanık Mantık Denetleyici Tasarımı .....	72
3.6.1. Kullanılan Genetik Operatörler.....	73
3.6.2. Bulanık Parametrelerin Genetik Olarak Kodlanması.....	74
3.7. Asenkron Motorun Hızının PI Denetleyici ile Kontrolü.....	77
3.8. Asenkron Motorun Hızının PID Denetleyici ile Kontrolü.....	78
3.9. Asenkron Motorun Hızının Genetik-PI Denetleyici ile Kontrolü.....	79
3.10. Asenkron Motorun Hızının Bulanık Mantık Denetleyicisi ile Kontrolü .....	80
3.11. Asenkron Motorun Hızının Genetik Tabanlı Bulanık Mantık	

Denetleyicisi ile Kontrolü .....	82
4. ARAŞTIRMA BULGULARI .....	84
4.1. UVDGM, Bağlaşık İndüktörlü Üç-Seviyeli Evirici ve Hız Kontrolcüsü Menüsü .....	86
4.2. Asenkron Motor Parametreleri ve Simülasyon Ayarları Menüsü .....	87
4.3. Genetik-PI Denetleyici Menüsü .....	88
4.4. Simülasyon Grafikleri Menüsü .....	89
4.5. Blok Diyagram Menüsü .....	90
4.6. Bulanık Mantık Denetleyicisi Menüsü .....	92
4.7. Genetik Tabanlı Bulanık Mantık Denetleyici Menüsü .....	96
5. TARTIŞMA VE SONUÇLAR .....	99
KAYNAKLAR .....	101
EKLER .....	108
EK A. Şekiller .....	109
EK B. Grafikler .....	113
ÖZGEÇMİŞ .....	118

## ÖZET

### Yüksek Lisans Tezi

# BAĞLAŞIK İNDÜKTÖRLÜ ÜÇ-SEVİYELİ EVİRİCİ İLE ASENKRON MOTORLARIN HIZ KONTROLÜ İÇİN GENETİK TABANLI BULANIK MANTIK DENETLEYİCİ SİMÜLATÖR TASARIMI

Alphan MAT

Süleyman Demirel Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektronik Bilgisayar Eğitimi Anabilim Dalı

Danışman: Yrd. Doç. Dr. Okan BİNGÖL

Asenkron motorlar (ASM) endüstride yaygın olarak kullanılmaktadırlar. ASM'lerin hız denetiminde genellikle dolaylı alan yönlendirmeli kontrol yöntemi tercih edilmektedir. Bu kontrol yönteminde değişkenlerin doğru bir şekilde bilinmesi gerekmektedir. Sisteme ait değişkenler matematiksel modelleme yapılabilecek kesinlikte bilinmeyebilir veya bu değişkenler zaman içinde değişiklik gösterebilir. Sistemlerin denetiminde, oransal-integral (PI) gibi klasik denetleyicinin performansı düşük olduğundan dolayı bulanık mantık (BM) ve genetik algoritma (GA) tasarım yöntemleri kullanılmaktadır. BM denetleyici yönteminde tasarım işlemi uzun zaman almakla beraber, denetleyici performansı uzman bilgisi ile sınırlı kalmaktadır. Bu yüzden BM denetleyici yapısı GA kullanılarak denetleyici performansı için en uygun çözüm sağlanmaktadır.

Bu çalışmada, genetik tabanlı bulanık mantık (GABM) denetleyicisi ile bağlaşıklık indüktörlü üç-seviyeli evirici temelli bir ASM'nin hız denetimi için, simülasyon yazılım programı hazırlanmıştır. Hazırlanan simülasyon yazılım programı Microsoft Visual Studio 2012 IDE'sinde Windows Presentation Foundation (WPF) alt yapısı kullanılarak, C# programlama dili ile yazılmıştır. Simülasyon yazılım programı, esnek bir yapıya ve grafiksel tabanlı bir arayüze sahiptir. Yapılan arayüz ile ASM'nin, eviricinin ve denetleyicinin parametreleri değiştirilerek sistemin farklı çalışma durumlarındaki tepkisi grafikler aracılığıyla izlenebilmektedir.

**Anahtar Kelimeler:** ASM, üç-seviyeli evirici, bağlaşıklık indüktör, uzay vektör, darbe genişlik modülasyonu(DGM), BM, GABM.

**2013, 118 sayfa**

## **ABSTRACT**

**M.Sc. Thesis**

### **THE SIMULATOR DESIGN OF GENETIC BASED FUZZY LOGIC CONTROLLER FOR INDUCTION MOTORS SPEED CONTROL WITH COUPLED INDUCTORS THREE-LEVEL INVERTER**

**Alphan MAT**

**Süleyman Demirel University  
Graduate School of Applied and Natural Sciences  
Department of Electronic and Computer Education**

**Supervisor: Asst. Prof. Dr. Okan BİNGÖL**

The induction motors (IM) are widely used in industry. Indirect field oriented control method is generally preferred at speed control of IM. The variables are necessary to know correctly in control method. Variables of the system, have done Mathematical modeling may not known definitely or these variables may change over time. Fuzzy logic (FL) and genetic algorithms (GA) desing methods are used because of classic controllers performance such proportional-integral (PI) at system controllers. Although design process is taken a long time at the method of FL controller, the performance of controller is limited by the expert knowledge. Therefore, the structure of the FL controller using GA provides the best solution for the performance of the controller.

In this study, the simulation software program is prepared for IM speed control with coupled inductors and three-level inverter with genetic based fuzzy logic (GAFL) controller. Have prepared simulation software program is written with C# programming language, using the infrastructure of Windows Presentation Foundation (WPF) at Microsoft Visual Studio 2012 IDE. Simulation software program has a flexible structure and the graphical based interface. By changing the IM interface, inverter and controller parameters with created interface, system response can be monitored by means of graphics at different working situations.

**Keywords:** IM, three-level inverter, coupled inductors, space vector, pulse width modulation (PWM), FL, GAFL.

**2013, 118 pages**



## TEŞEKKÜR

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli danışman hocam Yrd. Doç. Dr. Okan BİNGÖL'e teşekkürlerimi sunarım. Yazılım çalışmalarında yardımlarını esirgemeyen değerli arkadaşım bilgi işlem uzmanı Serdar PAÇACI'ya ve tezin araştırma sırasında sistemin matematiksel modelinde yardım eden arkadaşım Öğr. Gör. Mehmet Zahit UZUN'a teşekkür ederim.

Evirici sistemi çalışmalarında yardımcı olan değerli hocam Yrd. Doç. Dr. Ali SAYGIN'a ve literatür araştırmalarında yardımcı olan dostum Öğr. Gör. Alper KEREM'e teşekkür ederim.

Araştırmanın yürütülmesinde manevi yardımlarını gördüğüm bilişim teknolojileri öğretmeni arkadaşlarım Ebubekir AÇIKGÖZ ve Nafiz BÜLBÜL'e teşekkür ederim.

Tezimin her aşamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Alphan MAT  
ISPARTA, 2013

## ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. Üç-fazlı asenkron motorun manyetik eksen düzlemi.....	12
Şekil 3.2. abc - ekseninden dq - eksenine dönüşüm.....	15
Şekil 3.3. Asenkron motor değişkenlerinin aktarıldığı d - q referans düzlemleri.....	17
Şekil 3.4. Dolaylı vektör kontrolüne ait düzlem dönüşümleri.....	21
Şekil 3.5. İki seviyeli gerilim beslemeli evirici.....	24
Şekil 3.6. Üç-seviyeli diyot tutmalı evirici.....	26
Şekil 3.7. Üç-seviyeli diyot tutmalı eviricinin a) Çıkış faz b) Çıkış hat gerilimleri.....	26
Şekil 3.8. Üç-seviyeli kondansatör tutmalı evirici.....	27
Şekil 3.9. Bir fazlı seri bağlı tam köprü çok seviyeli evirici.....	29
Şekil 3.10. Bağlaşık indüktörlü üç-seviyeli evirici.....	30
Şekil 3.11. Bağlaşık indüktörlü üç-seviyeli eviricinin tek bacağındaki anahtarlama durumları.....	31
Şekil 3.12. Üç-seviyeli eviricinin çıkış gerilimlerinin anahtarlama durumları ve uzay vektörü.....	33
Şekil 3.13. Üç-seviyeli eviricinin gerilim uzay vektörleri.....	35
Şekil 3.14. A sektörüne ait uzay gerilim vektörleri.....	35
Şekil 3.15. A sektörüne ait dört bölge için üç fazın DGM dalga şekilleri.....	38
Şekil 3.16. $\alpha - \beta$ düzlemi ile g-h düzlemi arasındaki bağlantı.....	39
Şekil 3.17. Normalize edilmiş referans gerilim vektörünün maksimum uzunluğu.....	40
Şekil 3.18. Normalize edilmiş referans gerilim vektörünün A sektörü 1. bölgesi.....	40
Şekil 3.19. Klasik küme ve bulanık mantık kümesi.....	43
Şekil 3.20. Üçgen üyelik fonksiyonu.....	45
Şekil 3.21. Trapezoidal üyelik fonksiyonu.....	46
Şekil 3.22. Sigmoid üyelik fonksiyonu.....	47
Şekil 3.23. Gaussian üyelik fonksiyonu.....	47
Şekil 3.24. Max-Dot çıkarım tekniği.....	49
Şekil 3.25. Min-Max çıkarım tekniği.....	50
Şekil 3.26. Tsukamoto çıkarım tekniği.....	51
Şekil 3.27. Takagi-Sugeno çıkarım tekniği.....	51
Şekil 3.28. Maksimum üyelik tekniği.....	52
Şekil 3.29. Ağırlık merkezi tekniği.....	53
Şekil 3.30. Ağırlık ortalaması tekniği.....	53
Şekil 3.31. Maksimumların ortalaması tekniği.....	54
Şekil 3.32. BM denetleyicisinin yapısı.....	55
Şekil 3.33. Esnek programlamayı oluşturan programlama teknikleri.....	56
Şekil 3.34. Bir GA'da tipik kromozom dizgisi.....	62
Şekil 3.35. GA ile BM denetleyici parametrelerinin oluşturulması.....	73
Şekil 3.36. Bulanık veri tabanının bir kromozom içine kodlanması.....	76
Şekil 3.37. ASM'nin PI denetleyici ile hız denetimi.....	78
Şekil 3.38. ASM'nin PID denetleyici ile hız denetimi.....	79
Şekil 3.39. ASM'nin Genetik-PI denetleyici ile hız denetimi.....	80
Şekil 3.40. ASM'nin BM denetleyici ile hız denetimi.....	81

Şekil 3.41. ASM'nin GABM denetleyici ile hız denetimi .....	82
Şekil 4.1. Simülasyon programının ana penceresi .....	85
Şekil 4.2. ASM ve evirici parametreleri ayarları menüsü .....	88
Şekil 4.3. Genetik-PI denetleyici menüsü .....	89
Şekil 4.4. PI, PID ve Genetik-PI denetleyici için simülasyon grafikleri ekranı .....	90
Şekil 4.5. Blok diyagram menüsü.....	92
Şekil 4.6. BM denetleyicisi hız hatası üyelik fonksiyonu menüsü.....	93
Şekil 4.7. BM denetleyicisi hız hatası değişimi üyelik fonksiyonu menüsü ..	94
Şekil 4.8. BM denetleyicisi çıkış üyelik fonksiyonu menüsü .....	95
Şekil 4.9. BM denetleyicisi kural tabanı menüsü.....	96
Şekil 4.10. GABM denetleyicisi menüsü .....	98
Şekil A.1. Genetik-PI denetleyicisi akış diyagramı .....	109
Şekil A.2. BM denetleyicisi akış diyagramı.....	110
Şekil A.3. GABM denetleyicisi akış diyagramı.....	111
Şekil A.4. Simülasyon programının akış diyagramı.....	112
Şekil B.1. PI denetleyici için hız, akım, faz gerilimleri ve fazlar arası gerilimleri simülasyon grafik ekranı.....	113
Şekil B.2. PID denetleyici için hız, akım, faz gerilimleri ve fazlar arası gerilimleri simülasyon grafik ekranı.....	114
Şekil B.3. Genetik-PI denetleyici için hız, akım, faz gerilimleri ve fazlar arası gerilimleri simülasyon grafik ekranı.....	115
Şekil B.4. BM denetleyici için hız, akım, faz gerilimleri ve fazlar arası gerilimleri simülasyon grafik ekranı.....	116
Şekil B.5. GABM denetleyici için hız, akım, faz gerilimleri ve fazlar arası gerilimleri simülasyon grafik ekranı.....	117

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
Çizelge 3.1. Bağlaşık indüktörlü üç-seviyeli eviricinin anahtarlama durumları.....	33
Çizelge 3.2. Üç-seviyeli eviricinin gerilim vektörlerinin gruplandırılması ...	34
Çizelge 3.3. Farklı bölgelerdeki gerilim vektörlerinin zaman süreleri .....	37
Çizelge 3.4. g-h düzlemi kullanılarak sektör belirleme.....	40
Çizelge 3.5. A sektörü için bölge bulunması durumları .....	41
Çizelge 3.6. Genetik algoritmada kullanılan kromozom yapısı.....	75
Çizelge 3.7. Örnek kural tablosu .....	77

## SİMGELER VE KISALTMALAR DİZİNİ

AA	Alternatif Akım
ASM	Asenkron Motor
BDCM	Brushless Direct Current Motor
BM	Bulanık Mantık
ÇSE	Çok Seviyeli Evirici
DA	Doğru Akım
DGM	Darbe Genişlik Modülasyonu
DSP	Sayısal İşaret İşlemci
EMI	Elektromanyetik Parazitlenme
EMK	Elektro Motor Kuvveti
GA	Genetik Algoritmalar
GABM	Genetik Algoritma Tabanlı Bulanık Mantık
HD	Harmonik Bozulma
IGBT	Kapıdan Yalıtımlı Bipolar Transistör
NPC	Nötr Nokta Kenetlemeli Evirici
PI	Oransal-İntegral Denetleyici
PID	Oransal-İntegral-Türev Denetleyici
PMDC	Permanent Magnet Direct Current
PWM	Darbe Genişlik Modülasyonu
RMS	Etkin Değer
THD	Toplam Harmonik Bozulma
UVDGM	Uzay vektör darbe genişlik modülasyonu
WPF	Windows Presentation Foundation
$\alpha, \beta$	Kartezyen koordinat sisteminin koordinatları
$f_s$	Anahtarlama frekansı
$g, h$	Hekzagonal koordinat sisteminin koordinatları
$\theta'$	Referans gerilim vektörün faz açısı
$T_s$	Örnekleme periyodu
$k$	Modülasyon indeksi
$t_0$	Merkez vektör iletim süresi
$t_1, t_2, t_3, t_4, t_5$	Komşu vektörlerin iletim süreleri
$\omega$	Açısal hız
$\lambda$	Kaçak akı
$\theta_r$	Elektriksel açı
$N_r$	Rotor sipir sayısı
$N_s$	Stator sipir sayısı
$\lambda_{as}, \lambda_{bs}, \lambda_{cs}$	Stator faz akıları
$\lambda_{ar}, \lambda_{br}, \lambda_{cr}$	Rotor faz akıları
$r_s$	Stator faz direnci
$r_r$	Rotor faz direnci
$L_s$	Stator endüktansı
$L_{1s}$	Kaçak endüktans
$L_{ms}$	Mıknatıslanma endüktansı
$T_e$	Motorda üretilen elektromanyetik moment
$I_n$	n. akım harmoniği
$I_T$	Temel frekans bileşenindeki akımın RMS değeri
$\vec{V}_{ref}$	Referans gerilim vektörü

$V_{DC}$	Giriş DC gerilimi
$A_1, A_2, A_3, A_4$	Uzay vektör blok diyagramındaki 60°'lik üçgenler
$i_a, i_b, i_c$	Yük faz akımları
$i_{aU}$	A fazına ait üst sargı akımı
$i_{aL}$	A fazına ait alt sargı akımı
$i_{CM}$	Ortak mod akımı
$V_{A0}, V_{B0}, V_{C0}$	Yük faz gerilimleri
$d_X, d_Y, d_Z$	Anahtarların bekleme süresi
$S_1, S_3, S_5$	Bağlaşık indüktörlü üç-seviyeli evirici topolojisinin üst anahtarları
$S_2, S_4, S_6$	Bağlaşık indüktörlü üç-seviyeli evirici topolojisinin alt anahtarları

## 1. GİRİŞ

Günümüzde elektrik motorları, sürücü sistemler olarak endüstrinin çok önemli elemanlarından biridir. Asenkron motor (ASM)'lar düşük maliyet, ses ve atalete sahip olmaları, sağlam ve güvenilir yapıları, özellikle sincap kafesli tiplerinin az bakım gerektirmeleri, kirli ve tehlikeli ortamlarda çalıştırılabilmelerinden dolayı endüstride diğer elektrik motorlarına göre daha yaygın olarak kullanılmaktadırlar. Bunların yanı sıra geniş güç ve hız aralığına sahip olmaları da asenkron motorların diğer motorlara karşı önemli bir üstünlüğüdür ASM'ler genel olarak stator ve rotor olmak üzere iki kısımdan oluşmaktadır. Stator, ASM'nin duran kısmıdır. Rotor ise dönen kısmıdır (Del Toro, 1990; Bose, 2002).

Doğru akım (DA) makineleri hız kontrolünün kolay olması nedeni ile sanayide uzun süre yaygın olarak kullanılmıştır. DA makineleri bu avantajlarının yanında bakım ve üretim maliyetlerinin yüksek olması, çalışma şartlarında kolay aşınmaları gibi dezavantajları da vardır. Alternatif Akım (AA) makineler daha ucuz ve dayanıklı olmalarına karşın hız kontrollerinin zor olması nedeniyle bir dönem tercih edilmemiştir. AA makinelerin hız kontrolü için öncelikle skalar sürme yöntemleri geliştirilmiş ve uygulanmıştır. Bu yöntemler DA makinelerinin hız kontrol performansını elde edememiştir. DA makinelerin popülerliği AA makineler için vektör kontrollü sürme yönteminin bulunup uygulanmasıyla sona ermiştir. Vektör kontrollü sürme yönteminde AA makine DA makineye benzer şekilde akı ve momenti temsil eden iki büyüklük ile sürülebilmektedir. AA makineler içerisinde en çok tercih edileni ucuz ve az bakım gerektirdikleri için sincap kafesli olanlardır (Yıldız, 2008).

ASM'ler direkt olarak şebeke geriliminden beslendiğinde kutup sayıları ve şebeke frekansına göre belirli bir sabit hızda dönerler. ASM'lerin değişik hızlarda çalıştırılabilmesi için değişken genlikli gerilim ve frekans üretebilen kaynaklardan beslenmeleri gerekmektedir. Eviriciler tarafından beslendiklerinde, ASM'lerin hızını değiştirmek mümkün hale gelmektedir (Bose, 1997).

Şebekeden elde edilen alternatif akım (AA), doğrultucular ile doğru akım (DA)'a dönüştürülür. Daha sonra tekrar eviriciler ile AA' ya dönüştürülür. DA kaynağından beslenen sistemlerde gerilimin genliği, yaygın olarak darbe genişlik modülasyonu (DGM) ile ayarlanır. DA kaynağın çıkış akımı üzerinde, sinüsoidal giriş akımı üzerinden beslenmesi ile düşük seviyede harmonikler oluşur. Oluşan harmonikler ise çıkış gerilimindeki bozulmaları oluşturur. Bu durum sistemin denetimine bozucu etki yapmaktadır (Drury ve Grant, 1994; Saygın, 2004). Çıkış gerilimindeki harmonikler genel olarak iki durumda en aza indirgenebilir. Birinci durumda yüksek anahtarlama frekansına sahip yarı iletken anahtarlara farklı DGM teknikleri uygulanır. İkinci durumda ise eviricinin çıkış dalga şeklinin seviye sayısı arttırılır (Paçacı, 2011).

Çok seviyeli eviriciler (ÇSE), güç elektroniği alanındaki hızlı gelişmelerden dolayı birçok endüstriyel uygulamalarda kullanılmaya başlanmıştır. Nabae ve diğerleri yapmış oldukları çalışmalarda üç-seviyeli eviricilerde anahtarlama frekansını arttırmadan geriliminin çıkış dalga şeklindeki harmonik bileşenlerinin daha az olduğunu belirtmişlerdir (Nabae vd., 1981; Bhagwat vd., 1983; Lai ve Peng, 1996). Sonraki yıllarda bu evirici topolojisi üç seviyeli evirici kavramı olarak genişletilmiştir. Günümüzde eviricinin çıkış faz nötr gerilimlerine göre farklı seviyelerde ve farklı topolojilerde ÇSE yapıları mevcuttur. ÇSE topolojileri üzerine yapılan araştırmaların çoğu üç tip dönüştürücü üzerine yoğunlaşmıştır. Bunlar; diyot tutmalı evirici, kondanstor tutmalı evirici ve tek fazlı H-köprü eviricinin seri olarak bağlanması ile oluşturulan kaskad evirici topolojileridir (Kerem, 2012).

Son zamanlarda ÇSE topolojilerine alternatif bir topoloji geliştirilmiştir. Bu evirici topolojisinde altı adet kapıdan yalıtımlı bipolar transistör (IGBT) yarıiletken anahtar kullanılmakla beraber eviricinin her bir çıkış uçlarına çift sarımlı bobinler bağlanmaktadır. Bu topoloji, girişine uygulanan DA kaynak ve her bir çıkışa bağlanan 3 faz bölünmüş sargılı çift sarımlı bobinler aracılığı ile çok seviyeli çıkış gerilimleri üretmektedir (Ewanchuk vd., 2008; Salmon vd., 2009; Vafakhah, 2010).



Bölünmüş sargılı bobin üst ve alt anahtarlarla seri bağlı olduğundan, bağlaşıklık indüktörlü üç-seviyeli evirici topolojisinin önemli bir avantajı anahtarlama esnasındaki geçiş akımı için ölü-zaman gereksinimidir. Böylece ölü-zamanın olumsuz etkileri bastırılmış olur. Yük veriminin ve güç yoğunluğun potansiyel iyileştirilmesiyle elde edilen geliştirilmiş DGM gerilimi AA bobinlerdeki ve makine yüklerindeki yüksek frekans kayıplarını düşürür. Bu özellikleriyle, bağlaşıklık indüktörlü üç-seviyeli evirici topolojisi daha yüksek frekanslı DGM çıkışların temel olduğu düşük hassasiyetli ve yüksek frekans temel uygulamalarında kullanılan alternatif topolojilerden daha uygundur (Matsui, 2000; Ewanchuk vd., 2008; Salmon vd., 2009; Vafakhah, 2010).

Son yıllarda yarı iletken teknolojisinin ilerlemesi ile birlikte farklı DGM teknikleri geliştirilmiş ve böylece yeni statik güç dönüştürücüler üretilerek birçok alanda kullanılmaya başlanmıştır. Bunlardan biri de, ÇSE'dir. Bu eviriciler girişlerine uygulanan farklı DA gerilim seviyelerini birleştirerek sinüsoidal forma yakın bir çıkış gerilimi oluştururlar (Manjrekar vd., 2000; Wu ve He, 2000; Rodriguez vd., 2002; Bingöl, 2005; Tuncer, 2009).

ÇSE'lerde minimum harmonik bozulma ile istenen genlik ve frekansta gerilimler elde etmek için farklı DGM teknikleri kullanılmaktadır. İki ve çok seviyeli dönüştürücüler için şu anda en yaygın modülasyon tekniklerinden biri uzay vektör darbe genişlik modülasyon (UVDGM) tekniğidir. Bu teknik ile evirici çıkışında istenilen genlik ve frekansta üç-fazlı gerilimler elde edilebilmektedir (Zhang vd., 2000; Wei vd., 2003; Tuncer, 2004; Kerem, 2012).

Denetim sistemleri temelde açık ve kapalı döngü denetim sistemleri olarak iki ana gruba ayrılır. Açık döngü denetim sistemlerinde, denetim sinyali çıkıştan bağımsızdır. Kapalı döngü denetim sistemlerinde ise sistemin çıkışından girişine geri besleme yapılır. Sistemin girişine uygulanan referans sinyal ile sistemin çıkışından geri besleme yapılarak elde edilen gerçek sinyal karşılaştırılarak hata sinyali elde edilir. Denetleyici bu hata değerini giriş olarak alır ve kendi denetim sistemine göre bir çıkış değeri üretir. Bu çıkış değeri denetlenen sisteme giriş işareti olarak iletilir. Klasik denetleyiciler oransal denetim etkisi (P), integral

denetim etkisi (I) ve türev denetim etkisinin (D) bir arada kullanılmasıyla oluşturulur. En sık kullanılan klasik denetleyiciler oransal-integral denetleyici (PI) ve oransal-integral-türev denetleyici (PID) denetleyicidir. Klasik denetim yöntemleri ile yüksek performans elde edilmek istenirse, bu sistemlerin iyi tanımlanmış olması gerekir. Doğrusal olmayan yapıdaki sistemlerin matematiksel modellenmesi tam olarak yapılamamaktadır (Boz, 2011). Bu tip sistemlerin denetiminde yapay zekâ tekniklerinin kullanılması ile daha başarılı sonuçlar elde edilir (Paçacı, 2011).

Klasik küme teorisinde bir eleman kümenin ya elemanıdır ya da değildir. Kısmi üyelik kavramı klasik küme teorisinde yer almamaktadır. İnsanın karar verme ve çıkarım yapma yeteneğinden esinlenerek, bulanık küme teorisi ortaya çıkmıştır. Bulanık küme teorisinde bir elemanın kümeye ait olma derecesi vardır. Bu dereceye üyelik derecesi denilmektedir. BM denetleyicisinin tanımlanmasındaki işlem basamakları temel olarak şu şekildedir; denetleyicinin her bir giriş değeri için üyelik fonksiyon kümesi tanımlanır. Çıkış üyelik fonksiyonları tanımlanıp, kural tabanı oluşturulur. Karar verme ve son olarak da durulaştırma metotları belirlenir (Ekren, 2009; Elmas, 2007).

Genetik algoritmalar (GA), biyolojik süreci modelleyerek, fonksiyonları veya sistemleri optimize eden evrim algoritmalarıdır. GA'lar, optimizasyon ve arama problemlerini çözmek için kullanılan adaptif bir metottur. GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA'ların her bir ferdi, kromozomlar (bireyler) şeklinde temsil edilen popülasyonlardan oluşur. Popülasyonun uygunluğu, belirli kurallar dahilinde maksimize veya minimize edilir. Her yeni nesil, rasgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir (Angeline, 1995).

Klasik bulanık mantık denetleyici tasarımı deneme yanılmalara ve uzman bilgisinin denetim mekanizmalarına aktarılması esasına dayanır. Bu yöntem oldukça zaman alıcıdır ve tasarımın en iyi neticelere ulaşması bazen mümkün değildir. Basit yapıdaki bulanık mantık kontrolörler için deneme yanılma

yöntemi uygulanabilir. Fakat kompleks bulanık mantık denetleyicilerde denetleyici parametrelerinin fazla olması nedeniyle deneme yanılma yöntemi yetersiz kalmaktadır. Bu durumda bulanık mantık denetleyici tasarımı için farklı yöntemlere ihtiyaç duyulmaktadır. GA son yıllarda bulanık mantık denetleyici tasarımı probleminin çözümünde kullanılmaya başlanmıştır. Böylece GA'nın en iyi çözümleri bulma yeteneği ve bulanık mantık kontrolörlerin etkin proses kontrol yeteneği birleştirilmiştir. Bir bulanık model sayısız parametreye sahiptir, örneğin üyelik fonksiyonlarının şekli ve sayısı, ölçeklendirme faktörleri, dilsel değerlerin sayısı gibi. Tüm bu parametreler bulanık sistem üzerinde oldukça etkilidir. GA ile iyileştirilecek parametre sayısının fazla olması en iyi sonuçların elde edilmesini zorlaştırmaktadır. Bulanık sistemi en çok etkileyen parametrelerin GA ile iyileştirilmesi daha etkin bir iyileştirme neticesi elde edilmesine olanak sağlamaktadır (Yıldız, 2008).

Eğitimde teorik bilgilerin uygulama ile desteklenmesi, öğrenme hızını arttırmakta ve öğrenmenin kalıcı olmasını sağlamaktadır. Pratik becerileri kazanabilmek için pahalı laboratuvar sistemlerine gereksinim duyulmaktadır. Bu gereksinim, eğitim amaçlı olarak geliştirilen daha düşük maliyetli ve uygun öğrenme ortamı sağlayan simülatör programları ile giderilmeye çalışılmaktadır. Simülatörler, gerçek sistemlerin çalışma şartlarını bilgisayar ortamında taklit eden yazılımlardır. Eğitim amaçlı geliştirilen simülatör programları, farklı bilim dalları için yeterince esnek ve interaktif değildir (Akçayol vd., 2002; Gökbulut vd., 2006; Yiğit ve Elmas, 2008).

Çalışmanın amacı öğrencilerin laboratuvar ortamında yaptıkları ASM'nin hız kontrolü deneylerinin, zaman ve mekandan bağımsız olarak simülasyonunun yapılabilmesi için bu simülasyon programı hazırlanmıştır. Simülasyon programında öğrencinin öğrenmesi daha kolay ve hızlı olmaktadır. Laboratuvar ortamında zaman kriteri olduğundan dolayı yapılan bütün deneyler tam ve doğru anlaşılmamaktadır. Deneyden sonra hala anlaşılmayan kısımlar, simülasyon ortamında tekrar uygulama yapılarak kalıcı bir şekilde öğrenilebilir. Laboratuvar ortamında tehlikeli olan uygulamalar da simülasyon üzerinde

rahatlıkla denenebilir, sonuçları gözlemlenebilir. Simülasyon üzerinde uygulama yapan öğrenci gerçek laboratuvar ortamında zorluk çekmeyecektir.

Yapılan çalışmada PI, PID, Genetik-PI, BM denetleyiciler ile evirici temelli asenkron motorun hız kontrolü için literatürdeki tekniklerden farklı bir alan olan GABM denetleyici tekniği kullanılmıştır. Yapılan simülasyon çalışmaları sonucunda, klasik PI, ve bulanık mantık denetleyicilerinin tasarımı ve optimizasyonu için GABM denetleyici kullanımının oldukça iyi sonuçlar verdiği görülmüştür.

## 2. KAYNAK ÖZETLERİ

Literatürde, ASM sistemleri üzerine yapılmış kontrol çalışmaları ve kontrol aşamasında kullanılan GA, BM ve GABM denetleyici ile ilgili pek çok araştırma ve çalışma mevcuttur. Endüstriyel uygulamalarda, sistemlerin kontrolünü gerçekleştirmek için matematiksel modele ihtiyaç duyulmaktadır. Gerçekleştirilen ASM sistemlerinde kullanılan ASM ile ilgili matematiksel model bulma çalışmaları mevcuttur.

Lee ve Pang (1994), çalışmalarında; BM tabanlı fırçasız DA motor sürücülerini için yeni bir yöntem sunmuşlardır. Sunulan yeni yöntemde; PI denetleyici parametrelerinin bulunmasında bulanık mantık kullanılmış ve sunulan çözümün simülasyon sonuçlarında optimal sonuç verdiği görülmüştür.

Kinzel vd. (1994), bulanık denetleyicilerin tasarım ve optimizasyonu için değişik yapılarda GA geliştirmişlerdir. Çalışmalarında; bulanık denetleyicilerin otomatik olarak tasarlanması ve optimizasyonu için sistematik bir yaklaşım sunmuşlardır. Yapılan uygulama çalışmaları sonucunda, bulanık denetleyicilerin tasarımı ve optimizasyonu için genetik algoritmaların kullanımının oldukça iyi sonuçlar verdiği belirtilmiştir.

Tan ve Hu (1996), bulanık denetleyicilerin genetik algoritmalar kullanılarak tasarlanması isimli çalışmalarında; bulanık denetleyicilerin genetik algoritmalarla tasarlanmasına sistematik bir çözüm sunmuşlardır. Çalışmada; GA hakkında kısaca bilgi verilmekte ve bulanık denetleyicilerin tasarlanması anlatılmıştır.

Donescu vd. (1996), fırçasız DA motorlarda bulanık mantık hız kontrolü için, sistematik bir tasarım metodu isimli çalışmalarında; bulanık denetleyiciler için yeni bir tasarım yöntemi geliştirmişlerdir. Bu yöntemde göre; bulanık denetleyicinin kontrol yüzeyi, aynı uygulamada PI kontrol ile gerçekleştirilen kontrol yüzeyi referans alınarak bulunmaktadır. Bulunan başlangıç kontrol yüzeyinde, sistemin adım cevabına bakılarak parametrelerin ince ayarları

yapılmaktadır. Önerilen yöntemin çevrim dışı uygulamalarda kolayca yürütülebileceği belirtilmiştir.

Tang vd. (2001), optimal bulanık mantık PID denetleyici isimli çalışmalarında; geleneksel PID denetleyicilerin, genellikle lineer olmayan, derecesi yüksek, zaman gecikmeli ve matematiksel modeli iyi bilinmeyen sistemlerin kontrolünde yetersiz kaldığını ve PID parametrelerinin optimum değerlerinin bulunmasının geleneksel yollarla mümkün olmadığını belirtmişleridir. Bu nedenle yeni bir bulanık mantık PID denetleyici geliştirmişler ve parametrelerinin optimizasyonunda çok amaçlı genetik algoritmaları kullanmışlardır.

Arslan ve Kaya (2001), çalışmalarında; bulanık denetleyicilerin üyelik fonksiyonlarının yapısı ve şeklinin belirlenmesinde genetik algoritmaları kullandıklarını belirtmişlerdir. Tek girişli ve tek çıkışlı bir bulanık mantık sistem için bir genetik algoritma programı geliştirmişlerdir.

Altınten (2001), çalışmasında bulanık mantık denetleyici tasarımında genetik algoritma kullanmış, denetleyici performansını etkileyen parametreler doğal evrim sürecini taklit eden genetik algoritmaların etkin çözüm arama yeteneği ile belirlenmiştir.

Üstün (2001), çalışmasında yapay sinir ağları ve genetik algoritmalar kullanılarak DSP temelli asenkron motor kontrol sistemindeki PI katsayılarının optimizasyonu gerçekleştirilmiştir.

Uygur (2002), hız kontrolü sinirsel-bulanık yöntemler kullanılarak gerçekleştirilmiştir. Sürücü devresi olarak skalar metod kullanan bir inverter kullanılmış ve sinirsel-bulanık yöntemlerin başarısı incelenmiştir.

Kim ve Youn (2002), çalışmalarında; yüksek hız bölgelerinde fırçasız DA motorların kontrolünde, akım ve gerilim kontrol işlemleriyle beraber DGM sinyallerinin üretilmesinin, işlemci hızı açısından bir problem olarak karşımıza

çıkıldığını belirtmişlerdir. Bu problemi çözmek için, DGM sinyallerinin üretilmesi ve komütasyon işlemlerinin bağımsız olarak yapılması önermişlerdir.

Yun vd. (2003), GABM denetleyiciler için, değişik hibrit yöntemler geliştirmişlerdir. Yazarlar çalışmalarında; bulanık denetleyiciler için bir kaba optimizasyon ve bir lokal optimizasyon tekniği sunmuşlardır. Kaba optimizasyon tekniğinde; parametreleri geniş bir arama yüzeyinde aramışlar, bu parametreler ile bir bulanık denetleyici gerçekleştirmişler ve daha sonra parametrelerin ince ayarını yapmışlardır. Üç ayrı karmaşık optimizasyon problemi için 4 ayrı hibrit genetik algoritma uygulanmış ve sonuçları analiz etmişlerdir.

Çelik (2004), çalışmasında UVDGM ile yaygın olarak kullanılan birkaç DGM'lerinin ASM hız kontrolündeki performanslarını karşılaştırmıştır.

Gökbulut ve arkadaşları (2006), yaptıkları çalışmada, iki seviyeli evirici ile beslenen ASM' nin hız kontrolü için eğitim seti hazırlamışlardır. ASM' nin hız denetimi için klasik PI ve sinirsel bulanık denetleyici kullanılabilir. Eğitim setinde motor ve denetleyici parametreleri kullanıcı tarafından ayarlanabilmektedir.

Ewanchuk (2008), yaptığı çalışmasında üç fazlı yüksek hızlı makine uygulamaları için bağlaşıklık indüktörlü ÇSE topolojisini kullanarak, yüksek hızlı üç fazlı elektrik makinelerinin güç elektroniğinde uygulamasını kullanmıştır. Güç elektroniği temel kavramları ve AA için yüksek hızlı bir makine şartları ile elektrik makinesi tarafından üretilen momente bağlı güç yoğunluğu fiziksel parametrelerini geliştirilmiştir.

Kerem (2012), yapılan simülasyon çalışmasında yük olarak R-L yük ve ASM kullanılmıştır. Bağlaşıklık indüktörlü üç-seviyeli evirici ile ASM'nin hız denetimi yapılırken alan yönlendirmeli kontrol tekniği tercih edilmiştir. Hız kontrol ünitesinde PI denetleyici kullanılmıştır. Denetleyiciden elde edilen veriler kullanılarak bağlaşıklık indüktörlü üç-seviyeli eviricideki yarı iletken anahtarların

çalışması için gerekli sinyaller üretilmiştir. Bu sinyaller, uzay vektör darbe genişlik modülasyon tekniği ile elde edilmiştir.

Elmas ve Yiğit (2008), yaptıkları çalışmada, DGM' li üç fazlı bir eviricinin bilgisayar ile denetimi ile ilgili bir eğitim seti hazırlamışlardır. Çalışma yazılım ve donanım olarak iki bölümden oluşmaktadır. Yazılım bölümünde farklı DGM tekniklerinin evirici denetiminde yaptığı etkiler ile ASM' nin fazlarındaki akımların ve gerilimlerin grafikleri incelenebilmektedir.

Yiğit ve Elmas (2008), yaptıkları çalışmada, anahtarlamalı relüktans motorun hız kontrolü için klasik PI denetleyici ve genetik PI denetleyici kullanılabilen bir eğitim seti tasarlamışlardır. Eğitim setinde motorun ve denetleyicilerin parametreleri kullanıcı tarafından ayarlanabilmektedir.

Bingöl ve Paçacı (2010), yaptıkları çalışmada, klasik PI ve PID denetleyiciler ve bulanık mantık denetleyicisi ile sabit mıknatıslı DA (PMDC) motorun hız kontrolü için sanal laboratuvar tasarımı yapmışlardır. Eğitim setinde PMDC motorun ve denetleyicilerin parametreleri kullanıcı tarafından ayarlanabilmektedir. Simülasyon sonuçları grafikler yardımıyla incelenebilmektedir.



### **3. ASENKRON MOTORLAR VE EVİRİCİLER**

#### **3.1. Asenkron Motorlar**

Asenkron makinalar motor olarak stator sargıları ile aldığı elektrik enerjisini rotorundan dönme hareketi yaparak mekanik enerjiye çeviren ve jeneratör olarak, rotorundan aldığı dönme hareketi mekanik enerjisini, bazı koşullar altında, stator sargılarında elektrik enerjisine, çeviren elektro-mekanik makinalardır (Bodurođlu, 1981).

Son yıllarda, yarı iletken güç elektroniğindeki teknolojik gelişmeler ile asenkron makinaların kontrolü kolayca yapılabilmektedir. Bu gelişmelerle asenkron motorların hızının kontrolü, moment kontrolü ve yumuşak yol verme işlemleri ile bu makinalar, diğer üstün yapısal özellikleri ile doğru akım makinalarının endüstrideki yerini almaya başlamıştır (Koca, 2006).

##### **3.1.1. Üç fazlı asenkron motorun matematiksel modeli**

Bir sistemin fiziksel davranışının benzetimini yapmak için, matematiksel modelinin çıkarılması gereklidir. Matematiksel modeller, sistemin gerçek fiziksel davranışına olabildiğince yakın olmalıdır. Sistemin modelinin doğruluğunu artırmak için fiziksel büyüklükler skaler değil, vektörel olarak ele alınmalıdır. Bu yolla özellikle motorların geçici hal davranışlarında oluşacak hata önemli derecede azalacaktır (Güleç, 1999).

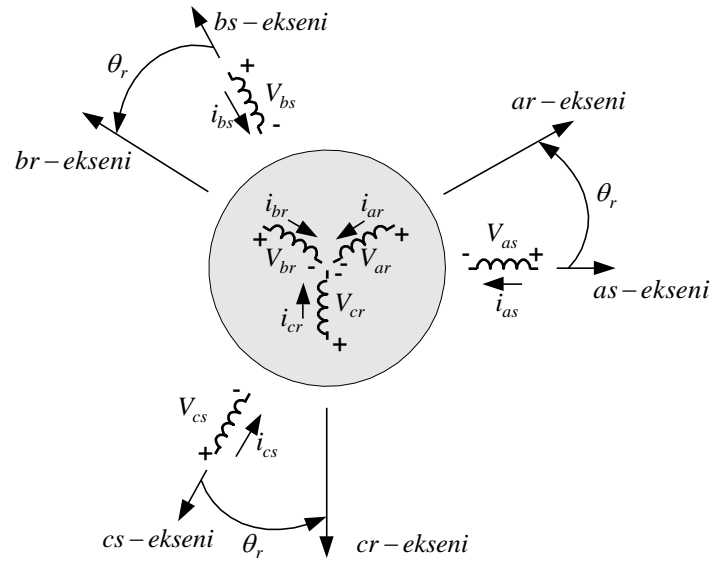
Üç fazlı asenkron motorun statorunda sargı eksenleri arasında uzayda 120°'lik açı olan üç adet bir fazlı sargı vardır. Bu üç sargı yıldız ya da üçgen olarak bağlanır. Faz sargılarını oluşturan bobinler stator oluklarına yerleştirilir.

Üç-fazlı ASM'nin matematiksel modeli oluşturulurken aşağıda belirtilen modelleme varsayımları kullanılacaktır;

- Stator sargıları stator olukların düzgün olarak yerleştirilmiş olup hava aralığındaki akının sinüsoidal olduğu,

- Manyetik devre doğrusal kabul edildiğinden doymanın oluşmadığı,
- Stator ve rotor sargılarının, direnç ve endüktansları frekans ve sıcaklık etkisi ile değişmediği,
- Akım yığılması olayının ihmal edildiği,
- Manyetik kısımların geçirgenliğinin sonsuz olduğu,
- Histerisiz ve fuko kayıplarının ihmal edildiği,
- Rotor çubuklarının, rotor eksenine göre simetrik olarak yayıldığı,

Stator ve rotor sargılarının dengeli olduğu ve stator sargılarının eşit olarak sarıldığı kabul edilir. Stator sargıları  $a_s$ ,  $b_s$  ve  $c_s$  olarak isimlendirilir. Stator sargılarının sarım sayıları ( $N_s$ ) birbirine eşittir. Sargılar arasında  $\pm 120^\circ$  faz farkı olup  $a_s$  sargısına simetrik olarak stator oyuklarına yerleştirilmiştir. Alt indis olarak "s" kullanılması stator veya durağan sargıları olduğu anlamına gelir. Aynı şekilde rotor sargıları,  $a_r$ ,  $b_r$  ve  $c_r$  olarak isimlendirilir. Alt indis olarak "r" kullanılması "rotor" veya "döner" sargıları olduğu anlamına gelir (Lipo ve Novotny, 1995). Şekil 3.1'de üç-fazlı asenkron motorun manyetik eksen düzlemi verilmiştir.



Şekil 3.1. Üç-fazlı asenkron motorun manyetik eksen düzlemi (Bingöl, 2005)

Şekil 3.1'deki stator ve rotor devrelerinin gerilim denklemleri Denklem 3.1'de gösterildiği gibi ifade edilir. Denklem 3.1'deki  $r_s$ ; stator faz direncini,  $\lambda_{as}$ ,  $\lambda_{bs}$ ,  $\lambda_{cs}$ ; stator faz akılarını ifade etmektedir.

$$\begin{aligned} v_{as} &= r_s i_{as} + \frac{d\lambda_{as}}{dt} \\ v_{bs} &= r_s i_{bs} + \frac{d\lambda_{bs}}{dt} \\ v_{cs} &= r_s i_{cs} + \frac{d\lambda_{cs}}{dt} \end{aligned} \quad (3.1)$$

Rotor gerilimleri ise Denklem 3.2'de gösterildiği gibidir. Burada  $r_r$ ; rotor faz direncini,  $\lambda_{ar}$ ,  $\lambda_{br}$ ,  $\lambda_{cr}$ ; rotor faz akılarını ifade etmektedir. Stator ve rotor fazlarındaki değişimler, akı değerlerinde birbirlerine etki etmektedir.

$$\begin{aligned} v_{ar} &= r_r i_{ar} + \frac{d\lambda_{ar}}{dt} \\ v_{br} &= r_r i_{br} + \frac{d\lambda_{br}}{dt} \\ v_{cr} &= r_r i_{cr} + \frac{d\lambda_{cr}}{dt} \end{aligned} \quad (3.2)$$

Motorun faz akıları akımlara bağlı olup Denklem 3.3'deki gibi ifade edilir.

$$\begin{aligned} \lambda_{abcs} &= \lambda_{abcs(s)} + \lambda_{abcs(r)} \\ \lambda_{abcr} &= \lambda_{abcr(s)} + \lambda_{abcr(r)} \end{aligned} \quad (3.3)$$

Burada Denklem 3.3 incelenecek olursa aşağıdaki eşitlikler elde edilir.

$$\lambda_{abcs(s)} = \begin{bmatrix} L_{as} & L_{abs} & L_{acs} \\ L_{abs} & L_{bs} & L_{bcs} \\ L_{acs} & L_{bcs} & L_{cs} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \quad (3.4)$$

$$\lambda_{abcs(r)} = \begin{bmatrix} L_{as,ar} & L_{as,br} & L_{as,cr} \\ L_{bs,ar} & L_{bs,br} & L_{bs,cr} \\ L_{cs,ar} & L_{cs,br} & L_{cs,cr} \end{bmatrix} \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \quad (3.5)$$

$$\lambda_{abcr(s)} = \begin{bmatrix} L_{ar,as} & L_{ar,bs} & L_{ar,cs} \\ L_{br,as} & L_{br,bs} & L_{br,cs} \\ L_{cr,as} & L_{cr,bs} & L_{cr,cs} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \quad (3.6)$$

$$\lambda_{abcr(r)} = \begin{bmatrix} L_{ar} & L_{abr} & L_{acr} \\ L_{abr} & L_{br} & L_{bcr} \\ L_{acr} & L_{bcr} & L_{cr} \end{bmatrix} \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \quad (3.7)$$

Statorun bir faz endüktansı, kaçak endüktans ( $L_{ls}$ ) ve mıknatıslanma endüktanslarının ( $L_{ms}$ ) toplamına eşittir. Bu durum Denklem 3.8'deki şekilde gösterilmiştir.

$$L_{as} = L_{bs} = L_{cs} = L_{ls} + L_{ms} \quad (3.8)$$

Stator faz sargıları ( $a_s - b_s$ ,  $b_s - c_s$  ve  $c_s - a_s$ ) arasındaki ortak endüktansın ifadesi ise Denklem 3.9'da gösterildiği gibi yazılır.

$$L_{abs} = L_{bcs} = L_{cas} = -\frac{L_{ms}}{2} \quad (3.9)$$

Stator sargılarının  $a_s$ ,  $b_s$  ve  $c_s$  fazlarına bağlı olarak akı bağlarının matrissel olarak ifadesi yazılırsa Denklem 3.10 elde edilir.

$$\lambda_{abcs(s)} = \begin{bmatrix} L_{ls} + L_{ms} & -\frac{L_{ms}}{2} & -\frac{L_{ms}}{2} \\ -\frac{L_{ms}}{2} & L_{ls} + L_{ms} & -\frac{L_{ms}}{2} \\ -\frac{L_{ms}}{2} & -\frac{L_{ms}}{2} & L_{ls} + L_{ms} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \quad (3.10)$$

Stator ve rotor sargıları arasında ortak bağ bulunmaktadır. Şekil 3.1'de rotor fazı  $a_r$  ve stator fazı  $a_s$  arasında  $\theta_r$  elektriksel açısı vardır ve  $\theta_r$  değişkendir. Benzer şekilde rotor fazları  $b_r$  ve  $c_r$  sırasıyla stator fazları  $b_s$  ve  $c_s$  arasında da  $\theta_r$  elektriksel açısı bulunur. Stator ile rotor arasındaki ortak endüktanslar, sargı eksenleri arasındaki açının kosinüsüdür. ( $a_s - a_r$ ), ( $b_s - b_r$ ) ve ( $c_s - c_r$ ) sargıları arasında  $\theta_r$  açısı vardır. Denklem 3.11 ve Denklem 3.12'de bu durum gösterilmiştir.

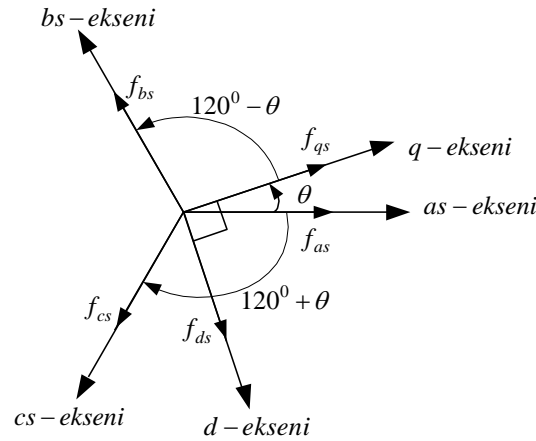
$$\lambda_{abcs(r)} = \frac{N_r}{N_s} L_{ms} \begin{bmatrix} \cos\theta_r & \cos(\theta_r + 2\pi/3) & \cos(\theta_r - 2\pi/3) \\ \cos(\theta_r - 2\pi/3) & \cos\theta_r & \cos(\theta_r + 2\pi/3) \\ \cos(\theta_r + 2\pi/3) & \cos(\theta_r - 2\pi/3) & \cos\theta_r \end{bmatrix} \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \quad (3.11)$$

$$\lambda_{abcs(s)} = \frac{N_r}{N_s} L_{ms} \begin{bmatrix} \cos\theta_r & \cos(\theta_r - 2\pi/3) & \cos(\theta_r + 2\pi/3) \\ \cos(\theta_r + 2\pi/3) & \cos\theta_r & \cos(\theta_r - 2\pi/3) \\ \cos(\theta_r - 2\pi/3) & \cos(\theta_r + 2\pi/3) & \cos\theta_r \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \quad (3.12)$$

Üç fazdan iki faza dönüşümde  $L_m = \frac{3}{2} L_{ms}$  alınması gerekir (Lipo ve Novotny, 1995).

### 3.1.2. Üç fazdan iki faza dönüşüm

Üç fazlı motorun altı sargısı arasında manyetik bir bağlantı vardır. Bu bağlantıların basitleştirilmesi için hava aralığı Manyeto Motor Kuvveti (MMK) iki dik düzlem yönünde bileşenlere ayrılır. Bu iki düzlem direct(dikine) ve quadrature (enine) olarak adlandırılır. Stator ve rotor değişkenleri bu iki düzleme aktarılır. Şekil 3.2'de üç fazlı eksen (abc) ile iki fazlı eksen (dq) arasındaki geometrik ilişki gösterilmiştir. Şekildeki a eksenini ile q eksenini arasında  $\theta$  açısı olup, d eksenini q eksenininin  $90^\circ$  gerisindedir (Paçacı, 2011).



Şekil 3.2. abc - ekseninden dq - eksenine dönüşüm (Paçacı, 2011)

Şekil 3.2'deki stator referans düzlemine ait dönüşümler olan 3 fazlı düzlemde iki fazlı düzleme dönüşüm matrisi olarak Denklem 3.13'de ve iki fazlı düzlemde üç fazlı düzleme ters dönüşüm ise Denklem 3.14'de gösterilmiştir.

Üç fazlı sistemdeki her bir sargının sarım sayısı N, iki fazlı sistemde ise  $\frac{3N}{2}$  dir. Dolayısıyla her iki sistemde de aynı MMK üretilmiş olur.

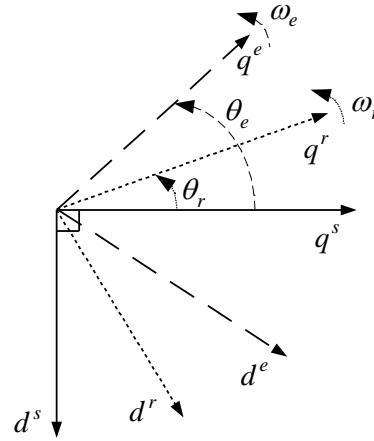
$$\begin{bmatrix} f_{qs} \\ f_{ds} \\ f_{os} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_{as} \\ f_{bs} \\ f_{cs} \end{bmatrix} \quad (3.13)$$

$$\begin{bmatrix} f_{as} \\ f_{bs} \\ f_{cs} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} f_{qs} \\ f_{ds} \\ f_{os} \end{bmatrix} \quad (3.14)$$

Sistem dengeli ise Denklem 3.13 ve Denklem 3.14'deki eşitliklerdeki  $f_o$  ifadesinin değeri sıfıra eşittir. ASM'ye ait faz gerilimleri veya akımlarının dengeli olmaması durumunda denklemlerde  $f_o$  bileşeninin de işlemlere girmesi gerekir. Aynı şekilde rotora ait değişkenlerin de yeni düzleme taşınması gerekir. Denklem 3.13 ve Denklem 3.14'deki dönüşümler rotor referans düzlemi için yapılır.

### 3.1.3. Asenkron motorun modelinde kullanılan referans düzlemleri

Elektrik motorları analizinde ve güç sistemlerinde ortak kullanılan referans düzlemleri keyfi, durağan, rotor ve senkron referans düzlemleridir. Referans düzlemlerinin kendi aralarında dönüşümü mümkündür. Motorun hız ve pozisyonu denetimine göre kullanılacak kontrol algoritması göz önüne alınarak referans düzlemlerinde dönüşüm yapılır. Bu durum Şekil 3.3'de gösterilmiştir. Şekildeki üst indislerden "s" durağan düzlemi, "r" rotor düzlemi, "e" ise senkron düzlemi ifade etmektedir.



Şekil 3.3. Asenkron motor değişkenlerinin aktarıldığı d - q referans düzlemleri (Krause vd., 1995)

#### Durağan referans düzlemi

Durağan referans düzlemindeki (stationary reference frame) gerçek eksen, stator faz sargılarından birinin eksenine bindirilir. Bu durumda referans düzlemin açısal hız değeri statora sabitlendiğinden dolayı, hem açısal hız  $\omega = 0$  hem de açısal pozisyon  $\theta = 0$  olur. Sabit eksen düzleminde değer alan değişkenler  $d^s - q^s$  ile ifade edilir (Krishnan, 2001). Açısal hız sıfır olarak kabul edilirse stator ve rotor kaçak akılarının denklemleri Denklem 3.15'deki gibidir.

$$\begin{aligned}
 \lambda_{qs}^s &= L_s i_{qs}^s + L_m i_{qr}^s \\
 \lambda_{ds}^s &= L_s i_{ds}^s + L_m i_{dr}^s \\
 \lambda_{qr}^s &= L_r i_{qr}^s + L_m i_{qs}^s \\
 \lambda_{dr}^s &= L_r i_{dr}^s + L_m i_{ds}^s
 \end{aligned} \tag{3.15}$$

Denklem 3.15'de stator endüktansı  $L_s = L_{ls} + L_m$  şeklinde ifade edilir ve kaçak endüktansı ile mıknatıslanma endüktansının toplamına eşittir. Aynı şekilde rotor endüktansının ifadesi de  $L_r = L_{lr} + L_m$  şeklinde ifade edilir ve rotor kaçak endüktansı ile mıknatıslanma endüktansının toplamına eşittir.

Stator ve rotor gerilimlerinin stator eksenindeki ifadesi Denklem 3.16'daki gibidir.

$$\begin{aligned}
 v_{qs}^s &= R_s i_{qs}^s + \frac{d\lambda_{qs}^s}{dt} \\
 v_{ds}^s &= R_s i_{ds}^s + \frac{d\lambda_{ds}^s}{dt} \\
 v_{qr}^s &= R_r i_{qr}^s - \omega_r \lambda_{dr}^s + \frac{d\lambda_{qr}^s}{dt} \\
 v_{dr}^s &= R_r i_{dr}^s + \omega_r \lambda_{dr}^s + \frac{d\lambda_{qr}^s}{dt}
 \end{aligned} \tag{3.16}$$

ASM'lerde üretilen elektromanyetik moment ifadesi Denklem 3.17'deki gibidir. Bu denklemdeki  $P$ ; kutup sayısını,  $L_m$ ; üç faz süzleminden iki faz düzlemine indirgenmiş stator mıknatıslanma endüktansını ifade eder.

$$T_e = \frac{3P}{2} L_m (i_{qs}^s i_{dr}^s - i_{ds}^s i_{qr}^s) \tag{3.17}$$

Motorun dengeli bir kaynaktan beslendiği varsayılırsa akımların toplamı sıfırdır. Bu durumda  $f_{as} + f_{bs} + f_{cs} = 0$  olduğu kabul edilir. İki fazlı  $d - q$  sistemi Denklem 3.18'deki gibi ifade edilir.

$$\begin{aligned}
 f_{qs} &= f_{as} \\
 f_{ds} &= \frac{1}{\sqrt{3}} (f_{cs} - f_{bs})
 \end{aligned} \tag{3.18}$$

### Senkron referans düzlemi

ASM stator ve rotor değişkenleri senkron referans düzleme aktarılır. Bu durum Şekil 3.2'de  $\theta = \theta_e$  ve  $\omega = \omega_e$  olması durumudur. Senkron eksen takımında dönüşümler için motor senkron elektriksel hız ve pozisyon değeri kullanılır. Stator ve rotor akılarının eşitlik ifadeleri Denklem 3.19'deki gibidir. Denklem 3.20'de ise gerilimlerin senkron eksen düzlemindeki ifadeleri verilmiştir.



$$\begin{aligned}
\lambda_{qs}^e &= L_s i_{qs}^e + L_m i_{qr}^e \\
\lambda_{ds}^e &= L_s i_{ds}^e + L_m i_{dr}^e \\
\lambda_{qr}^e &= L_r i_{qr}^e + L_m i_{qs}^e \\
\lambda_{dr}^e &= L_r i_{dr}^e + L_m i_{ds}^e
\end{aligned} \tag{3.19}$$

$$\begin{aligned}
v_{qs}^e &= R_s i_{qs}^e + \omega_r \lambda_{ds}^e + p \lambda_{qs}^e \\
v_{ds}^e &= R_s i_{ds}^e + \omega_r \lambda_{qs}^e + p \lambda_{ds}^e \\
v_{qr}^e &= R_r i_{qr}^e + (\omega_e - \omega_r) \lambda_{dr}^e + p \lambda_{qr}^e \\
v_{dr}^e &= R_r i_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e + p \lambda_{dr}^e
\end{aligned} \tag{3.20}$$

ASM'de üretilen elektromanyetik moment ifadesi Denklem 3.21'de verilmiştir.

$$T_e = \frac{3P}{2} L_m (i_{qs}^e i_{dr}^e - i_{ds}^e i_{qr}^e) \tag{3.21}$$

Denklem 3.21'deki elektromanyetik moment ifadesi ASM'nin senkron hızda dönen düzlem sistemindeki elektriksel davranışlarından çıkartılmıştır. Denklem 3.22'de ise; elektromanyetik moment ifadesi, ASM'nin senkron hızda dönen düzlem sisteminde dinamik davranışından çıkartılmıştır. Bu denklemdeki  $J$  atalet momenti,  $\omega_m$  mekaniksel hız,  $B$  sürtünme katsayısı ve  $T_L$  yük momentidir.

$$T_e = J \frac{d\omega_m}{dt} + B\omega_m + T_L = \frac{2}{P} J \frac{d\omega_r}{dt} + B\omega_r + T_L \tag{3.22}$$

### Referans düzlemleri arası dönüşüm

ASM hız kontrolü için kullanılacak denetleyicilerde değişik referans düzlem dönüşümleri gerekebilir. Durağan düzlemde senkron referans düzleme dönüşüm genellikle alan yönlendirmeli kontrol tekniğinde gerekli olmaktadır. Aynı şekilde senkron referans düzlemde durağan referans düzleme dönüşüm de gereklidir (Bingöl, 2005). Durağan referans düzlemde senkron referans düzleme dönüşüm Denklem 3.23'de gösterildiği gibidir. Senkron referans

düzleminden tekrar durağan referans düzlemine çevrim ise Denklem 3.24'de gösterilmiştir.

$$\begin{bmatrix} f_{qs}^e \\ f_{ds}^e \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} f_{qs}^s \\ v_{ds}^s \end{bmatrix} \quad (3.23)$$

$$\begin{bmatrix} f_{qs}^s \\ f_{ds}^s \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} f_{qs}^e \\ f_{ds}^e \end{bmatrix} \quad (3.24)$$

### 3.1.4. Dolaylı vektör kontrol yöntemi

Dolaylı vektör kontrol yönteminde rotor miline yerleştirilen pozisyon veya hız algılayıcısı kullanılır. Şekil 3.4'de dolaylı vektör kontrolüne ait düzlem dönüşümleri gösterilmiştir.  $d^s - q^s$  eksenleri stator eksenine sabitlenmiş olup durağan haldedir. Rotor değişkenleri  $d^r - q^r$  ise rotor eksenine sabitlenmiş olup  $\omega_r$  hızında dönmektedir. Senkron referans eksenini  $d^e - q^e$  ile rotor referans eksenini arasında pozitif kayma açısı  $\theta_{sl}$  vardır. Dolayısıyla senkron pozisyon ( $\theta_e$ ), rotor pozisyonu ( $\theta_r$ ) ile kayma pozisyonunun toplamı veya senkron hız rotor hızı ile kayma hızının toplamının integraline eşittir. Bu durum Denklem 3.25'de gösterilmiştir.

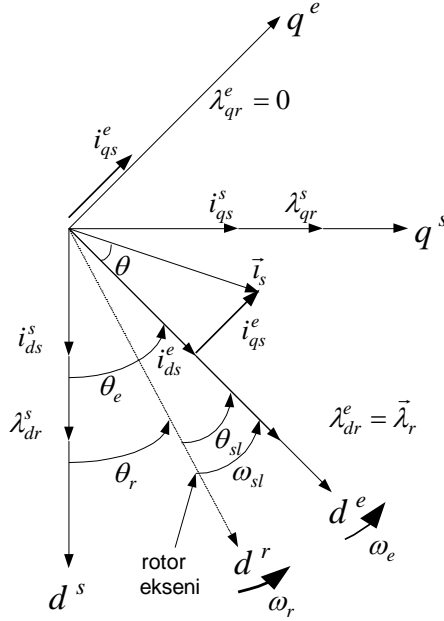
$$\theta_e = \int \omega_e dt = \int (\omega_r + \omega_{sl}) dt = \theta_r + \theta_{sl} \quad (3.25)$$

Senkron referans düzlemi için stator ve rotor gerilim eşitlikleri Denklem 3.20'de gösterilmiştir. Sincap kafesli asenkron motorda, rotor devresi kısa devre edildiği için rotor gerilimleri sıfır olarak alınır. Rotor devresi gerilimleri Denklem 3.26'daki gibidir.

$$\begin{aligned} R_r i_{qr}^e + (\omega_e - \omega_r) \lambda_{dr}^e + p \lambda_{qr}^e &= 0 \\ R_r i_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e + p \lambda_{dr}^e &= 0 \end{aligned} \quad (3.26)$$

Senkron hız durumundan rotor hızı çıkarıldığında kayma hız değeri elde edilir. Bu durum Denklem 3.27'de gösterilmiştir.

$$\omega_{sl} = \omega_e - \omega_r \quad (3.27)$$



Şekil 3.4. Dolaylı vektör kontrolüne ait düzlem dönüşümleri (Bingöl, 2005)

Senkron referans düzlemi için akı denklemleri göz önüne alınarak ve rotor devresi akı eşitliğini kullanarak rotor devresine ait akımlar Denklem 3.28'deki gibi ifade edilir.

$$\begin{aligned} i_{dr}^e &= \frac{1}{L_r} \lambda_{dr}^e - \frac{L_m}{L_r} i_{ds}^e \\ i_{qr}^e &= \frac{1}{L_r} \lambda_{qr}^e - \frac{L_m}{L_r} i_{qs}^e \end{aligned} \quad (3.28)$$

Alan yönlendirmeli kontrolde, rotor akı vektörü  $\vec{\lambda}_r^e$  birbirinden bağımsız  $\lambda_{dr}^e$  ve  $\lambda_{qr}^e$  değerlerinin vektörel bileşkesinden oluşmaktadır. Denetlenecek değişken sayısının bire indirilmesi için akı vektörlerinden  $\lambda_{qr}^e$  sıfıra eşitlenir. Akının sıfır olması türevini de sıfır yapacaktır. Bu durumda rotor akı vektörü, akının sadece  $d^e$  bileşeninden oluşacaktır. Denklemler yeniden yazılacak olursa kayma hız değeri Denklem 3.29'daki gibi elde edilir.

$$\omega_{sl} = \frac{R_r L_m}{L_r \overline{\lambda}_r^e} i_{qs}^e \quad (3.29)$$

Senkron dönen referans düzleminde rotor mıknatıslanma akımı Denklem 3.30'da verilmiştir.

$$i_{mr}^e = \frac{\overline{\lambda}_r^e}{L_m} \quad (3.30)$$

Bu durumda  $i_{dr}^e$  ifadesi yeniden düzenlenecek olursa Denklem 3.31'deki gibi yazılabilir.

$$i_{dr}^e = \frac{L_m}{L_r} i_{mr}^e - \frac{L_m}{L_r} i_{ds}^e \quad (3.31)$$

$i_{ds}^e$  ifadesi yeniden düzenlenecek olursa Denklem 3.32'deki gibi yazılabilir.

$$i_{ds}^e = i_{mr}^e - \frac{L_r}{R_r} p i_{mr}^e \quad (3.32)$$

Denklem 2.23.'deki elektromanyetik moment ifadesindeki rotor akımları yerine motor akımları kullanılarak tekrar yazılırsa Denklem 3.33 elde edilir.

$$T_e = \frac{3 P L_m}{2 L_r} (\lambda_{dr}^e i_{qs}^e - \lambda_{qr}^e i_{ds}^e) \quad (3.33)$$

$\lambda_{qr}^e$  değerinin sifıra eşitlenerek değişken sayısının bire indirgeneceği daha önce anlatılmıştı. Bu durum Denklem 3.33'e uygulanırsa  $i_{qs}^e$  akımı Denklem 3.34'deki gibi olur.

$$i_{qs}^e = \frac{T_e}{\frac{3 P L_m^2}{2 L_r} i_{ds}^e} \quad (3.34)$$

Denklem 3.33'deki elektromanyetik moment ifadesi ise Denklem 3.35'deki gibi yazılır.

$$T_e = \frac{3P}{2} \frac{L_m^2}{L_r} (i_{qs}^e i_{ds}^e) \quad (3.35)$$

### 3.2. Üç-Seviyeli Evirici ve Modülasyon Teknikleri

Eviriciler, girişindeki DA gerilimini istenilen değer ve frekansta AA gerilimine dönüştüren ve bu gerilimi çıkış olarak veren çeviricilerdir. Girişindeki DA gerilimi sabit ya da değişken olabilir.

Eviriciler çalışma alanlarına göre ikiye ayrılırlar. Bunlar, bir fazlı ve üç fazlı eviricilerdir. Bir fazlı eviriciler genellikle kesintisiz güç kaynaklarında (UPS) kullanılırlar. Üç fazlı eviricilerden gerilim beslemeli eviriciler alternatif akım motor sürücü sistemleri, kesintisiz güç kaynakları ve indüksiyon ısıtma sistemlerinde kullanılmaktadırlar. Üç fazlı eviricilerden akım beslemeli olanlar ise büyük güçlü alternatif akım motor sürücü sistemlerinde kullanılırlar (Bose, 2002).

Eviriciler, beslenme özelliklerine göre de gerilim kaynaklı ve akım kaynaklı eviriciler olmak üzere iki kısımda incelenirler (Asker vd., 2009).

Eviriciler sinüsoidal olmayan bir çıkış gerilimi üretirler. Üç fazlı eviricilerde çıkış gerilimleri simetriktir. Eviricinin çıkış gerilimine Fourier analizi yapıldığında temel bileşenlere ek olarak harmonik bileşenlerinin de olduğu görülmektedir (Bingöl, 2005). Çıkış gerilimindeki harmonikler genel olarak iki durumda en aza indirgenebilir. Birinci durumda yüksek anahtarlama frekansına sahip yarı iletken anahtarlara farklı DGM teknikleri uygulanır. İkinci durumda ise eviricinin çıkış dalga şeklinin seviye sayısı artırılır (Nabae vd., 1981).

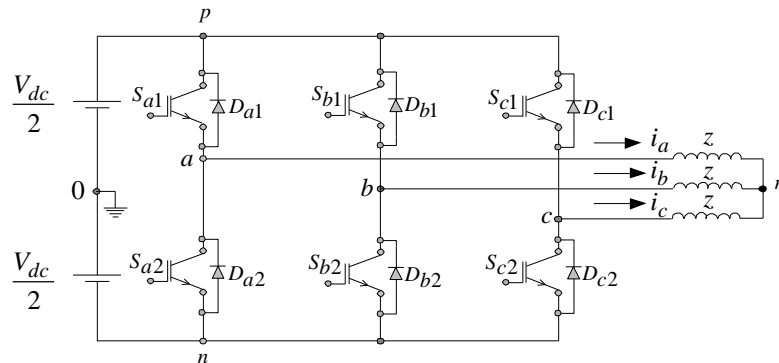
Gerilim beslemeli eviriciler genel olarak kare dalga eviriciler ve DGM' li eviriciler olarak ikiye ayrılırlar (Tuncay vd., 2003).

Kare dalga eviricilerin çıkış geriliminin genliği, girişindeki doğru akım geriliminin ayarlanmasıyla sağlanır. Dolayısıyla evirici sadece çıkış geriliminin frekansını kontrol eder. Çıkış geriliminin şekli kare dalga biçimindedir ki bu tip eviricilere gerilim beslemeli kare dalga eviriciler denmiştir. Gerilim beslemeli kare dalga eviricileri içerdikleri yüksek harmonikler nedeniyle daha çok küçük ve orta güçlü motor uygulamalarında kullanılmışlardır (Bingöl, 2005).

Darbe genişlik modülasyonlu (DGM) eviricilerin girişindeki DA gerilimi hemen hemen sabittir. Çünkü giriş gerilimi şebeke geriliminin bir doğrultucu tarafından doğrultulmasıyla sağlanır. Bu sebeple evirici çıkış geriliminin hem genliğini hem de frekansını ayarlamak zorundadır. Ayarlama işlemi DGM ile yapıldığı için bu tür eviricilere DGM' li eviriciler denmiştir (Holtz, 1994; Bose, 2002).

### 3.2.1. İki seviyeli gerilim beslemeli evirici

İki seviyeli eviricide her faz kolunda iki adet yarıiletken anahtar bulunmaktadır. Anahtarlar tüm periyot süresince sırayla iletme ya da kesime gitmektedir. Bu işlemden dolayı eviricideki anahtarlama kayıpları artmaktadır. Çıkış gerilimleri iki seviyeden oluşmaktadır. Bu gerilim seviyeleri  $\frac{V_{dc}}{2}$  ve  $-\frac{V_{dc}}{2}$  'dir. Şekil 3.5'de iki seviyeli gerilim beslemeli eviricinin temel devre şeması verilmiştir (Paçacı, 2011).



Şekil 3.5. İki seviyeli gerilim beslemeli evirici (Paçacı, 2011)

### 3.2.2. Üç-seviyeli gerilim beslemeli eviriciler

ÇSE'ler orta ve yüksek gerilim beslemelerinde, motor sürücü sistemlerinde, güç dağıtımında ve güç kalitesinin önemli olduğu yerlerde kullanılırlar. Bu tip eviricilerin en belirgin avantajı, eviricinin anahtarlama frekansı arttırılmadan çıkış dalga gerilimindeki harmonik bileşenlerinin daha az olmasıdır (Nabae vd., 1981; Holtz, 1994).

ÇSE'lerin temel avantajları aşağıda özetlenmiştir (Zhang, 1995; Celanovic, 2000).

- Yüksek güçlü uygulamalarda kullanılırlar.
- Statik ve dinamik gerilim paylaşımının karmaşıklığı olmaksızın anahtarlar seri bağlanarak daha yüksek gerilim isteyen uygulamalarda kullanabilirler.
- Aynı anahtarlama frekansında çok-seviyeli eviricinin çıkış dalga şeklinin harmonik analizi iki seviyeli eviricilere göre daha üstündür. Böylece daha düşük anahtarlama frekanslarında çalışılabilir.
- Motor sargıları üzerindeki gerilim oranı  $dv/dt$  düşüktür.

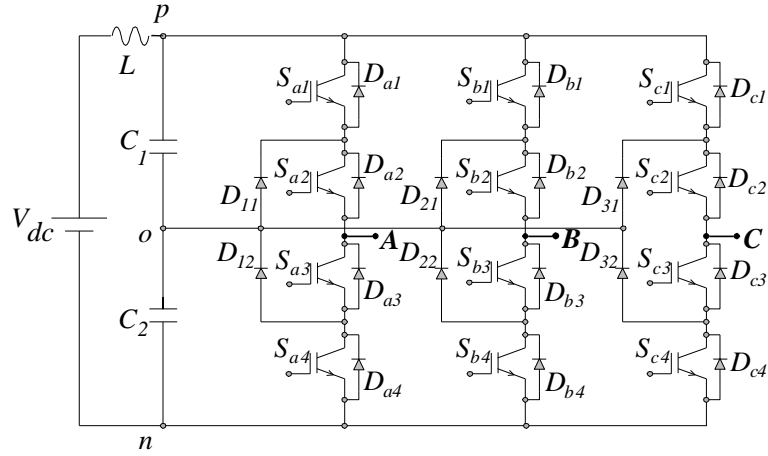
Yaygın olarak kullanılan çok seviyeli gerilim beslemeli eviriciler diyot tutmalı çok seviyeli evirici, kondansatör tutmalı çok seviyeli evirici ve seri bağlı tam köprü çok seviyeli eviricilerdir.

#### *Diyot tutmalı çok seviyeli eviriciler*

İlk pratik çok-seviyeli evirici 1981 yılında Nabae ve arkadaşları tarafından üç-seviyeli nötr-nokta tutmalı (NPC) evirici olarak önerilmiştir (Nabae vd., 1981).

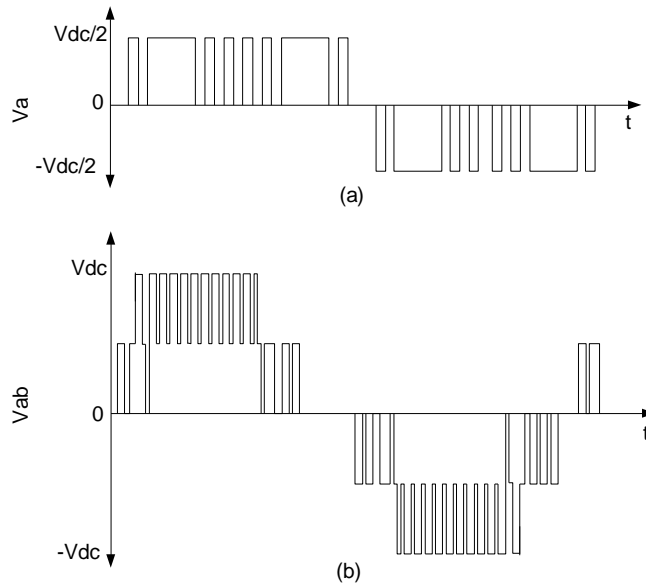
Diyot tutmalı çok seviyeli eviricilerde birbirine seri bağlanmış kondansatörler arasındaki düğüm noktası ile giriş gerilimi çeşitli seviyelere bölünür.  $n$  seviyeli diyot tutma eviricisi,  $n$  seviyeli çıkış faz gerilimine ve  $2n-1$  seviyeli çıkış hat

gerilimine sahiptir. Yine  $n$  seviyeli bir eviricide  $n - 1$  kondansatör ve her bir faz için  $(n - 1)(n - 2)$  tutma diyotu kullanılır.



Şekil 3.6. Üç-seviyeli diyot tutmalı evirici (Paçacı, 2011)

Şekil 3.6'da üç seviyeli diyot tutmalı eviricinin devre şeması gösterilmiştir. Şekilde görüldüğü gibi, eviricinin her bir fazında iki tutma diyotu ( $D_{11}$ ,  $D_{12}$ ), dört anahtar ( $S_{a1}$ ,  $S_{a2}$ ,  $S_{a3}$ ,  $S_{a4}$ ) ve dört bypass diyotu ( $D_{a1}$ ,  $D_{a2}$ ,  $D_{a3}$ ,  $D_{a4}$ ) yer almaktadır.



Şekil 3.7. Üç-seviyeli diyot tutmalı eviricinin a) Çıkış faz b) Çıkış hat gerilimleri (Paçacı, 2011)

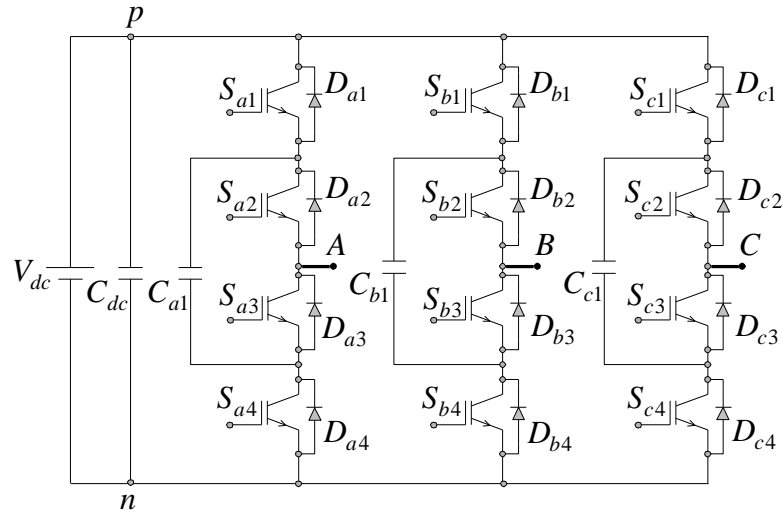


Şekil 3.7'de ise üç seviyeli diyot tutmalı bir eviriciye ait çıkış faz ve çıkış hat gerilimleri gösterilmiştir.

### Kondansatör tutmalı çok seviyeli eviriciler

Diyot-tutmalı çok-seviyeli eviriciye alternatif olarak 1992 yılında kondansatör-tutmalı evirici Meynard ve Foch tarafından önerilmiştir (Meynard vd., 1992).

Kondansatör tutmalı eviricinin yapısı diyot tutmalı eviriciye benzemektedir. Farkı tutma diyotları yerine tutma kondansatörlerinin kullanılmasıdır.  $n$  seviyeli kondansatör tutmalı evirici,  $n$  seviyeli kondansatör tutmalı evirici,  $n$  seviyeli çıkış faz gerilimine ve  $2n - 1$  seviyeli çıkış hat gerilimine sahiptir. Her bir kondansatör anahtarlama elemanı olarak aynı gerilim oranına sahiptir.  $n$  seviyeli bir evirici için DC hat üzerinde  $n - 1$  kondansatöre ve her bir faz için de  $\frac{(n-1)(n-2)}{2}$  yardımcı kondansatöre gereksinim duyulur. Üç seviyeli kondansatör tutmalı eviricinin devre şeması Şekil 3.8'de gösterilmiştir.



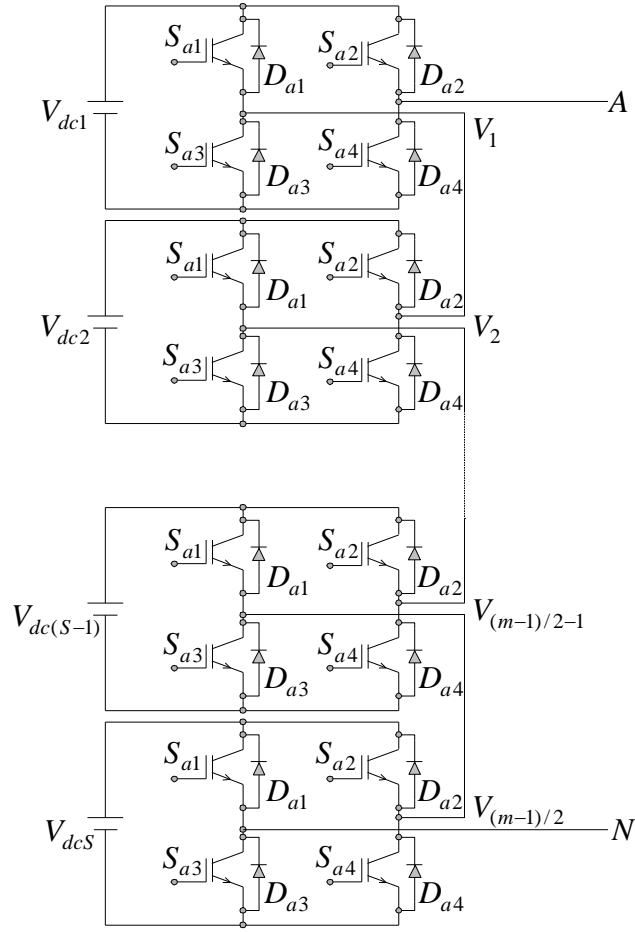
Şekil 3.8. Üç-seviyeli kondansatör tutmalı evirici (Paçacı, 2011)

### Seri bađlı tam köprü çok seviyeli eviriciler

Marchesoni ve arkadaşları tarafından diđer iki çok-seviyeli eviriciye alternatif olarak seri bađlı tam köprü çok-seviyeli eviriciyi önermişlerdir (Marchesoni vd., 1990).

Eviricinin çıkışında istenilen gerilimi elde etmek için birbirinden bađımsız, birden fazla bir fazlı tam köprü bađlı olan eviricilerin beslemeleri olan DA kaynaklar bulunmaktadır. Farklı seviyedeki eviricilerin AA gerilim çıktıları birbirleri ile seri bağlanmışlardır. Şekil 3.9'da bir fazlı n seviyeli seri bađlı tam köprü çok seviyeli evirici gösterilmiştir.

Bu evirici tipinde ekstra gerilim dengeleme kondansatörlerine ve tutma diyotlarına gerek yoktur. Bir fazlı tam köprü bađlı eviricide dört anahtar ( $S_1, S_2, S_3$  ve  $S_4$ ) vardır. Anahtarların farklı durumları ile her bir evirici seviyesi üç farklı gerilim çıkışı ( $V_{dc}, 0, -V_{dc}$ ) elde edilir. Tam köprü bađlı eviricilerin her bir farklı seviyelerdeki AA çıkışları seri bağlanmıştır. Dolayısıyla bu gerilimin dalga şeklinin toplamı eviricinin çıkış gerilimini verir. Çıkış faz gerilim seviyelerinin sayısı önceki iki eviriciden farklı olarak tanımlanmıştır. Bu eviricinin çıkış faz gerilim seviyelerinin sayısı  $n = 2s + 1$  olarak belirlenmektedir. Burada "s" DA kaynak sayısını ifade etmektedir.

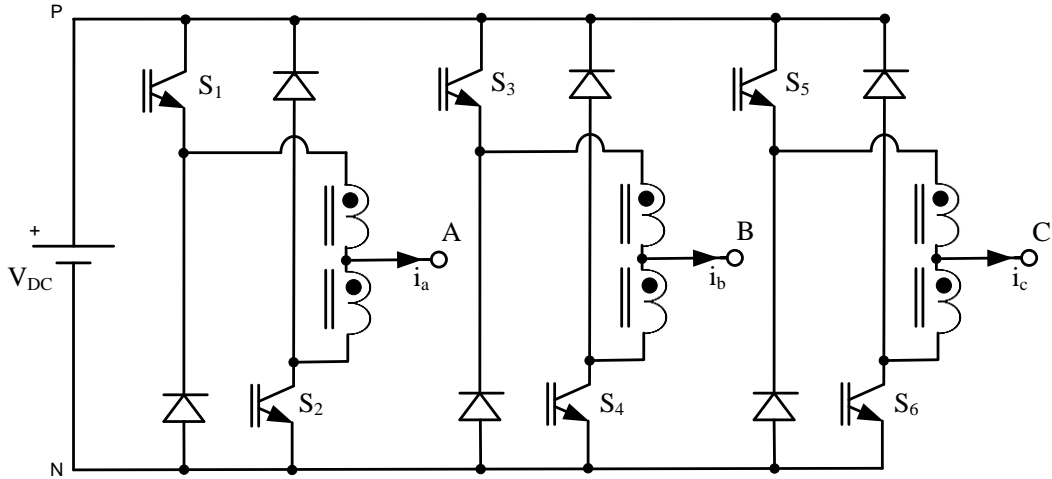


Şekil 3.9. Bir fazlı seri bağlı tam köprü çok seviyeli evirici (Paçacı, 2011)

### 3.2.3. Bağlaşık indüktörlü üç-seviyeli eviriciler

Çok seviyeli evirici topolojilerine alternatif bir topoloji geliştirilmiştir. Bu evirici topolojisinde altı adet IGBT yarı iletken anahtar kullanılmakla beraber eviricinin her bir çıkış uçlarına çift sarımlı bobinler bağlanmaktadır. Bağlaşık indüktörlü üç-seviyeli evirici Şekil 3.10'da verilmiştir. Bu topoloji, girişine uygulanan DA kaynak ve her bir çıkışa bağlanan 3 faz bölünmüş sargılı çift sarımlı bobinler aracılığı ile çok seviyeli çıkış gerilimleri üretmektedir.  $S_1$  ve  $S_2$  anahtarlarının her ikisinin de iletimde ya da kesimde olduğu durumlarda, çıkış terminal gerilimi olan  $V_{AN}$ 'den  $+\frac{V_{dc}}{2}$  lik bir gerilim elde edilmektedir.  $S_1$  anahtarı kesimde,  $S_2$  anahtarı iletimde olduğunda  $V_{AN}$  negatif DA barasına bağlanmış olur ve her hangi bir gerilim üretilmez.  $S_1$  anahtarı iletimde,  $S_2$  anahtarı kesimde olduğunda ise  $V_{AN}$  pozitif DA barasına bağlı olduğundan dolayı  $+V_{DC}$  gerilimi üretilir (Ewanchuk vd., 2008; Salmon vd., 2009; Vafakhah, 2010).

Bölünmüş sargılı bobin üst ve alt anahtarlarla seri bağlı olduğundan, bağlaşık indüktörlü üç-seviyeli evirici topolojisinin önemli bir avantajı anahtarlama esnasındaki geçiş akımı için ölü-zaman gereksinimidir. Böylece ölü-zamanın olumsuz etkileri bastırılmış olur. Yük veriminin ve güç yoğunluğun potansiyel iyileştirilmesiyle elde edilen geliştirilmiş DGM gerilimi AA bobinlerdeki ve makine yüklerindeki yüksek frekans kayıplarını düşürür. Bu özellikleriyle, bağlaşık indüktörlü üç-seviyeli evirici topolojisi daha yüksek frekanslı DGM çıkışların temel olduğu düşük hassasiyetli ve yüksek frekans temel uygulamalarında kullanılan alternatif topolojilerden daha uygundur (Matsui, 2000; Ewanchuk vd., 2008; Salmon vd., 2009; Vafakhah, 2010).



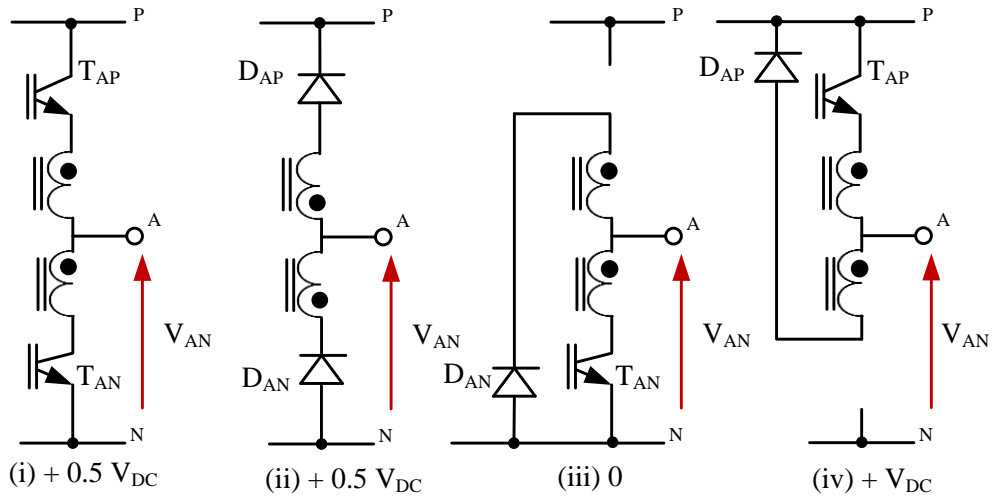
Şekil 3.10. Bağlaşık indüktörlü üç-seviyeli evirici (Bircenas vd., 2002)

### 3.2.4. Çok seviyeli eviricilerde DGM teknikleri

DGM tekniği ilk olarak 1960'lı yılların başlarında statik eviricilerin kontrolü uygulamasında kullanılmışlardır. Daha sonraları DGM tekniği güç elektroniği alanında yaygın olarak kullanılmaya başlanmıştır. DGM tekniği ile yarım periyot içinde anahtarlama elemanları bir çok kez uygun açılarda ilettime sokulup çıkartılarak evirici çıkış gerilimindeki harmonikler bastırılabilir. Eviricilerde DGM tekniği üç temel gruba ayrılabilir (Li, 1999). Bunlar; sinüsoidal DGM, histerisiz DGM ve uzay vektör DGM'dir.

### 3.2.5. Baęlaşık indüktörlü üç-seviyeli evirici ve uzay vektör DGM

Şekil 3.11’de gösterildięi gibi baęlaşık indüktörlü üç-seviyeli eviricinin çıkış terminal gerilimi olan  $V_{AN}$ ’den;  $+\frac{V_{dc}}{2}$ ,  $V_{DC}$  ve 0 gerilim deęerleri elde edilir. Standart eviricilerde üst ve alt anahtarların açma ve kapanma zamanları DC bara kısa devresini önleyici niteliktedir. Tek bacaktaki her bir anahtar dięer anahtar ile baęımsız çalışamaz. Fakat baęlaşık indüktörlü üç-seviyeli evirici topolojisinde üst ve alt anahtarlar orta-gerilim noktasını oluşturmak için birbirinden baęımsız davranmamak şartı ile aynı anda açılıp kapanabilirler (Vafakhah, 2010).



Şekil 3.11. Baęlaşık indüktörlü üç-seviyeli eviricinin tek bacağındaki anahtarlama durumları (Bircenas vd., 2002)

Şekil 3.11(i) ve Şekil 3.11(ii)’de anahtarlarının her ikisinin de iletimde ya da kesimde olduęu durumları gösterirken, 3.11(iii) ve 3.11(iv)’de ise anahtarlardan birinin iletimde dięerinin kesimde olduęu durumları göstermektedir (Kerem, 2012).

Şekil 3.11(i) ve 3.11(ii)’de gösterildięi gibi, baęlaşık indüktörlü üç-seviyeli evirici topolojisi üçüncü gerilim seviyesine bölünmüş sargılı bobin üzerinde oluşan gerilim bölünmesi ile ulaşmaktadır. Her bir bacakta üst ve alt anahtarların her ikisi de aynı anda iletime geçtiğinde ya da kesime gittiğinde,

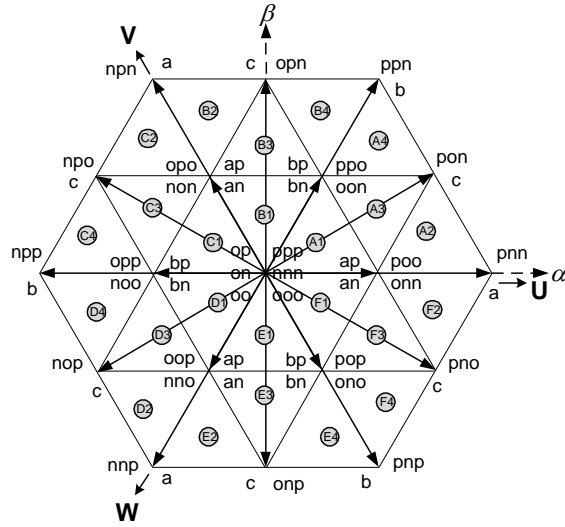
üçüncü gerilim seviyesi olan  $\frac{V_{dc}}{2}$  ye ulaşılır. Şekil 3.11(i)'deki gibi anahtarların her ikisi de iletimde olduğunda,  $V_{DC}$  gerilimi bölünmüş sargılı bobin üzerinden akar (Kerem, 2012).

Bu iki bobin arasındaki bağlantı oranı bir olup her iki bobin de eş tasarımlı olduğundan, gerilim bobinler üzerinde eşit dağılır ve çıkış terminalinde  $0.5 V_{DC}$  elde edilir. Aynı şekilde, Şekil 3.11(ii)'deki gibi anahtarlar kesimde olduğunda, çift sarımlı bobin sargı geriliminin aynı kutuplu olduğu düşünülürse, bobin üzerinde  $-V_{DC}$  oluşur. Ama bu kez, bobin yönüne bağlı olarak, çıkıştan  $+0,5 V_{DC}$  elde edilir (Vafakhah, 2010).

Şekil 3.11(iii) ve 3.11(iv)'deki gibi anahtarlardan birinin iletim diğerinin kesimde olduğu durumda, çift sarımlı bobin uçları kısa devre olur ve bobinin iki ucu ya üst DC baraya (P) ya da alt DC baraya (N) bağlanır (Kerem, 2012).

Çeşitli anahtarlama durumlarına ilişkin çıkış terminal gerilimi (0-Anahtar kesimde, 1-Anahtar iletimde) Çizelge 3.1'de verilmiştir. Her bir fazın 4 anahtarlama durumu vardır. Her bir fazdaki anahtarlama durumlarının kombinasyonları üç fazlı çıkış sinyali üretmek için kullanılmaktadır (Vafakhah, 2010).

Anahtarlama durumlarına göre Şekil 3.12'de gerilim vektörleri genliklerine göre dört gruba ayrılmaktadır. Bunlar, Çizelge 3.2'de gösterilmiştir (Bousquest-Monge vd., 2004).



Şekil 3.12. Üç-seviyeli eviricinin çıkış gerilimlerinin anahtarlama durumları ve uzay vektörü (Bingöl, 2005)

Çizelge 3.1. Bağlaışık indüktörlü üç-seviyeli eviricinin anahtarlama durumları (Vafakhah, 2010)

Anahtarlama Durumu	$S_{x1}$	$S_{x2}$	$S_{x3}$	$S_{x4}$	$S_{x5}$	$S_{x6}$	$V_{x0}$
p	1	1	1	1	1	1	$\frac{V_{dc}}{2}$
n	0	0	0	0	0	0	$\frac{V_{dc}}{2}$
o	0	1	0	1	0	1	0
o	1	0	1	0	1	0	$V_{dc}$

Şekil 3.13'de dıştaki altıgen altı adet sektöre sahiptir. Bunlar A, B, C, D, E ve F olarak isimlendirilmişlerdir. Her sektörde de 1, 2, 3 ve 4 olarak adlandırılan dört adet bölge bulunur. Dolayısıyla toplam bölge sayısı 24' tür.

Çizelge 3.2.'de üç-seviyeli eviricinin gerilim vektörlerinin gruplandırılması gösterilmiştir.

Çizelge 3.2. Üç-seviyeli eviricinin gerilim vektörlerinin gruplandırılması

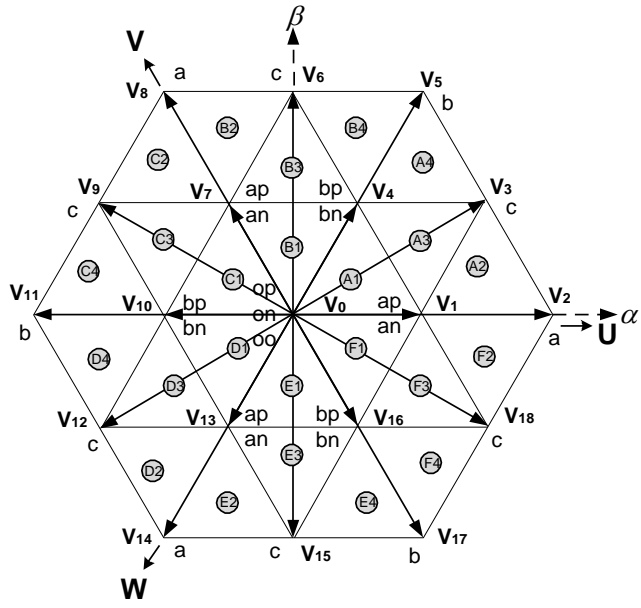
Gerilim Vektörleri	Grupları	Anahtarlama Durumları	Genlikleri
Büyük Gerilim Vektörleri	a	pnn,npn,nnp	$\frac{2V_{dc}}{3}$
	b	ppn,npp,pnp	
Orta Gerilim Vektörleri	c	pon,opn,npo,nop,onp,pno	$\frac{V_{dc}}{\sqrt{3}}$
Küçük Gerilim Vektörleri	ap,an	poo-onn,opo-non,oop-noo	$\frac{V_{dc}}{3}$
	bp,bn	ppo-oon,opp-noo,pop-ono	
Sıfır Gerilim Vektörleri	op,on,oo	ppp,nnn,ooo	0

Gerilim vektörünün her bir boyutu (sıfır vektör, küçük vektör, orta vektör ve büyük vektör) için olası bobin konfigürasyonları Çizelge 3.2’de verilerek, bobin endüktansının genliğine göre gruplandırılmıştır.

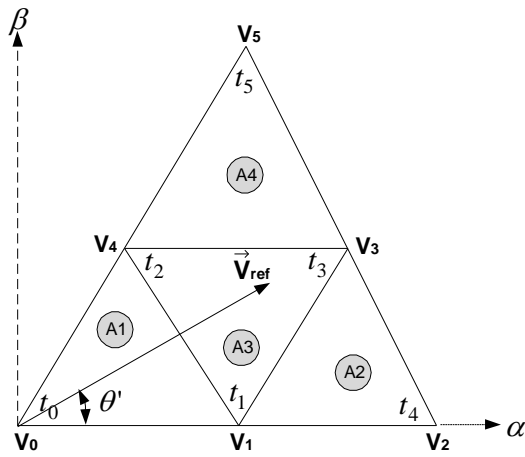
#### Gerilim vektör sürelerinin bulunması

Şekil 3.13’de üç seviyeli eviricinin gerilim uzay vektörleri gösterilmektedir. Örneğin referans gerilim uzay vektörü A sektörünün 3. bölgesinde ise, Şekil 3.14’e göre referans gerilimlerinin büyüklükleri Denklem 3.36’daki ve komşu vektörlerin uygulama süresi Denklem 3.40’daki gibi ifade edilir.





Şekil 3.13. Üç-seviyeli eviricinin gerilim uzay vektörleri



Şekil 3.14. A sektörüne ait uzay gerilim vektörleri

A sektörünün 3. bölgesindeki gerilim vektörlerinin genlikleri Denklem 3.36'da gösterilmiştir.

$$\begin{aligned}
V_1 &= \frac{1}{3}V_{dc}e^{j0} \\
V_3 &= \frac{1}{\sqrt{3}}V_{dc}e^{j\frac{\pi}{6}} \\
V_4 &= \frac{1}{3}V_{dc}e^{j\frac{\pi}{6}} \\
V_{ref} &= Ve^{j\theta'} \tag{3.36}
\end{aligned}$$

$T_s$  örnekleme zamanı içerisinde gerilim vektörlerinin zaman süreleri Denklem 3.37'de gösterilmiştir.

$$\begin{aligned}
V_{ref}T_s &= V_1t_1 + V_3t_3 + V_4t_2 \\
T_s &= t_1 + t_3 + t_2 \tag{3.37}
\end{aligned}$$

Denklem 3.36, Denklem 3.37'de yerine konulursa Denklem 3.38 elde edilir.

$$\begin{aligned}
V_{ref}(\cos\theta' + jsin\theta')T_s \\
&= \frac{V_{dc}t_1}{3} + \frac{V_{dc}\left(\cos\frac{\pi}{6} + jsin\frac{\pi}{6}\right)t_3}{\sqrt{3}} + \frac{V_{dc}\left(\cos\frac{\pi}{3} + jsin\frac{\pi}{3}\right)t_2}{3} \tag{3.38}
\end{aligned}$$

Denklem 3.38 gerçık ve sanal kısımlarına ayrıldığında Denklem 3.39 elde edilir.

$$\begin{aligned}
\text{Gerçık} : V_{ref} \cos\theta' T_s &= \frac{V_{dc}t_1}{3} + \frac{V_{dc} \cos\frac{\pi}{6} t_3}{\sqrt{3}} + \frac{V_{dc} \cos\frac{\pi}{3} t_2}{3} \\
\text{Sanal} : V_{ref} \sin\theta' T_s &= \frac{V_{dc} \sin\frac{\pi}{6} t_3}{\sqrt{3}} + \frac{V_{dc} \sin\frac{\pi}{3} t_2}{3} \tag{3.39}
\end{aligned}$$

Gerilim vektörlerinin zaman süreleri Denklem 3.40'da verilmiştir.

$$\begin{aligned}
t_1 &= T_s (1 - 2k \sin \theta') \\
t_2 &= T_s \left[ 1 - 2k \sin \left( \frac{\pi}{3} - \theta' \right) \right] \\
t_3 &= T_s \left[ 2k \sin \left( \frac{\pi}{3} + \theta' \right) - 1 \right]
\end{aligned} \tag{3.40}$$

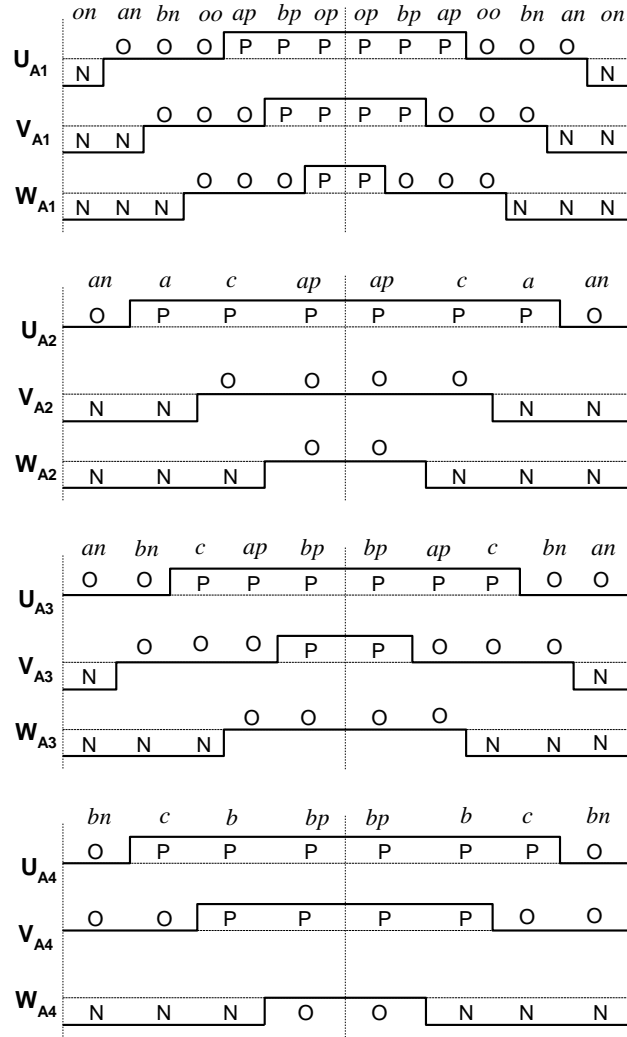
Denklem 3.40'daki  $k$  ifadesi  $\sqrt{3} \frac{V_{ref}}{V_{dc}}$  dir.

Çizelge 3.3. Farklı bölgelerdeki gerilim vektörlerinin zaman süreleri

Bölgeler	Vektörler	Zaman Süreleri
A1, B1, C1, D1, E1, F1	op, on, oo	$t_0 = T_s \left[ 1 - 2k \sin \left( \theta' + \frac{\pi}{3} \right) \right]$
	ap, an	$t_1 = 2kT_s \sin \left( \frac{\pi}{3} - \theta' \right)$
	bp, bn	$t_2 = T_s \left[ 1 - 2k \sin \left( \theta' + \frac{\pi}{3} \right) \right]$
A2, B2, C2, D2, E2, F2	ap, an	$t_1 = 2T_s \left[ 1 - k \sin \left( \theta' + \frac{\pi}{3} \right) \right]$
	c	$t_3 = 2kT_s \sin \theta'$
	a	$t_4 = T_s \left[ 2k \sin \left( \frac{\pi}{3} - \theta' \right) - 1 \right]$
A3, B3, C3, D3, E3, F3	ap, an	$t_1 = T_s [1 - 2k \sin \theta']$
	bp, bn	$t_2 = T_s \left[ 1 - 2k \sin \left( \frac{\pi}{3} - \theta' \right) \right]$
	c	$t_3 = T_s \left[ 2k \sin \left( \theta' + \frac{\pi}{3} \right) - 1 \right]$
A4, B4, C4, D4, E4, F4	bp, bn	$t_2 = 2T_s \left[ 1 - k \sin \left( \theta' + \frac{\pi}{3} \right) \right]$
	c	$t_3 = 2kT_s \sin \left( \frac{\pi}{3} - \theta' \right)$
	b	$t_5 = T_s [2k \sin(\theta' - 1)]$

Her bir sektörün zaman süreleri Çizelge 3.3'de özetlenerek verilmiştir. Burada  $T_s$  örnekleme periyodu,  $k$  modülasyon indeksi ve  $\theta'$  ise 'a' gerilim vektörüne en yakın açı değeridir (Yamanaka vd., 2002).

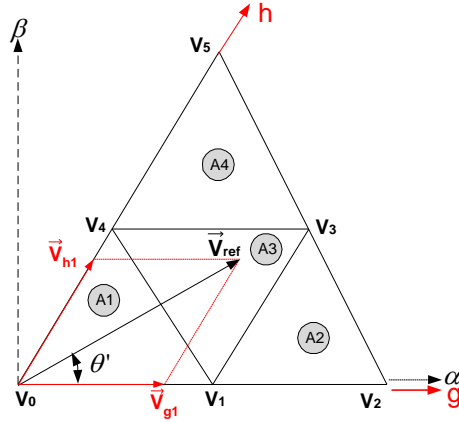
Şekil 3.15’da A sektörüne ait dört bölge için üç fazın DGM dalga şekilleri gösterilmiştir. Üç fazın DGM dalga şekilleri Çizelge 3.2’den yararlanarak çizilmiştir. Burada dalga şekilleri için simetrik dalga şekilleri kullanılmıştır (Mondal vd., 2002).



Şekil 3.15. A sektörüne ait dört bölge için üç fazın DGM dalga şekilleri

$V_\alpha$  ve  $V_\beta$  gerilim değerlerin karşılaştırılmasıyla, referans gerilim vektörünün hangi sektörün hangi bölgesinde olduğu bulunur. Bulunan  $t_1, t_2, t_3, t_4$  ve  $t_5$  süreleri her sektördeki bölgenin gerilim vektör iletim süreleridir. Hızlı olması açısından  $\alpha - \beta$  düzlemi üzerindeki  $V_\alpha$  ve  $V_\beta$  referans gerilim bileşenleri bir başka düzlem olan  $g - h$  düzlemi üzerine dönüştürülür. Dönüştürme işleminin eşitliği Denklem 3.41’de gösterildiği gibidir.  $\alpha - \beta$  ve  $g - h$  düzlemleri Şekil 3.16’da gösterilmiştir (Celanovic ve Boroyevich, 2000; 2001).

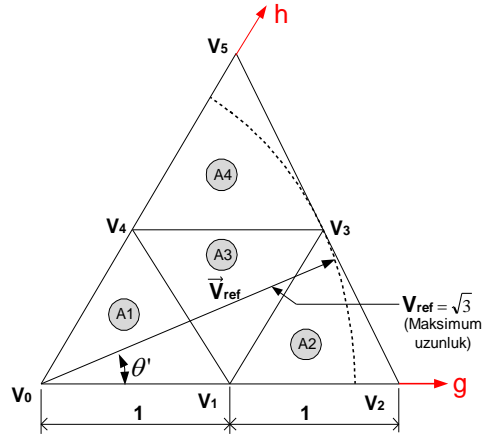
$$V_{ref\ g,h} = \begin{bmatrix} 1 & -\frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{3}} \end{bmatrix} V_{ref\ \alpha,\beta} \quad (3.41)$$



Şekil 3.16.  $\alpha - \beta$  düzlemi ile g-h düzlemi arasındaki bağlantı

Altıgenin içerisindeki tüm üçgenlerin uzunlukları bir birim olarak alınırsa referans gerilim vektörünün genliği de normalize edilmek zorundadır. Teoride normalize edilmiş referans gerilim vektörünün maksimum uzunluğu iki birim değerindedir. Ancak kararlı durum çalışma şartlarında referans gerilim vektörünün uzunluğu  $\sqrt{3}$  ile sınırlandırılmıştır. Bu durum Şekil 3.17’de gösterilmiştir (Bingöl, 2005).

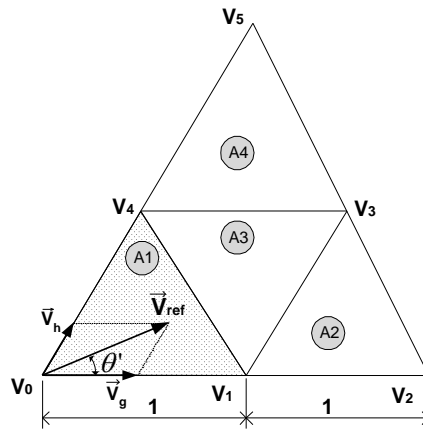
Çizelge 3.4’de  $g - h$  düzlemlerini kullanarak referans gerilim vektörünün hangi sektörde olduğu özetlenmiştir (Pou vd., 2002).



Şekil 3.17. Normalize edilmiş referans gerilim vektörünün maksimum uzunluğu (Bingöl, 2005)

Çizelge 3.4. g-h düzlemi kullanılarak sektör belirleme

$g - h$ Düzlemi	Sektör
$V_g \geq 0 V_h \geq 0$	A
$V_g < 0 V_h \geq 0 V_g + V_h \geq 0$	B
$V_g < 0 V_h \geq 0 V_g + V_h < 0$	C
$V_g < 0 V_h < 0$	D
$V_g \geq 0 V_h < 0 V_g + V_h < 0$	E
$V_g \geq 0 V_h < 0 V_g + V_h \geq 0$	F



Şekil 3.18. Normalize edilmiş referans gerilim vektörünün A sektörü 1. bölgesi

Çizelge 3.4'deki şartlar yardımıyla sektör bulunduktan sonra sektör içinde hangi bölgede olduğunu bulmak gerekir. Çizelge 3.5'de  $V_g$  ve  $V_h$  normalize edilmiş büyüklükleri birbiriyle karşılaştırılarak A sektöründeki bölgelerin bulunması özetlenmiştir. Normalize edilmiş referans gerilim vektörü A Sektörü birinci bölgede olup Şekil 3.18'de gösterilmiştir. Diğer sektörler için de bölge bulunması benzer şekildedir.

Çizelge 3.5. A sektörü için bölge bulunması durumları

Durum	Bölge
$V_g > 1$	A1
$V_g \leq 1, V_h \leq 1, V_g + V_h > 1$	A2
$V_h > 1$	A3
$V_g \leq 1, V_h \leq 1, V_g + V_h \leq 1$	A4

### 3.3. Bulanık Mantık

Bazı olayları açıklamak için kesin tanımlamalarda bulunabilmek imkânsızdır ve olaylar çoğu kere belirsizlikler ve doğrusal olmama özellikleri taşır. Terimler ya da ölçüler kesin olarak tanımlanıp ölçülemediğinden dolayı insanlar çoğu zaman belirsiz (kesin olmayan) ifadeler kullanırlar. İşte bulanık mantık bazı sorulara basitçe evet-hayır cevabı verilemeyen durumları kapsar. Bulanıklığın ve bulanık mantığın temeli de budur. Bulanık mantığın gücü basit şeyleri basit tutmaktır (Karaođlan, 2007).

Bulanık küme teorisi ilk olarak 1965 L. A. Zadeh tarafından ortaya atılmıştır. Polonyalı mantıkçı Jan Lucasiewicz 1920'li yıllarda önermelerin ikili mantığı 0 ve 1 değerleri arasında, kesirli değerlerde alabileceği mantık ilkelerini açıkladı. Bulanık mantık bir denetleyici olarak Mamdani ve Assilian tarafından bir buhar makinesinin, buhar basıncı ve motor hızının kontrolünde kullanılmıştır (Şafak, 2005).

Günümüzde bulanık mantığın uygulandığı pek çok alan mevcuttur. Kontrolü yapılan sistemlerin ortak özelliği, sistemlerin doğrusal olmaması veya sistemin matematiksel modelinin çok karmaşık veya hiç elde edilememesidir (Topuz, 2002).

Bulanık mantık, bulanık üyelik kavramı kullanarak insanın düşünme biçimi modellemeye çalışır ve dilsel değişkenler kullanır. Bulanık denetleyici, özellikle geleneksel sayısal tekniklerle analizi çok karmaşık olan sistemlerde ya da elde edilebilir bilgi kaynakları sayısal olmayan veya kesin olmayan sistemlerde çok etkindir (Baba, 1995).

### 3.3.1. Bulanık küme kuramı ve bulanık mantık

Matematikte küme, ayırt edilebilen belirli özellikleri olan nesnelere bütünüyle kavranmış topluluğu olarak tarif edilir. Bir kümeden bahsedildiği zaman onun bütün olarak kavranmış veya anlaşılabilir olması gerekir. Bir başka ifadeyle, küme kendisine neyin ait olup olmadığı konusunda herhangi bir şüphenin bulunmadığı topluluktur. İşte bahsedilen bu topluluğa veya nesnelere bu kümenin elemanları denir (Karaođlan, 2007).

Klasik küme teorisinde herhangi bir A kümesinde üyelerin kümeye ait olup olmaması aşağıdaki gibi bir karakteristik fonksiyonla ifade edilir (Denklem 3.42) (Karaođlan, 2007).

$$\mu_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases} \quad \forall x \in X \quad (3.42)$$

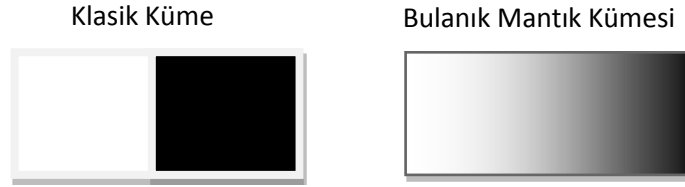
Herhangi, x elemanlarından oluşan X uzayı düşünöldüğünde, X uzayındaki A bulanık kümesi [0,1] aralığında gerçek sayı tanımlayan  $\mu_A(x)$  üyelik fonksiyonu ile oluşturulur.  $\mu_A(x)$  her x değerinin A kümesindeki üyelik derecesini verir. Klasik kümelerde bir kümeye üyelik 1 ya da 0 (üye ya da üye değil) olarak tanımlanır. Buradan yola çıkarak üyelik derecesi olarak 1 ya da 0 iki değer alabilen bir üyelik fonksiyonu kullanılmasıyla bulanık küme tanımlama mantığı ile klasik kümelerinde tanımlanabildiği görölmektedir. Bulanık kümeler,



elemanlarının  $[0,1]$  aralığında üyelik derecesi olan ve klasik kümeleri kapsayabilen kümelerdir denilebilir (Yıldız, 2008).

Bu tip kümelerde kısmen üyelik kavramı yer alır.  $X$  evrensel kümeyi,  $x$  ise bu kümeye ait genel bir elemanı göstermektedir. Bu tür kümelerde, söz konusu evrenin elemanlarının bir  $A$  bulanık kümesine ait olma derecesini temsil etmek için üyelik fonksiyonları belirlenir.  $X$  evrensel kümesi içerisinde  $A$  bulanık kümesinin üyelik fonksiyonu  $\mu_A(x)$  ile gösterilir ve Denklem 3.42'deki gibi ifade edilir.

Klasik küme ile bulanık küme mantığı Şekil 3.19'da gösterilmiştir.



Şekil 3.19. Klasik küme ve bulanık mantık kümesi (Ekren, 2009)

Bulanık denetleyici, algılayıcılardan gelen bilgileri, bulanık üyelik fonksiyonlarını kullanarak dilsel değişkenlere çevirir. Bu işleme bulanıklaştırma denir. Bu işlemde sonra kural tabanı ve veri tabanı içeren çıkarım biriminde, bulanık "Eğer... Öyle ise..." (If - Then) kurallarına göre değerlendirilerek mantıksal çıkarımlar, bulanık çıkışlar elde edilir. Elde edilen bulanık çıkış değerleri, durulaştırma yöntemlerinden biri kullanılarak kesin bir sayısal değere çevrilir. Elde edilen bu sayısal değer kontrol işaretini oluşturur ve sisteme uygulanır (Erdal, 1998).

Bulanık denetleyici genel olarak 4 ana bölümden oluşmaktadır.

- 1- Bulanıklaştırma birimi
- 2- Kural tabanı birimi
- 3- Çıkarım birimi
- 4- Durulaştırma birimi

Bulanık denetleyici tasarım işlemi genellikle bir deneme-yanılma döngüsünden oluşur (Topuz, 2002).

### **Bulanıklaştırma birimi**

Algılayıcılardan gelen sayısal bilgiyi, dilsel değişkenlere çevrildiği birimdir (Erdal, 1998). Kesin olmayan ölçümleri ifade eden her giriş için, bulanıklaştırma işlemi tanımlanır (Topuz, 2002). Bulanıklaştırma küme içerisindeki tanımlanan dilsel ifadeler ile ölçülen bulanık olmayan girişlerini haritalama işlemidir ve haritalanmış veriyi uygun dilsel terimleri üyelik değerlerine çevirir (Tong, 2000).

Bulanık denetleyicilerde çıkarım işlemi, bulanık değerlerle yapıldığı için, giriş vektörünün gerçek değerlerinin, bulanık değerlere dönüştürülmesi gerekmektedir. Matematiksel ifade ile bulanıklaştırma, giriş değer uzayını alt etki alanlarına ayırma işlemidir. Bulanık denetleyici tasarımının bir parçası olan alt etki alanlarına ayırma işlemi, sistem çıkış durum değişkenleri hakkında bazı bilgiler gerektirir. Literatürde üçgen, trapezoidal, sigmoid, gaussian, quadratik ve cos fonksiyonu formunda değişik üyelik fonksiyonları tanımlanmaktadır (Aydoğdu, 2006).

### **Üyelik fonksiyonları**

Üyelik fonksiyonu, giriş ve çıkış değerlerinin dilsel olarak tanımlanmasını sağlar. Ayrıca üyelik fonksiyonu, giriş ve çıkış değerlerini 0 ile 1 arasında tanımlayan geometrik bir ifadedir.  $X$  evrensel kümesindeki  $A$  bulanık kümesi sıralı çiftler kümesi şeklinde gösterilir. Bu durum Denklem 3.43'de gösterilmiştir. Buradaki  $\mu_A(x)$  üyelik fonksiyonunu ifade etmektedir.

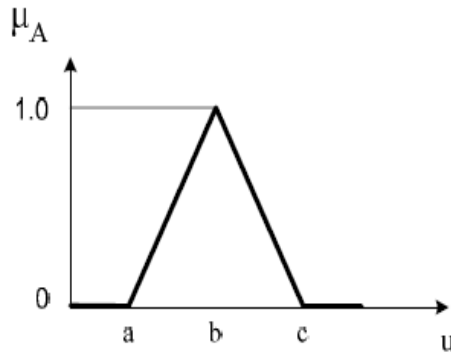
$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (3.43)$$

En sık kullanılan dört üyelik fonksiyonu vardır. Bunlar; üçgen üyelik fonksiyonu, trapezoidal üyelik fonksiyonu, sigmoid üyelik fonksiyonu ve gaussian üyelik fonksiyonudur.

### Üçgen üyelik fonksiyonu

Üçgen üyelik fonksiyonu, üç noktanın üçgen oluşturacak şekilde birleştirilmesi ile oluşturulur. Bu üç noktanın  $x$  eksenindeki değerleri  $a$ ,  $b$  ve  $c$  değişkenleri ile ifade edilir. Üçgen üyelik fonksiyonu Denklem 3.44'deki gibi tanımlanır ve Şekil 3.20'de gösterilmiştir.

$$\text{Üçgen}(u; a, b, c) = \begin{cases} 0 & u \leq a \\ \frac{(u - a)}{(b - a)} & a \leq u \leq b \\ \frac{(c - u)}{(c - b)} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (3.44)$$

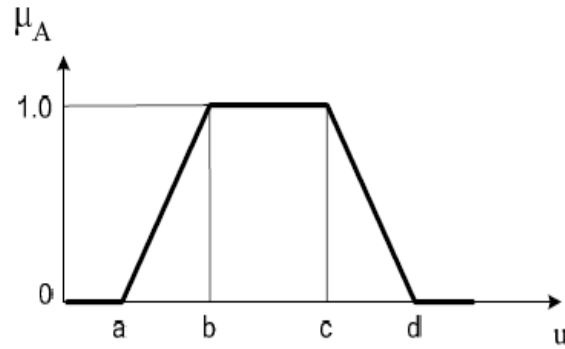


Şekil 3.20. Üçgen üyelik fonksiyonu

### Trapezoidal üyelik fonksiyonu

Trapezoidal üyelik fonksiyonuna yamuk üyelik fonksiyonu da denmektedir. Trapezoidal üyelik fonksiyonu, dört noktadan oluşur. Bu dört noktanın  $x$  eksenindeki değerleri  $a$ ,  $b$ ,  $c$  ve  $d$  değişkenleri ile ifade edilir. Trapezoidal üyelik fonksiyonu Denklem 3.45'deki gibi tanımlanır ve Şekil 3.21'de gösterilmiştir.

$$\text{Trapezoidal}(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{(u-a)}{(b-a)} & a \leq u < b \\ 1 & b \leq u \leq c \\ \frac{(d-u)}{(d-c)} & c < u \leq d \\ 0 & u > d \end{cases} \quad (3.45)$$

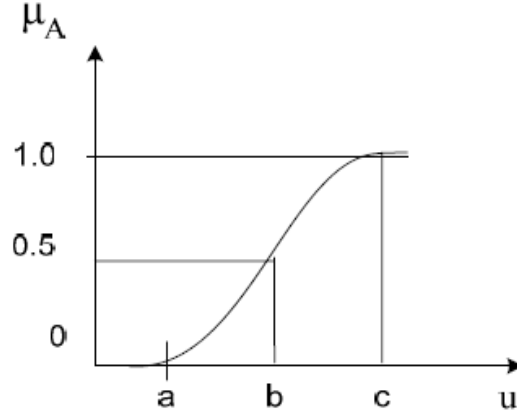


Şekil 3.21. Trapezoidal üyelik fonksiyonu

### Sigmoid üyelik fonksiyonu

Sigmoid üyelik fonksiyonu, üç noktadan oluşur. Bu üç noktanın x eksenindeki değerleri a, b ve c değişkenleri ile ifade edilir. Sigmoid üyelik fonksiyonu Denklem 3.46'daki gibi tanımlanır ve Şekil 3.22'de gösterilmiştir.

$$\text{Sigmoid}(u; a, b, c) = \begin{cases} 0 & u < a \\ 2 \left[ \frac{(u-a)}{(c-a)} \right]^2 & a \leq u \leq b \\ 1 - 2 \left[ \frac{(u-c)}{(c-a)} \right]^2 & b \leq u \leq c \\ 1 & u > c \end{cases} \quad (3.46)$$

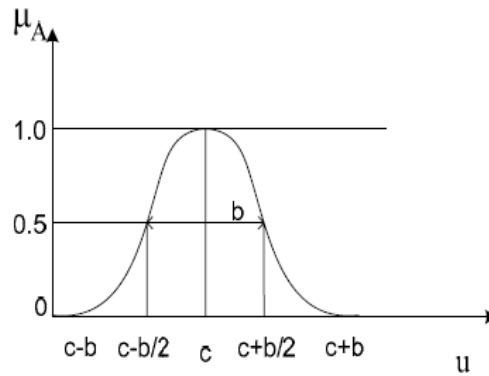


Şekil 3.22. Sigmoid üyelik fonksiyonu

Gaussian üyelik fonksiyonu

Gaussian üyelik fonksiyonu, iki noktadan oluşur. Bu iki noktanın  $x$  eksenindeki değerleri  $b$  ve  $c$  değişkenleri ile ifade edilir. Gaussian üyelik fonksiyonu Denklem 3.47'deki gibi tanımlanır ve Şekil 3.23'de gösterilmiştir.

$$\text{Gaussian}(u; b, c) = \begin{cases} \text{Sigmoid}\left(u; c - b, \frac{(c - b)}{2}, c\right) & u \leq c \\ 1 - \text{Sigmoid}\left(u; c, \frac{(c + b)}{2}, c + b\right) & u \geq c \end{cases} \quad (3.47)$$



Şekil 3.23. Gaussian üyelik fonksiyonu

### ***Bulanık küme teorisi***

Bulanık küme, klasik kümedeki açık ya da kapalı veya uzun ya da kısa gibi iki durumlu denetim değişken değerlerini az açık, az kapalı veya biraz uzun, biraz kısa gibi yumuşak değişken değerleri ile ifade eder. BM temelde insanın düşünüş şeklini aldığı için, bu mantığın küme elemanları da insan mantığına yakın elemanlarından oluşmaktadır (Şen, 2001).

Bir bulanık küme  $[0,1]$  arasında değerler alan üyelik fonksiyonu  $\mu_A(x)$  tarafından tanımlanır. Bir klasik kümenin üyelik fonksiyonu sadece iki değer yani 0 ve 1 değerlerini alabilirken bir bulanık kümenin üyelik fonksiyonu 0 ile 1 arasında herhangi bir sayı değeri alabilir (Karaođlan, 2007).

### **Kural tabanı birimi**

Kural tabanı, bulanık denetleyici davranışı belirleyen kontrol kurallarını içerir. Uzman kişinin bilgi ve tecrübelerinden yararlanılarak oluşturulmuş, bulanık kontrolörün davranışları belirleyen kontrol kurallarıdır (Erdal, 1998).

Kural tabanı eđer... ise... (if... then) biçiminde oluşturulmuş dilsel tanımlamalardır. Kurallardaki “eđer” terimi giriş değişkenlerinin bulanık ifadeleri, “ise” terimi her kurula ait kontrol davranışının sayısal değerlerini içermektedir. Örnek bir kural aşağıdaki gibi gösterilebilir.

**Eđer** Hata = sıfır ve hata değişimi = sıfır **ise** kontrol sinyali = 0.

Kural tabanının oluşturulmasında kullanılan beş değişik yöntem aşağıda maddeler halinde sıralanmıştır (Erdal, 1998).

1. Uzman kişinin bilgi ve tecrübesine göre bulanık model oluşturulur.
2. Sistem çalıştırılarak davranışları gözlenir. Gözlenen davranışlara göre bulanık model oluşturulur.
3. Sistemin matematik modelinden hareketle bulanık model oluşturulur.

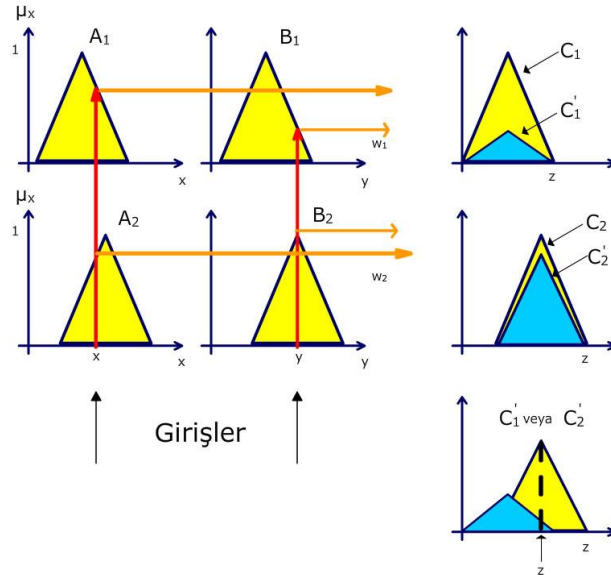
4. Kurallar denetleyici tarafından öğretilir.
5. Kurallar genetik algoritma gibi bir optimizasyon algoritması kullanılarak üretilebilir.

### Çıkarım birimi

Çıkarım işleminin amacı, giriş değişkenlerini, ilgili bulanık kurallar ile birleştirerek bulanık kontrol işaretinin bulunmasıdır. Farklı çıkarım modelleri bulunmaktadır. Çıkarım işleminde yaygın olarak dört farklı çıkarım tekniği kullanılır. Bunlar; Max-Dot, Min-Max, Tsukamoto ve Takagi-Sugeno çıkarım teknikleridir.

### Max-Dot çıkarım

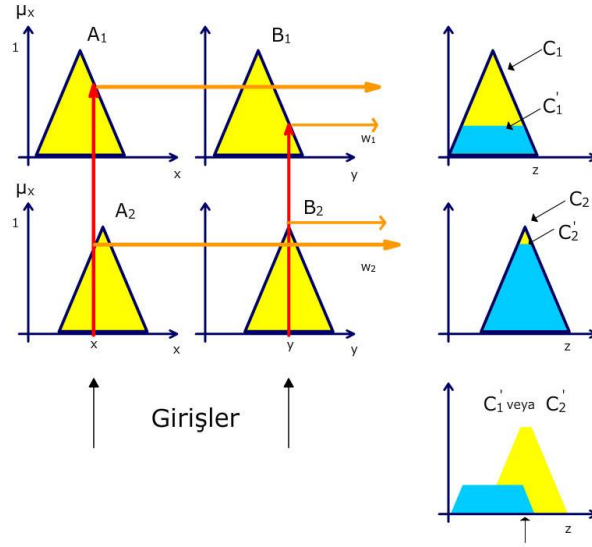
Her bir giriş değeri, ait olduğu üyelik fonksiyonundaki üyelik derecesine bağlı olarak ilgili bulanık kümeyi yeniden ölçeklendirir. Çıkış değeri, ölçeklenmiş bu kümeler içerisindeki maksimum değer alınarak bulunur. Şekil 3.24'de Max-Dot çıkarım tekniğine ait üyelik fonksiyon grafikleri verilmiştir (Elmas, 2007).



Şekil 3.24. Max-Dot çıkarım tekniği

### Min-Max çıkarım

Her bir giriş değeri için ait olduğu üyelik fonksiyonundaki üyelik derecesine bağlı olarak ilgili bulanık kümenin üyelik değerleri üstündeki kısmı kesilir. Bu metotta Mamdani'nin minimum operatörü kullanılır. Şekil 3.25'de Min-Max çıkarım tekniğine ait üyelik fonksiyon grafikleri verilmiştir (Elmas, 2007).

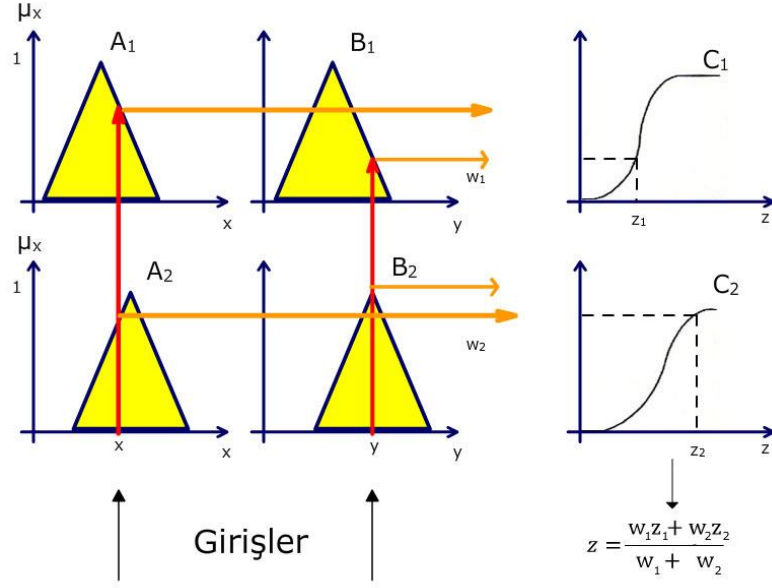


Şekil 3.25. Min-Max çıkarım tekniği

### Tsukamoto çıkarım

Bu çıkarım tekniğinde çıkış üyelik fonksiyonu tek yönlü artan bir fonksiyon olarak seçilir. Çıkış değeri ise her bir kuralın keskin çıkış değerinin ağırlık ortalaması alınarak bulunur. Şekil 3.26'da Tsukamoto çıkarım tekniğine ait üyelik fonksiyon grafikleri verilmiştir (Elmas, 2007).

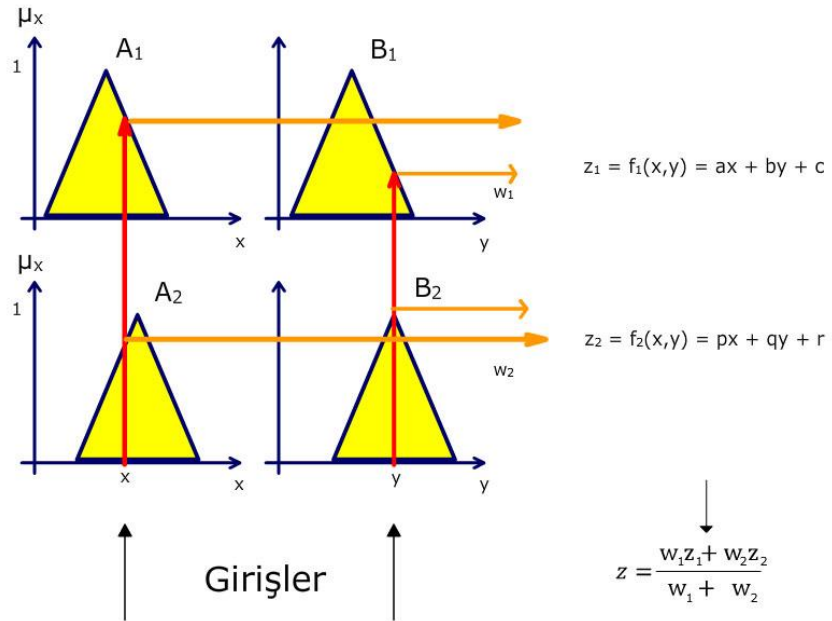




Şekil 3.26. Tsukamoto çıkarım tekniği

Takagi-Sugeno çıkarım

Her bir kuralın çıkışı giriş değerlerinin doğrusal birleşimiyle bulunur. Keskin çıkış değeri ise ağırlık ortalaması alınarak bulunur. Takagi-Sugeno çıkarım Şekil 3.27’de gösterilmiştir (Elmas, 2007).



Şekil 3.27. Takagi-Sugeno çıkarım tekniği

### Durulařtırma birimi

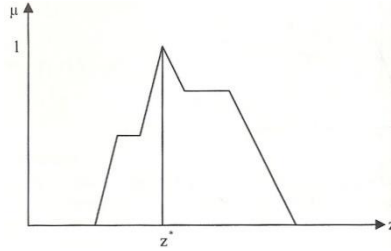
Bulanık ıkarım biriminin ıkıřı bulanık bir kmedir. Bu sonucun tekrar sisteme uygulanması iin sayısal bir deęere dnřtrlmesi gerekir. Durulama adımımda karar verme adımımdan gelen bulanık bilgiden bulanık olmayan ve denetlenen sisteme uygulanacak gerek deęer elde edilir. Durulařtırma adımımda sık kullanılan drt teknik vardır. Bunlar; maksimum yelik, aęırlık merkezi, aęırlık ortalaması ve maksimumların ortalaması teknięidir.

### Maksimum yelik teknięi

Bulanıklařtırma adımımda elde edilen bulanık kmeler ierisindeki yelik derecesi en byk olan deęer alınır. Bu durum Denklem 3.48'de gsterilmiřtir (Elmas, 2007).

$$\mu_C(z^*) \geq \mu_C(z) \quad z \in Z \quad (3.48)$$

$z^*$  ıkıř deęerinin elde edilmesi řekil 3.28'de gsterildięi gibidir.



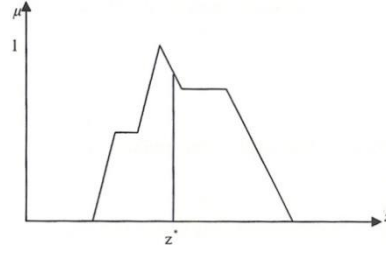
řekil 3.28. Maksimum yelik teknięi

### Aęırlık merkezi teknięi

Bulanıklařtırma adımımda elde edilen bulanık kmelerin aęırlık merkezi hesaplanarak bulunur. Denklem 3.49'daki gibi ifade edilir (Elmas, 2007).

$$z^* = \frac{\int \mu_C(z)z \, dz}{\int \mu_C(z) \, dz} \quad (3.49)$$

$z^*$  çıkış değerinin elde edilmesi Şekil 3.29'da gösterildiği gibidir.



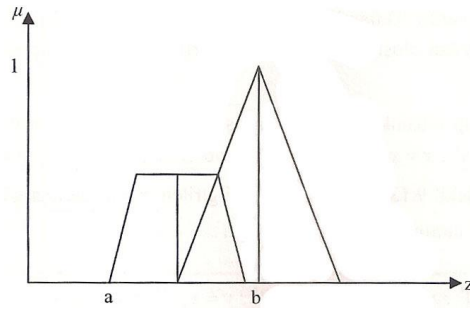
Şekil 3.29. Ağırlık merkezi tekniği

### Ağırlık ortalaması tekniği

Bu teknik ile girişlerden elde edilen bütün bulanık değerler ile üyelik değeri kullanılarak durulama yapılmaktadır. Denklem 3.50'deki gibi ifade edilir (Elmas, 2007).

$$z^* = \frac{\sum \mu_c(\bar{z})\bar{z}}{\sum \mu_c(\bar{z})} \quad (3.50)$$

$z^*$  çıkış değerinin elde edilmesi Şekil 3.30'da gösterildiği gibidir.



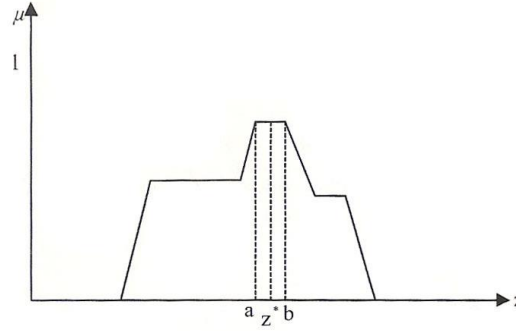
Şekil 3.30. Ağırlık ortalaması tekniği

### Maksimumların ortalaması tekniği

Maksimum üyelik fonksiyon yöntemi ile ilişkilidir. Bu teknik maksimum üyelik derecesi tek bir nokta olmayan bulanık kümeler için kullanılır. Denklem 3.51'deki gibi ifade edilir (Elmas, 2007).

$$z^* = \frac{a + b}{2} \quad (3.51)$$

$z^*$  çıkış değerinin elde edilmesi Şekil 3.31'de gösterildiği gibidir.

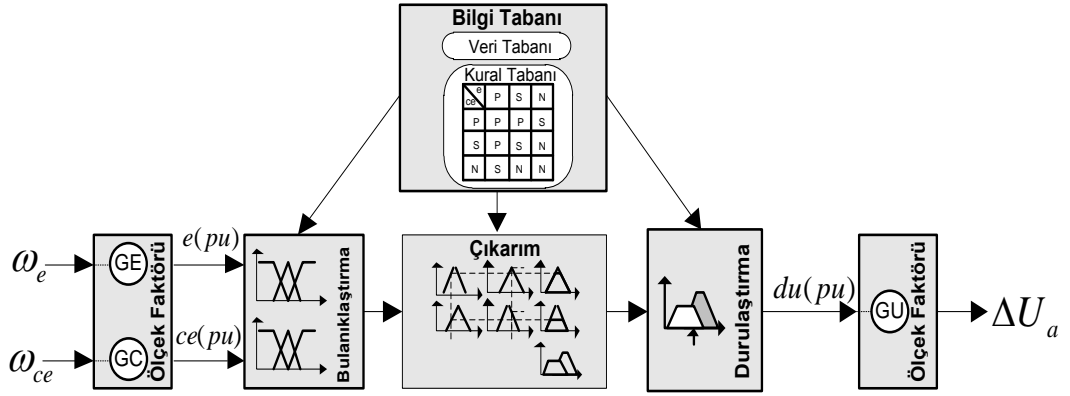


Şekil 3.31. Maksimumların ortalaması tekniği

### **3.3.2. Bulanık mantık denetleyici**

Denetleyici tasarımının matematiksel modele ihtiyaç duyulmadan gerçekleştirilmesi ve lineer olmayan sistemlerde oldukça iyi sonuçlar vermesi nedeniyle, bulanık denetleyiciler, son yıllarda birçok uygulamada sıklıkla karşımıza çıkmaktadır. Temelde, bir bulanık mantık sistemin en belirgin özelliği, herhangi bir lineer olmayan fonksiyonun, bu yöntemle yeterince doğruluk derecesine sahip bir fonksiyona grafiksel olarak yaklaşımıdır (Aydoğdu, 2006).

BM denetleyici temelde, uzman bir sistem operatörünün bilgi, deneyim, sezgi ve denetim stratejisini denetleyici tasarımında bilgi tabanı olarak oluşturmaktır. Denetim işlemi, karmaşık ve klasik denetim yöntemleri ile değil, bilgi ve tecrübeye dayanan sözel kurallarla gerçekleştirilir.



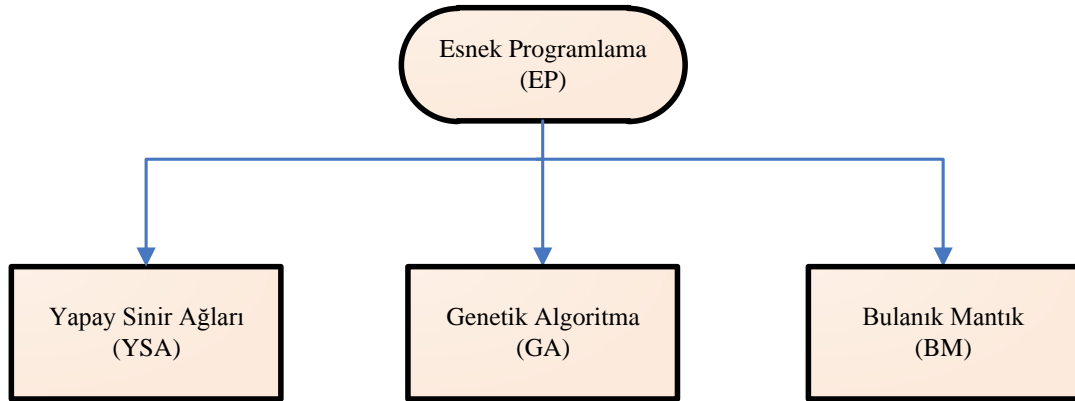
Şekil 3.32. BM denetleyicisinin yapısı (Bingöl ve Paçacı, 2010)

Şekil 3.32'de görülen BM denetleyicisinin yapısı aşağıda belirtilen özelliklere sahiptir.

- Giriş ve çıkış ölçeklendirme katsayıları ( $GE$ ,  $GC$ ,  $GU$ ) sabit sayılardır.
- Kural tabanı denetim süresi boyunca değişmez ve kurallar arası etkileşim yoktur.
- Üyelik fonksiyonları sabittir ve denetim süresi boyunca değişmezler.
- Kuralların sayısı giriş değişkenleri için tanımlanan üyelik fonksiyonlarının sayısına bağlıdır.
- Durulama ve kuralların sonuçlarını hesaplama yöntemi sabittir.

BM denetleyiciler, bulanıklaştırma birimi, bilgi tabanı (kural ve veri tabanı), çıkarım birimi ve durulaştırma biriminden oluşmaktadır.

### 3.4. Geleneksel Tasarımda Karşılaşılan Temel Problemler



Şekil 3.33. Esnek programlamayı oluşturan programlama teknikleri (Topuz, 2002)

Bulanık sistemde en önemli kavram evrensel yaklaşım teoremidir. Bu teorem, kontrol mühendisliğinde bulanık sistemlerin pratik başarısı için bir tanımlama sağlar (Şekil 3.33). Bununla beraber teorem, kayda değer bir dezavantaja sahiptir. Mevcut teorem, lineer olmayan sistemler için bir bulanık mantık çözüm olduğunu gösterir, fakat çözümün nasıl bulunacağını göstermez. Pratikte, bulanık denetleyicinin tasarım aşamaları, deneme ve yanılma yaklaşımı içermektedir. Deneme-yanılma yaklaşımı ile oluşabilen tasarım problemleri aşağıdaki gibi sıralanır (Aydoğdu, 2006).

- Kontrol edilen sistem karmaşık bir yapı olursa, uzman deneyimi ile kural tablolarının oluşturulması zordur.
- Çok girişli, çok çıkışlı bulanık denetleyicilerin tasarım ve ayarı yapılamayacak kadar çok zor ve sıkıcıdır.
- Güvenilir uzman bilgi tabanı mümkün olmayabilir.
- Uzman bilgisi bile ince ayarlar için yetersiz olabilir.
- Parametre veya bozucu etkilerde oluşan bazı önemli değişiklikler, uzman deneyimi dışında olabilir.
- Uzman kişinin bilgisi, deneyimi ve sezgisi ile optimal bulanık denetleyici tasarımı garanti edilemez.

- Optimal olmayan bir bulanık denetleyici ile diğer denetleyicilerin performans karşılaştırması geçersizdir.

Otomatik olarak bir sistemin tasarımını gerçekleştirmek için, denetleyicide bulunan optimizasyon problemi, uygulanan optimizasyon tekniğiyle çözümlenmelidir. Bir bulanık denetleyici, genellikle, matematiksel modeli bilinmeyen parametre elemanlarına sahip lineer olmayan bir yapıdadır. Modeli zor elde edilen bulanık kurallar ve bulanık kümelerden oluşur. Diğer taraftan, çok sayıdaki tasarım problemine sahip olması, optimizasyon işleminin büyük bir yüzeyde yapılmasını gerektirir. Bu da optimizasyon işlemini oldukça karmaşık yapar ve kesin olmayan sonuçlara götürebilir. Karmaşıklığın derecesi arama yüzeyine bağlı olarak artar. Bu yüzden, eğer arama yüzeyinde en iyi sonucu veren tepe dışında başka bir tepede işlem başlatılırsa hesaplamaya dayalı optimizasyon tekniği hatalı sonuç verir. Üstelik arama yüzeyi iyi bilinmediğinden, var olan süreksizlik noktaları, eldeki bu problemi gradyan metodu ile çözmeyi zorlaştırır. Üstelik, bulanık denetleyicilerin optimizasyon problemi ardışık değildir. Yani, elimizdeki problemin parametrelerinin aynı anda paralel olarak optimize edilmesini gerektirir (Aydoğdu, 2006).

Yapay sinir ağları, bulanık mantık ve evrim algoritmalarının birlikte, birbirlerini destekler şekilde kullanılması esnek programlamayı oluşturmaktadır. Bu yöntemlerde, genetik algoritma doğal genetiği, yapay sinir ağları insanın sinir ve öğrenme yapısını, bulanık mantık ise insanın düşünme biçimini modelleyerek çözüm üretilmesi sağlanması amaçlanmıştır. Şekil 3.35’de esnek programlamayı oluşturan yöntemler görülmektedir (Topuz, 2002).

Bu çalışmada; bulanık denetleyicinin optimizasyonu otomatik olarak yapılacaktır. Önerilen çözüm, önceden belirtilen bir performans indeksi temelinde en optimal çözümü sunmaktır. Sunulan sistem herhangi bir önsezi tabanına dayanmayıp, bulgusal bilgilerle çalışan algoritmalarından oluşmaktadır.

### **3.5. Genetik Algoritmalar ile Optimizasyon**

Genetik algoritmalar (GA), doğadaki biyolojik evrim temeline dayanan optimizasyon tekniğidir. GA ilk defa 1975 yılında Michigan Üniversitesi'nde John Holland ve arkadaşları tarafından tanıtılmıştır. İlk uygulaması, boru hatları ile ilgili bir problemin çözümü amacıyla David Goldberg tarafından yapılmıştır (Goldberg, 1989). İlk tanıtımından beri kullanılan bu yaklaşım, doğal ve yapay sistem bilimlerinin gelişimine önemli katkılar sağlamıştır (Ashrafzadeh, 1996).

GA ile optimizasyon yapabilmek için, optimize edilecek olan parametreler tanımlanmalı ve bu parametreler optimizasyon işlemi için uygun biçimde kodlanmalıdır. GA, kodlama biçimine göre ikili kodlu GA'lar (Binary coded GA) ve gerçek kodlu GA'lar (Real coded GA) olmak üzere ikiye ayrılır. GA'larda kodlama, algorithmada yürütülen matematiksel işlemler açısından oldukça önemlidir (Aydoğdu, 2006).

Genetik algoritmalarla optimizasyonda, optimize edilecek olan parametrelerin ilk olası çözümleri için ya rasgele bir değer kümesi üretilir ya da tasarımcı tarafından verilen bir başlangıç çözüm kümesi tanımlanır. Genetik algoritma çözümlerini değerlendirmek için, bir uygunluk fonksiyonu tanımlanır ve her bir bireyin uygunluk değeri hesaplanır. En uygun çözümü bulmak için, en iyilerin hayatta kalması mantığıyla temel genetik algoritma adımları yürütülerek yeni nesil çözüm kümeleri elde edilir. Genetik algoritmalarda her bir birey, arama uzayında özel bir noktaya karşılık gelir. Gradyan tabanlı yaklaşımda olmayan bu özellik, yerel optimumların bulunması noktasında genetik algoritmaları etkili kılmaktadır (Aydoğdu, 2006).

#### **3.5.1. Genetik algoritmalar**

GA, biyolojik süreci modelleyerek, fonksiyonları veya sistemleri optimize eden evrim algoritmalarıdır. GA'lar, optimizasyon ve arama problemlerini çözmek için kullanılan adaptif bir metottur. GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA'ların



her bir ferdi, kromozomlar (bireyler) şeklinde temsil edilen popülasyonlardan oluşur. Popülasyonun uygunluğu, belirli kurallar dâhilinde maksimize veya minimize edilir. Her yeni nesil, rasgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir (Angeline, 1995).

GA, doğal seçme ve doğal genetik kurallara dayanan bir arama türüdür. Doğal seçme, doğa koşullarına en fazla uyum sağlamış olan canlının neslini devam ettirmesi, uyum sağlayamamış olan türlerin ise elenmesidir. Canlılar nesilden nesile genlerini aktarırken, bu genler de doğal genetik kurallara göre başka genlerle çaprazlanır, değişime uğrar ve yeni genleri oluştururlar. Genetik algoritmalar da tabiattaki bu iki oluşumu birleştirerek en iyiyi (optimal noktayı) arar. Bir önceki neslin en uyumlu fertleri kullanılarak yeni neslin üyeleri oluşturulur. Bu yöntem sayesinde, klasik yöntemlerle çözülmesi çok zor kimi zaman da imkânsız olan problemler çözülebilmektedir (Karaođlan, 2007).

Olası çözümlerin yeni bir popülasyonu, şimdiki nesillerden en iyi bireylerin seçilmesi ve yeni birey kümesi oluşturmak için çiftleştirilmeleri yoluyla oluşturulur. Aslında, nesil karakteristiklerinin iyileştirilmesi, daha iyi bireylerin değiştirilmesi ve karıştırılması yoluyla yapılır. Optimizasyon açısından bakıldığında, başlangıç popülasyonunun rastgele üretilmesi yoluyla arama uzayında büyük bir alan araştırılır ve daha sonra uygun bireylerin çiftleştirilmesi yoluyla arama uzayının en uygun bölgesi tespit edilir ve sonuçta genetik algoritma muhtemel global optimum noktasını bulabilir. Eğer genetik algoritma iyi tasarlanmış ise popülasyon, optimal çözüm noktasına yaklaşacaktır (Çunkaş, 2004).

İhtimal içeren doğası nedeniyle genetik algoritmalar, global optimal çözümü bulmayı daima garanti edemez fakat, genellikle bu sağlanır. GA'ların avantajları aşağıdaki gibi sıralanabilir.

- Sürekli ve ayırık parametreleri optimize etmesi,
- Türevsel hesaplamalar gerektirmemesi,
- Amaç fonksiyonunu geniş bir spektrumda araştırması,

- Çok sayıda parametrelerle çalışma imkânı olması,
- Paralel PC'ler kullanılarak çalıştırılabilmesi,
- Karmaşık amaç fonksiyonu parametrelerini, lokal minimum veya maksimumlara takılmadan optimize edebilmesi,
- Çoklu amaç fonksiyonları ile çalışabilmesi.

Her problemin çözümü için GA kullanmak iyi bir yol değildir. Birkaç parametrelili analitik fonksiyonun çözümünde klasik metotlar daha hızlıdır. Bu nedenle bu gibi durumlarda, nümerik metotlar tercih edilmelidir. Paralel bilgisayarlar kullanılırsa GA daha hızlı sonuç verebilir. Bu çalışmada, optimizasyon için gerçek kodlu GA kullanılmış ve kısaca incelenmiştir (Aydoğdu, 2006).

### **3.5.2. Gerçek kodlu genetik algoritmalar**

Problemlerin çözümünde rakamların hassasiyeti önemli bir yer tutmaktadır. İkili kodlu GA'da parametrelerin "1" ve "0" ile ifade edilmesi, kromozom boyutlarını oldukça artırdığından, hassasiyet sınırlı olmaktadır. Bunun yerine, gerçek rakamlarla kodlama yapabilen gerçek kodlu GA'yı kullanmak avantajlıdır. Gerçek kodlu GA, hem daha hassas hem de PC belleğinde daha az yer kaplamaktadır. Literatürde, gerçek kodlu GA'ların ikili kodlu GA'ya göre daha hızlı çalıştığı ve global optimumu daha kısa sürede bulduğu belirtilmektedir (Cunkaş ve Akkaya, 2002).

Gerçek kodlu GA, bazı literatürlerde "Sürekli Parametrelili GA" olarak da bilinmektedir. Bu çalışmada "Gerçek Kodlu GA" tanımlaması kullanılmıştır. Gerçek kodlu GA, ikili kodlu GA'ya çok benzemektedir. Aralarındaki en önemli farklılık parametrelerin "1" ve "0" 'lar yerine gerçek rakamlarla ifade edilmesidir.

### **3.5.3. Parametrelerin kodlanması ve amaç fonksiyonu**

Genetik algoritmayı diğer arama metotlarından ayıran en önemli özellik, genetik algoritmanın parametreler yerine parametrelerin kodlanmış şekli ile işlem

yapmasıdır. Bu nedenle, genetik algoritma uygulamasında ilk adım, çözülecek problemin çözüm uzayını en uygun biçimde temsil edecek kodlama yönteminin seçilmesidir (Şafak, 2005).

GA için bir problemin uygun şekilde tanımlanması, bir parametre kümesi şeklinde yapılmaktadır. GA'da optimize edilecek olan her bir parametre, bir gen ( $gen_n$ ) ile ifade edilmektedir. Gerçek kodlu GA'da parametrenin yapısına göre her bir gen, on tabanlı bir sayı dizisiyle temsil edilir. Parametreleri temsil eden genler topluluğuna kromozom denir.  $N_{par}$  adet parametreye veya gene sahip problemde bir kromozom,  $1 \times N_{par}$  boyutlu matris biçiminde aşağıdaki gibi yazılır (Aydoğdu, 2006).

$$\text{Kromozom} = [\text{gen}_1, \text{gen}_2, \text{gen}_3, \dots, \text{gen}_{N_{par}}]$$

Gerçek kodlu GA'da parametrelerin hepsi "ondalık" rakamlarla ifade edilir ve her bir kromozomun bir uygunluk değeri (C) vardır. Tanımlanan  $f(.)$  amaç fonksiyonuna göre uygunluk değerleri,

$$C = f(\text{kromozom}) = f(\text{gen}_1, \text{gen}_2, \text{gen}_3, \dots, \text{gen}_{N_{par}})$$

şeklinde ifade edilir. Örneğin iki parametrelili bir optimizasyon için amaç fonksiyonu Denklem 3.52;

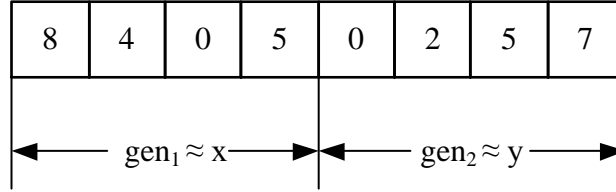
$$f(x, y) = x \sin(4x) + 1,1y \sin(2y) \quad (3.52)$$

ve parametre sınırları  $0 \leq x < 10$ ,  $0 \leq y < 10$  olarak tanımlansın. Verilen bu problemde optimize edilecek parametreler x ve y ifadeleridir. Yukarıda gösterildiği gibi kromozom,

$$\text{Kromozom} = [\text{gen}_1, \text{gen}_2] \approx [x, y]$$

şeklinde ifade edilebilir. Bu problem, çok sayıda lokal minimuma sahip karmaşık bir optimizasyon problemidir ve klasik metotlarla bu fonksiyonun global minimumunu bulmak zordur. Bu optimizasyon probleminde her bir gen, on

tabanlı 4 rakam ile temsil edilirse, kromozom yapısı Şekil 3.34’de görüldüğü gibi 8 adet on tabanlı sayıdan oluşacaktır. Şekilde verilen kromozom yapısında, parametrelerin gerçek değerleri, parametre sınırları içine normalize edilmelidir (Aydoğdu, 2006).



Şekil 3.34. Bir GA’da tipik kromozom dizgisi (Aydoğdu, 2006)

### 3.5.4. Başlangıç popülasyonu

Genetik algoritma, en uygun çözümü bulmak için birden çok noktadan aramaya başlamaktadır. Bunun için başlangıç değerlerini oluşturmak önemlidir. Oluşturulmuş olan ilk popülasyon genellikle rasgele bir şekildedir. Ancak önceden elde edilmiş bilgiye dayalı veya sezgisel olarak da üretilebilmektedir (Kallel ve Schoenaur, 1997).

GA’da olası çözüm kümesi, tek bir kromozom ile ifade edilmez. Algoritmanın yürütüleceği ortam kapasitesi ve parametre karakteristikleri dikkate alınarak bir kromozom kümesi oluşturulur ve popülasyon olarak tanımlanır. GA’nın çalışmaya başlayabilmesi için,  $N_{ipop}$  kromozomdan oluşan başlangıç popülasyonuna ihtiyaç vardır ve başlangıç popülasyonu,  $N_{par} \times N_{ipop}$ ’luk bir matris ile temsil edilir. GA’da başlangıç popülasyonunun oluşturulması iki şekilde yapılır. Birincisinde; eğer programcı, geniş arama yüzeyinde lokal bir bölgede sonuç arayacaksa, o bölgeye yakın noktalardan oluşan bir başlangıç popülasyonu tanımlayabilir. Bu, algoritmanın daha kısa sürede sonuca ulaşmasını sağlar ve birçok çalışmada, ince ayar (fine-tuning) için kullanılmaktadır. İkincisinde; başlangıç popülasyonunda kromozomları oluşturan genler, Denklem 3.53 ile verilen rasgele sayı üretici ile elde edilir.

$$iPOP = (P_H - P_L) \times Rasgele\{N_{ipop}, N_{par}\} + P_L \quad (3.53)$$

Burada;  $P_H$ , parametrenin üst sınır değeri,  $P_L$ , parametrenin alt sınır değeri ve  $Rasgele\{N_{ipop}, N_{par}\}$  ise  $N_{ipop} \times N_{par}$  olacak şekilde matris formunda 0-1 arasında üretilen rasgele sayıdır. Bu tanımlamalar her parametre (gen) için kullanılabilir (Aydoğdu, 2006).

### 3.5.5. Uygunluk değerlerinin hesaplanması

Parametrelerin bu şekilde değerleri belirlendikten sonra, amaç fonksiyonundan yararlanılarak uygunluk değerleri (C) hesaplanır. Denklem 3.52'de görüldüğü gibi, örnekte verilen amaç fonksiyonunun x ve y den oluşan iki tane parametresi vardır. Bu parametrelerin sınır değerleri,  $P_L = 0$  ve  $P_H = 10$ 'dur. Amaç fonksiyonu karmaşık bir yapıya sahip olduğundan dolayı, popülasyon sayısının yüksek tutulması iyi sonuç vermektedir. Başlangıç popülasyon sayısı  $N_{ipop} = 24$  seçilirse, popülasyon matrisi  $24 \times 2$  olmaktadır. Başlangıç popülasyonunun büyüklüğü, araştırma uzayının daha geniş seçilmesini sağlamaktadır. GA döngüsünün sona erdirilebilmesi için, algoritma başlangıcında, bir yakınsama testi veya döngü sayısı tanımlanmalıdır (Aydoğdu, 2006).

### 3.5.6. Değerlendirme ve doğal seçim

Kodlama yöntemi belirlenip başlangıç popülasyonu oluşturulduktan sonra, yeni popülasyona geçebilmek için seçimin nasıl yapılacağı belirlenmelidir. Seçim, bireylerin üreme için seçildiği bir işlemdir. Seçim operatörünün yüksek uyuma sahip bireylerin sonraki kuşağa kopyalanma olasılığını artırması istenir. Böylece seçim, arama uzayının başarılı olunacağı umulan bölgelerinde yoğunlaşır. Sonraki kuşak için üremeye katılacak bireylerin popülasyondan nasıl seçileceği ve her bir bireyin kaçar tane yavruya sahip olacağı belirlenmelidir. Seçim işleminden sonra işlem havuzu hazırlanmış olur ve bu bireylere çaprazlama ve mutasyon uygulanır. Bu işlemlerden sonra yeni nesil elde edilir (Karaoğlan, 2007).

GA'lerde gelecek nesilde, hangi kromozomun yer alacağını belirlemek gerekir. Gerçek kodlu GA'da, toplam  $N_{ipop}$  kadar kromozom olduğu düşünülürse, gelecek iterasyonda kullanılmak üzere  $N_{pop}$  kadar kromozom tutulur ve geri kalanı dikkate alınmaz.  $N_{pop}$  kromozomları, kendi aralarında  $N_{iyi}$  ve  $N_{kötü}$  olarak büyükten küçüğe doğru sıralamaya tabi tutulur.  $N_{iyi}$ , eşleştirme havuzuna atılırken,  $N_{kötü}$  hariçte tutulur (Haupt, 1998).

Doğal seçim, algoritmanın her bir iterasyonunda gerçekleşir. Burada ne kadar kromozomun elde tutulacağına karar vermek biraz keyfidir. Kromozomlarda gelecek nesil içerisinde, yer alma mücadelesi başlamaktadır. Popülasyon içerisinde bütün kromozomların seçilmesinin gelecek nesile çok fazla katkısı olmamaktadır. GA'da değişik seçim metotları vardır.

*a) Ağırlıklı seçim metodu:* İlk olarak kromozomların uygunluk değerleri amaç fonksiyonunda hesaplanır. Hesaplanan uygunluk değerleri, en küçükten en büyüğe doğru sıralanır.  $N_{pop}$  adet kromozom tutulur, geriye kalanlar atılır. Burada  $N_{pop}$  değeri  $N_{ipop}$ 'a kadar olabilir. Genelde popülasyonun %50'sinin seçilmesi ( $N_{pop}=N_{ipop}/2$ ) uygun seçenektir. Seçilen  $N_{pop}$ 'un yarısı  $N_{iyi}$ , yarısı da  $N_{kötü}$  olarak ayrılır.  $N_{iyi}$  olanlar eşleştirme havuzuna konulurken  $N_{kötü}$  olanlar eşleştirme havuzundan atılır.

*b) Eşik değer seçim metodu:* GA'da doğal seçim için kullanılan bir başka yöntem, bir eşik uygunluk değeri tanımlamak ve bu tanımlanan değerle, her bir bireyin uygunluk değerini karşılaştırmaktır. Eşik değerden küçük olan kromozomlar yaşamaya devam eder, eşik değeri aşan kromozomlar ölür. Bu tekniğin avantajı, popülasyonun sıralanmasına gerek olmamasıdır.

*c) Rulet tekerleği seçim metodu:* Diğer bir yöntem olan Rulet tekerleği seçim metodunun uygulanması için, ilk önce kromozomların toplam uygunluk değeri hesaplanır. Kromozomlar, toplam uygunluk değerine bölünerek her bir kromozom için 0-1 arasında değişen seçim ihtimalleri bulunur. Daha sonra, kümülatif ihtimaller hesaplanır. Popülasyon sayısı kadar 0-1 arasında "rasgele" sayılar üretilir. Üretilen rasgele sayı, birinci kromozomun kümülatif seçim ihtimalinden küçük ise, birinci kromozom seçilir. Eğer değilse, ikinci

kromozomun veya diğerlerinin kümülatif ihtimalleriyle karşılaştırılarak hangisinden küçükse o kromozom seçilir. Böylece Rulet seçim metodu gerçekleştirilmiş olur (Aydoğdu, 2006).

### 3.5.7. Eşleştirme

Doğal seçim yoluyla elde edilen  $N_{iyi}$  kromozomların bulunduğu eşleştirme havuzundan, yeni nesil üretmek için iki tane kromozom seçilir. Eşleştirme, seçilen kromozomlar arasında gerçekleştirilir. Eşleştirmenin değişik yöntemleri vardır.

a) *Yukarıdan aşağıya doğru eşleştirme:* Listenin en üstünden başlanır ve eşleştirme için seçilen  $N_{iyi}$  kromozomların sayısına ulaşana kadar devam edilir. Böylece kromozom<sub>2i-1</sub> ile kromozom<sub>2i</sub>  $i=1,2,3$  için eşleştirilir. Bu örnekte kromozom<sub>1</sub> ile kromozom<sub>2</sub>, kromozom<sub>3</sub> ile kromozom<sub>4</sub> ve kromozom<sub>5</sub> ile kromozom<sub>6</sub> eşleştirilmiştir. Bu metot, doğal bir yöntem olmamakla beraber yeni başlayanlar için kolay bir yoldur.

b) *Rasgele eşleştirme:* Kromozomların eşleştirilmesinde rasgele sayı üretici kullanılır. Kromozomlar 1'den başlayarak  $N_{iyi}$ 'ye kadar sıralanır. Birinci eşleştirmeyi bulmak için iki adet rasgele sayı üretilir (Denklem 3.54).

$$Kromozom = Roundup\{N_{iyi} \times rasgele\ sayı\} \quad (3.54)$$

Burada roundup fonksiyonu, sayıları en yüksek sayıya yuvarlar. Örneğin; rasgele olarak üretilen 6 sayı 0.1535, 0.6781, 0.0872, 0.1936, 0.7021 ve 0.3933 ise bu sayılar, 6 ile çarpılıp bir üst tamsayıya yuvarlanarak; 1, 5, 1, 2, 5, 3 değerleri elde edilir. Buna göre kromozom<sub>1</sub>-kromozom<sub>5</sub>, kromozom<sub>1</sub>-kromozom<sub>2</sub> ve kromozom<sub>3</sub>-kromozom<sub>5</sub> eşleştirilecektir.

c) *Ağırlıklı rasgele eşleştirme:* Bu yöntem, eşleştirme havuzundaki kromozomların, amaç fonksiyonunda hesaplanan uygunluk değerlerine göre seçilme ihtimallerini dikkate alır. En düşük uygunluk değerine sahip kromozom,

eşleştirme ihtimali en yüksek kromozom olurken en yüksek uygunluk değerine sahip kromozom eşleştirme ihtimali en düşük olarak düşünülür. Üretilen rasgele sayı, hangi kromozomun seçileceğini belirler. Bu tip uygulamaya “Rulet tekerleği ağırlık yöntemi” de denir. Bu yöntem kendi arasında ikiye ayrılır.

*i. Sıralama (Rank-Based) ağırlık:* Kromozomlar büyükten küçüğe doğru sıralanır ve Denklem 3.55 kullanılarak sıralamadaki yerlerine göre kromozomların olasılıkları  $P_n$  hesaplanır.

$$P_n = \frac{N_{iyi} - n + 1}{\sum_{n=1}^{N_{iyi}} n} = \frac{6 - n + 1}{1 + 2 + 3 + 4 + 5 + 6} = \frac{7 - n}{21} \quad (3.55)$$

Burada  $n$ , sıra sayısını göstermektedir. Bu yöntemle uygunluk değerlerinden bağımsız olarak, sadece sıralamayı dikkate alan  $P_n$  normalize değerleri elde edilmektedir.

Bu yöntemde, kromozomların seçilmesinde kümülatif ihtimaller kullanılır. Rasgele olarak 0-1 arasında bir sayı üretilir. Bu sayı, kümülatif ihtimallerin en başından başlanarak sırayla karşılaştırılır. Kümülatif ihtimal değeri, üretilen rasgele sayıdan büyükse, eşleştirme havuzu için kromozom seçilir. Bu yaklaşım tarzı doğaya daha uygundur.

*ii. Değer ağırlık:* Kromozomların uygunluk değerleri amaç fonksiyonundan hesaplanır. Eşleştirme havuzunda  $N_{iyi+1}$ 'inci sırada bulunan kromozomun amaç fonksiyonunda hesaplanan uygunluk değeri, bütün eşleştirme havuzundaki kromozomlardan çıkartılır. Eşleştirme havuzundaki her bir kromozomun normalize uygunluk değerleri ve seçilme ihtimalleri Denklem 3.56 ve Denklem 3.57'deki gibi hesaplanır.

$$C_n = f(kromozom_n) - f(kromozom_{N_{iyi+1}}) \quad (3.56)$$

$$P_n = \left| \frac{C_n}{\sum_{p=1}^{N_{iyi}} C_p} \right| \quad (3.57)$$



Daha önce üretilen rasgele sayılar kullanılırsa kromozom<sub>1</sub>-kromozom<sub>3</sub>, kromozom<sub>1</sub>-kromozom<sub>1</sub> ve kromozom<sub>3</sub>-kromozom<sub>1</sub> eşleşmeleri görülür. Kromozomların eşleştirilmesinde, uygunluk değerlerine göre kromozomların seçilme ihtimalleri ağırlıklı rasgele eşleştirme metodu kullanılarak elde edilmiştir.

### 3.5.8. Çaprazlama

Çaprazlama işlemi, ikili kodlu genetik algoritmalarda tek nokta çaprazlama veya iki nokta çaprazlama gibi yöntemler ile yapılmaktadır. Gerçek kodlu genetik algoritmalarda ise, çaprazlama için karıştırma yöntemleri daha iyi sonuç vermektedir. Karıştırma yönteminde; iki parametrenin değerleri karıştırılır ve Denklem 3.58 kullanılarak yeni nesiller elde edilir (Radcliff, 1991).

$$P_{yeni} = \beta P_{an} + (1 - \beta)P_{bn} \quad (3.58)$$

Burada,  $\beta$ ; 0 ve 1 arasında üretilen rasgele sayı,  $P_{an}$ ; Anne kromozomun n. Parametresi,  $P_{bn}$ ; Baba kromozomun n. parametresidir.

İkinci nesil,  $\beta$  yerine  $1-\beta$  konularak ve birinci neslin tümleyeni alınarak bulunur.  $\beta=1$  ise,  $P_{an}$  baskın gelir ve  $P_{bn}$  ölür.  $\beta=0$  ise  $P_{bn}$  baskın gelir ve  $P_{an}$  ölür.  $\beta=0.5$  olduğu zaman (Davis, 1991), sonuç iki parametrenin ortalaması olarak ortaya çıkar. Bu metodun iyi işlediği Michalewicz (1994) tarafından gösterilmiştir. Karıştırma işlemi için hangi parametrenin seçileceği belirlenmelidir. Bazen çaprazlama noktalarının solundan veya sağından başlanarak bütün parametreler için lineer bir kombinasyon işlemi yapılır. Çaprazlama noktası rastgele seçilebilir. Parametreler, farklı  $\beta$  değerleri kullanılarak karıştırılabilir. Parametrelerin sınır aralıkları kullanılırsa, iki kromozomun bilgileri etkili olarak karıştırılır.

Gerçek kodlu GA karıştırma metotlarının en basiti, "lineer çaprazlama metodu" dur (Wright, 1991). Bu metotla iki kromozomdan üç tane yeni nesil üretilir (Denklem 3.59, Denklem 3.60 ve Denklem 3.61).

$$P_{yeni1} = 0.5P_{an} + 0.5P_{bn} \quad (3.59)$$

$$P_{yeni2} = 0.5P_{an} + 0.5P_{bn} \quad (3.60)$$

$$P_{yeni3} = -0.5P_{an} + 1.5P_{bn} \quad (3.61)$$

Parametre sınırları dışında kalan herhangi bir nesil dikkate alınmaz. Sonraki popülasyonda kromozomun hayatını diğer iki nesil devam ettirir. Dikkat edilecek olursa  $\beta$  pek fazla değişmemektedir.

Diğer bir çaprazlama metodu olan “Heuristic çaprazlama” da ise  $\beta$ , 0-1 arasında değer alır ve yeni nesil Denklem 3.62.’ye göre elde edilir (Michalewicz, 1994).

$$P_{yeni} = \beta(P_{an} - P_{bn}) + P_{an} \quad (3.62)$$

Denklem 3.62’ye göre, izin verilen bazı değerlerin dışında üretilen yeni nesiller ihmal edilir. Algoritma, yeni  $\beta$  değeri üreterek işleme devam eder.

“Karma çaprazlama metodu” (blend crossover, BLX- $\alpha$ ) bazı parametrelerin  $\alpha$  değerlerini belirleyerek işleme başlar (Eshelman ve Shafter, 1993).  $\alpha$  değeri, parametrelerin dış sınırlarını belirler ve bu sınırlara göre yeni nesiller üretilir.

“Kuadratik çaprazlama” gibi metotlar, amaç fonksiyonu için sayısal uygunluk gerçekleştirmeye çalışırlar. Sayısal uygunluğun gerçekleştirilebilmesi için üç adet kromozoma ihtiyaç duyulmaktadır (Adewuya, 1996).

Heuristik çaprazlama metoduyla dış değer bulma (extrapolasyon) metodunun kombinasyonundan oluşan metot, en çok kullanılan çaprazlama metodudur (Haupt, 1998). Bu metotla, ikili kodlu GA’daki çaprazlamaya daha çok yaklaşılmaktadır. Burada ilk olarak çaprazlama noktası rasgele seçilir (Denklem 3.63).

$$\alpha = roundup\{rasgele \times N_{par}\} \quad (3.63)$$

$$Kromozom_1 = [P_{a1}, P_{a2}, \dots, P_{a\alpha}, \dots, P_{aNpar}]$$

$$\text{Kromozom}_2 = [P_{b1}, P_{b2}, \dots, P_{b\alpha}, \dots, P_{bNpar}]$$

Burada a ve b indisleri sırasıyla anne ve baba kromozomlarını tanımlamak amacıyla kullanılmaktadır. Bu metotta, Denklem 3.64 ve Denklem 3.65 kullanılarak yeni nesiller elde edilir.

$$P_{yeni1} = P_{a\alpha} - \beta [P_{a\alpha} - P_{b\alpha}] \quad (3.64)$$

$$P_{yeni2} = P_{b\alpha} - \beta [P_{a\alpha} - P_{b\alpha}] \quad (3.65)$$

Burada  $\beta$ , 0-1 arasında bir değerdir. Üretilen bu yeni nesiller kromozomda yerlerine konularak işlem tamamlanır.

$$\text{Nesil}_1 = [P_{a1}, P_{a2}, \dots, P_{yeni1}, \dots, P_{bNpar}]$$

$$\text{Nesil}_2 = [P_{b1}, P_{b2}, \dots, P_{yeni2}, \dots, P_{aNpar}]$$

### 3.5.9. Mutasyon

Genetik algoritmalar, bazen çok hızlı yakınsarlar. Yakınsama sonucunda global noktalar bulunursa sonuç iyi, lokal noktalar bulunursa beklenen sonuç elde edilememiş demektir. Hızlı yakınsamadan kurtulmanın yolu, araştırma uzayında, mutasyon aracılığıyla yeni çözümler elde etmektir. İkili kodlu genetik algortmada, mutasyon oranı %1 - %5 arasında değiştirilirse iyi sonuçlar elde edilir. Gerçek kodlu GA'da ise mutasyon oranı daha yüksektir. Parametrelerin toplam sayısı ile mutasyon oranı çarpılarak mutasyona girecek parametre sayısı tespit edilir. Bir matriste, satır ve sütunu ifade edecek şekilde rasgele rakamlar üretilerek mutasyon gerçekleştirilir.

### 3.5.10. Elitizm

Elitizm, önceki popülasyon içerisindeki en iyi uygunluk değerine sahip bireyin, olduğu gibi yeni popülasyona taşınması işlemidir. En iyi birey, çaprazlama ve mutasyon gibi diğer genetik işlemlerden etkilenmeden, doğrudan yeni popülasyona aktarılır. Bu işlem, GA'da sıkça uygulanan bir yöntemdir. Çalışma

için yazılan algoritmada istenirse, seçme suretiyle elitizm işlemi yapılabilmektedir.

### 3.5.11. Optimizasyon işlemi

Optimizasyon, bir şeyin daha iyisini gerçekleştirme işlemidir. Ayrıca bilim adamları tarafından ortaya konan yeni bir fikrin geliştirilmesine yardımcı olur.

Optimizasyon işlemi; fikirleri ilerletmek için ortaya atılan bilgileri kullanarak, başlangıç parametrelerini değiştirmektir. Eğer fikri etkileyen parametreler elektronik formata dönüştürülürse, bilgisayar mükemmel bir optimizasyon aracı olabilir.

Bir deney düzeneğinde, en uygun sonucu elde edebilmek için, süreç giriş karakteristiklerinin ayarlanması da bir optimizasyon işlemidir. Optimize edilen parametrelerin uygunluk değerleri, amaca göre tanımlanan uygunluk fonksiyonu ile belirlenir. Eğer süreç deneysel ise, giriş parametreleri fiziksel büyüklüklerden oluşur (Pierre, 1992).

Çoğu optimizasyon metotları, sistemin kök veya sıfırlarının bulunması temeline dayanır. Matematiksel yaklaşımda kök bulmak için, fonksiyonun sıfır olduğu yerler araştırılırken, optimizasyon da türevin sıfır olduğu yerleri araştırmak gerekmektedir. Oysa türev hesabı her zaman kolay bir iş değildir. Teknik problemlerin çoğu, köklerini bulmak üzere formülize edilebilir. Fakat bir kısım optimizasyon yöntemleri bu kökleri bulmada yetersiz kalmaktadır.

Optimizasyonda diğer bir zorluk; elde edilen bir sonucun, global veya lokal bir çözüm olup olmadığının belirlenmesidir. Lineer olmayan bir fonksiyonun da minimumunu bulmak oldukça zordur. Bu tip problemler ya lineer bir yaklaşımla ya da optimizasyon bölgesini küçük bir bölge ile sınırlamakla çözülür.

Optimizasyon algoritmaları altı gruba ayrılabilir. Örneğin; dinamik optimizasyon problemi sınırlı veya sınırsız olabilir. Bazı parametreler ayrık veya sürekli olarak tanımlanabilir.

1) *Deneme-yanılma optimizasyonu*; süreç hakkında çok fazla bilgi olmaksızın çıkışı etkileyen parametrelerin ayarlanmasıdır. Örneğin TV 'deki en iyi görüntü ve ses, deneme yanılma yoluyla ayarlanır. Deneysel çalışma yapanlar ve çoğu büyük kâşifler bu yolu kullanmışlardır.

2) *Tek ve çok parametrelili optimizasyon*; optimize edilecek sadece bir parametre varsa, optimizasyon bir boyutludur. Birden fazla parametreye sahip fonksiyon için çok boyutlu optimizasyon gereklidir. Boyut sayısı artarsa, optimizasyonun zorluk derecesi de artar. Çok boyutlu optimizasyon metodu, bir boyutlu optimizasyon metodu temel alınabilir.

3) *Statik ve dinamik optimizasyon*; statik optimizasyon zamandan bağımsızdır, dinamik optimizasyon ise zamana bağlı olarak çıkış üretir. Örneğin; bir şehrin kenar mahallesinde oturan bir insanın, merkezdeki işine gitmesi için bir çok yol olduğu kabul edilirse en iyi yolun hangisi olduğu sorgulanabilir. Mesafe açısından bakılacak olursa problem statiktir. Çözüm, harita ve arabanın kilometre/saat'i kullanılarak bulunabilir.

4) *Sürekli ve ayrık parametrelili optimizasyon*; sürekli parametreler sonsuz değer alırken ayrık parametreler sınırlı değerler alırlar. Örneğin yapılacak işler bir liste halinde verilmiş olsun. Bu işlerin yapılması bir birinden bağımsız olduğundan ayrık parametrelili düşünülebilir. Ayrık parametrelili optimizasyon, kombinasyonel optimizasyon olarak da adlandırılır.

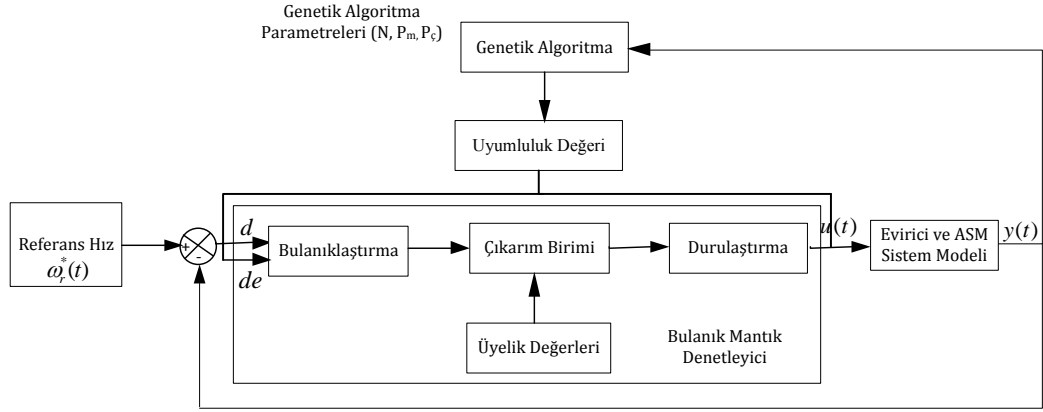
5) *Sınırlı ve sınırsız optimizasyon*; sınırlı optimizasyon, parametreleri bir tanım aralığında değerlendirir. Sınırsız optimizasyonda ise, parametreler herhangi bir değerde olabilir. Değişkenlerin sınırları kaldırılarak sınırlı parametreler sınırsız parametrelere çevrilirler. Çoğu nümerik optimizasyon rutinleri sınırsız parametrelerle çalışırlar. Sınırlı optimizasyonda sınırlar ve denklemler lineer ise, optimizasyon programı lineer program olarak adlandırılır. Sınırlar ve maliyet denklemleri lineer değilse, program da lineer değildir.

6) *Rasgele ve minimum araştırma algoritmaları*; bazı algoritmalar parametrelerin başlangıç değerlerini ayarlayarak, uygunluk değerlerini minimize etmeye çalışır. Bu araştırma tekniği hızlı olmakla beraber, lokal minimumlara ulaşabilir. Bunlar nümerik metotlara dayanan klasik optimizasyon algoritmalarıdır. Burada bir parametreden hareketle diğer parametreyi tespit etmek, bazı deterministik adımlarla gerçekleştirilmektedir. Diğer taraftan rasgele metotlar; parametrelerin optimum çözümünü bulmada ihtimal hesaplarını kullanırlar. Bu metotlar yavaş olmakla birlikte global minimumu bulmada daha başarılıdırlar.

Yukarıdaki gruplandırmanın yanında optimizasyon metotlarını; deterministik metotlar ve istatistiksel metotlar olmak üzere iki ana gruba ayırmak mümkündür. Deterministik optimizasyon metotları, lokal maksimum veya minimuma yakınsayan algoritmalarıdır. Türevsel hesaplamalar veya türevsel yaklaşımlar deterministik metotlara örnek verilebilir. Rastgele araştırma algoritmaları gibi istatistiksel metotlar ise, global minimum veya maksimumu bulmada bazı stratejileri ve rastgele sayıları kullanırlar. PC'lerin işlem hızlarındaki artışla birlikte, rastgele araştırma algoritmaları, değişik alanlarda sıkça kullanılmaktadır (Haupt, 1998).

### **3.6. Genetik Algoritma Tabanlı Bulanık Mantık Denetleyici Tasarımı**

Bu bölümde, aksiyon ağırlık değerleri genetik algoritma tarafından bulunan bir bulanık denetleyicinin tasarlanması açıklanacaktır. Şekil 3.35'de böyle bir denetleyicinin yapısı görülmektedir.



Şekil 3.35. GA ile BM denetleyici parametrelerinin oluşturulması (Şafak, 2005)

Böyle bir kontrol sistemini gerçekleştirmek için üç önemli problemin çözülmesi gerekir. Bunlar:

- Kullanılacak genetik operatörlerin belirlenmesi,
- Bulanık parametrelerin genetik olarak kodlanması,
- Sistem çıkış performansının genetik algoritma uygunluk değerine dönüştürülmesi.

### 3.6.1. Kullanılan Genetik Operatörler

Genetik algorithmada temel olarak popülasyon büyüklüğü, çaprazlama, mutasyon ve seçim gibi genetik operatörler kullanılır. Gerçekleştirdiğimiz sistemde kullanmış olduğumuz genetik operatörler ve değerleri şöyledir:

*Popülasyon büyüklüğü:* Genetik algorithmada popülasyon büyüklüğünün performansı oldukça etkilediği bilinmektedir. Popülasyon büyüklüğünün çok yüksek değerlerinde bilgisayarın işlem gücü ve sonuç bulma zamanı oldukça artmakta, çok küçük popülasyon değerlerinde ise genetik algoritma istenilen sonucu bulamamaktadır. Bu yüzden farklı deneyler yapılmış ve en uygun popülasyon büyüklüğü olarak  $N=1000$  seçilmiştir.

*Çaprazlama:* Tek noktadan çaprazlama operatörü kullanılmıştır. Çaprazlama operatörünün olasılık değeri yapılan deneyler sonucunda  $P_c = 4$  olarak tespit edilmiştir.

*Mutasyon:* Bit mutasyon operatörü kullanılmıştır. Mutasyon operatörünün olasılık değeri de yapılan deneyler sonucunda  $P_m = 0.1$  olarak tespit edilmiştir.

*Üreme:* Gerçekleştirmiş olduğumuz genetik algoritma yapısında bütünüyle yer değiştirme üreme operatörü kullanılmıştır. Bu üreme yöntemine ilave olarak Elistik model kullanılmıştır. Kullanılan Elistik model ile genetik algoritmanın performansının oldukça arttığı gözlenmiştir.

*Seçim:* Seçim operatörü için Rulet çarkı ve Turnuva yöntemleri kullanılmıştır. Yapılan deneylerde Turnuva yönteminin performansının daha iyi olduğu gözlenmiştir. Fakat çok yüksek turnuva büyüklüğü değerlerinde (popülasyon büyüklüğüne eşit olduğunda) genetik algoritmanın doyuma gittiği gözlenmiştir. Deneyler sonucunda bulunan değerler literatürdeki benzer çalışmalarla uygunluk göstermektedir.

### 3.6.2. Bulanık Parametrelerin Genetik Olarak Kodlanması

Genetik algoritma çözülecek problemin kendisi ile değil kodlanmış şekli üzerinden işlem yapar. Dolayısıyla çözülecek problemin kodlanma biçimi, genetik algoritma performansı üzerinde oldukça önemli etki yapar. Gerçekleştirdiğimiz çalışmada, literatürde en yaygın olarak kullanılan Ağırlıklı İkili Kodlama (Weighted Binary Code) biçimi kullanılmıştır. Bu kodlama biçiminde, bütün parametreler ikili kodlanmış bir dizi şeklinde değerlendirilir. Bir gerçel sayının ikili kodlanması Denklem 3.66'da gösterilmiştir.

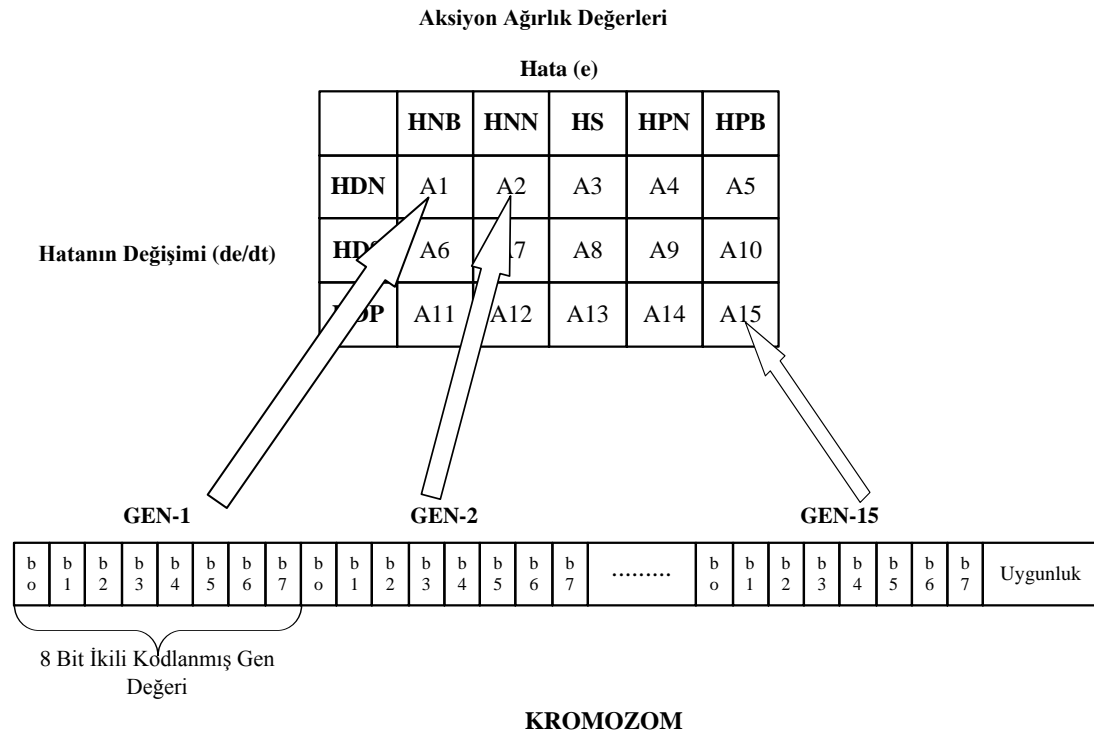
$$X_{Real} = X_{min} + \frac{X_{max} - X_{min}}{2^L - 1} \left( \sum_{i=0}^{L-1} b[i] * 2^{L-i-1} \right) \quad (3.66)$$



Burada  $x_{\min}$ : gerçel sayının min değeri,  $x_{\max}$ : gerçel sayının maksimum değeri, L: dizi boyu, olarak verilmiştir.

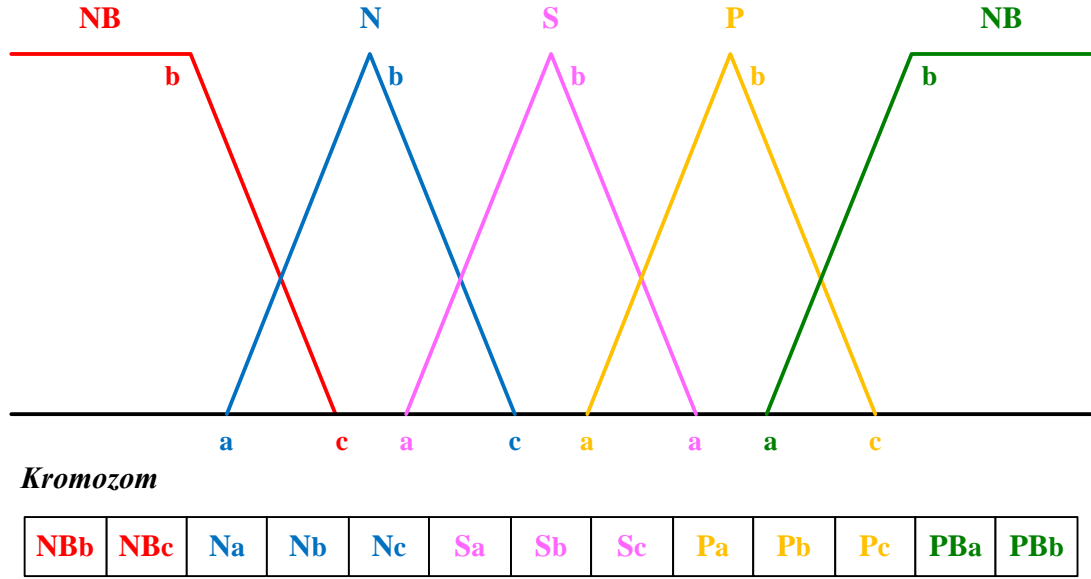
Tasarladığımız bulanık denetleyici Sugeno tipinde bir denetleyicidir. Buna göre oluşturulan kromozom yapısı Çizelge 3.6'da verilmiştir. Burada; A1, A2, ..., An değerleri gerçel sayılardır ve aksiyon ağırlıklarını verirler. Kural ağırlık tablosunda toplam 15 adet kural ağırlığı vardır. Böylece bütün ağırlıkları kodlamak için, ağırlığı 8 bitlik ikili bir dizi ile kodlanarak 120 bitlik ikili bir dizi oluşturulur.

Çizelge 3.6. Genetik alıritmada kullanılan kromozom yapısı (Şafak, 2005)



Bulanık kurallar, bilinen ve sabit olan kurallardır. Genetik algoritmanın rolü giriş ve çıkış üyelik fonksiyonlarını ayarlamaktır. Böylece, sistem için seçilen denetleyici parametreleri, değişik çalışma durumları için maksimum uygunluk değerlerini arar. En uygun denetleyici değerleri, sistem parametre değışim değerlerine adaptasyon ve öğrenme kapasitesinin izin verdiği kadarıyla çevrimiçi olarak kullanılır (Dessouky, 2002).

Genetik algoritmada, parametreler optimizasyon için kromozomların içine kodlanır. Her bir kromozom çözüm için birer adaydır. Kromozomlar bir dizi olarak tanımlanır. Örnek olarak bir bulanık değişkenin 7 adet alt bulanık kümesi olsun. NB, NY, NK, S, PK, PY, PB sırasıyla 1, 2, ..., 7 kadar kodlanır. Kromozomun uzunluğu kural sayısına eşittir (Tong, 2000).



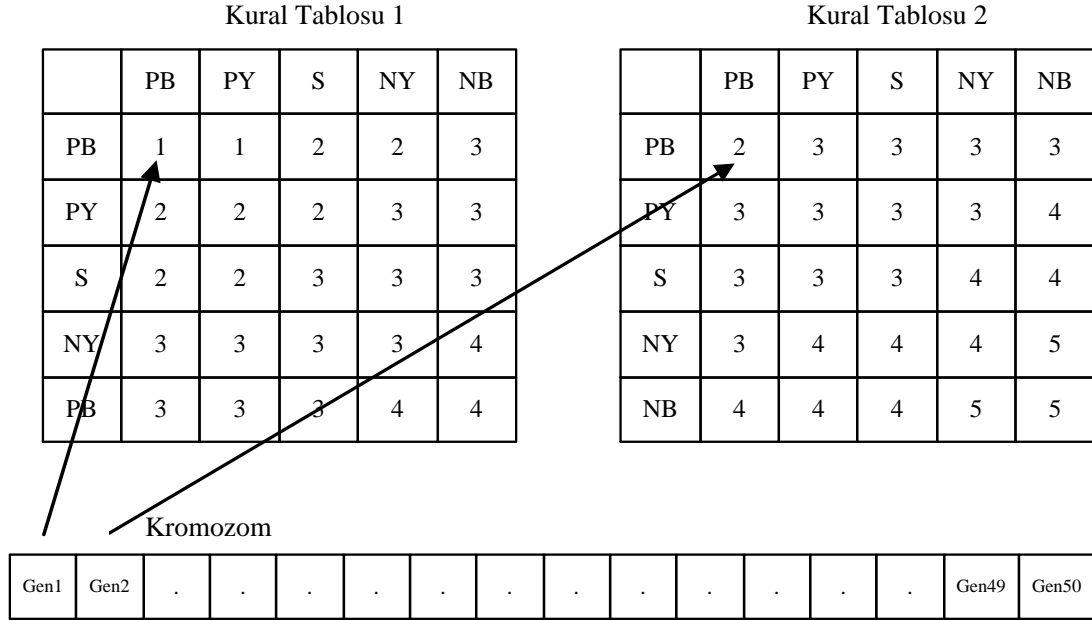
Şekil 3.36. Bulanık veri tabanının bir kromozom içine kodlanması (Şafak, 2005)

Kullanılan her bir kromozom bütün üyelik fonksiyonlarının tiplerini ve sınırlarını içerir. Şekil 3.36'da bu şekilde oluşturulmuş bir kromozom görülmektedir. Her bir kromozoma, bulanık denetleyici uygulanır ve istenilen performans ölçüsünü ne derece gerçekleştirdiğine bakılarak kromozoma uygunluk değerleri atanır.

Bulanık denetleyici kural tablosu bir kromozomun içerisine kodlanır. Genetik algoritma, öğrenme işlemini gerçekleştirir. Oluşturulmuş veri tabanı gerekmektedir. Bir jenerasyon süreci içinde, herhangi bir kromozom içinde, kodlanmış olarak bulanık denetleyici kural tablosuna ait bir değer bulunmasa, o kuralın kullanılmadığı kabul edilerek kural sayısını optimize etmek mümkündür. Çizelge 3.7'de kural tablolarına göre kodlanmış bir kromozom görülmektedir. Kromozomdaki birinci gen 1. kuralı göstermektedir. Kromozom

uzunluğu kural tablosunun uzunluğuna bağlıdır. Çizelge 3.7’de verilen kural sayısı 50 tanedir. Kromozom içerisindeki gen sayısı da 50 olacaktır.

Çizelge 3.7. Örnek kural tablosu (Şafak, 2005)

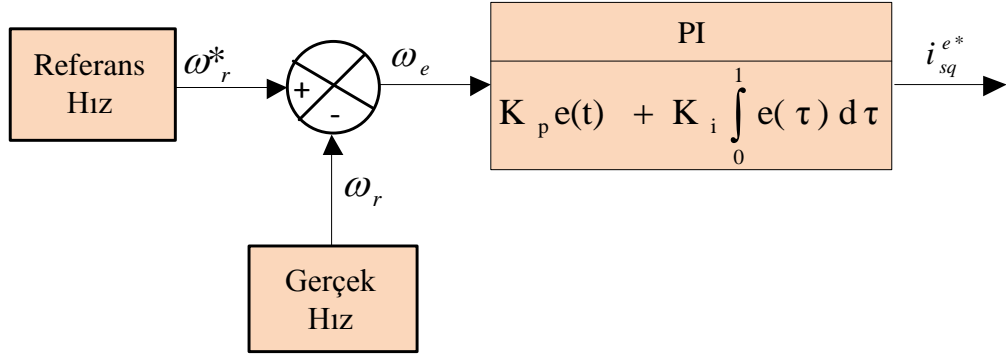


Gerçekleştirilen yapıda bulanık denetleyicinin aksiyon değerleri genetik algoritma tarafından bulunduğundan, sadece aktivasyon değerleri bir kromozom şeklinde kodlanmıştır.

### 3.7. Asenkron Motorun Hızının PI Denetleyici ile Kontrolü

Klasik PI denetleyicinin giriş değişkeni, referans hız ile motorun gerçek hızı arasındaki hız hatası olarak tanımlanmıştır. Çıkış değişkeni olarak da senkron döner eksen akımının  $q$  fazı olan  $i_{sq}^e$  seçilmiştir. Denklem 3.67’de hız hatası verilmektedir. Denklemdeki  $k$  ifadesi,  $k$ ’inci örnekleme anını ifade etmektedir. Şekil 3.37’de PI denetleyiciye ait blok şema verilmiştir.

$$\omega_e(k) = \omega_r^*(k) - \omega_r(k) \quad (3.67)$$



Şekil 3.37. ASM'nin PI denetleyici ile hız denetimi

PI denetleyici iki temel bileşenden oluşur. Bunlar; oransal (proportional) ve integral bileşenleridir. PI denetim algoritmasında bu bileşenler birer katsayı ile ifade edilirler. Bu katsayılar oransal katsayı ( $K_p$ ) ve integral katsayı ( $K_i$ ) olarak adlandırılır. PI denetimin matematiksel ifadesi Denklem 3.68'de gösterildiği gibidir.

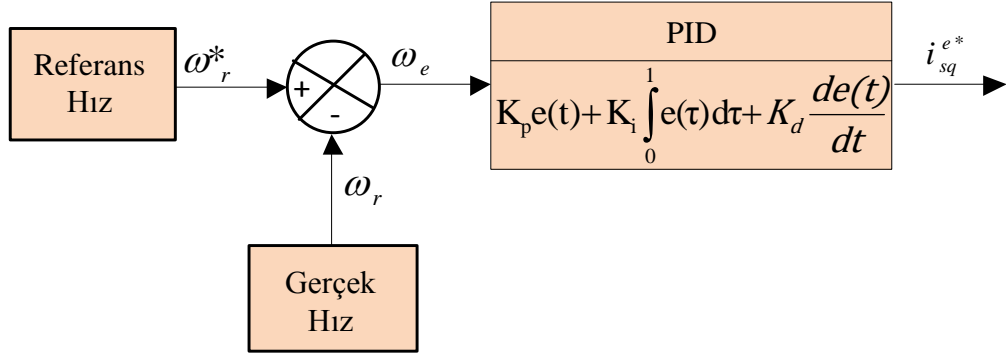
$$\Delta i_{sq}^e = K_p \omega_e(k) + K_i \sum_{j=0}^k \omega_{ej} \quad (3.68)$$

PI denetleyicinin bilgisayar uygulamasındaki diferansiyel ifadesi Denklem 3.69'da gösterildiği gibidir.

$$i_{sq}^e(k) = i_{sq}^e(k-1) + \Delta i_{sq}^e(k) \quad (3.69)$$

### 3.8. Asenkron Motorun Hızının PID Denetleyici ile Kontrolü

Klasik PID denetleyicinin giriş değişkeni, referans hız ile motorun gerçek hızı arasındaki hız hatası olarak tanımlanmıştır. Çıkış değişkeni olarak da senkron döner eksen akımının  $q$  fazı olan  $i_{sq}^e$  seçilmiştir. Şekil 3.38'de PID denetleyiciye ait blok şema verilmiştir.



Şekil 3.38. ASM'nin PID denetleyici ile hız denetimi

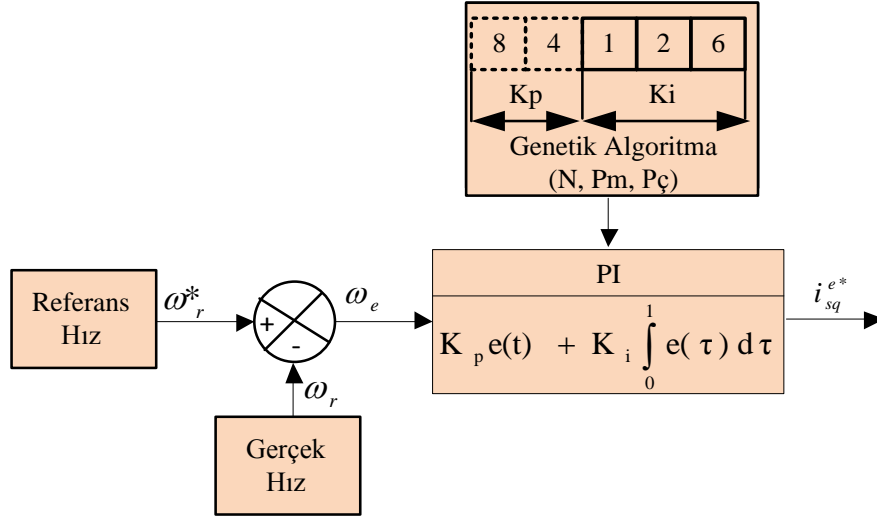
PID denetleyici üç temel bileşenden oluşur. Bunlar; oransal (proportional), integral ve türevsel (derivative) bileşenleridir. PID denetim algoritmasında bu bileşenler birer katsayı ile ifade edilirler. Bu katsayılar oransal katsayı ( $K_p$ ), integral katsayı ( $K_i$ ) ve türevsel katsayı ( $K_d$ ) olarak adlandırılır. PID denetimin matematiksel ifadesi Denklem 3.70'de gösterildiği gibidir.

$$\Delta i_{sq}^e = K_p \omega_e(k) + K_i \sum_{j=0}^k \omega_{ej} + K_d (\omega_e(k) - \omega_e(k-1)) \quad (3.70)$$

### 3.9. Asenkron Motorun Hızının Genetik-PI Denetleyici ile Kontrolü

Genetik-PI denetleyicinin giriş değişkeni, referans hız ile motorun gerçek hızı arasındaki hız hatası olarak tanımlanmıştır. Çıkış değişkeni olarak da senkron döner eksen akımının  $q$  fazı olan  $i_{sq}^e$  seçilmiştir.

Genetik bir PI kontrol denetleyicisi GA ile kullanılabilir. Evrim ve genetik prensiplerin seçimi ve adapte denetleyici parametreleri  $K_p$  ve  $K_i$  için kromozomlar ondalık sayı ile kodlanmıştır. Şekil 3.39'da Genetik-PI denetleyiciye ait blok şema verilmiştir.



Şekil 3.39. ASM'nin Genetik-PI denetleyici ile hız denetimi

Genetik-PI denetleyici iki temel bileşenden oluşur. Bunlar; oransal (proportional) ve integral bileşenleridir. Genetik-PI denetim algoritmasında bu bileşenler birer katsayı ile ifade edilirler. Bu katsayılar oransal katsayı ( $K_p$ ) ve integral katsayı ( $K_i$ ) olarak adlandırılır. Genetik-PI denetimin matematiksel ifadesi Denklem 3.71'de gösterildiği gibidir. Bu denklemde  $\omega_e$  motorun hız hatası,  $\omega_{ce}$  hız hatasındaki değişim ve  $T$  örnekleme zamanını ifade etmektedir.

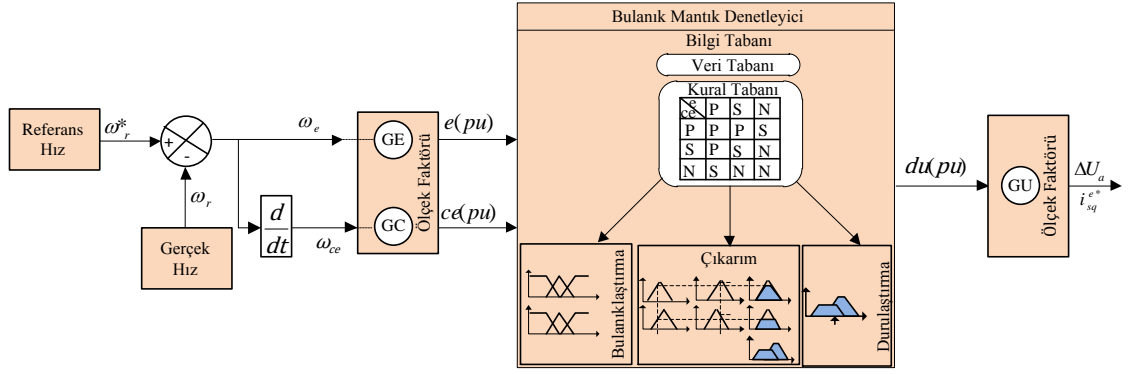
$$F = \frac{1}{\omega_e^2 + \omega_{ce}^2 + (K_p \cdot \omega_{ce} + K_i \cdot \omega_e \cdot T)^2} \quad (3.71)$$

### 3.10. Asenkron Motorun Hızının Bulanık Mantık Denetleyicisi ile Kontrolü

Bulanık mantık denetleyicisinin giriş değişkenleri referans hız ile motorun gerçek hızı arasındaki hız hatası ve hız hatasındaki değişim olarak tanımlanmıştır. Hız hatası Denklem 3.67'deki gibi ifade edilmektedir. Hız hatasındaki değişim ise Denklem 3.72'de gösterildiği gibidir.

$$\omega_{ce}(k) = \omega_e(k) - \omega_e(k-1) \quad (3.72)$$

Şekil 3.40'da bulanık mantık denetleyiciye ait blok şema verilmiştir.



Şekil 3.40. ASM'nin BM denetleyici ile hız denetimi

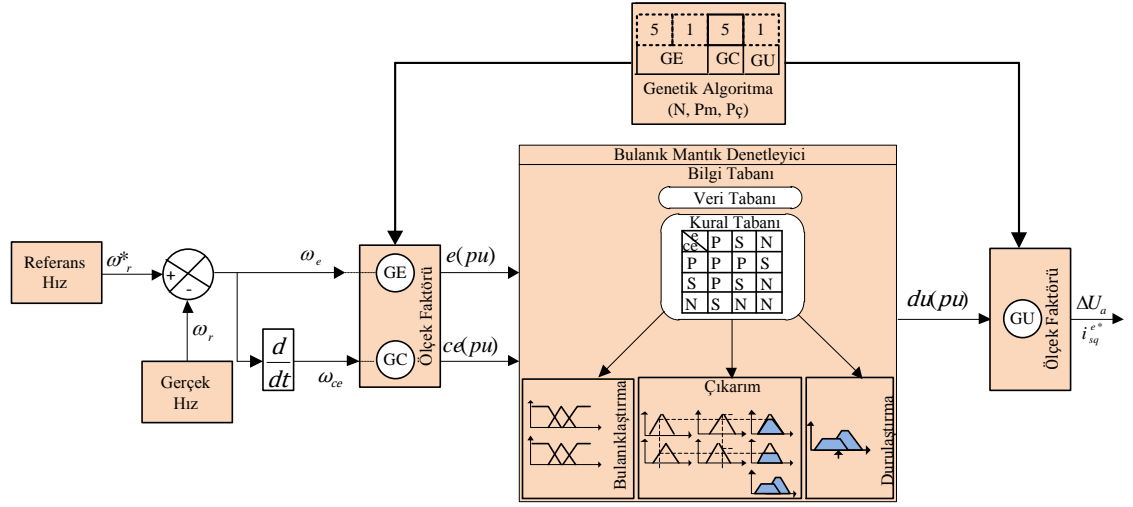
Tasarımda esneklik sağlamak ve denetleyiciyi doğru bir şekilde ayarlamak için değişkenler, birim değer yani -1 ile 1 aralığında temsil edilmektedirler. Denetleyicinin giriş ve çıkış değişkenlerinin birim değer olması için ölçeklendirme faktörleri kullanılır. Bu ölçekleme faktörleri GE, GC ve GU olarak isimlendirilirler. Birim değer olarak açıklanan değişkenler Denklem 3.73'de verilmiştir. Birim değeri ifade etmek için "bd" ifadesi kullanılmıştır.

$$\begin{aligned}
 e(bd) &= \frac{\omega_e(k)}{GE} \\
 ce(bd) &= \frac{\omega_{ce}(k)}{GC} \\
 du(bd) &= \frac{\Delta i_{sq}^e(k)}{GU}
 \end{aligned} \tag{3.73}$$

Hız denetiminde öncelikle asenkron motorun hız değeri örneklenir. Belirlenen referans hız değerine göre hız hatası ve hız hatasındaki değişim hesaplanmaktadır. Hız hatası ve hız hatasındaki değişim için belirlenen bulanık kümeler ve üyelik fonksiyonları kullanılarak üyelik fonksiyonlarına üyelik dereceleri bulunur. Çıkarım işlemi yapılarak kural tabanından her bir kurala bağlı denetim fonksiyonundaki değişim belirlenir. Durulaştırma yöntemi kullanılarak gerçek değişim hesaplanır. Hesaplanan gerçek değişim değeri motoru kontrol etmek için kullanılır.

### 3.11. Asenkron Motorun Hızının Genetik Tabanlı Bulanık Mantık Denetleyicisi ile Kontrolü

Genetik tabanlı bulanık mantık denetleyicisinin giriş değişkenleri referans hız ile motorun gerçek hızı arasındaki hız hatası ve hız hatasındaki değişim olarak tanımlanmıştır. Hız hatası Denklem 3.67’de ve hız hatasındaki değişim ise Denklem 3.72’de gibi ifade edilmektedir. Şekil 3.41’de genetik tabanlı bulanık mantık denetleyiciye ait blok şema verilmiştir.



Şekil 3.41. ASM'nin GABM denetleyici ile hız denetimi

Bulanık mantık denetleyici için verilen ölçekleme faktörleri GE, GC ve GU birim değer olarak açıklanan değişkenler Denklem 3.73’de gösterilmiştir.

Hız denetiminde öncelikle asenkron motorun hız değeri örneklenir. Belirlenen referans hız değerine göre hız hatası ve hız hatasındaki değişim hesaplanmaktadır. Hız hatası ve hız hatasındaki değişim için belirlenen bulanık kümeler ve üyelik fonksiyonları kullanılarak üyelik fonksiyonlarına üyelik dereceleri bulunur. Çıkarım işlemi yapılarak kural tabanından her bir kurala bağlı denetim fonksiyonundaki değişim belirlenir. Durulaştırma yöntemi kullanılarak gerçek değişim hesaplanır. Daha sonra GA operatörleri kullanarak yeni nesil üretilir ve popülasyonun her bir üyesi için BM denetleyici ölçekleme faktörleri GE, GC ve GU değerleri, Denklem 3.74’de verilen uygunluk değerine göre  $\alpha$  ve  $\beta$  birer sabit sayı,  $tr$  ASM'nin hızlanma zamanı ve  $\sum e_{\omega}$  toplam hız

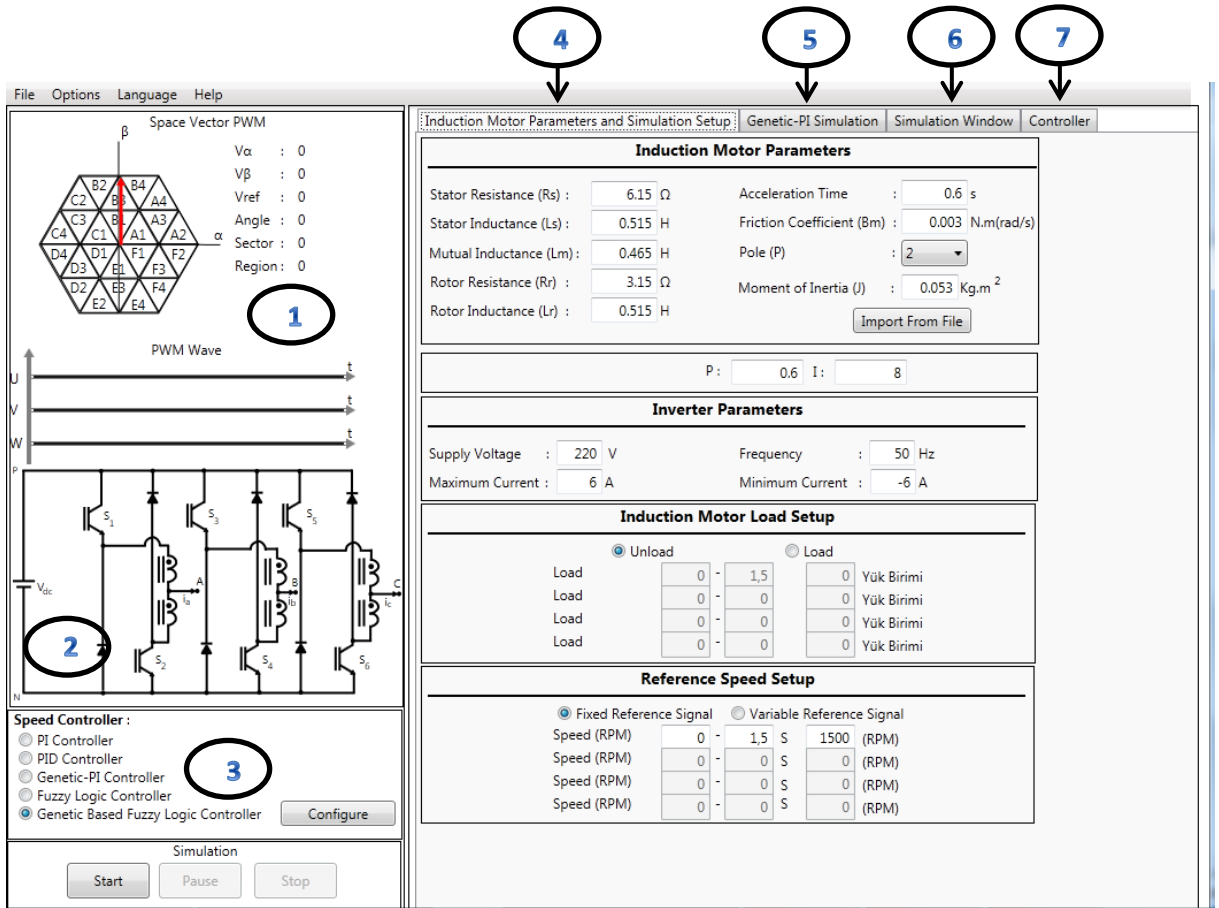


hatası olarak belirlenir. Hesaplanan BM denetleyici ölçekleme faktörleri GE, GC ve GU değerlerine uygunluk değeri atanarak ilgili değeri motoru kontrol etmek için kullanılır.

$$F = \frac{1}{1 + \alpha tr + \beta \sum |e_{\omega}|} \quad (3.74)$$

#### 4. ARAŞTIRMA BULGULARI

Bu tez çalışmasında ASM'nin hız kontrolünü gerçekleştirmek için klasik PI ve PID denetleyici, Genetik-PI, BM ve GABM denetleyicilerinin kullanılacağı bir simülasyon programı hazırlanmıştır. Simülasyon programı Microsoft Visual Studio 2012 IDE' sinde WPF altyapısı kullanılarak, C# programlama dili ile yazılmıştır. Simülasyon programı esnek bir yapıya ve grafiksel bir kullanıcı ara yüzüne sahiptir. Motor ve denetleyici parametreleri kolaylıkla değiştirilerek farklı çalışma durumlarında devre tepkileri grafikler yardımıyla izlenebilmektedir. Programın kullanılabilmesi için .NET Framework 4.0 yüklenmiş, Windows 7 ve üzeri işletim sistemine sahip bilgisayarlar gereklidir. Programın ana penceresi Şekil 4.1'de gösterilmiştir. Simülasyon programı "7" ana bölümden oluşmaktadır. "1" numaralı bölümde UVDGM simülasyonu, "2" numaralı bölümde bağlaşık indüktörlü üç-seviyeli evirici simülasyonu, "3" numaralı bölümde hız kontrolcüsü ve simülasyon işlemleri seçim menüsü, "4" numaralı bölümde ASM parametreleri ve simülasyon ayarı menüsü, "5" numaralı bölümde Genetik-PI simülasyon menüsü, "6" numaralı bölümde simülasyon penceresi menüsü, "7" numaralı bölümde BM ve GABM kontrol denetleyicisi menüleri yer almaktadır.



Şekil 4.1. Simülasyon programının ana penceresi

Ana pencerenin üst kısmında “File - Dosya”, “Options- Seçenekler”, “Language - Dil” ve “Help - Yardım” menüleri yer almaktadır. “File - Dosya” menüsünün altında programdaki verilerin dışarıya aktarılması için “Export – Dışa Aktar” menüsü, dışarıdaki verilerin programa aktarılması için “Import – İçe Aktar” menüsü ve programdan çıkış yapmak için “Exit - Çıkış” menüsü yer almaktadır. “Options - Seçenekler” menüsünde simülasyon programının ana parametreleri ayarlanır. “Language - Dil” menüsü ile hazırlanan simülasyon programının iki farklı dilde kullanılabilmesini sağlar. “Help - Yardım” menüsü ise hazırlanan simülasyon programının kullanılabilmesi için bilgi verir. Hazırlanan simülasyon programı Türkçe ve İngilizce olarak kullanılabilir.

#### 4.1. UVDGM, Baęlaşık İndüktörlü Üç-Seviyeli Evirici ve Hız Kontrolcüsü Menüsü

“1” numaralı bölümde UVDGM bilgilerinin gösterildięi  $V_\alpha$ ,  $V_\beta$ ,  $V_{ref}$ , açısı, sektör ve bölge alanları yer almaktadır. Bu bölümde uzay vektör düzleminin şekli ve üç seviyeli evirici için kullanılan 24 bölge gösterilmiştir. Simülasyon süresi boyunca referans vektörünün düzlemdeki şekli görülebilmektedir. Aynı zamanda referans vektör hangi bölgede yer alırsa o bölgenin arka rengi açık mavi yapılarak hangi bölgenin kullanıldığını kullanıcıya gösterilmiştir. Uzay vektör düzleminin sol tarafında  $\alpha$ ,  $\beta$  ve referans vektörlerinin değerlerinin yanında, referans vektörün açısı, hangi sektörde olduğu ve hangi bölgede yer aldığı bilgileri de kullanıcıya gösterilmiştir. Bu bölümün alt kısmında ise DGM dalga şekilleri yer almaktadır. Burada gösterilen dalga şekilleri üç seviyeden oluşmaktadır ve pozitif, sıfır ve negatif olarak adlandırılırlar. Her bir faz için çıkış sinyalleri eviricide yer alan IGBT güç anahtarlarını kontrol ettiği için renklendirme de bu durum göz önüne alınarak yapılmıştır. Eviricide kullanılan renkler pozitif gerilim seviyesi için kırmızı, sıfır gerilim seviyesi için turkuaz ve negatif gerilim seviyesi için mavi renk kullanılmıştır. Her bir faz için çıkış sinyalleri örnekleme periyodu içerisinde eviriciye uygulanır.

“2” numaralı bölümde baęlaşık indüktörlü üç-seviyeli evirici devre şemasının yapısı, anahtarlama ve fazların durumlarını göstermektedir. Simülasyon süresince hangi IGBT güç anahtarlarının iletimde hangilerinin kesimde olduğu bilgisi renkler yardımıyla kullanıcıya gösterilmiştir. Siyah renkteki IGBT güç anahtarları, anahtarın kesimde olduğunu gösterir. İletim durumu için ise 3 farklı renk kullanılmıştır. Bu renkler kırmızı, mavi ve turkuaz rengidir. Üç ayrı rengin kullanılma nedeni, asenkron motorun fazına uygulanan üç farklı gerilim olmasıdır. Giriş gerilimi  $V_{dc}$  olarak alındığında bu gerilimler  $\frac{V_{dc}}{2}$ , 0 ve  $-\frac{V_{dc}}{2}$  dir ve sırasıyla pozitif, sıfır ve negatif olarak adlandırılmıştır. Bu sebeple pozitif gerilimi oluşturacak anahtarlama için kırmızı renk, sıfır gerilimi oluşturacak anahtarlama durumu için turkuaz ve negatif gerilim değeri oluşturacak anahtarlama durumu için mavi renk kullanılmıştır.

“3” numaralı bölümde hız kontrol denetleyici seçimi ve simülasyon işlemlerinin ayarlanabileceği bölümdür. Bu bölümdeki listeden PI, PID, Genetik-PI, BM ve GABM Kontrol denetleyicileri seçilebilir. Denetleyicinin parametre ayarları “Configure - Yapılandırma” tuşu yardımıyla gerçekleştirilir. PI ve PID denetleyiciler dışındaki denetleyici türleri seçildiğinde ise yeni bir sekme penceresi açılarak, seçilen denetleyici ile ilgili parametrelerin ayarlanabileceği bölümler gösterilir. “3” numaralı bölümde diğer bir seçenek ise simülasyonun başlatılıp, duraklatılabileceği ve durdurulabileceği düğmeler bulunmaktadır.

#### **4.2. Asenkron Motor Parametreleri ve Simülasyon Ayarları Menüsü**

“4” numaralı bölümde ASM parametreleri ve simülasyon ayarı menüsü Şekil 4.2’de gösterilmiştir. Bu bölümün “a” numaralı kısmında kontrol edilmek istenen ASM’nin parametrelerinin ayarlandığı bölüm mevcuttur. Bu kısımda yer alan “Import From File - Dosyadan İçer Aktar” seçeneği ile dışarıdan daha önce kaydedilmiş ASM parametreleri programa aktarılabilir. Aynı işlem “File - Dosya” menüsü altında yer alan “Import – İçer Aktar” menüsünden de yapılabilir. Bu bölümün “b” numaralı kısmında PI ve PID klasik denetleyicilere ait P, I ve D katsayılarının değerleri ayarlanmaktadır. “c” numaralı kısmında evirici parametrelerinin ayarlanabileceği bölüm yer almaktadır. “d” numaralı kısmında ise ASM yük ayarı bölümünde ise, motorun yük var iken ya da boşta olarak çalıştırılacağı ayarlanır. Yük var iken seçeneği seçildiğinde istenen aralıklarda istenen değerde motora yük bindirilebilir. Eklenen yük değerinin hangi aralıklarda ve hangi değerde uygulanacağı seçilir. Eklenen yük değeri değiştirilebilir ya da iptal edilebilir. Yüksüz seçeneği seçildiğinde ise motor boşta olarak çalıştırılmaktadır.

4

Induction Motor Parameters and Simulation Setup		Genetic-PI Simulation	Simulation Window
<b>Induction Motor Parameters</b>			
Stator Resistance (Rs) :	<input type="text" value="6.15"/> Ω	Acceleration Time :	<input type="text" value="0.6"/> s
Stator Inductance (Ls) :	<input type="text" value="0.515"/> H	Friction Coefficient (Bm) :	<input type="text" value="0.003"/> N.m(rad/s)
Mutual Inductance (Lm) :	<input type="text" value="0.465"/> H	Pole (P) :	<input type="text" value="2"/>
Rotor Resistance (Rr) :	<input type="text" value="3.15"/> Ω	Moment of Inertia (J) :	<input type="text" value="0.053"/> Kg.m <sup>2</sup>
Rotor Inductance (Lr) :	<input type="text" value="0.515"/> H	<input type="button" value="Import From File"/>	
P: <input type="text" value="0.6"/> I: <input type="text" value="8"/>			
<b>Inverter Parameters</b>			
Supply Voltage :	<input type="text" value="220"/> V	Frequency :	<input type="text" value="50"/> Hz
Maximum Current :	<input type="text" value="6"/> A	Minimum Current :	<input type="text" value="-6"/> A
<b>Induction Motor Load Setup</b>			
<input checked="" type="radio"/> Unload		<input type="radio"/> Load	
Load	<input type="text" value="0"/> - <input type="text" value="1,5"/>	<input type="text" value="0"/>	Yük Birimi
Load	<input type="text" value="0"/> - <input type="text" value="0"/>	<input type="text" value="0"/>	Yük Birimi
Load	<input type="text" value="0"/> - <input type="text" value="0"/>	<input type="text" value="0"/>	Yük Birimi
Load	<input type="text" value="0"/> - <input type="text" value="0"/>	<input type="text" value="0"/>	Yük Birimi

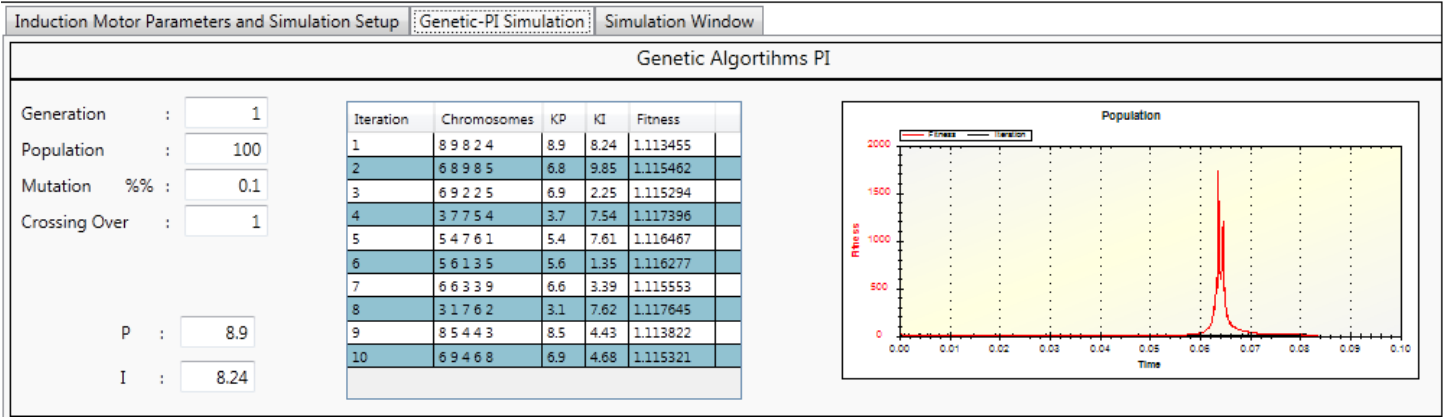
Şekil 4.2. ASM ve evirici parametreleri ayarları menüsü

### 4.3. Genetik-PI Denetleyici Menüsü

“5” numaralı bölümde Genetik PI simülasyon menüsü Şekil 4.3’de gösterilmiştir. Bu kısımda yer alan “Generation – Nesil”, “Population - Popülasyon”, “Mutation - Mutasyon” ve “Crossing Over - Çaprazlama” gibi GA parametrelerinin değerleri ayarlanmaktadır. GA ile iterasyon sayısına bağlı olarak kromozom yapısı oluşturulmaktadır. Genetik PI denetleyici ayarı için beş basamaklı bir kromozom yapısı tanımlanmıştır. Örneğin bu uygulamada Ziegler ve Nichols yöntemi ile  $K_p$  ve  $K_i$ , değerleri belirli bir çalışma noktası için ayarlanmış denetleyici katsayıları sırasıyla, 8.4 ve 1.72 olarak bulunmuştur. Oluşturulan kromozomun ilk iki değeri  $K_p$  için, son üç değeri ise  $K_i$  için kodlanmıştır [ $K_p K_i$ ] = {84172}. Kodlama yöntemi ondalık sayı ve her parametre ögesi arasında 0-9

arası rakamdan oluşmaktadır. Uzman bilgisine ihtiyaç duyulmadan GA kullanılarak Genetik-PI denetleyicisi ile oluşturulan popülasyonun üyeleri klasik PI denetleyicisinde olduğu gibi ASM'ye P ve I katsayısı şeklinde uygulanabilmektedir. GA yoluyla belirlenen yüksek uyumluluk değerleri elde edildikten sonra, bu yüksek uyumluluk değerleri içerisinde en uygun olan değerlerin Kp ve Ki katsayıları PI denetleyicisi yapısına yerleştirilerek direkt olarak ASM'ye uygulanmaktadır. Ayrıca verilen grafikte ise zamana bağlı olarak iterasyon sayısına göre uyumluluk değeri gözlemlenebilmektedir.

4



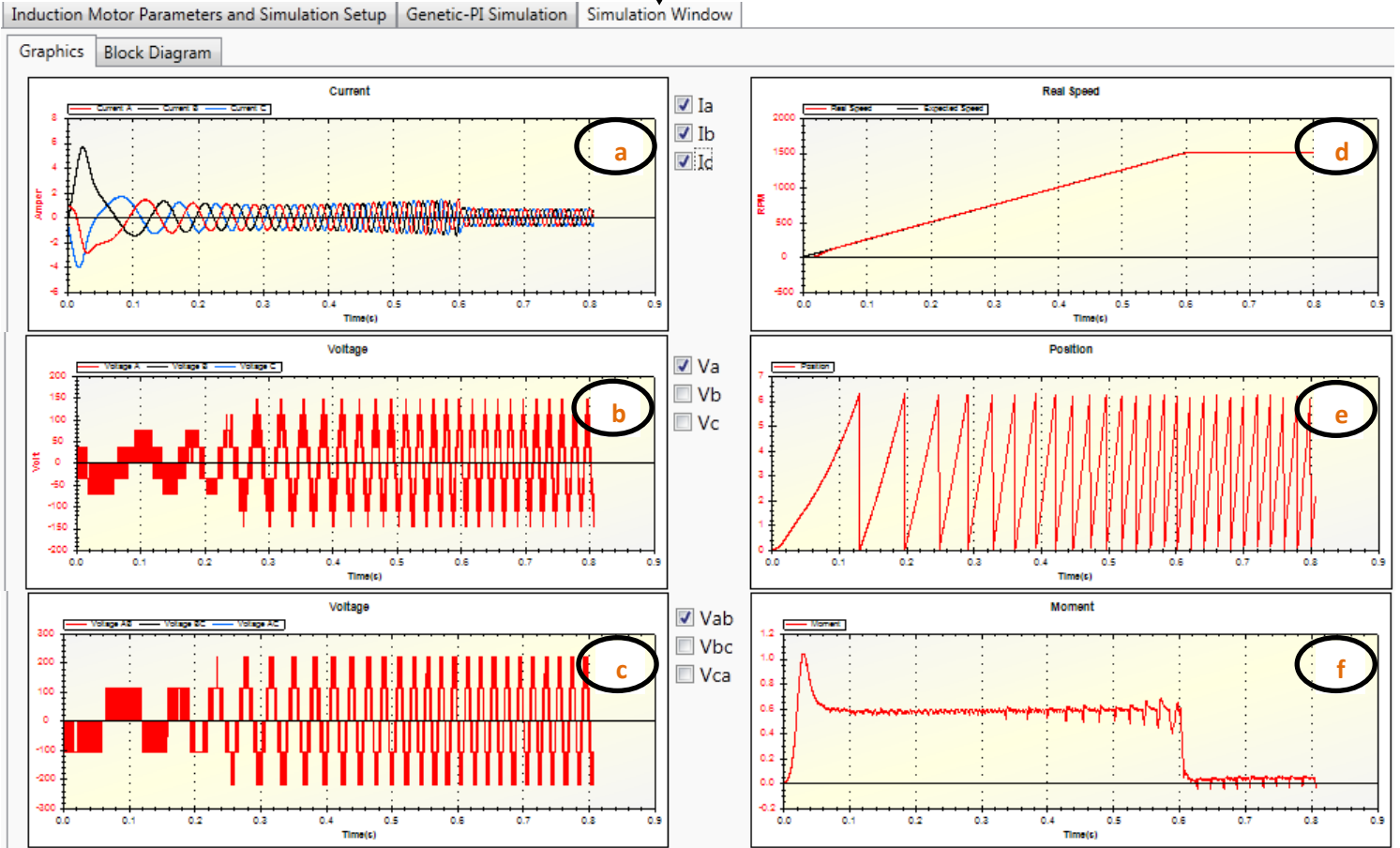
Şekil 4.3. Genetik-PI denetleyici menüsü

#### 4.4. Simülasyon Grafikleri Menüsü

“6” numaralı bölümde simülasyon pencere menüsü grafikler sekmesi için iki farklı grafik gösterim ekranı hazırlanmıştır. PI, PID ve Genetik-PI denetleyici için hazırlanan simülasyon grafikleri ekranı Şekil 4.4’de gösterilmiştir. Bu bölümün “a” numaralı kısmındaki grafik ekranı motorun  $I_a$ ,  $I_b$  ve  $I_c$  fazlar arası akımlarının simülasyon grafikleridir. Grafiğin yanında yer alan seçenekler yardımı ile grafiği görüntülenmek istenilen faz akımları seçilebilir. Bu kısımda tüm faz akımların seçimi, sadece bir faz akımın seçimi ya da herhangi iki faz akımın seçimi kullanıcı tarafından yapılabilmektedir. Bu bölümün “b” numaralı kısmındaki grafik ekranı  $V_{an}$ ,  $V_{bn}$  ve  $V_{cn}$  faz gerilimlerinin grafiklerine yer verilmiştir. Akım grafiklerinde olduğu gibi faz gerilim grafiklerinin gösterimi kullanıcının seçimine bağlıdır. Bu bölümün “c” numaralı kısmındaki grafik ekranı  $V_{ab}$ ,  $V_{bc}$  ve

$V_{ca}$  fazlar arası gerilimlerinin yer aldığı grafiklerdir. Bu bölümün “d” numaralı kısmındaki grafik ekranı referans hızın ve motorun gerçek hızının gösterildiği grafikdir. Bu grafikte referans hız siyah renkle, motorun gerçek hızı ise kırmızı renkle ifade edilmiştir. Bu bölümün “e” numaralı kısmındaki grafik ekranı motorun konumunu gösteren pozisyon grafiğidir. Bu bölümün “f” numaralı kısmındaki grafik ekranı ise motorun moment değişiminin zamana göre değerini gösteren grafiği temsil etmektedir.

6



Şekil 4.4. PI, PID ve Genetik-PI denetleyici için simülasyon grafikleri ekranı

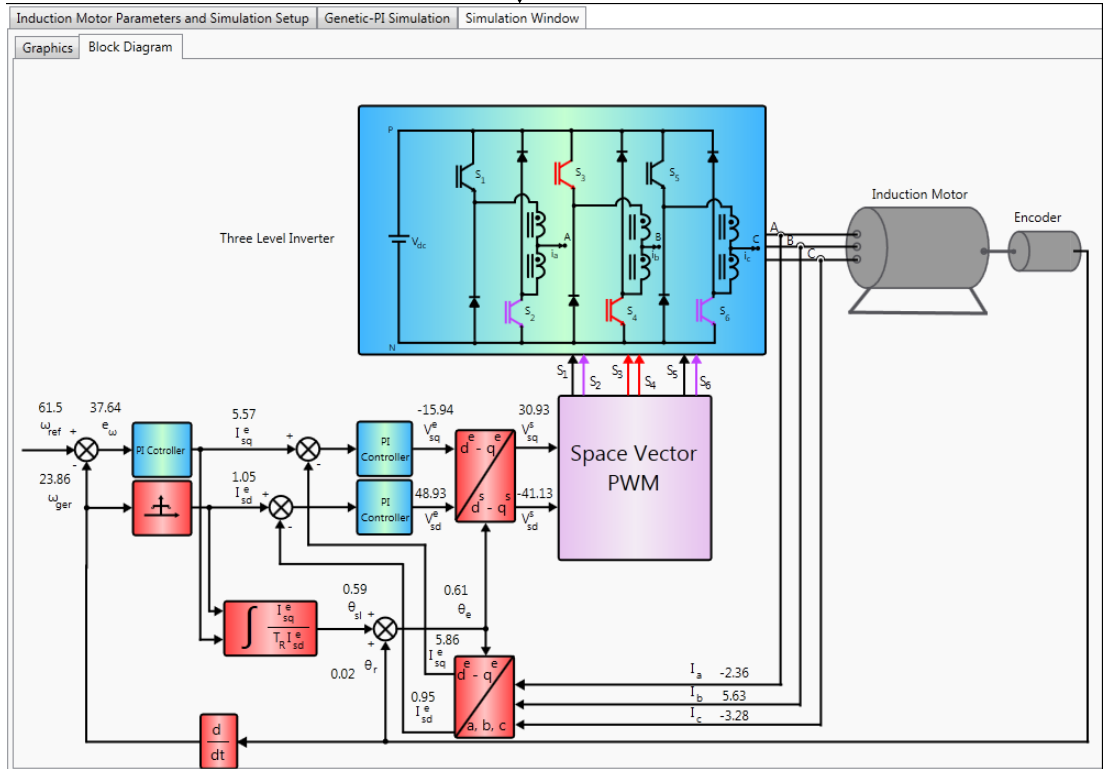
#### 4.5. Blok Diyagram Menüsü

“6” numaralı bölümde simülasyon pencere menüsü blok diyagram sekmesi Şekil 3.1’de gösterilen bağlaşıklı indüktörlü üç-seviyeli evirici tarafından beslenen ASM’nin denetim blok şeması programın arayüzüne aktarılmıştır. Blok



diyagram sekmesi Şekil 4.5’de gösterilmiştir. Bu amaçla simülasyon süresince blok diyagramda gösterilen değerlerin anlık olarak izlenebilmesi sağlanmıştır. Anlık değerler genellikle veri etiketlerinin üzerinde gösterilmiştir. Bağlaşık indüktörlü üç-seviyeli eviricinin simülasyon anındaki anahtarlama durumları ve üretilen DGM sinyalleri blok diyagramda gösterilmiştir. Simülasyon penceresi menüsü grafikler sekmesinde sınırlı sayıda grafik gösterimleri yapıldığından dolayı blok diyagram sekmesinde bu sınırlılık ortadan kaldırılmıştır. Blok diyagram sekmesindeki veri etiketlerine tıkladığında ayrı bir pencerede, simülasyon süresince ilgili değişkenin aldığı değerler grafiksel olarak gösterilmiştir. Açılan bu pencere kapatılıncaya veya simülasyon süresi bitinceye kadar ilgili değişkenin anlık değerleri grafiğe aktarılmaktadır. Bu amaçla simülasyon pencere menüsü grafikler sekmesinde izlenemeyen değişken değerlerinin grafikleri görüntülenebilmektedir. Veri etiketlerine tıklanarak birden fazla grafik gösterim ekranı penceresi açmak mümkündür. Bu sekmedeki bir başka durum ise, UVDGM ile bağlaşık indüktörlü üç-seviyeli evirici arasındaki bağlantıların 3 ayrı gruba ayrılmasıdır. Birinci grup eviricinin  $S_a$  anahtarlarına, ikinci grup eviricinin  $S_b$  anahtarlarına üçüncü grup ise eviricinin  $S_c$  anahtarlarına bağlanmıştır. Bu gruplarda üç farklı renk kullanılmıştır. İlgili bağlantılar bağlaşık indüktörlü üç-seviyeli eviricideki bağlı olan IGBT güç anahtarlarının renk durumları, aynı renktedir.

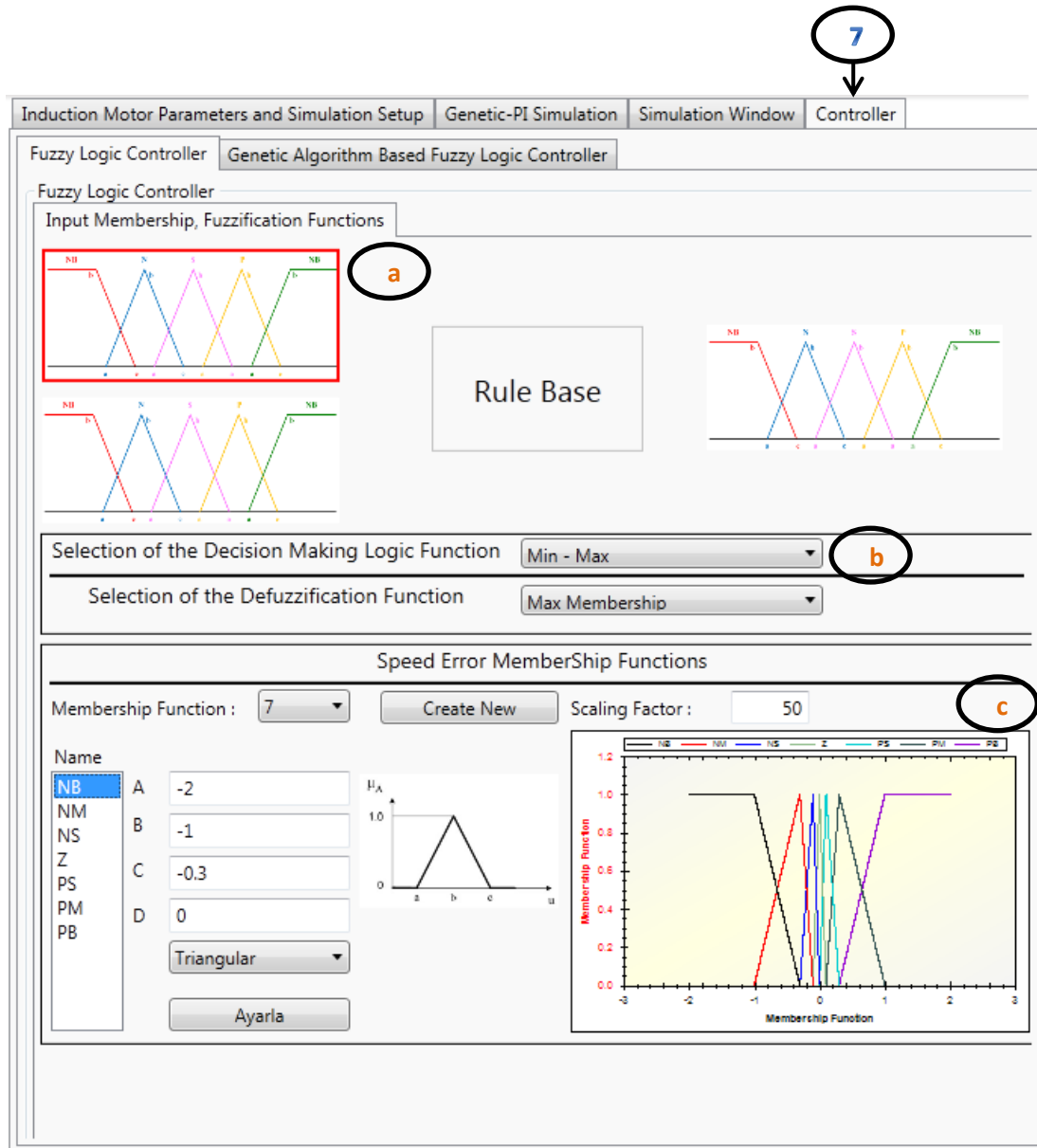
6



Şekil 4.5. Blok diyagram menüsü

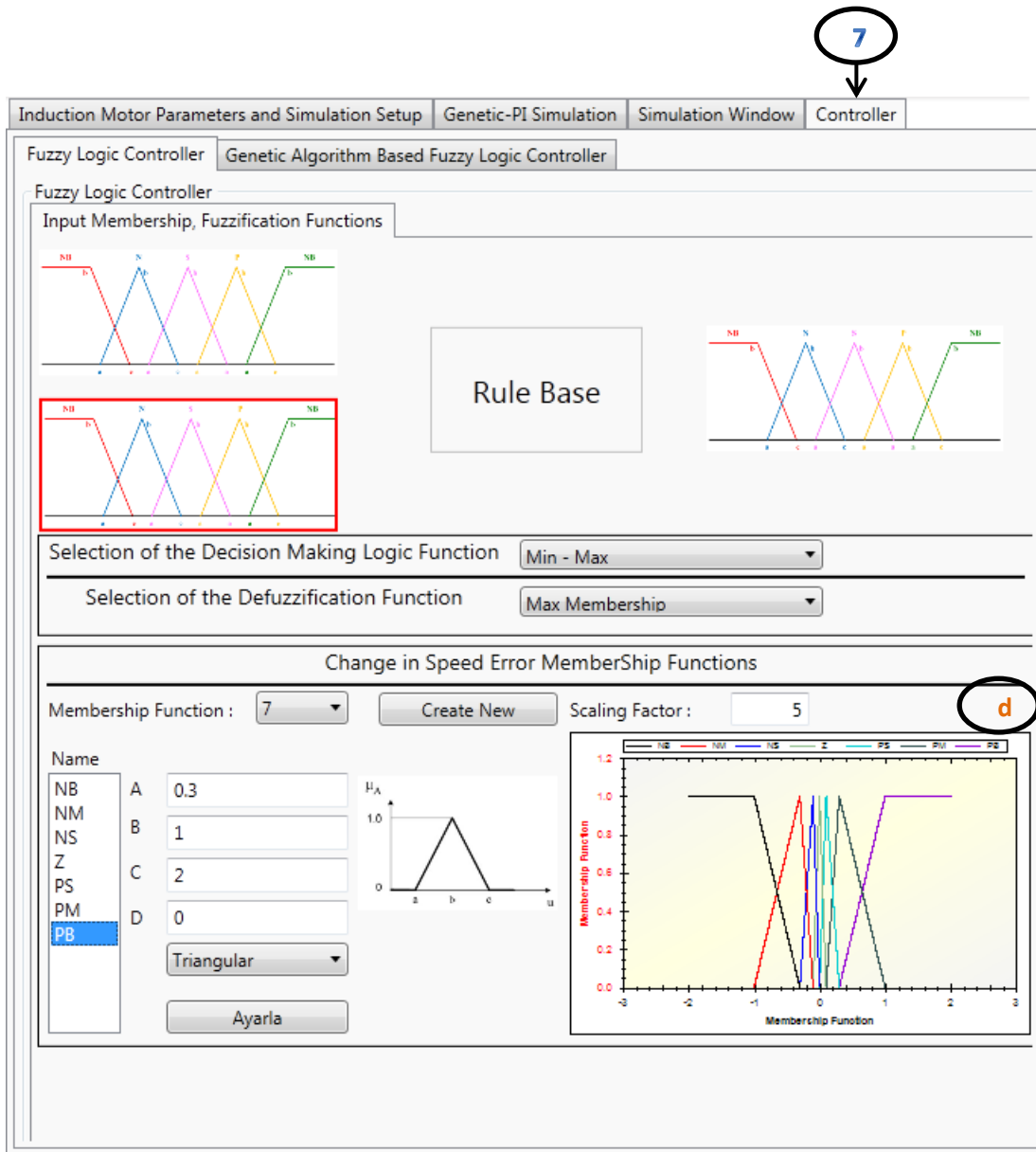
#### 4.6. Bulanık Mantık Denetleyicisi Menüsü

“7” numaralı bölümde kontrol menüsü BM denetleyicisi sekmesi Şekil 4.6 ve Şekil 4.7’de gösterilmiştir. Bu bölümün “a” numaralı kısmında motorun hız hatası, motorun hız hatası değişimi, çıkış üyelik fonksiyonlarının ve kural tabanının tanımlanması işlemidir. Üyelik fonksiyonlarının sayısı için üç, beş ve yedi değerleri seçilebilir. Ayrıca her bir üyelik fonksiyonu için dört fonksiyon tipi kullanıcının seçimine sunulmuştur. Bunlar triangular, trapezoidal, gaussian ve sigmoid üyelik fonksiyonlarıdır. Bu bölümün “b” numaralı kısmında bulanıklaştırma metodlarının seçimi yer almaktadır. Min-Max, Max-Product veya Tsukamoto bulanıklaştırma metodlarından biri seçilebilmektedir. Bu bölümün “c” numaralı kısmında motorun hız hatası üyelik fonksiyonunun sayısı, üyelik fonksiyonu tipi ve ölçekleme faktörünün belirlenmesi bu bölümde yapılmaktadır. Ayrıca motorun hız hatası üyelik fonksiyonunun değerleri grafiksel olarak izlenebilmektedir.



Şekil 4.6. BM denetleyicisi hız hatası üyelik fonksiyonu menüsü

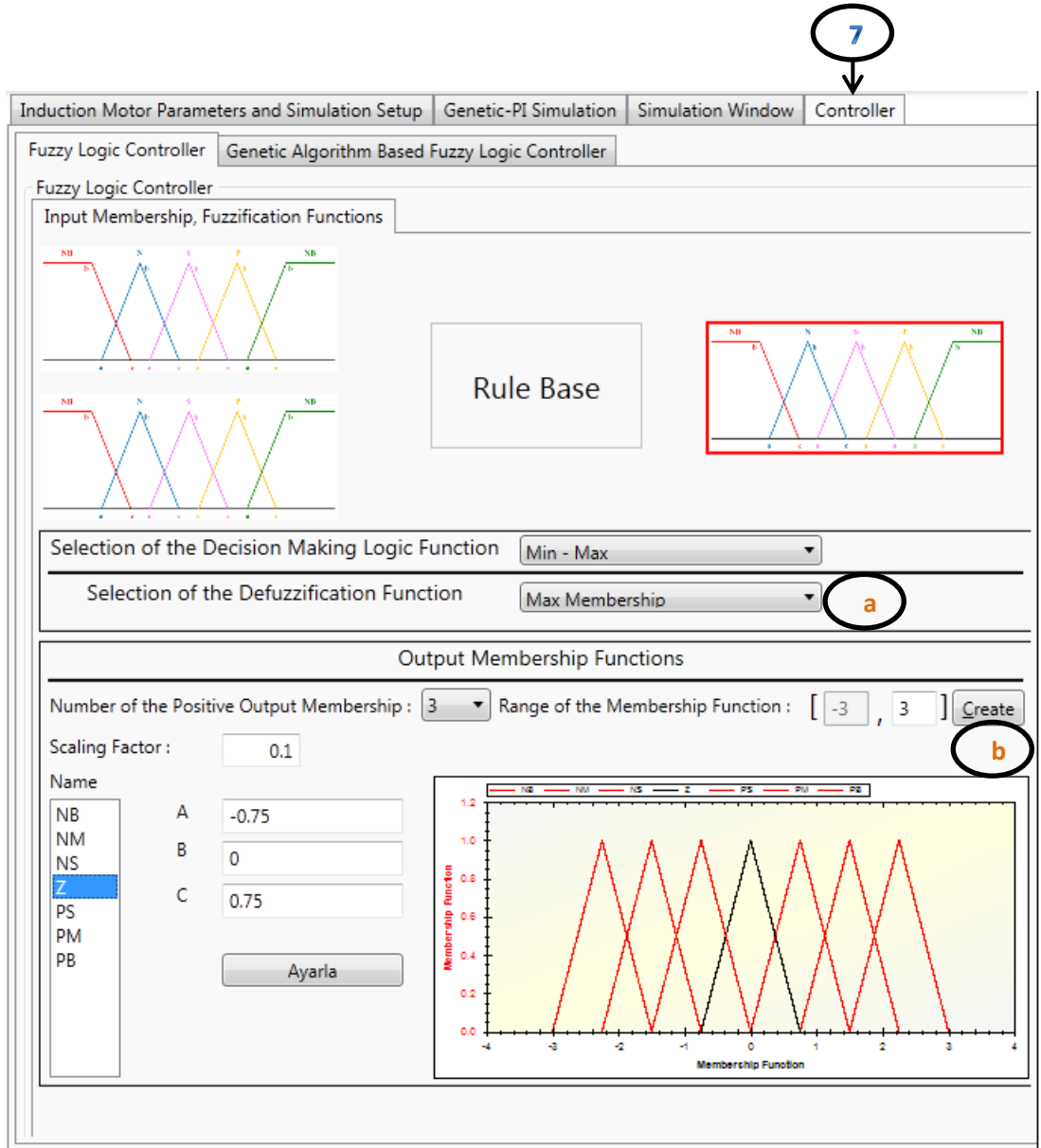
Bu bölümün “d” numaralı kısmında motorun hız hatası değişimi üyelik fonksiyonunun sayısı, üyelik fonksiyonu tipi ve ölçekleme faktörünün belirlenmesi bu bölümde yapılmaktadır. Yine motorun hız hatası değişimi üyelik fonksiyonunun değerleri grafiksel olarak izlenebilmektedir.



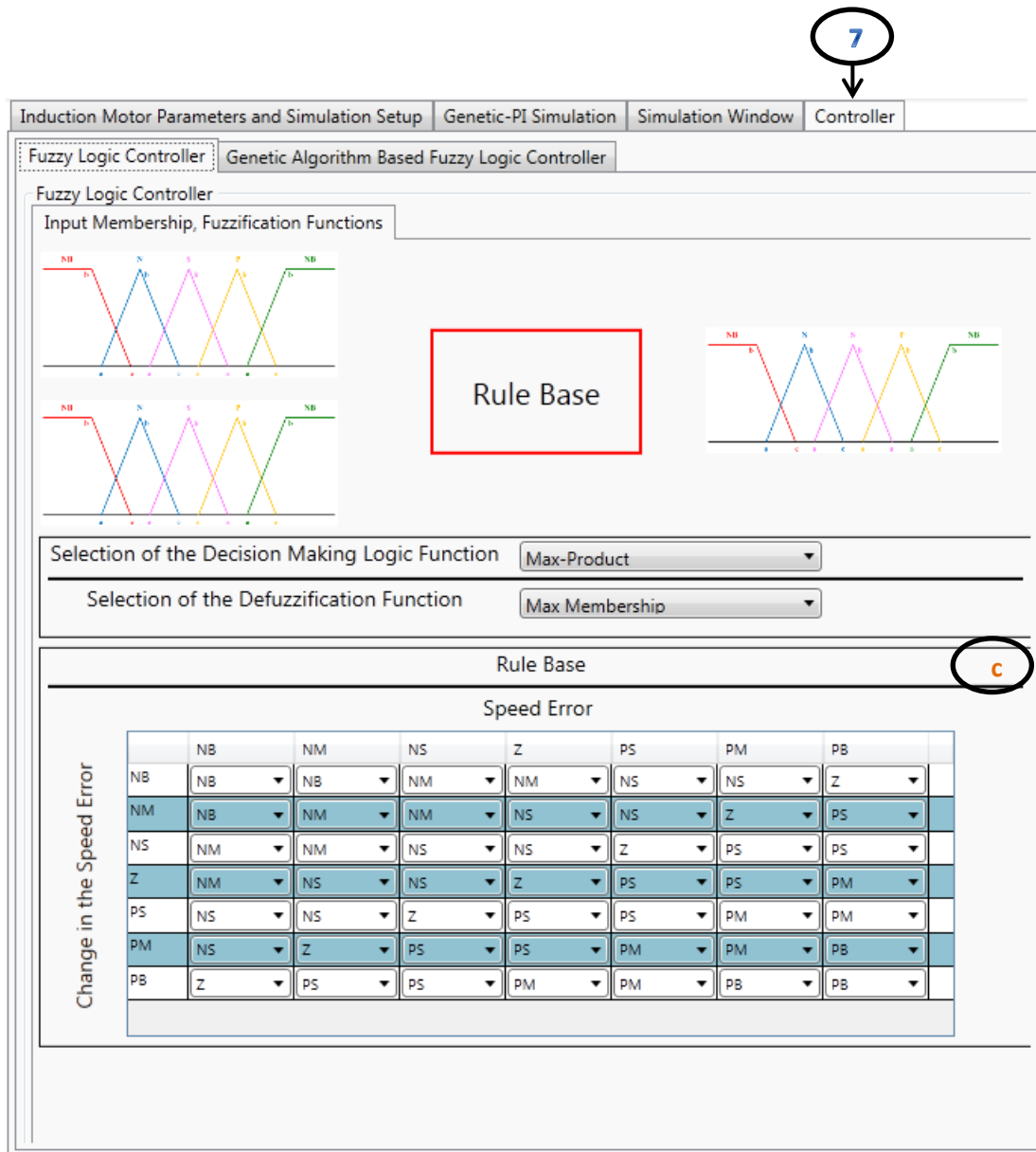
Şekil 4.7. BM denetleyicisi hız hatası değişimi üyelik fonksiyonu menüsü

Şekil 4.8 ve Şekil 4.9’da BM denetleyicisi için ayarlanması gereken durulaştırma fonksiyonunun seçimi, çıkış üyelik fonksiyonu parametreleri ve kural tabanı özellikleri gösterilmiştir. Bu bölümün “a” numaralı kısmında Max Membership, Middle of Maxima, Center of Gravity ve Weighted Average durulaştırma fonksiyonlarından seçim işlemi yapılmaktadır. Bu bölümün “b” numaralı kısmında çıkış üyelik fonksiyonu ayarlanmaktadır. Çıkış üyelik fonksiyonunun ayarlanmasında pozitif çıkış üyelik fonksiyonu sayısı seçilmektedir. Örneğin bu değer üç olarak seçilirse, üç pozitif, üç negatif ve bir de sıfır olmak üzere yedi adet çıkış üyelik fonksiyonu oluşturulur. Üyelik fonksiyonu aralığı da bu

bölümde yapılmaktadır. “Create - Oluştur” düğmesi aracılığıyla çıkış üyelik fonksiyonu oluşturulur. Durulaştırmadan sonra gerçek denetim değerinin hesaplanması için kullanılan ölçeklendirme faktörü de bu bölümde ayarlanmaktadır. Aynı zamanda çıkış üyelik fonksiyonunun değerleri grafiksel olarak izlenebilmektedir. Giriş ve çıkış üyelik fonksiyonları tanımlandıktan sonra kural tabanı ayarı işlemine geçilmektedir. Bu bölümün “c” numaralı kısmında kural tabanının ayarlanmasında çıkış üyelik fonksiyonunun isimleri açılır listeden seçilmektedir.



Şekil 4.8. BM denetleyicisi çıkış üyelik fonksiyonu menüsü



Şekil 4.9. BM denetleyicisi kural tabanı menüsü

#### 4.7. Genetik Tabanlı Bulanık Mantık Denetleyicisi Menüsü

“7” numaralı bölümde Kontrol menüsü GABM denetleyicisi sekmesi genetik tabanlı bulanık mantık kontrolü gerçekleştirmek için tasarlanmıştır. Bu kısımda yer alan “Generation – Nesil”, “Population - Popülasyon”, “Mutation - Mutasyon” ve “Crossing Over - Çaprazlama” gibi GA parametrelerinin değerleri ayarlanmaktadır. GA ile iterasyon sayısına bağlı olarak kromozom yapısı oluşturulmaktadır. GABM denetleyici ayarı için iki basamaklı bir kromozom yapısı tanımlanmıştır. Bu uygulamada kutup atama ve Ziegler ve Nichols

yöntemi ile BM denetleyici hız hatası  $G_e$ , hız hatası değişimi  $G_c$  ve çıkış  $G_u$  ölçeklendirme faktörleri değerleri belirli bir çalışma noktası için ayarlanmış BM denetleyici  $G_e$ ,  $G_c$  ve  $G_u$  katsayıları sırasıyla, 51, 5 ve 0.1 olarak bulunmuştur. GABM denetleyici sistemini gerçekleştirmek için BM denetleyici  $G_e$  hız hatası ölçeklendirme faktörü değeri kutup atama yöntemi,  $G_c$  hız hatası değişimi ölçeklendirme faktörü ve  $G_u$  çıkış ölçeklendirme faktörü değerleri ise Ziegler ve Nichols yöntemi kullanılarak belirlenmiştir. Oluşturulan kromozomun iki değeri  $G_e$  için, ikinci değeri ise  $G_c$  için ve ilk değeri  $G_u$  olarak kodlanmıştır [ $G_e G_c G_u$ ] = {51}. Kodlama yöntemi ondalık sayı ve her parametre ögesi arasında 0-9 arası rakamdan oluşmaktadır. Uzman bilgisinin yeterli olmadığı durumlarda GA kullanılarak GABM denetleyicisi ile oluşturulan popülasyonun üyeleri BM denetleyicisinde olduğu gibi durulaştırmadan sonra gerçek denetim değerinin hesaplanması için kullanılan ölçeklendirme faktörü  $G_e$ ,  $G_c$  ve  $G_u$  katsayıları da bu bölümde ayarlanmaktadır. GA yoluyla belirlenen yüksek uyumluluk değerleri elde edildikten sonra, bu yüksek uyumluluk değerleri içerisinde en uygun olan değer  $G_e$ ,  $G_c$  ve  $G_u$  ölçeklendirme faktörü değerleri, gerçek denetim değerinin GABM denetleyicisi yapısına yerleştirilerek ASM'ye uygulanmaktadır. Gerçekleştirdiğimiz GABM denetleyicisi programının ekran görüntüsü Şekil 4.10'da görülmektedir.

7

Induction Motor Parameters and Simulation Setup Genetic-PI Simulation Simulation Window Controller

Fuzzy Logic Controller Genetic Algorithm Based Fuzzy Logic Controller

Genetic Based Fuzzy Logic Controller

Generation : 1  
Population : 100  
Mutation %% : 0.1  
Crossing Over : 1

Iteration	GE	GC	GU	Fitness
1	17	1	0.7	1.103423E-06
2	43	4	0.3	1.103423E-06
3	52	5	0.2	1.103423E-06
4	33	3	0.3	1.103423E-06
5	89	8	0.9	1.103423E-06
6	43	4	0.3	1.103423E-06
7	98	9	0.8	1.103423E-06
8	55	5	0.5	1.103423E-06
9	99	9	0.9	1.103423E-06
10	67	6	0.7	1.103423E-06
11	92	9	0.2	1.103423E-06
12	44	4	0.4	1.103423E-06
13	44	4	0.4	1.103423E-06
14	79	7	0.9	1.103423E-06
15	52	5	0.2	1.103423E-06
16	92	9	0.2	1.103423E-06
17	28	2	0.8	1.103423E-06
18	11	1	0.1	1.103423E-06
19	67	6	0.7	1.103423E-06
20	45	4	0.5	1.103423E-06
21	69	6	0.9	1.103423E-06
22	75	7	0.5	1.103423E-06
23	71	7	0.1	1.103423E-06
24	54	5	0.4	1.103423E-06
25	86	8	0.6	1.103423E-06
26	62	6	0.2	1.103423E-06
27	33	3	0.3	1.103423E-06

Şekil 4.10. GABM denetleyicisi menüsü



## 5. TARTIŞMA VE SONUÇLAR

Teorik öğrenilen bilgilerin uygulama ile desteklenmesi, öğrenme hızını arttırmakta ve öğrenmenin kalıcı olmasını sağlamaktadır. Donanım üzerine yapılan uygulamalar için pahalı laboratuvar sistemlerine gereksinim duyulmaktadır. Bu sebeple, tamamen eğitim destek olarak geliştirilen daha düşük maliyetli ve uygun öğrenme ortamı sağlayan simülatör ya da eğitim seti programları kullanılmaktadır. Bilişim ortamında ücretli ya da ücretsiz olarak GA, BM, ve GABM denetleyiciler ilgili simülasyon programları mevcuttur. Fakat bu simülasyon programları sınırlı sistemler için geliştirilmiş olup elektrik makinelerinin uygulama alanları için uygun değildir. En yaygın olarak kullanılanlarından birisi, MathworksInc. tarafından geliştirilen MATLAB/Simulink programıdır. Bu yazılım paketi ile modelleme ve kontrol ile ilgili birçok tasarım yapılabilir de kullanması belirli bir bilgi birikimi ve deneyim gerektirmektedir.

Endüstriyel uygulamalarda bağlaşıklık indüktörlü üç-seviyeli eviricinin kullanımı sırasında en iyi PI katsayılarının elde edilmesi için yapay zekâ uygulamalarına ihtiyaç duyulmaktadır. Yapay zekâ tekniklerinden olan yapay sinir ağı, BM veya GA kullanılarak en iyi PI katsayısı elde edilebilir. Asenkron motorun hız denetiminde klasik PI, PID denetleyici, Genetik PI, BM ve GABM denetleyicilerden yararlanılmıştır. Yapılan çalışmada BM ve GABM denetleyici ile daha düzgün çıkış akımları ve hız kontrolü sağlanabilmektedir.

Yapılan sistemin amacı öğrencilerin laboratuvar ortamında yaptıkları ASM'nin hız kontrolü deneylerinin, zaman ve mekandan bağımsız olarak simülasyonunun yapılabilmesi için bir simülasyon programı tasarlamaktır. Çalışma sonucunda PI, PID, Genetik-PI, BM, GABM denetleyicileri ile bağlaşıklık indüktörlü üç-seviyeli evirici temelli asenkron motorun hız kontrolü için, simülasyon programı hazırlanmıştır. Hazırlanan simülasyon Visual Studio 2012 IDE' sinde WPF alt yapısı kullanılarak, C# programlama dili ile yazılmıştır. Simülasyon esnek bir yapıya sahip olup kullanımı kolay bir grafiksel arayüze sahiptir. Simülasyon programı ile ASM'nin, eviricinin ve denetleyicilerin

parametreleri değiştirilerek sistemin farklı çalışma durumlarındaki tepkisi grafikler yardımıyla izlenebilmektedir. Simülasyon Türkçe ve İngilizce olarak kullanılabilir.

Simülasyon programının hazırlanması 8 aşamada gerçekleştirilmiştir. İlk aşamada ASM'nin matematiksel modeli araştırılmış ve bulunan model, programa aktarılmıştır. İkinci aşamada bağlaışık indüktörlü üç-seviyeli eviricinin çalışması araştırılıp gerilim seviyeleri belirlenmiştir. Üçüncü aşamada uzay vektör DGM araştırılarak, DGM'nin bağlaışık indüktörlü üç-seviyeli evirici ile birlikte kullanımı programlanmıştır. Dördüncü aşamada klasik PI ve PID denetleyiciler ile ASM'nin hızı kontrol edilmiştir. Beşinci aşamada GA destekli Genetik PI denetleyicisi araştırılıp programa aktarılmıştır. Altıncı aşama BM denetleyicisinin yapısı araştırılıp aynı zamanda yazılımsal olarak gerçekleştirilmiştir. Yedinci aşamada denetleyici tekniklerinden GABM denetleyici yapısı araştırılıp, bütün aşamalar tek bir arayüz üzerinde birleştirilip programa aktarılmıştır.

Sonuç olarak Şekil B.1'de PI, Şekil B.2'de PID, Şekil B.3'de Genetik-PI, Şekil B.4'de BM ve Şekil B.5'de ise GABM denetleyicileri için yapılan örnek simülasyonun, hız, akım ve gerilim grafikleri verilmiştir. Grafiklerde görülen siyah renk referans hızı, kırmızı renk ise gerçek hızı ifade etmektedir. Yapılan simülasyonda BM denetleyicisi, GABM denetleyicisinin sisteme uyguladığı kontrol değerine göre en uygun hale getirilmiştir. Bu nedenle BM ve GABM denetleyicilerinin hız grafikleri benzerdir. Şekil B.4 ve Şekil B.5 incelendiğinde BM ve GABM denetleyicilerinin klasik PI, PID ve Genetik-PI tabanlı denetleyicilere göre, gerçek hızın referans hıza daha hızlı ulaştığı görülmektedir.

Grafiklerde simülasyonu yapılan ASM'ye uygulanan referans hızlar görülmektedir. Referans hızlar için hızlanma zamanı 0.6 saniye olarak seçilmiştir. Ayrıca 0.8 - 0.9 saniyeleri arasında motora 2 nm'lik bir yük uygulanmıştır.

## KAYNAKLAR

- Adewuya, A.A., 1996. New Methods in Genetic Search with Real Valued Chromosomes, Massachusetts Institute of Technology, M.S. Thesis, Cambridge.
- Akar, M., 2005. Bulanık Mantık Yöntemiyle Bir Servo Motorun Kontrolü ve Geleneksel Yöntemlerle Karşılaştırılması, Marmara Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Akçayol, M.A., Cetin, A., Elmas, Ç., 2002. An Educational Tool for Fuzzy Logic Controlled BDCM, 33-42.
- Angeline, P.J., 1995. Evolution Revolution: An introduction to the Special Track on Genetic and Evolutionary Programming, IEEE Expert Intelligent Systems and Their Applications, 6-10.
- Altınten, A., 2001. Genetik Algoritma Kullanarak Fuzzy Kontrol Yöntemi ile Bir Stiren Polimerizasyon Reaktörünün Kontrolü, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- Arslan, A., Kaya, M., 2001. Determination of Fuzzy Logic Membership Functions Using Genetic Algorithms, Elsevier Fuzzy Sets and Systems, 297-306.
- Ashrafzadeh, F., 1996. Genetic Based Auto-Design of Fuzzy Controllers, The University of Galgary, Ph.D. Thesis, Alberta.
- Asker, M.E., Özdemir, M., Bayındır, M.İ., 2009. Sinüsoidal DGM İle Uzay Vektör DGM Yöntemlerinin Karşılaştırmalı İncelemesi, 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), 7s, Karabük.
- Aydoğdu Ö., Akkaya R., 2005. DSP Based Fuzzy Control of a Brushless DC Motor Without Position and Speed Sensors. Proc. of 4<sup>th</sup> Int. Advanced Technologies Symposium, 182-187, September 28-30, Konya.
- Aydoğdu, Ö., 2006. Fırçasız Doğru Akım Motorlarının Genetik Tabanlı Bulanık Denetleyici ile Sensörsüz Kontrolü, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 171s, Konya.
- Baba, A.F., 1995. İTÜ Triga Mark-II Reaktörünün Bulanık Kontrolü, Marmara Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- Bhagwhat, P.M., Stefanovic, R.V., 1983. Generalized Structure of a Multilevel PWM Inverter, IEEE Transactions On Industry Applications, 1057-1069.
- Bingöl, O., 2005. Yapay Sinir Ağı ile Modellenen Alan Yönlendirmeli Bir Asenkron Motorun Üç Seviyeli Evirici İle Hız Denetimi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 178s, Ankara.

- Bingöl, O., Paçacı, S., 2010. A Virtual Laboratory for Fuzzy Logic Controlled DC Motors, International Journal of Physical Sciences, 2493-2502.
- Bircenas, E., Ramirez, S., Cardenas, V., Echavarria, R., 2002. Cascade Multi-Level Inverter with Only One DC Source, VIII. IEEE International Power Electronics Congress, CIEP 2002, 171-176.
- Bodurođlu, T., 1981. Elektrik Makinaları Dersleri, Beta Yayım Dađıtım A. Ő., İstanbul.
- Bose, B., 1997. Power Electronics and Variable Frequency Drives, IEEE Press Marketing, 639p, New York.
- Bose, B.K., 2002. Modern Power Electronics and AC Drives, Prentice Hall, 737p, New Jersey.
- Bousquest-Monge, B., Bordonau, J., Boroyevich, D., Somavilla, S., 2004. The Nearest Three Virtual Space Vector PWM A Modulation for the Comprehensive Neutral-Point Balancing in the Three-Level NPC Inverter, IEEE Power Electronics Letters, 11-15.
- Celanovic, N., 2000. Space Vector Modulation and Control of Multilevel Converters, Doctor of Philosophy in Electrical Engineering and Computer Engineering, 11-18, Blackburg, Virginia.
- Celanovic, N., Boroyevich, D., 2000. A Comprehensive Study of Neural- Point Voltage Balancing Problem in Three-Level Neutral-Point-Clamped Voltage Source PWM Inverters, IEEE Transactions On Power Electronics, 242-249.
- Celanovic, N., Boroyevich, D., 2001. A Fast Space-Vector Modulation Algorithm for Multilevel Three-Phase Converters, IEEE Transactions On Industry Applications, 637-641.
- ÇunkaŐ, M., 2004. Elektrik Motorlarında Genetik Algoritma ile Tasarım Optimizasyonu, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, Konya.
- Çelik, H., 2004. Uzay Vektör Darbe Genişlik Modülasyonu ile Üç Fazlı Asenkron Motorun Hız Kontrolü, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Elazığ.
- Davis, L., 1991. The Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York.
- Del Toro, V., 1990. Basic Electric Machines, Prentice-Hall, Englewood Cliffs.

- Dessouky, A., 2002. Adaptive Fuzzy Control for High Performance Induction Motor Drives, Royal Military College of Canada, Ph. D. in Electrical Engineering, Canada.
- Donescu, V., Neacsu, D.O., Griva, G., Profumo, F., 1996. A Systematic Design Method for Fuzzy Controller for Brushless DC Motor Drives, Proc. of the 27th. Annual IEEE Power Electronics Specialists Conf., 689-694, Baveno.
- Drury, W., Grant, D., 1994. Variable Speed Drives The Future, Power Engineering Journal, 27-34.
- Ekren, O., 2009. Bir Soğutma Grubunda Kompresör Hızının ve Elektronik Genleşme Vanasının Bulanık Mantık Algoritma ile Kontrolü. Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 154s, İzmir.
- Elmas, Ç., 2007. Yapay Zeka Uygulamaları. Seçkin Yayıncılık, 425s, Ankara.
- Elmas, Ç., Yiğit, T., 2008. Eviriciler için Bilgisayar Temelli Eğitim Laboratuvarı. Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Dergisi, 21, 128-140.
- Erdal, H., 1998. Bir İklimlendirme Odasının Bulanık Mantık Yöntemi ile Kontrolü, Marmara Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- Ewanchuk, J., 2008. A Six-Switch Multi-level Inverter Topology for Three Phase High Speed Machine Applications, MSc Thesis, Department of Electrical and Computer Engineering, 36-53, Edmonton, Alberta.
- Ewanchuk, J., Salmon, J., Knight, A., 2008. Performance of a High Speed Motor Drive System Using a Novel Multi-Level Inverter Topology, Industry Applications, IEEE Transactions.
- Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Publishing Inc., ABD.
- Gökbulut, M., Bal, C., Dandil, B., 2006. A Virtual Electrical Drive Control Laboratory: Neuro-Fuzzy Control of Induction Motors, Computer Applications in Engineering Education, 211-221.
- Gülez, K., 1999. Asenkron Motorun DSP (Sayısal İşaret İşleyici) Tabanlı Bir Kontrol Sistemi Kullanarak YSA (Yapay Sinir Ağları) İle Performansının Arttırılması, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- Haupt, R.L., Haupt, S.E., 1998. Practical Genetic Algorithms, A Willey Interscience Publication, USA.

- Holtz, J., 1994. Pulse width Modulation for Electronic Power Conversion, Proceedings of the IEEE, 1194-1214.
- Kallel, L., Schoenaur, 1997. Alternative Random Initialization in Genetic Algorithm, Proceeding of The Seventh International Conference on Genetic Algorithms, Morgan Kaufman, ABD.
- Karaođlan, Ö.G., 2007. Kontrol Sistemleri İçin Bulanık PID Kontrolörlerin Genetik Algoritmalar Yardımıyla Ayarlanması, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 103s, İstanbul.
- Kayaalp, K., 2007. Asenkron Motorlarda Veri Madenciliđi ile Hata Tespiti, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 86s, Isparta.
- Kerem, A., 2012. 6 Anahtarlı 3-Seviyeli Evirici Tasarımı, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 111s, Ankara.
- Kinzel, J., Klawonn, F., Kruse, R., 1994. Modifications of Genetic Algorithms for Designing and Optimizing Fuzzy Controllers, Proc. First IEEE Conf. on Evolutionary Computation, 28-33, Orlando.
- Knight, A.M., Ewanchuk, J., Salmon, J. C., 2008. Coupled Three-Phase Inductors for Interleaved Inverter Switching, Magnetics, IEEE Transactions on.
- Koca, Z., 2006. Üç Fazlı Asenkron Motorların Yapay Sinir Ağlarıyla Vektör Esaslı Hız Kontrolü, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 61s, Kahramanmaraş.
- Krause, P.C., Wasynczuk, O., Sudhoff, S.D., 1995. Analysis of Electrical Machinery, Institute of Electrical and Electronics Engineers, IEEE Press Marketing, 133-150, New York.
- Krishnan, R., 2001. Electrivity Motor Drives Modeling Analysis and Control, PrenticeHall, 652p, New Jersey.
- Lai, J.S., Peng, F.Z., 1996. Multilevel Converters – A New Breed of Power Converters, IEEE Transactions on Industry Applications, 509-517.
- Lee, C.K., Pang, W.H., 1994. A Brushless DC Motor Speed Control System Using Fuzzy Rules, Proc. of the 5th. IEE Int. Conf. on Power Electronics and Variable Speed Drives, 399p, London.
- Li, L., 1999. Optimal PWM Techniques in Voltage Source Inverters, Polytechnic University, Ph. D., 117p, New York.
- Lipo, T.A., Novotny, D.W., 1995. Dynamics and Control of AC Drives, McGrawHill, 352p, New York.

- Manjrekar, M.D., Steimer, P.K., Lipo, T.A., 2000. Hybrid Multilevel Power Conversion System: a Competitive Solution for High-Power Applications, IEEE Transactions on Industry Applications, 834-841.
- Marchesoni, M., Mazzuchalli, M., Tenconi, S., 1990. A Non Conventional Power Converter for Plasma Stabilization, IEEE Trans. On Power Electronics.
- Matsui, K., Kawata, Y., Ueda, F., 2000. Application of Parallel Connected NPC-PWM Inverters with Multi-Level Modulation for AC Motor Drive, Power Electronics, IEEE Transactions 901-907.
- Meynard, T.A., Foch, H., 1992. Multi-Level Conversion: High Voltage Choppers and Voltage-Source Inverters, IEEE Power Electronics Specialists Conference, 397-403.
- Michalewicz, Z., 1994. Genetic Algorithms + Data structures = Evolution Programs, 2nd ed. Springer-Verlag, New York.
- Mondal, S.M., Pinto, J.O., Bose, B.K., 2002. A Neural Network Based Space Vector PWM Controller for a Three-Level Voltage Fed Inverter Induction Motor Drive. IEEE Transactions on Industry Applications, 660-669.
- Nabae, A., Takahashi, I., Akagi, H., 1981. A New Neutral Point Clamped PWM Inverter, IEEE Transactions on Industry Applications, 518-523.
- Paçacı, S., 2011. Yapay Sinir Ağları, Bulanık Mantık ve Sinirsel Bulanık Denetleyiciler ile Asenkron Motorların Hız Denetimi için Simülatör Tasarımı, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 105s, Isparta.
- Pierre, D.A., 1992. Optimization in McGraw-Hill Encyclopedia of Science and Technology, McGraw-Hill, New York.
- Radcliff, N.J., 1991. Formal Analysis and Random Respectful Recombination, In Proc. of Fourth Int. Conf. on Genetic Algorithms, Morgan Kaufman, San Mateo, CA.
- Rodriguez, J., Lai, J.S., Peng, F.Z., 2002. Multilevel Inverters: a Survey of Topologies, Controls, and Applications, IEEE Transactions on Industrial Electronics, 724-738.
- Saygın, A., 2004. Asenkron Motor Hız Kontrolünün Matris Konverter ile Hızının Denetlenmesi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 153s, Ankara.
- Şafak, C., 2005. Pnömatik Motorun Hız Kontrolü için Genetik-Bulanık Kontrolör Tasarımı, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 125s, İstanbul.

- Salmon, J., Ewanchuk, J., Knight, A., 2009. PWM Inverters Using Split-Wound Coupled Inductors, Industry Applications, IEEE Transactions, 2001-2009.
- Şekkeli, M., Yıldız, C., Özçalık, H.R., 2007. Bulanık Mantık ve PI Denetimli DC-DC Konvertör Modellenmesi ve Dinamik Performans Karşılaştırması, 4. Otomasyon Sempozyumu, Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kurupelit Yerleşkesi.
- Şen, Z., 2001. Bulanık Mantık ve Modelleme İlkeleri, Bilge Kültür Sanat, 172s, İstanbul.
- Tan, G.V., Hu, X., 1996. on Designing Fuzzy Controllers Using Genetic Algorithms, Proc. of 5th IEEE Int. Conf. on Fuzzy Systems, 905-911, New Orleans.
- Tang, K.S., Man, K.F., Chen, G., Kwong, S., 2001. An Optimal Fuzzy Controller, IEEE Transactions on Industrial Electronics, 757-765.
- Tong, C.P., 2000. Symbiosis of Fuzzy Systems and Genetic Algorithms for Modelling and Control, the Hong Kong Polytechnic University, Ph. D. İn Engineering, Hong Kong.
- Topuz, V., 2002. Bulanık Genetik Proses Kontrolü, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- Tuncay, N., Gökaşan, M., Boğosyan, S., 2003. Güç Elektroniği Çeviriciler, Uygulamalar ve Tasarım, Mohan, N., Undeland, T.M., Robbins, W., Literatür Yayıncılık, 896s, İstanbul.
- Tuncer, S., 2004. Uzay Vektör Darbe Genişlik Modülasyonu Kullanan Beş Seviyeli İnverter Tasarımı ve Uygulaması, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 56s, Elazığ.
- Tuncer, S., 2009. Çok Seviyeli Eviricilerde Taşıyıcı Dalga Şekli Değişimlerinin Çıkış Gerilimi Üzerine Etkilerinin İncelenmesi, Gazi Üniversitesi Müh. Mim. Fak. Dergisi, 613-628.
- Uygur, A.F., 2002. Bir Asenkron Motor Hız Denetim Sisteminin Neuro-Fuzzy Yöntemlerle Gerçekleştirilmesi, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kahramanmaraş.
- Üstün, S.V., 2001. Asenkron Motorun DSP (Sayısal İşaret İşleyici) Temelli Kontrolünde Yapay Sinir Ağları ve Genetik Algoritmalar Kullanılarak PI Katsayılarının Optimizasyonu, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- Vafakhah, B., Masiala, M., Salmon, J., Knight, A.M., 2009. Space-Vector PWM for Inverters with Split-Wound Coupled Inductors, in Electric Machines and Drives Conference, IEMDC '09. IEEE International.



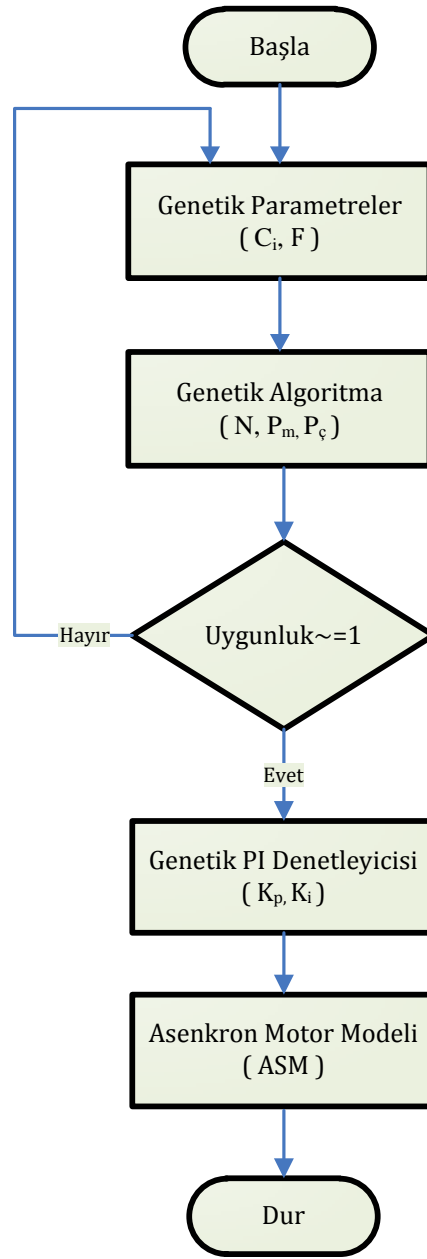
- Vafakhah, B., 2010. Multilevel Space Vector PWM for Multilevel Coupled Inductor Inverters, Department of Electrical and Computer Engineering, Edmonton, Ph.D., 133p, Alberta.
- Wei, S., Wu, B., Li, F., Liu, C., 2003. A general Space Vector PWM Control Algorithm for Multilevel Inverters, A Lie Power Electronics Conference and Exposition, APEC'03. Eighteenth Annual IEEE, 562-568.
- Wright, A., 1991. Genetic Algorithms for Real Parameter Optimization, Foundations of Genetic Algorithms 2. Rawlins G.J.E., 205-218, Morgan Kaufman, San Mateo, CA.
- Wu, H., He, X., 2000. Research on PWM Control of a Cascade Multilevel Converter, Power Electronics and Motion Control Conference, Proceedings, PIEMC The Third International, 1099-1103.
- Yamanaka, K., Hava, A.M., Kirino, H., Tanaka, Y., Koya, N., Kume, T., 2002. A Novel Neutral Point Potential Stabilization Technique Using the Information of Output Current Polarities and Voltage Vector, IEEE Transactions on Industry Applications, 1572-1580.
- Yıldız, C., 2008. Genetik Algoritma Destekli Bulanık Denetim Kullanarak Vektör Esaslı Asenkron Motor Kontrolü, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 78s, Kahramanmaraş.
- Yiğit, T., Elmas, Ç., 2008. An Educational Tool For Controlling SRM. Computer Applications in Engineering Education, 268-279.
- Yun, Y., Gen, M., Seo, S., 2003. Various Hybrid Methods Based on Genetic Algorithm with Fuzzy Logic Controller, Journal of Intelligent Manufacturing, 401-419.
- Zadeh, L.A., 1965. Fuzzy Sets. Informat and Control, 338-353.
- Zhang, J., 1995. High Performans Control of a Three-level IGBT Inverter Fed AC Drive, Industry Applications Conference, Thirtieth IAS Annual Meeting Record of the IEEE, 22-28.
- Zhang, H., Jouanne, A.V., Dai, S., Wallace, A.K., Wang, F., 2000. Multilevel Inverter Modulation Schemes to Eliminate Common-Mode Tages, IEEE Transactions on Industry Applications, 1645-1653.

## **EKLER**

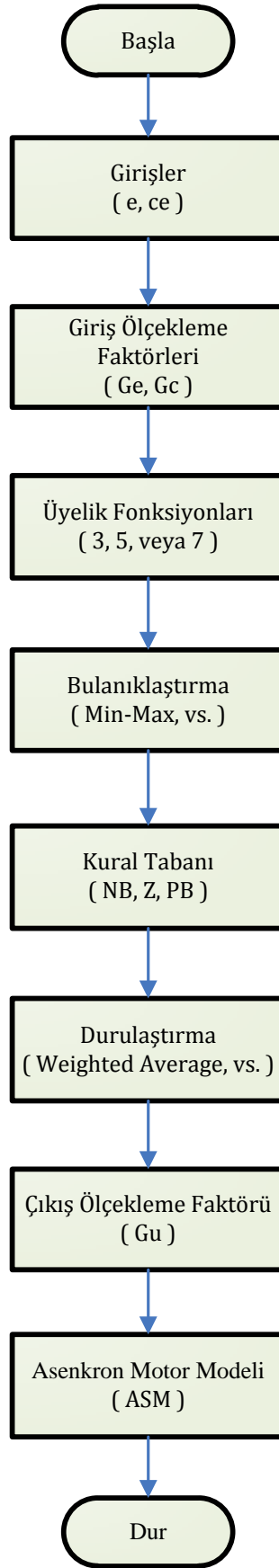
**EK A.** Şekiller

**EK B.** Grafikler

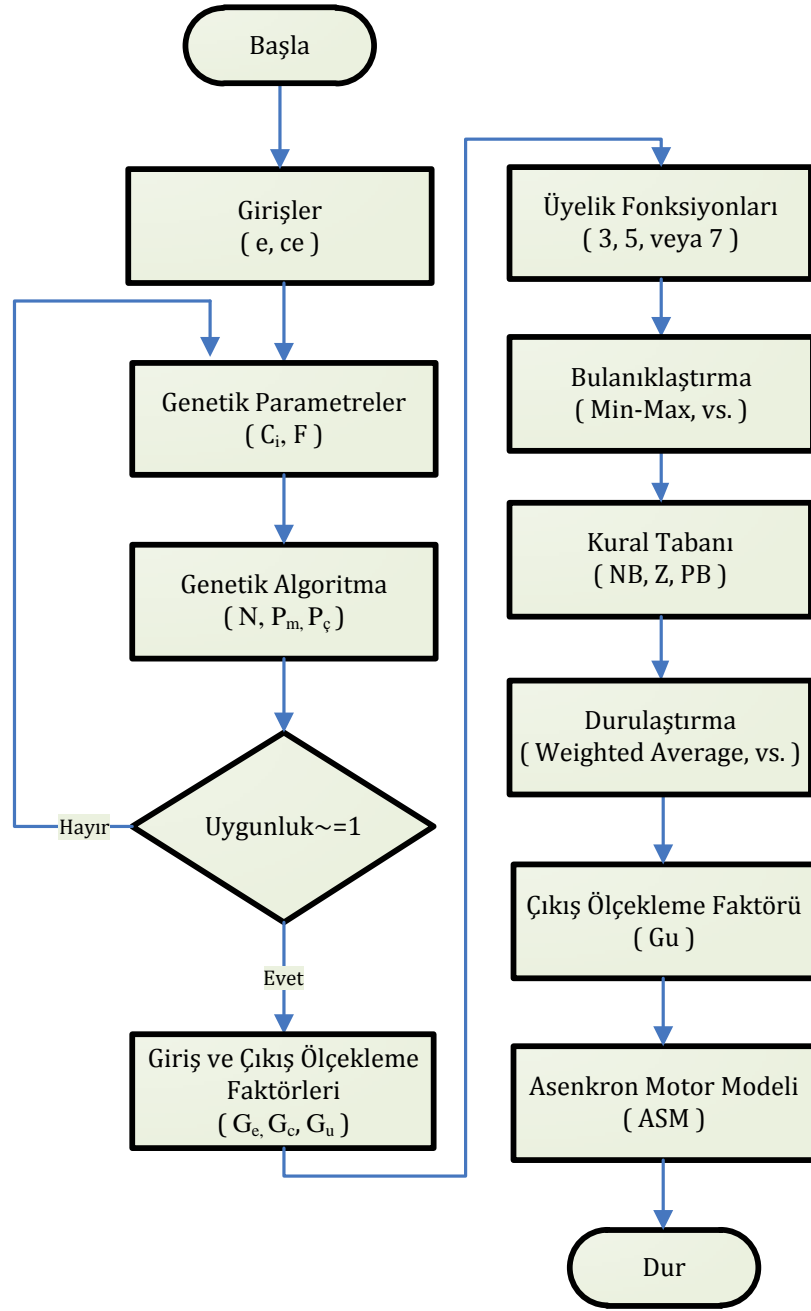
## EK A. Şekiller



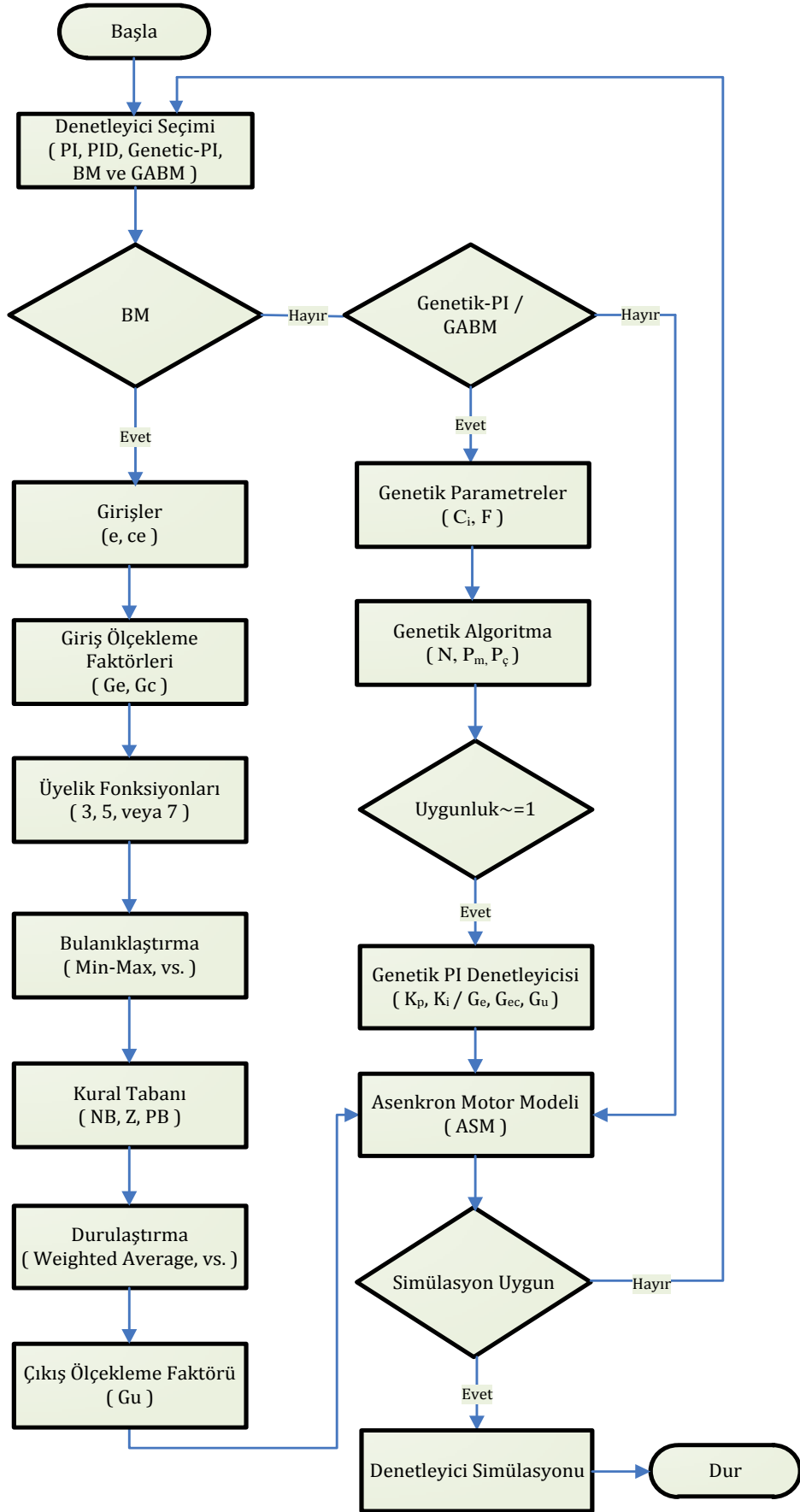
Şekil A.1. Genetik-PI denetleyicisi akış diyagramı



Şekil A.2. BM denetleyicisi akış diyagramı

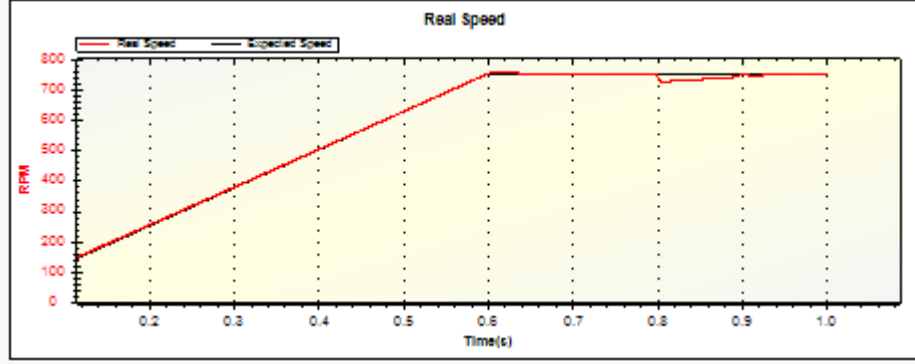


Şekil A.3. GABM denetleyicisi akış diyagramı

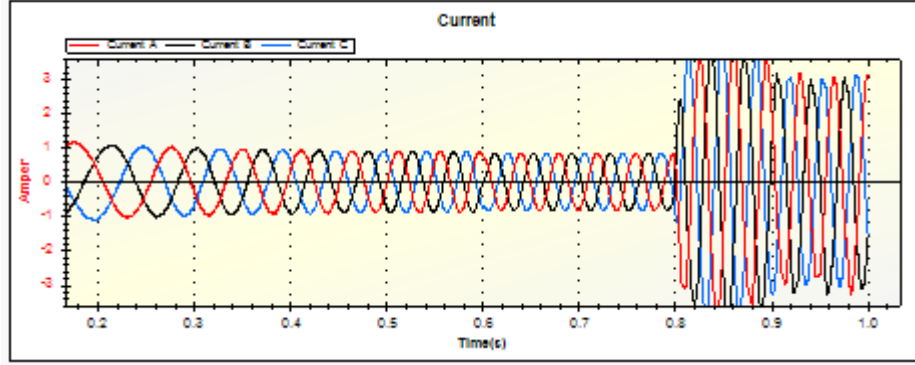


Şekil A.4. Simülasyon programının akış diyagramı

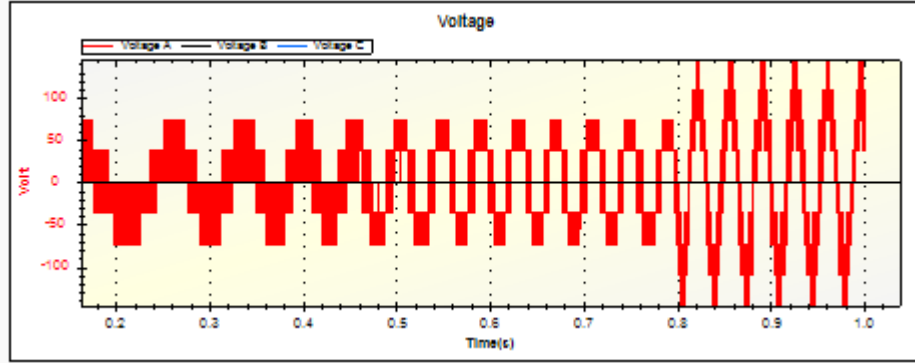
## EK B. Grafikler



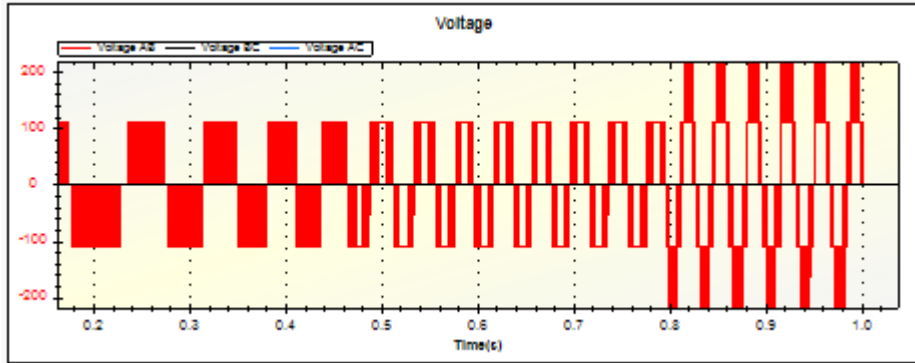
(a)



(b)

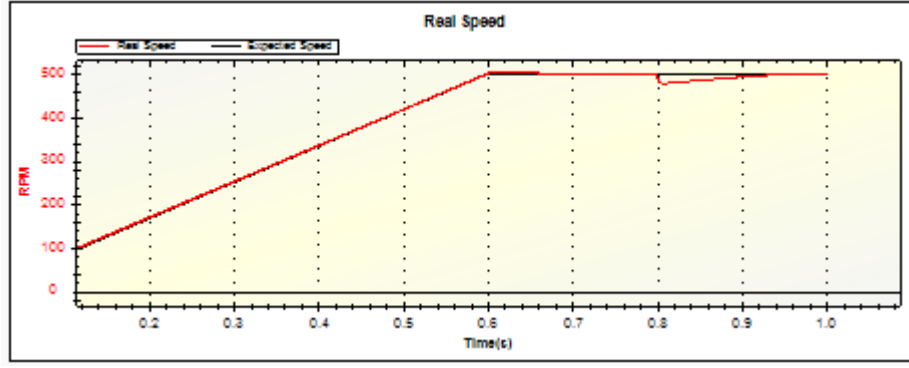


(c)

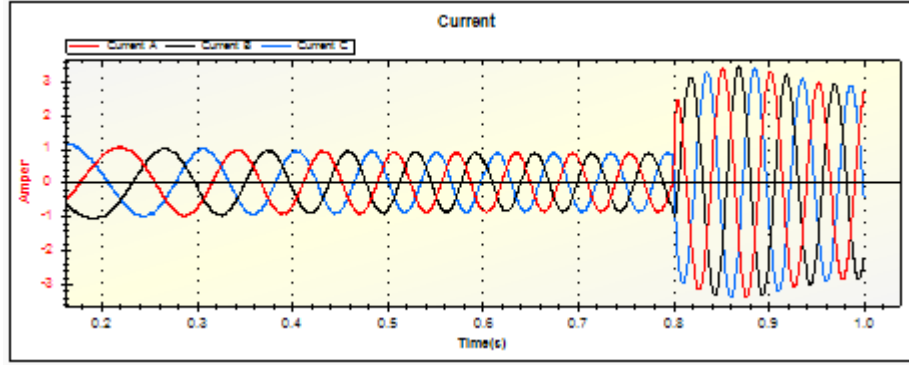


(d)

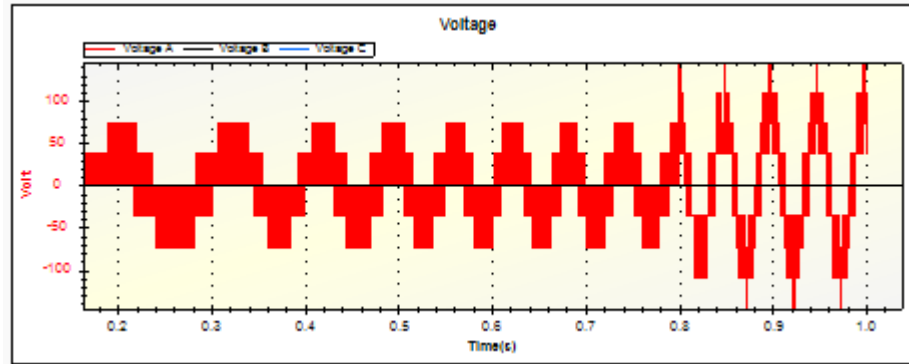
Şekil B.1. a) PI denetleyici için hız, b) akım, c) faz gerilimleri ve d) fazlar arası gerilimleri simülasyon grafik ekranı



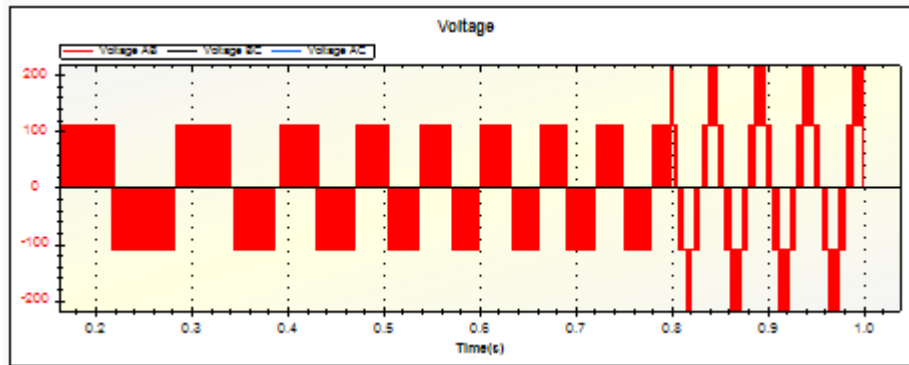
a)



b)



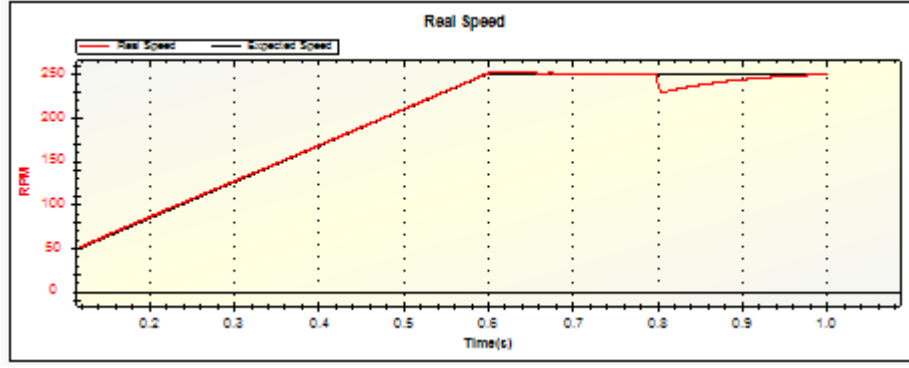
c)



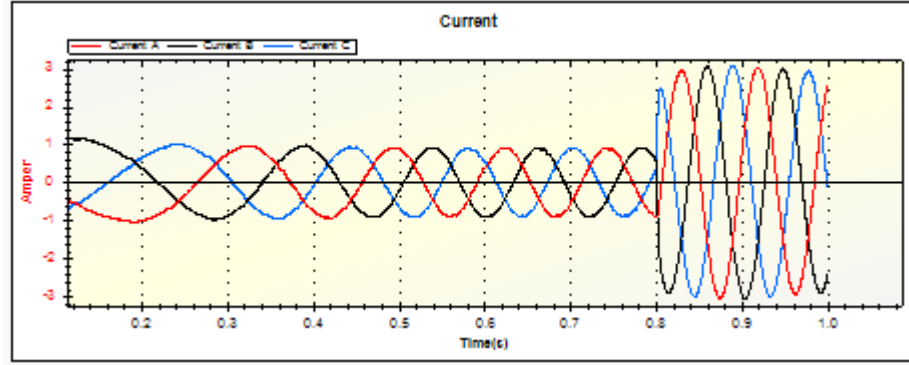
d)

Şekil B.2. a) PID denetleyici için hız, b) akım, c) faz gerilimleri ve d) fazlar arası gerilimleri simülasyon grafik ekranı

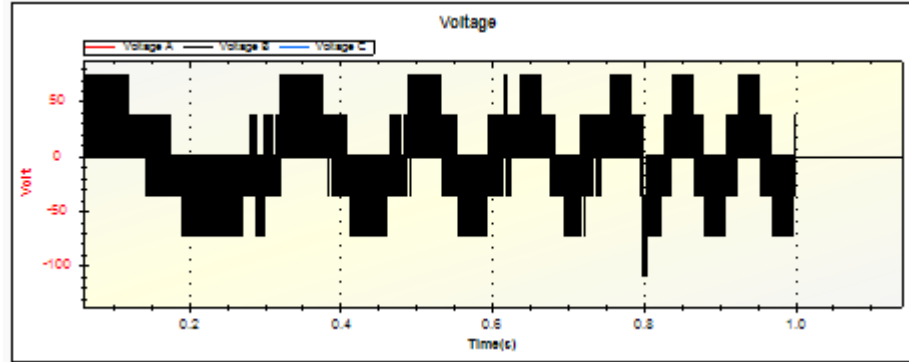




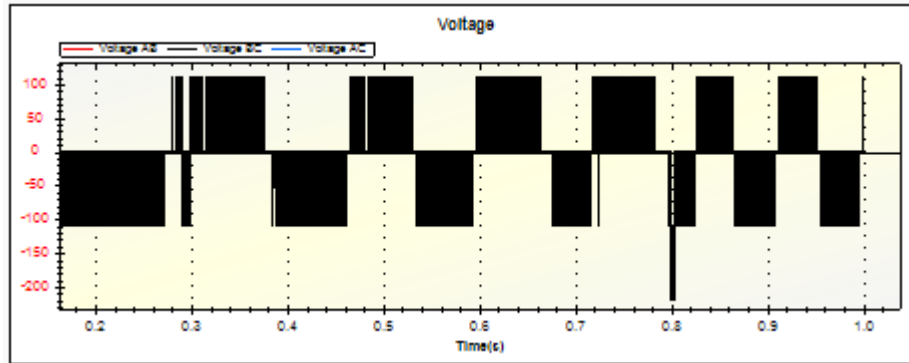
a)



b)

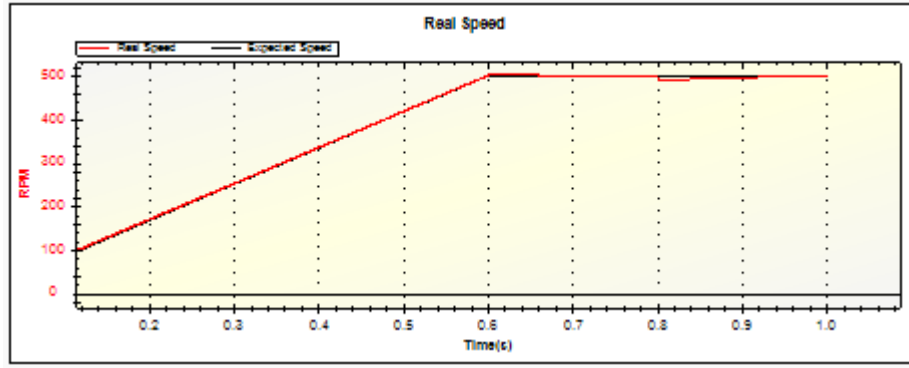


c)

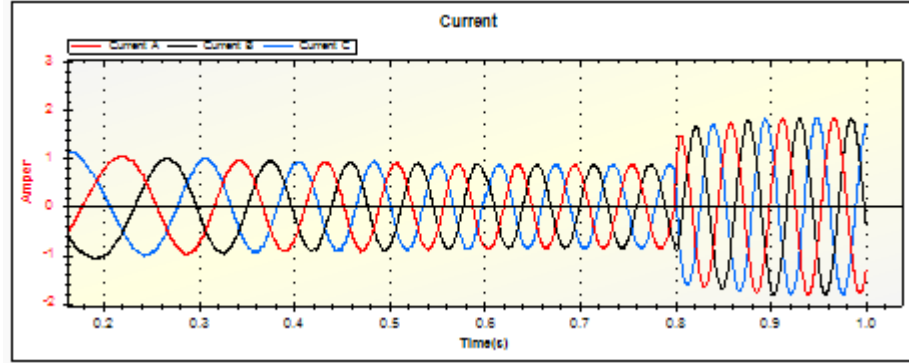


d)

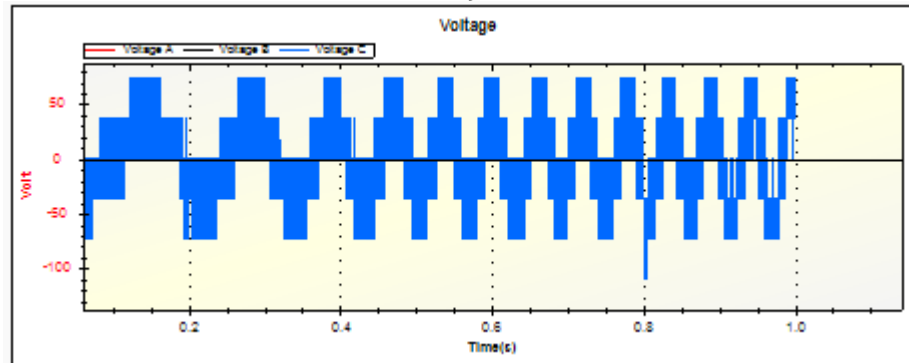
Şekil B.3. a) Genetik-PI denetleyici için hız, b) akım, c) faz gerilimleri ve d) fazlar arası gerilimleri simülasyon grafik ekranı



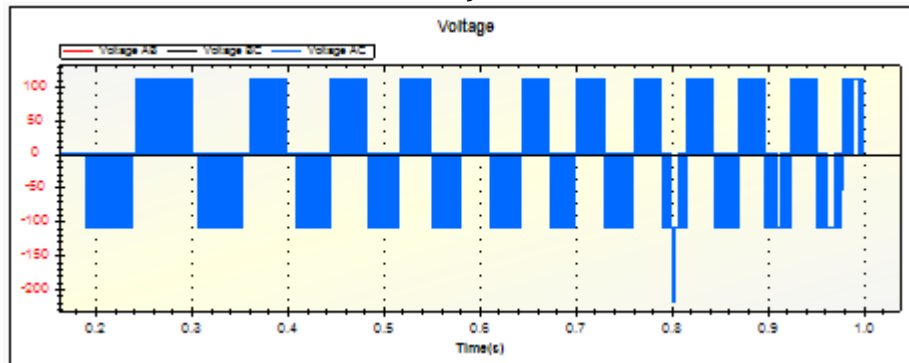
a)



b)

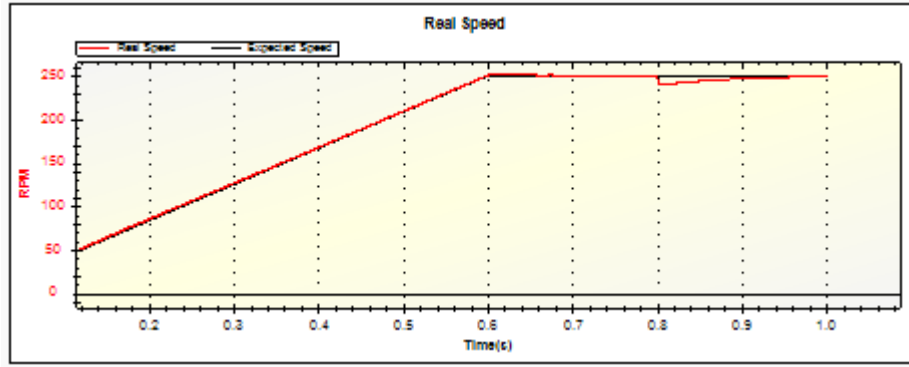


c)

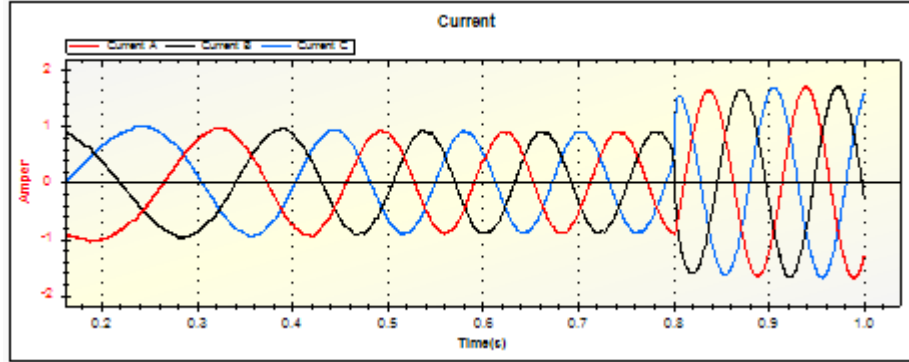


d)

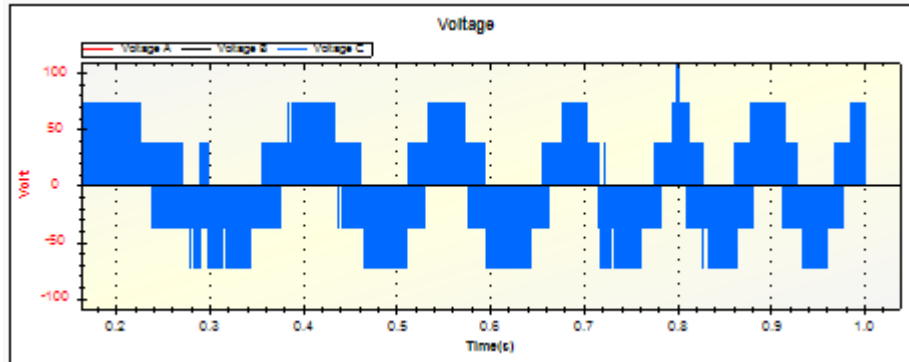
Şekil B.4. a) BM denetleyici için hız, b) akım, c) faz gerilimleri ve d) fazlar arası gerilimleri simülasyon grafik ekranı



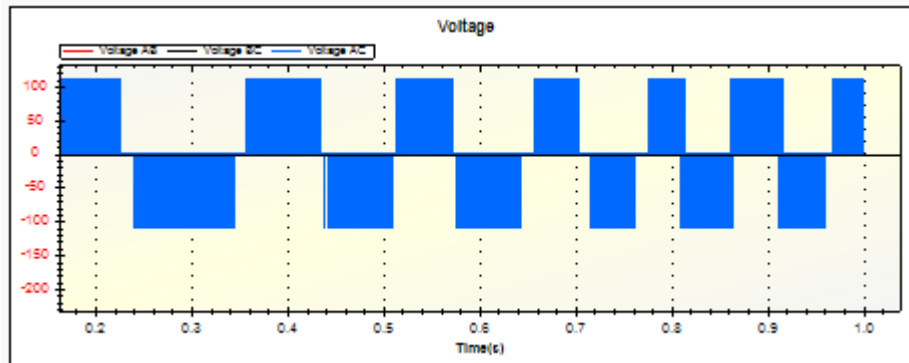
a)



b)



c)



d)

Şekil B.5. a) GABM denetleyici için hız, b) akım, c) faz gerilimleri ve d) fazlar arası gerilimleri simülasyon grafik ekranı

## ÖZGEÇMİŞ

Adı Soyadı : Alphan MAT  
Doğum Yeri ve Yılı : Konya, 1984  
Medeni Hali : Bekâr  
Yabancı Dili : İngilizce  
E-posta : alphanmat@yahoo.com



## Eğitim Durumu

Lise : Konya Fatih Teknik ve Endüstri Meslek Lisesi Bilgisayar Yazılım, 2001  
Ön Lisans : Selçuk Üniversitesi, Teknik Bilimler Meslek Yüksek Okulu, Bilgisayar Programcılığı, 2005  
Lisans : Süleyman Demirel Üniversitesi, Teknik Eğitim Fakültesi, Bilgisayar Sistemleri Öğretmenliği, 2008  
Yüksek Lisans : Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Bilgisayar Eğitimi, 2013

## Mesleki Deneyim

Bilfaks Bilişim Hizmetleri	2004-2005
Selçuk Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi	2007-2008
Konya Fatih Teknik ve Endüstri Meslek Lisesi Bilişim Teknolojileri Öğretmeni	2009-..... (halen)