T.C
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FIRÇASIZ DOĞRU AKIM MOTORLARININ GENETİK TABANLI BULANIK DENETLEYİCİ İLE SENSÖRSÜZ KONTROLÜ

Ömer AYDOĞDU

DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI
Konya, 2006

T.C

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

FIRÇASIZ DOĞRU AKIM MOTORLARININ GENETİK TABANLI BULANIK DENETLEYİCİ İLE SENSÖRSÜZ KONTROLÜ

Ömer AYDOĞDU

DOKTORA TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez 21/07/2006 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

……	……	……
Prof.Dr.Novruz ALLAHVERDİ	Prof.Dr.Bekir KARLIK	Yrd.Doç.Dr.Abdullah ÜRKMEZ
Üye	Üye	Üye

……	……
Yrd.Doç.Dr.Ramazan AKKAYA	Yrd.Doç.Dr.Yüksel ÖZBAY
Danışman	Üye

ÖZET

Doktora Tezi

FIRÇASIZ DOĞRU AKIM MOTORLARININ GENETİK TABANLI BULANIK DENETLEYİCİ İLE SENSÖRSÜZ KONTROLÜ

Ömer AYDOĞDU

Selçuk Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd.Doç.Dr. Ramazan AKKAYA
2006, 171 sayfa

Jüri: Prof.Dr. Novruz ALLAHVERDİ
Prof.Dr. Bekir KARLIK
Yrd.Doç.Dr. Abdullah ÜRKMEZ
Yrd.Doç.Dr. Ramazan AKKAYA
Yrd.Doç.Dr. Yüksel ÖZBAY

Fırçasız Doğru Akım (DC) motorları, yüksek verim, yüksek kalkınma momenti ve sessiz çalışma gibi özelliklerinden dolayı gittikçe geniş bir kullanım alanı bulmaktadır. Fırçasız DC motorların çalıştırılabilmesi için, bir kontrol sistemine ve rotor pozisyonunu belirleyen sensörlere ihtiyaç duyulmaktadır. Ancak, sensörlerin maliyet, yer ve kararsızlık gibi dezavantajlarından dolayı, son yıllarda sensörsüz hız kontrolü önem kazanmıştır. Ayrıca, yapay zeka tekniklerindeki gelişmeyle birlikte, fırçasız DC motorların bulanık mantık, YSA ve genetik algoritma tabanlı sensörsüz hız kontrolü konusunda değişik çalışmalar yapılmaktadır.

Bu çalışmada, fırçasız DC motorların genetik tabanlı bulanık denetleyici ile sensörsüz hız kontrolü simüle edilmiş ve pratik olarak gerçekleştirilmiştir. Fırçasız DC motor, trapezoidal zıt emk dalga şekline göre yeniden modellenmiş ve simülasyon işlemlerindeki hesaplama hatalarını azaltmak için dördüncü dereceden Runge-Kutta metodu kullanılmıştır. Fırçasız DC motor hız kontrol sistemi için ilk olarak uzman kişi bilgi ve becerisine dayanan bilgi tabanlı bir bulanık denetleyici tasarlanmıştır. Bu tip tasarımda, sistematik bir yaklaşım olmadığı için denetleyicinin başarısı, bilgi tabanıyla sınırlı kalmakta ve denetleyici tasarlanmış olsa bile optimum çözüm garanti edilememektedir. Bu dezavantaj, bulanık denetleyicilerin kullanım alanını sınırlandırmaktadır. Bu problemleri çözmek için, denetleyici tasarımında bilgi tabanı yerine, denetleyici parametrelerinin gerçek kodlu genetik algoritma ile optimize edildiği yeni bir kontrol sistemi geliştirilmiştir.

Geliştirilen kontrol sisteminin optimizasyonunu sağlamak için, çok amaçlı bir performans indeksi tanımlanmış ve genetik algoritmada minimize edilmesi gereken amaç fonksiyonu olarak kullanılmıştır. Tanımlanan performans indeksine göre, denetleyici parametrelerinin genetik algoritma ile optimizasyonunda iki farklı yaklaşım ortaya konulmuştur. Bu yaklaşımlardan ilki, çok sayıdaki bulanık denetleyici parametresinin uygun biçimde gruplandırıldığı ve sırasıyla her bir gruptaki parametrelerin genetik algoritma ile belirlendiği ardışık yaklaşımdır. İkincisi ise, bulanık denetleyicinin tüm parametrelerinin genetik algoritma ile aynı anda belirlendiği eş zamanlı yaklaşımdır. Geliştirilen optimal bulanık denetleyiciler ile fırçasız DC motorun hız kontrolü simüle edilerek, geleneksel bulanık denetleyiciye ve birbirlerine göre üstünlükleri tartışılmıştır. Sonuçta, eş zamanlı yaklaşımın diğerlerine göre daha iyi sistem cevabına sahip olduğu görülmüştür.

Simülasyon çalışmalarında başarıyla yürütülen geleneksel ve optimal bulanık denetleyici algoritmaları, uygun boyut ve kod yapısına dönüştürülerek ADSP-21992 sayısal işaret işlemcisine (DSP) yüklenmiş ve fırçasız DC motorun sensörsüz hız kontrolü deneysel olarak gerçekleştirilmiştir. Deneysel çalışmalardan elde edilen sonuçların, simülasyon çalışmalarından elde edilen sonuçlarla uyum içinde olduğu görülmüştür.

Anahtar Kelimeler: Fırçasız DC Motor, Bulanık Denetleyici, Genetik Algoritma, Optimal Kontrol, Sensörsüz Kontrol, DSP, Performans İndeksi.

ABSTRACT

PhD Thesis

SENSORLESS CONTROL OF BRUSHLESS DC MOTORS BY MEANS OF GENETIC BASED FUZZY CONTROLLER

Ömer AYDOĞDU

Selcuk University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assist.Prof.Dr. Ramazan AKKAYA
2006, 171 pages

Jury: Prof.Dr. Novruz ALLAHVERDİ
Prof.Dr. Bekir KARLIK
Assist.Prof.Dr. Abdullah ÜRKMEZ
Assist.Prof.Dr. Ramazan AKKAYA
Assist.Prof.Dr.Yüksel ÖZBAY

Brushless DC motors are finding more widespread use due to their high efficiency, high starting torque and noiseless operation characteristics. The operation of brushless DC motors requires a control system and sensors to estimate rotor position. However, owing to some of their drawbacks such as cost, space requirement and instability, sensorless speed control has recently gained importance. Especially with the improvements in artificial intelligence techniques, various studies are implemented about the sensorless speed control of brushless DC motors based on fuzzy logic, ANNs and genetic algorithm.

In this study, sensorless speed control of brushless DC motors with genetic-based fuzzy controller has been simulated and practically implemented. Brushless DC motor has been remodeled according to trapezoidal back emf waveform and in order to reduce calculation error in simulation process, fourth order Runge-Kutta method has been utilized. For brushless DC motor speed control system, firstly a knowledge-based fuzzy controller depending on the knowledge and capability of a human expert has been designed. In this type of design, because no systematic approach exists, the success of the controller is limited by the knowledge-base and an optimum solution in the designed controller is not guaranteed. This drawback limits the implementation fields of the fuzzy controllers. To overcome these drawbacks in controller design, in place of knowledge-base, a new method in which controller parameters are optimized by real coded genetic algorithm has been developed.

In accomplishing optimization of developed control system, a multi-objective performance index has been defined and it has been used as objective function to be minimized in genetic algorithm. According to the defined performance index, two approaches have been presented in genetic algorithm optimization of controller parameters. The first one is a sequential approach, by which parameters are appropriately grouped and parameters in each group is determined by genetic algorithm sequentially. The second one is a simultaneous approach by which all fuzzy controller parameters are determined simultaneously. The brushless DC motor speed control has been simulated for the developed optimal fuzzy controller and their superiorities have been compared with each other and with the conventional fuzzy controller. As a result, simultaneous approach has been found to possess a better system response than the others.

The successful implementations of simulated conventional and optimal fuzzy controller algorithms have been transformed into appropriate size and code structure and programmed into ADSP-21992 digital signal processor, and sensorless speed control of brushless DC motor has been experimentally implemented. The results obtained from the experimental studies have been found to be in compliance with the results obtained from the simulation studies.

Keywords: Brushless DC Motor, Fuzzy Controller, Genetic Algorithm, Optimal Control, Sensorless Control, DSP, Performance Index.

ÖNSÖZ

Son yıllarda, değişken hızlı sürücü sistemlerinde meydana gelen gelişmelere paralel olarak fırçasız DC motorlar; yüksek moment, yüksek güç/ağırlık oranı, yüksek verim ve sessiz çalışma gibi özellikleri sebebiyle geniş bir kullanım alanı bulmuştur. Ancak, fırçasız DC motorların kontrolünde, yapısı gereği rotor pozisyon bilgisine ve inverter sistemine ihtiyaç vardır. Bu nedenle çalıştırılması ve kontrolü oldukça zordur.

Bir sistemin matematiksel modelinden ziyade, davranışını temel alan bulanık denetleyiciler, son zamanlarda kontrol sistemlerinde önemli bir yapay zeka aracı olarak karşımıza çıkmaktadır. Lineer olmayan yapısı ve bir çok lineer olmayan fiziksel sistemde kullanılması nedeniyle bulanık denetleyicilerin, fırçasız DC motor kontrolünde kullanılması da kaçınılmaz olmuştur.

Bulanık denetleyicilerin tasarımında belirli metotlar kullanılmasına rağmen, prosedürler cebirsel kriterlerle ifade edilemediği için, tasarımcının bilgi ve becerisi, biraz da şansı, tasarım işleminde belirleyici olmaktadır. Tasarımcı uzman kişi, bilgi ve becerisi dahilinde denetleyiciyi tasarlamış olsa bile, bu işlem çok sayıda deneme gerektirdiği için uzun zaman almakta ve optimal çözüm sunacağı garanti edilememektedir. Bu dezavantaj, bulanık denetleyicilerin yaygınlaşmasına en büyük engel olarak görülmektedir.

Bulanık denetleyici tasarımına sistematik bir yöntem sunmak, bulanık denetleyici ile optimal kontrol sağlamak en önemli amaçtır. Bu noktada, bulanık denetleyicilerin tasarım ve optimizasyonu karşımıza çıkmaktadır. Bulanık denetleyicilerin çok parametreli yapısı dikkate alındığında, bir başka yapay zeka aracı olan genetik algoritmaların kullanılabileceği düşünülebilir. Ancak, genetik algoritmaların gerçek zaman uygulamalarında kullanımının çok kısıtlı olması, en büyük dezantaj olarak karşımıza çıkmaktadır. Buna rağmen, bilgisayar teknolojisindeki gelişmelere paralel olarak genetik algoritmalarla ilgili çalışmalarda bir artış kaydedilmekte ve her geçen gün yeni uygulama alanlarına tatbik edildiği görülmektedir.

Bulanık denetleyicilerin, genetik algoritmalar ile optimal tasarımının simülasyon ortamında yapılması ve pratik olarak uygulanması mümkündür. Pratik uygulamada, parametrelerin hassas bir şekilde ayarlanması gibi işlemler tekrar genetik algoritmalar ile gerçekleştirilebilir.

Simülasyon ortamında geliştirilen sistemlerin pratik hayata uygulanması, teknolojinin gelişmesi ve sanayileşme açısından oldukça önemlidir. Pratik uygulamalarda yazılıma dayalı gelişmiş ve hızlı elemanların kullanımı, denetleyici sistemlerinde yapay zeka gibi yöntemlerin kullanımına imkan vermekte, gelişmeye sürekli açık esnek donanımlar ortaya çıkarmaktadır. Yapılan bu çalışma, konu ile ilgili teorik ve pratik çalışma yapan kişi ve ilgili firmalara referans olacak ve katkı sağlayacaktır.

Doktora çalışmalarım süresince hiçbir yardımını esirgemeyen ve çalışmalarımı büyük bir titizlikle yönlendiren danışmanım Yrd.Doç.Dr. Ramazan AKKAYA'ya, tez izleme komitesi üyeleri Prof.Dr.Novruz ALLAHVERDİ'ye ve Yrd.Doç.Dr.Abdullah ÜRKMEZ'e teşekkür ederim. Yoğun çalışmalarım esnasında bana sabırla destek veren değerli eşim ve kızıma ayrıca teşekkür ederim.

Bu çalışma, Selçuk Üniversitesi Bilimsel Araştırma Projeleri (BAP) tarafından desteklenmiştir. (Proje No:2003/051)

İÇİNDEKİLER

ÖZET iii
ABSTRACT v
ÖNSÖZ vii
İÇİNDEKİLER ix
SİMGELER VE KISALTMALAR xii
Simgeler xii
Kısaltmalar xv
1. GİRİŞ 1
1.1 Konunun Tanıtımı 4
1.2 Çalışmanın Amacı ve Önemi 5
2. KAYNAK ARAŞTIRMASI 10
3. MATERYAL VE METOT 17
3.1 Fırçasız DC Motorlar 17
3.1.1 Mekanik yapısı ve tipleri 18
3.1.1.1 Rotor çeşitleri 20
3.1.1.2 İç ve dış rotorlu fırçasız DC motorlar 21
3.1.1.3 Fırçasız DC motorlarda kullanılan sabit mıknatıslar 23
3.1.1.4 Fırçasız DC motor kontrolünde kullanılan sensörler 24
3.1.2 Çalışma prensibi 25
3.1.3 Matematiksel modeli 29
3.1.4 Fırçasız DC motor sürücüleri 30
3.1.5 Fırçasız DC motorların pozisyon sensörsüz kontrolü 32
3.2 Bulanık Denetleyiciler 34
3.2.1 Denetleyici tasarımında temel ilkeler 34
3.2.2 Bulanık denetleyicilerin temel yapısı 36
3.2.2.1 Normalizasyon parametreleri 36
3.2.2.2 Bulanıklaştırma ve durulaştırma 37
3.2.2.3 Çıkarım mekanizması 38
3.2.3 Bulanık denetleyici tipleri 39

3.2.4 Bulanık denetleyici karakteristikleri ……..…..……. 41
3.2.5 Bulanık denetleyicilerin tasarımı ……..……..... 43
3.2.6 Bulanık denetleyici tasarım parametreleri ……...…….…… 45
3.2.6.1 Normalizasyon parametrelerinin etkileri ……. 45
3.2.6.2 Üyelik fonksiyonları ve kuralların etkileri …….. 47
3.2.7 Geleneksel tasarımda karşılaşılan temel problemler.…….…… 49
3.3 Genetik Algoritmalar ile Optimizasyon …… 51
3.3.1 Genetik algoritmalar ……..…….. 52
3.3.2 Gerçek kodlu genetik algoritmalar ……..……... 53
3.3.2.1 Parametrelerin kodlanması ve amaç fonksiyonu ……. 55
3.3.2.2 Başlangıç popülasyonu …… 56
3.3.2.3 Uygunluk değerlerinin hesaplanması……... 56
3.3.2.4 Değerlendirme ve doğal seçim …… 57
3.3.2.5 Eşleştirme …… 59
3.3.2.6 Çaprazlama …….. 62
3.3.2.7 Mutasyon ……. 64
3.3.2.8 Elitizm …….. 66
3.3.3 Optimizasyon işlemi ...…..……... 66
3.4 Sayısal İşaret İşlemciler (DSP) ve ADSP-21992 ……...…… 71

4. BULANIK DENETLEYİCİLERLE FIRÇASIZ DC MOTORLARIN SENSÖRSÜZ KONTROLÜ…….. 74
4.1 Bulanık Denetleyicili Fırçasız DC Motor Kontrol Sistemi ……..... 74
4.2 Fırçasız DC Motorların Simülasyon Modeli …….…… 75
4.2.1 Runge-Kutta yöntemi ….……. 76
4.3 Fırçasız DC Motor Sürücülerde Akım Kontrol Döngüsünün Modellenmesi… 81
4.3.1 Referans akım üreteci…….. 82
4.3.2 Üç faz PWM inverter ve histerezis akım kontrol modeli…… 82
4.3.3 Akım kontrol döngüsü simülasyonu …….. 85
4.4 Geleneksel Bulanık Denetleyicinin Tasarlanması ……. 90
4.4.1 Giriş ve çıkışlar…… 92
4.4.2 Normalizasyon faktörleri …… 93
4.4.3 Bulanıklaştırma ve durulaştırma ……. 94

4.4.4 Çıkarım mekanizması ……. 97
4.5 Simülasyon Sonuçları……. 102
4.6 ADSP 21992 Tabanlı BLDC Motor Sensörsüz Kontrol Uygulama Devresi … 106
4.6.1 Üç faz PWM inverter …….. 107
4.6.2 Akım detektörü ……... 108
4.6.3 Sensörsüz kontrol ve zıt emk detektörü …….. 109
4.6.4 Sistemin çalışması …….. 110
4.7 Deneysel Sonuçlar ...…….. 113

5.GENETİK TABANLI BULANIK DENETLEYİCİ İLE FIRÇASIZ DC MOTORLARIN SENSÖRSÜZ KONTROLÜ …… 117
5.1 Genetik Tabanlı Bulanık Denetleyicinin Tasarlanması …….…… 117
5.1.1 Temel bulanık denetleyici …….……....……. 119
5.1.2 Performans indeksinin tanımlanması …..…… 121
5.1.3 Optimize edilecek denetleyici parametreleri ve optimizasyon yaklaşımları .. 123
5.1.4 Denetleyici parametrelerinin genetik algoritmada kodlanması …….. 126
5.2 Genetik Tabanlı Bulanık Denetleyici ile Fırçasız DC motorların Sensörsüz Kontrolü ……..…...… 129
5.2.1 Gerçek kodlu GA algoritması …..……..…… 130
5.2.2 Simülasyon sonuçları …….. 133
5.2.2.1 Ardışık optimizasyon yaklaşımı simülasyon sonuçları …….…….. 133
5.2.2.2 Eş zamanlı optimizasyon yaklaşımı simülasyon sonuçları ……. 142
5.2.3 Deneysel sonuçlar .. …… 147

6. SONUÇ VE ÖNERİLER..... 151
6.1 Sonuçlar..... 151
6.2 Öneriler..... 153

7. KAYNAKLAR..... 155

EK-A ADSP-21992 İşlemciler ile Proje Geliştirme …….…… 160
EK-B Bulanık Denetleyiciyi GA ile Optimize Eden Simülasyon Programı ……. 168
EK-C Bulanık Denetleyici ADSP Yazılım Kodu ...……... 170

SİMGELER VE KISALTMALAR

Simgeler

B	Motor sürtünme katsayısı
B_f	Sabit mıknatıs akı yoğunluğu
B_r	Artık mıknatıslanma
C	Uygunluk değerleri
d-q	İki boyutlu eksen takımı
E	Zıt emk gerilimi
e	Hata vektörü
e_A, e_B, e_C	Anlık zıt-emk gerilimleri
e_{ia}, e_{ib}, e_{ic}	Üç faz akım hataları
e_s	Tek bobinde indüklenen zıt emk
$f_A(\theta)$, $f_B(\theta)$, $f_C(\theta)$	Üç faz zıt-emk şekil fonksiyonları
f_z	Zıt-emk detektörü çıkış sinyali frekansı
h_b	İnverter histerezis aralığı
H_c	Koersif kuvvet
H1, H2, H3	Hall sensörleri
I	Akım
I^*	Referans akım
i_A, i_B, i_C	Faz akımları
i^*_A, i^*_B, i^*_C	Referans faz akımları
I_{base}	İnverter eşik akımı
I_{dc}	İnverter kaynak akımı
J	Motor atalet momenti
J_{in}	Performans indeksi
J_1	Geçici durum performans indeksi
J_2	Maksimum aşma performans indeksi
J_3	Sürekli durum performans indeksi
k	Motor sabiti

k_e	Zıt emk sabiti
k_t	Moment sabiti
kW	Kilowatt
K_P	Oransal kazanç katsayısı
K_I	İntegral katsayısı
K_D	Türev katsayısı
ℓ	Rotor uzunluğu
L_m	Karşılıklı indüktans
L_s	Motor sargı indüktansı
MW	Megawatt
N	Motor yıldız noktası
n	İşlem adımı
N_e	e-için normalizasyon faktörü
$N_{\dot{e}}$	$\dot{e}$ -için normalizasyon faktörü
N_u	u -için normalizasyon faktörü
N_{ipop}	Başlangıç populasyonu
N_{par}	Optimize edilecek parametre sayısı
n_r	Rotor hızı (dev/dak)
N_s	Bir fazın sarım sayısı
P_{ipop}	0-1 arası rasgele sayı
P_{mut}	Mutasyon olasılığı
P_H	Populasyon üst sınırı
P_L	Populasyon alt sınırı
P_o	Anlık güç
p	Motor kutup sayısı
Q	Hata fonksiyonu
r	Rotor yarıçapı
R	Sargı direnci
R_f	Bulanık denetleyicide dönüşüm fonksiyonu
R_s	İnverter akım ölçme direnci
S	Sıfır
s	Laplas operatörü

T^*	Referans moment
T_e	Elektriksel moment
T_L	Yük momenti
u	Durum uzayı analizinde giriş vektörü
v_A, v_B, v_C	Faz gerilimleri
v_{A0}, v_{B0}, v_{C0}	Kaynağa göre motor gerilimleri
V_{dc}	Kaynak gerilimi
v_{N0}	Kaynağa göre motor yıldız noktası gerilimi
ω^*	Referans hız (rad/sn)
ω_r	Rotor hızı (rad/s)
w_t	Geçici durum cevabı ağırlık katsayısı
w_{ss}	Sürekli durum cevabı ağırlık katsayısı
w_p	Maksimum aşma ağırlık katsayısı
x	Durum vektörü
y	Çıkış vektörü
$\mathbf{y}^*$	Referans giriş vektörü
μ	Üyelik değeri
$\hbar(\mathbf{e})$	Durum uzayı analizinde dönüşüm fonksiyonu
$\otimes$	Kartezyen çarpım
o	Bulanık çıkarım işlemi
β	Çaprazlama olasılığı
θ	Rotor pozisyonu
λ_s	Bobin akısı
ϕ	Manyetik akı

Kısaltmalar

ADC	Analog Sayısal Dönüştürücü
ALU	Aritmetik Lojik Birimi
Al-Ni-Co	Alüminyum-Nikel-Kobalt
BLDC	Fırçasız Doğru Akım
CAN	Ağ Alan Denetleyici Kapısı
CNC	Bilgisayarlı Nümerik Kontrol
CPU	Merkezi İşlemci Birimi
DC	Doğru Akım
DSP	Sayısal İşaret İşlemci
emk	Elektromotor kuvvet
DD	Dilsel Değişken
DM	Data Hafızası
DMA	Data Hafızası Adres Yolu
DMD	Data Hafızası Data Yolu
GA	Genetik Algoritma
I/O	Giriş/Çıkış
Nd-Fe-B	Neodmiyum-İron-Boron
MAC	Matematiksel İşlemci
MOSFET	Metal Oksit Silikon Alan Etkili Transistor
MSPS	Saniyedeki Örnek Sayısı
NB	Negatif Büyük
NO	Negatif Orta
NK	Negatif Küçük
PK	Pozitif Küçük
PO	Pozitif Orta
PB	Pozitif Büyük
PM	Program Hafızası
PMA	Program Hafızası Adres Yolu
PMD	Program Hafızası Data Yolu
P	Oransal

PI	Oransal-İntegral
PD	Oransal-Türevsel
PID	Oransal-İntegral-Türevsel
PWM	Darbe Genişlik Modülasyonu
S	Sıfır
SPI	Seri Çevresel Arabirim
SPORT	Seri Port Arabirimi
YSA	Yapay Sinir Ağları

1. GİRİŞ

Dünyada, enerji kaynaklarının hızla tükenmesi sonucu enerji fiyatlarının artmasıyla, elektrik motorlarının verimlerinin artırılması ve bu motorların kontrolünde minimum enerji tüketiminin sağlanması öncelikli amaç haline gelmiştir. Tüm dünyada üretilen elektrik enerjisinin ortalama %60'ı, elektrik motorları tarafından tüketilmektedir. Bunun yanında, motor kontrol uygulamalarının %75'inde değişken hızlı sürücüler kullanılmaktadır. Bu nedenle, elektrik motor ve sürücülerinin yapısal gelişimi büyük önem kazanmıştır. Bazı ülkeler, enerji tasarrufu programları çerçevesinde, elektrik motor ve sürücülerinin geliştirilmesi ve değişken hızlı sürücülerin yaygınlaştırılması için destek sağlamaktadır.

Elektrik motorlarının üretiminde kullanılan manyetik malzemelerdeki hızlı gelişmeler, motor sürücü sistemlerinde kullanılan elektronik elemanların yeterli yüksek hızlara ulaşmaları ve maliyetlerinin düşmesi, sabit mıknatıslı elektronik komütasyonlu özel elektrik motorlarının ortaya çıkmasına neden olmuştur. Bu tip motorlardan en yaygın olanı, son yıllarda kullanımı giderek artan fırçasız DC motorlardır. Fırçasız DC motorlar, özellikle küçük güçlü kontrol motorları olarak üretilmektedir. Çünkü bu motorlar; yüksek moment, yüksek güç/ağırlık oranı, yüksek verim, sessiz çalışma, yoğun sargı yapısı, güvenilirlik ve düşük bakım giderleri açısından oldukça avantajlıdır. Bu avantajlarından dolayı fırçasız DC motorlar; bilgisayarlarda, uzay araçlarında, askeri ekipmanlarda, otomotivde, endüstride ve ev gereçlerinde yaygın olarak kullanılmaktadır. Fırçasız DC motorların kullanımının, önümüzdeki 5 yıl içerisinde katlanarak artacağı belirtilmektedir (Shao 2003).

Fırçasız DC motorların çalıştırılması ve kontrolü oldukça zordur. Üstelik, bu motorların hız ve moment gibi temel değişkenlerinin, akım ve gerilim gibi değişkenlere lineer olarak bağımlı olmaması sebebiyle, matematiksel olarak modellenmesi zordur. Bu nedenle, bulanık denetleyiciler gibi lineer olmayan bir denetleyici ile kontrol edilmeleri gerekmektedir. Fırçasız DC motorlar, yapıları gereği elektronik olarak kontrol edilir ve doğru akım komütasyonu için kontrol

işleminde rotor pozisyon bilgisi gerektirirler. Son zamanlarda araştırmacılar, maliyetleri ve güvenilirlik problemlerinden dolayı pozisyon sensörlerini elimine etmek için, fırçasız DC motorların pozisyon sensörsüz kontrolü üzerine çalışmalar yapmaktadır. Bu çalışmalarla birlikte, birçok uygulamada, fırçasız DC motorların sensörsüz kontrolü daha fazla yaygınlaşacaktır (Matsui ve Shigyo 1992, Matsui 1993, Kim 2003).

Günümüzde, modern elektrik motor sürücüleri; modern kontrol teorisi, elektrik motor teorisi, güç elektroniği, sinyal işleme ve mikrobilgisayarlar gibi elektrik mühendisliğinin önemli disiplinlerini içine alan, karmaşık ve gelişmeye sürekli açık bir yapı haline gelmiştir. Bu yapılara son zamanlarda yapay zeka yöntemleri de dahil olmuştur. Yapay zeka yöntemlerinin mühendislik alanlarında hızla yaygınlaşması ve modern elektrik motor sürücülerine uygulanması, zeki denetim sistemlerini ortaya çıkarmıştır (Aydoğdu ve Akkaya 2005).

Yapay zeka, insanın düşünme işleminin makine benzetimidir (Holland 1975). İlk olarak yapay zeka terimi, 1956'da Dartmouth kolejinde gerçekleştirilen bir konferansta kullanılmıştır. Bu konferansta yapay zeka; zekanın kullanımını gerektiren işlerde, insanın düşünme işlemine benzeyen bilgisayar uygulamaları olarak tanımlanmıştır. L.A. Zadeh tarafından 1960'ların ortalarında, yapay zekanın bir dalı olarak, uzman sistemleri destekleyen Bulanık Mantık (Fuzzy Lojik) teorisi ortaya atılmıştır (Zadeh 1965). Bu teoride temel olarak; Boole mantığında kullanılan 1 (Doğru) ve 0 (Yanlış) ifadelerinin, aslında insanın düşünme tarzını tamamen yansıtmadığı, insanın düşünme tarzında doğal olarak bulanık veya kesin olmayan ifadelerin olduğu belirtilmektedir. Son yıllarda bulanık mantık, fırçasız DC motorlar gibi lineer olmayan ve tam olarak modellenemeyen sistemlerin kontrol edilmesinde önemli bir yapay zeka aracı olarak görülmektedir. Bulanık mantık yöntemi birçok alanda yaygın olarak uygulanmaktadır. Bunlar; motor kontrol, proses kontrol, model tahmini ve sistem tanımlama gibi kontrol sistemleri, tıbbi tanımlama ve teşhis, pazarlamacılıkta stok kontrolü ve sipariş, tarım, askeri bilimler ve benzeri alanlar olarak sayılabilir.

Bulanık mantık sistemler, temelde insana benzeyen bir bilgi tabanına sahip olsa da, uzman bir insanın yerine konulamaz. Çünkü insan, yeni şeyleri öğrenme

kabiliyetine ve önceki bilgilerini en iyi şekilde yenileme özelliğine sahiptir. Bu aklımıza şu soruyu getirebilir. Bulanık mantık tabanlı bir sistem, yeni bilgiler elde edip bilgi tabanını yenileyebilir mi? Bu soruya cevap ararken, bulanık denetleyicilerin otomatik olarak tasarlanması veya parametrelerinin otomatik olarak bulunması fikri oluşabilir. Bulanık denetleyicilerin, belirtilen kriterler dahilinde otomatik olarak tasarlanmasında bir başka yapay zeka aracı olan genetik algoritmalar kullanılabilir (Chen 2002).

Genetik algoritmalar (GA), doğadaki biyolojik evrimden esinlenen ihtimal tabanlı bir optimizasyon yaklaşımıdır. Diğer optimizasyon tekniklerinin aksine genetik algoritma, bir amaç fonksiyonu ile tanımlanan çok boyutlu bir hiper uzayda arama yapabilir. Genetik algoritmalar, diğer optimizasyon tekniklerinin tüm özelliklerini sağlamanın yanında, diğer tekniklerin çözümünde zorlandığı karmaşık problemlerde de oldukça etkilidir (Goldberg 1989). Örneğin, genetik algoritmalar; arama uzayında, arama yönü için türev hesaplarına bağımlı olmadığından, sürekli olmayan ve hesaplanamayan hiper uzaylarda başarılı bir şekilde çalışmaktadır. Üstelik; genetik algoritmalar özellikle de lineer olmayan veya birden fazla lokal optimumlara sahip karmaşık hiper uzaylarda optimumu bulmada başarılıdır. Gerçekte, gradyan tabanlı tekniklerde, herhangi bir optimum noktaya doğru arama yapıldığı için, bu noktanın en iyi çözüm olmama ihtimali vardır. Yani; bu yöntemler, birden fazla yerel optimuma sahip arama yüzeylerinde hata verebilir. Genetik algoritmalarda ise, her döngüde, önceki çözümlerden bağımsız yeni çözümler üretildiği için, birden fazla lokal optimuma sahip karmaşık arama yüzeylerinde bile iyi sonuçlar vermektedir. Bundan dolayı, global optimum noktasının bulunmasında daha etkilidir. Karmaşık bir problem olan bulanık denetleyicilerle fırçasız DC motorların optimal kontrolünde, optimizasyon için genetik algoritmaların kullanımının uygun olduğu görülmektedir (Kinzel ve ark. 1994).

En uygun geçici ve sürekli durum cevaplarının elde edilebilmesi için kontrol sistemlerinde, sistem performansını ölçen bir performans kriteri tanımlanır. Kontrol sistemlerinde, performans ölçümleri kullanılarak kendi kendini ayarlayabilen denetleyiciler tasarlanabilmektedir. Bu çalışmada; bulanık denetleyici parametrelerinin optimizasyonu için genetik algoritmalar kullanılmıştır. Böylece,

bulanık denetleyicide tanımlanan performans kriteri, genetik algoritmalarda amaç fonksiyonu olarak alınıp, çok parametreli ve geniş bir optimizasyon yüzeyinde arama yapılmıştır.

1.1 Konunun Tanıtımı

Fırçasız DC motorlar, son zamanlarda kullanım alanları hızla yaygınlaşan özel elektrik makineleridir. Uygulamalarda kullanımının giderek artacağı göz önüne alındığında, bu motorların çalıştırılması ve en hassas bir şekilde kontrol edilmesi önem kazanmaktadır. Lineer olmayan karakteristiklerinden dolayı fırçasız DC motorların kontrolü, lineer olmayan denetleyiciler ile yapılmalıdır. Fırçasız DC motorların kontrolünde, lineer olmayan karakteristiği ve tasarımında, sistem modelinden ziyade sistem davranışını temel almasından dolayı bulanık denetleyicilerin kullanımı uygun olmaktadır.

Fırçasız DC motorların kontrolünde, iki ayrı kapalı çevrim kullanılır. Birincisi, iç kontrol çevrimidir ve burada motor akımı veya momenti kontrol edilir. Bu amaçla, motor faz sargılarının akımları sürekli takip edilerek, referans akımı izlemesi sağlanır. İkincisi ise, dış kontrol çevrimidir. Bu çevrimde, motor hızı kontrol edilir. Geri besleme olarak, pozisyon sensörlerinden alınan gerçek motor hız bilgisi, hız denetleyicisine uygulanır. Sensörsüz kontrolde ise geri besleme sinyalleri, pozisyon sensörü kullanılmadan motor akım veya gerilimlerinden elde edilir ve hız denetleyiciye uygulanır. Daha sonra, hata sinyali doğrultusunda motor için bir kontrol sinyali üretilir.

Bulanık denetleyicilerin tasarlanmasında karşılaşılan en temel problem, çok sayıdaki denetleyici parametresinin belirlenmesi ve optimal bir bulanık denetleyici için bu parametrelerin en uygun değerlerde seçilmesi gereklidir. Bu nedenle, bulanık denetleyicinin optimizasyonu çok önemlidir.

Bu çalışmada, bulanık denetleyici parametrelerinin optimizasyonu için, evrim algoritmaları içerisinde üstün özelliğe sahip olan genetik algoritmalar kullanılmıştır. Genetik algoritmalar, doğal seçim mekaniğine ve doğal genetiğe bağlı araştırma algoritmalarıdır. Araştırma işlemi, canlıların biyolojik gelişmesine benzemektedir. Canlılar, doğarlar ve üreyerek yeni nesillerini yetiştirirler. Doğal seçim mekaniğine

bağlı olarak, canlı grubu içerisindeki en iyi bireyler hayatta kalarak, bir sonraki neslin oluşturulmasında rol oynarlar. Her yeni nesil, rasgele bilgi değişimi ile oluşturulan diziler içinden hayatta kalanların birleştirilmesi ile elde edilir.

Genetik algoritmalar, çok parametreli sistemlerin optimizasyonunda kullanılabildiği gibi, çoklu amaç fonksiyonlarının optimizasyonunda da başarılı bir şekilde kullanılabilmektedir. Genetik algoritmaların, gerçek zamanlı sistemlerde kullanılmasının en büyük dezavantajı; on-line uygulamalarda kullanımının çok zor ve kısıtlı olmasıdır. Buna rağmen bazı kontrol sistemlerinde kısmen on-line uygulamalar mevcuttur (Ahn ve ark. 2000, Wander ve ark. 2000). Genetik algoritmanın diğer bir özelliği de; optimize edilecek olan tüm parametrelerin aynı anda paralel olarak bulunabilmesidir. Bu özellik sayesinde, birbirine bağımlı parametrelerin optimizasyonu başarılı bir şekilde gerçekleştirilmektedir. Bu özelliklerinden dolayı GA'lar, bulanık denetleyicilerin optimizasyonu için oldukça uygundur.

Verilen bir problemi çözmek için yapılması gereken ilk şey, problemi tüm ayrıntılarına kadar tanımak, çözüm metotlarını incelemek, kullanılacak materyali belirlemek ve özelliklerine hakim olmaktır. Yapılan çalışmada; P4 tabanlı kişisel bir bilgisayar, ADSP-21992 Ez-Kit Lite DSP uygulama kartı, VisualDSP++ çalışma programı, Amatek 119003-01 fırçasız DC motor, fırçasız DC motor sürücüsü, besleme kaynakları ve diğer elektronik arabirim devreleri kullanılarak, sistem deneysel çalışmalar için pratik olarak gerçekleştirilmiştir.

1.2 Çalışmanın Amacı ve Önemi

Bu çalışmanın amacı; fırçasız DC motorların sensörsüz kontrolü için genetik tabanlı bir bulanık denetleyicinin tasarlanarak simüle edilmesi ve kontrol sisteminin DSP tabanlı bir işlemci ile pratik olarak gerçekleştirilmesidir. Bunu sağlamak için; bulanık denetleyicilerin tasarımında yeni ve güçlü bir yöntem geliştirmek ve aynı zamanda amaç fonksiyonu temelinde genetik algoritma ile optimal bir çözüm bulmak amaçlanmıştır. Lineer olmayan ve çok parametreli bir problem olduğundan dolayı fırçasız DC motorların sensörsüz kontrolü, genetik algoritmalar ile optimize edilen bulanık denetleyiciler ile gerçekleştirilmiştir. Yapılan çalışmada aşağıdaki hedefler temel alınmıştır.

Birincisi; kullanımı giderek yaygınlaşan fırçasız DC motorların sensörsüz kontrolüdür. Bilimsel alanda yapılan birçok çalışmada halen, değişik tip motorların sensörsüz kontrolü üzerine araştırmalar devam ederken, en optimum, en verimli ve en ekonomik çözüme ulaşabilme gayretleri sürmektedir. Ekonomik ve ergonomik çözümler için, hızlı ve ucuz işlemciler üretilirken, sensörler gibi elemanlar kaldırılarak yazılım ile çözüme ulaşılması hedeflenmektedir.

Bilindiği gibi sürücü uygulamalarında, doğru ve hassas bir kontrol sağlamak için sistemden, hız veya pozisyon gibi bilgilerin geri besleme yoluyla alınması gerekmektedir. Sistem bu nedenle, sensörlere ve yeni bağlantılara ihtiyaç duymaktadır. Fakat bazı uygulamalarda bu oldukça zordur. Bu nedenle sensörlerin elimine edilmesi gerekir. Ayrıca hızlı işlemcilerin ortaya çıkmasıyla, bazı bilgilerin programlar yoluyla elde edilmesi mümkündür. Artık optimum mühendislik çözümlerinde; yeni kablo bağlantılarına, pahalı sensörlere ve kontrol elemanlarına, esnek olmayan sistemlere yer yoktur. Bu nedenle önerilen çalışma, hem halen uygulanan sistemlerin incelenmesi hem de daha değişik yöntemlerin uygulanabilirliği açısından oldukça önemlidir.

İkincisi; simülasyon çalışmalarında kullanılmak üzere fırçasız DC motorların matematiksel modelinin en doğru biçimde elde edilmesidir. Özel elektrik motorları arasında yer alan fırçasız DC motorlar, yapıları gereği trapezoidal zıt emk dalga şekline sahiptirler. Ancak, literatürde verilen bazı simülasyon çalışmalarında, zıt emk modeli sinüsoidal olarak ele alınmakta, analizler ve simülasyonlar buna göre yapılmaktadır. Halbuki, zıt emk modelinin trapezoidal şekilde ele alınması, analiz ve simülasyonların buna göre yapılması daha doğru bir yaklaşımdır. Simülasyon çalışmaları için diğer önemli bir nokta, motor modelinin nümerik çözümlerinde hesaplama hatasının en aza indirilmesidir. Bu amaçla, fırçasız DC motorun simülasyon modelinde, Runge-Kutta yöntemi kullanılarak daha etkili bir çözüm elde edilmiştir.

Üçüncüsü; Endüstriyel uygulamalarda kullanımı oldukça yaygınlaşan bulanık denetleyicilerin, genetik algoritmalar kullanılarak optimal biçimde tasarlanmasıdır.

Bulanık denetleyici tasarımı, beş farklı aşamadan oluşur. Bu aşamalar; normalizasyon, bulanıklaştırma, kuralların çıkarımı, durulaştırma ve denormalizasyondur. Sistemin dinamik performansı ve denetleyici parametreleri arasındaki gerçek ilişki bilinmediği için, özel uygulamalarda bir bulanık denetleyicinin en iyi şekilde tasarımında kullanılan sistematik bir yaklaşım yoktur. Bundan dolayı, bazı noktalarda bulanık denetleyici tasarım işlemi, deneme ve yanılma yaklaşımıyla yapılmaktadır. Yani, bulanık denetleyici tasarımı, tamamen tasarımcının kabiliyeti ve deneyimlerine bağlı kalmaktadır. Bazı deneme yanılma yaklaşımları çok sayıda tekrarlama gerektirir ve bu yüzden sıkıcı olup uzun zaman alır. Ayrıca, bulanık denetleyicinin giriş/çıkış sinyal sayısı arttıkça, işlem daha da zorlaşır ve kabul edilebilir çözüm imkansızlaşır. Sistem dinamiklerini iyi bir şekilde bilen uzman olmayışı veya uzman bilgisinin yetersiz kalması gibi bazı durumlar da olabilir. Tüm bu durumlarda, bulanık denetleyici tasarımının ilk aşamasını bile gerçekleştirmek mümkün olmayabilir. Bu bakış açısının dışında, deneme yanılma yoluyla bir bulanık denetleyici tasarlansa bile, tasarlanan denetleyicinin optimal bir sonuç verebileceği garanti değildir. Her ne kadar başarılı uygulamaları bulunsa da, yukarıdaki sınırlamalar, kontrol problemlerinde bulanık kontrolün daha yaygın olarak kullanılmasını engellemektedir. Bu sınırlamaları gidermek için, optimizasyon yöntemlerinden çok sayıdaki parametreyi aynı anda optimize etme özelliği olan genetik algoritmalar kullanılabilir. Böyle bir uygulamada genetik algoritma, kullanıldığı sistemi otomatikleştirir ve aynı zamanda bulanık denetleyici tasarım işlemini önceden tanımlanmış bir performans indeksine göre optimize eder. Bu nedenle, çalışma ile önerilen denetleyicinin kullanımı belki de birçok alanda insanlığa kolaylık sağlayacak ve bilime katkıda bulunacaktır.

Uygulamalarda, optimizasyon tekniklerinin büyük bir çoğunluğu gradyan tabanlıdır. Bu tekniklerle, kısıtlı sayıdaki parametre optimize edilmektedir. Bu çalışmada, optimizasyon amacıyla genetik algoritmalar kullanılmaktadır. GA'lar ile çok amaçlı, çok parametreli sistemlerin optimizasyonu mümkündür. Bu çalışmanın orijinal bir yönü de; normalizasyon faktörleri, üyelik fonksiyonları ve çıkarım kuralları gibi tüm parametrelerin aynı anda optimize edilmesi ve parametrelerin birbirlerine olan etkilerinin incelenmesidir. Bu, bulanık kontrol sistemlerinin performansını da etkilemektedir.

Dördüncüsü; Kontrol cihazları arasında dünyada yaygın olarak kullanılmaya başlanan, yüksek işlem hızına sahip DSP elemanlarının kullanılmasıdır. DSP kontrol elemanları, motor sürücüleri için oldukça uygundur. Bu elemanlar, elektrik motorlarının kontrolünde yüksek hesaplama gücü sağlamasının yanında, dış ortamlarla ilişki kuran arabirimleri de içermektedirler. Bu nedenle DSP'ler oldukça etkili bir kontrol sağlamaktadır.

Sonuç olarak; geliştirilen sistem, sadece fırçasız DC motorların kontrolünde değil, aynı zamanda bulanık kontrol gerektiren tüm lineer olmayan sistemlerde kullanılabilecek yapıdadır. Çalışma, bu özellikleriyle ele alındığında, endüstriyel uygulamalara katkısının yanında, gerek bireysel araştırmalar ve gerekse akademik çalışmalar yönünden bilime katkı sağlayacaktır.

Bu tez çalışması toplam yedi bölümden oluşmaktadır.

Birinci bölümde; tezin konusu genel olarak tanıtılmış, tezin önemi ve amacı hakkında kısaca bilgi verilmiştir.

İkinci bölümde; konu ile ilgili günümüze kadar yapılan çalışmaların özetleri verilmiştir.

Üçüncü bölümde; Tezde kullanılacak materyal ve metot sunulmuştur. Deneysel donanım için gerekli materyallerin ayrıntıları verilmiş, uygulanacak metotlar ve fırçasız DC motorlar detaylı olarak anlatılmıştır. Fırçasız DC motorların pozisyon sensörsüz kontrolünde kullanılan yöntemler açıklanmıştır. Bulanık denetleyiciler detaylı olarak incelenmiş ve tasarım aşamaları verilmiştir. Genetik algoritmaların yapısı ve çalışması açıklanmıştır. Uygulamada kullanılan sayısal işaret işlemci (DSP) hakkında kısaca bilgi verilmiştir.

Dördüncü bölümde; Fırçasız DC motorların matematiksel modellerinden faydalanılarak, fark denklemleri şeklinde ayrık zaman modelleri elde edilmiş ve simülasyonlar için uygun forma sokulmuştur. Simülasyonlar yardımıyla, fırçasız DC motorların karakteristiği incelenmiştir. ADSP-21992 işlemci kullanılarak gerçekleştirilen kontrol devresi tasarlanmıştır. Geleneksel yollardan tasarlanan bulanık denetleyiciler kullanılarak, fırçasız DC motorların sensörsüz kontrolü, simüle

edilmiş ve pratik olarak gerçekleştirilmiştir. Bulanık denetleyicilerin tasarımında karşılaşılan zorluklar ortaya konulmuştur.

Beşinci bölümde; Genetik algoritmalar (GA) kullanılarak bulanık denetleyicinin optimal tasarımı gerçekleştirilmiştir. Bu amaçla ilk olarak; optimal bulanık denetleyicinin tasarımında optimize edilecek denetleyici parametreleri belirlenerek, parametrelerin denetleyici üzerindeki etkileri incelenmiştir. Ayrıca, değişik simetri özellikleri kullanılarak denetleyici parametre sayısı azaltılmıştır. Sayısı azaltılan denetleyici parametreleri, genetik algoritmalar için gerçek kodlu olarak kodlanmıştır. Denetleyici parametrelerinin, GA ile optimizasyonunda izlenecek yöntemler belirlenmiştir. Başlangıçta; parametreler ardışık olarak optimize edilmiş, daha sonra eş zamanlı yaklaşım ile tüm parametrelerin aynı anda optimizasyonu yapılmıştır. GA tabanlı optimal bulanık denetleyici ile fırçasız DC motorların sensörsüz kontrolü simüle edilerek pratik olarak gerçekleştirilmiş ve elde edilen sonuçlar değerlendirilmiştir.

Altıncı bölümde; yapılan tez çalışmasında elde edilen sonuçlar tartışılmış ve ileride bu konuda çalışma yapacak olan araştırmacılara bazı önerilerde bulunulmuştur.

Yedinci bölümde; tezde faydalanılan kaynaklar verilmiştir.

2. KAYNAK ARAŞTIRMASI

Literatürde, fırçasız DC motorların sensörsüz kontrolü ile ilgili çeşitli metotlar verilmiştir. Ayrıca fırçasız DC motorların, bulanık mantık yöntemi ile kontrolü üzerine değişik çalışmalar mevcuttur. Ancak, fırçasız DC motorların genetik tabanlı bulanık denetleyicilerle kontrolü ile ilgili pek az çalışmaya rastlanmaktadır. Tezde faydalanılan bazı önemli çalışmaların literatür özetleri aşağıda verilmiştir.

Zadeh (1965), bulanık kümeler isimli çalışmasında; matematiğin bir alt dalı olarak bulanık küme teorisini ortaya atmış ve teori ile ilgili tanımlamalar yapmıştır. Ayrıca çalışmada, bulanık kümelerle cebirsel işlemler ve özellikleri verilmiştir. İlk olarak Zadeh tarafından ortaya konulan bulanık mantığın uygulaması ise, genellikle yapay zeka sistemlerinde karşımıza çıkmaktadır.

Matsui ve Shigyo (1992), fırçasız DC motorların pozisyon sensörsüz kontrolü isimli çalışmalarında; fırçasız DC motorların çalıştırılması ve moment kontrolü için, neden rotor pozisyon sensörüne ihtiyaç duyulduğunu açıklamışlardır. Ayrıca; uygulamada kullanılan pozisyon sensörlerinin dezavantajları belirtilmiştir. Yapılan çalışmada yazarlar, fırçasız DC motorların pozisyon sensörsüz kontrolü için yeni bir yöntem önermişlerdir. Önerilen yöntemde rotor pozisyonu, gerçek bir motor ile önerilen bir model arasındaki akım farkı kullanılarak elde edilmiştir. Buradaki akım farkı iki bileşene ayrılarak, bu bileşenlerin biri pozisyon, diğeri hız tahmini için kullanılmıştır. Yeni yöntemin çalışması, önce simülasyonla, daha sonra uygulama devresinde incelenmiştir. Deneysel sonuçlarda, yöntemin hem geçici durum hem de sürekli durum koşullarında, istenilen kontrol karakteristiklerini sağladığı belirtilmektedir.

Hemati ve Leu (1992), fırçasız DC motorların karakteristik modelleri ile ilgili çalışmalarında; manyetik doyumun oluştuğu çalışma bölgelerinde, düzgün olmayan hava aralığı nedeniyle oluşan modelleme problemlerini ele almışlardır. Doyum anında oluşan relüktans değişimlerinin modele etkileri incelenmiş ve modelleme problemleri deneysel olarak tanımlanmıştır. Bu tanımlamalar dikkate alınarak, yeni bir model önerilmiş ve modelin doğruluğu kontrol edilmiştir.

Matsui (1993), çalışmasında; genel olarak motor kontrol uygulamalarında sensör kullanmanın dezavantajlarını belirtmiş ve özellikle de fırçasız DC motor sürücülerinde sensör elemanının niçin elimine edilmesi gerektiğini irdelemiştir. Ayrıca çalışmasında; motor gerilim ve akım modellerini kullanarak, sinüsoidal akı dağılımında yüksek kalitede moment kontrolü sağlandığını göstermiştir. Yapılan uygulama ile fırçasız DC motorun geçici ve sürekli çalışma şartları izlenerek karşılaştırılmıştır.

Luk ve Lee (1994), fırçasız DC motor sürücüleri için etkili bir modelleme isimli çalışmalarında, fırçasız DC motor ve sürücü sistemi için MATLAB ortamında yeni bir simülasyon modeli önermişlerdir. Önerilen model sayesinde, tasarım aşamasında maliyetin azaldığı ve MATLAB ortamında hazırlandığı için kullanımının oldukça kolay olduğu belirtilmiştir.

Lee ve Pang (1994), çalışmalarında; bulanık mantık tabanlı fırçasız DC motor sürücüleri 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 ve arkadaşları (1994), bulanık denetleyicilerin tasarım ve optimizasyonu için değişik yapılarda genetik algoritmalar 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; genetik algoritmalar hakkında kısaca bilgi verilmekte ve bulanık denetleyicilerin tasarlanması anlatılmaktadır. Ters sarkaç probleminde tasarlanan denetleyici deneysel olarak incelenmiştir.

Donescu ve arkadaşları (1996), fırçasız DC motorlarda bulanık mantık hız kontrolü için, sistematik bir tasarım metodu isimli çalışmalarında; bulanık denetleyiciler için yeni bir tasarım yöntemi geliştirmişlerdir. Bu yönteme 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 belirtilmektedir.

Herrera ve arkadaşları (1998), çalışmalarında; öğrenebilen bulanık denetleyiciler için bir genetik öğrenme yöntemi sunmuşlardır. Bu yöntem üç aşamadan oluşmaktadır. Birincisi; bulanık kuralların genetik olarak üretilmesidir. İkincisi; eğer varsa uzman kuralları ile üretilen bulanık kuralların birleştirilmesi ve gereksiz bulanık kuralların iptal edilmesidir. Üçüncüsü; bulanık kuralların üyelik fonksiyonlarının ayarlanmasıdır. Öğrenme işleminin bu üç aşaması uygun genetik algoritmalar kullanılarak gerçekleştirilmiştir.

Bonvin ve Perriard (2000), çalışmalarında; fırçasız DC motorların sinüzoidal olmayan zıt emk'dan dolayı oluşan regülatör dalgalanmalarını yok etmek için, yeni bir metot önermişlerdir. Bunun için, fırçasız DC motorların d-q modellerinden yararlanarak vektör kontrolünü yapmışlardır. Gerçekleştirilen yeni regülatör yapısıyla, geleneksel regülatör yapıları karşılaştırılmıştır.

Wu ve Liu (2000), çalışmalarında; bulanık denetleyici tasarımında, üyelik fonksiyonlarının ve bulanık kurallarının aynı anda genetik algoritmalar yoluyla bulunması için, yeni bir yaklaşım önermişlerdir. Optimize edilecek parametreler olarak; üyelik fonksiyonlarının konumu, sağ ve sol genişlikleri ve bulanık kurallar ele alınmıştır. Bu parametreler, bir ölçekleme metodu kullanılarak gerçek kodlu kromozomlara dönüştürülmüş ve genetik algoritma için mutasyon, çaprazlama ve yeniden üreme işlemlerinden geçirilmiştir. Önerilen metodun geçerliliğini ve uygulanabilirliğini göstermek için bir simülasyon çalışması yapılmıştır.

Tang ve arkadaşları (2001), optimal bulanık mantık PID denetleyici isimli çalışmalarında; geleneksel PID denetleyicilerin, genellikle lineer olmayan, derecesi yüksek, zaman gecikmeli ve matematiksel modeli iyi bilinemeyen 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.

Aslan 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.

Toliyat ve Gopalarathnam (2002), tarafından güç elektroniği el kitabı için hazırlanan bölümde; fırçasız DC motorların yapıları, sabit mıknatısları, stator sargıları, motor karakteristikleri, matematiksel modelleri, güç elektroniği dönüştürücüleri, motorun sensörlü ve sensörsüz pozisyon ölçümleri, hız-moment karakteristikleri ve değişik uygulamaları verilmiştir.

Kim ve Youn (2002), çalışmalarında; yüksek hız bölgelerinde fırçasız DC motorların kontrolünde, akım ve gerilim kontrol işlemleriyle beraber PWM sinyallerinin üretilmesinin, işlemci hızı açısından bir problem olarak karşımıza çıktığını belirtmişlerdir. Bu problemi çözmek için, PWM sinyallerinin üretilmesi ve komütasyon işlemlerinin bağımsız olarak yapılması önerilmiştir. Bu amaçla, DC hat üzerine bir anahtarlama elamanı yerleştirmişlerdir. Önerilen yönteme ait donanım şemasını vermişlerdir.

Ming-Chen ve Chou-Chen (2002), çalışmalarında; genetik algoritmalar ile bulanık mantık üyelik fonksiyonlarının otomatik olarak gerçekleştirilmesi ve bulanık kuralların elde edilmesi için yeni bir yöntem sunmuşlardır. Bu yönteme göre; ilk olarak eğitim dataları ağırlık fark kümeleme yöntemi kullanılarak kümelere ayrılmış ve her bir küme için bir bulanık mantık kural üretilmiştir. Daha sonra, üretilen bulanık mantık üyelik fonksiyonlarının optimizasyonu, genetik algoritmalar ile yapılmıştır. Yazarlar, önerilen yöntemin varolan yöntemlere göre daha avantajlı olduğunu belirtmişlerdir.

Çunkaş ve Akkaya (2002), İkili kodlu genetik algoritmalarla Gerçek kodlu genetik algoritmaların karşılaştırılmasını, örnek test fonksiyonlarını kullanarak yapmışlar. Gerçek kodlu genetik algoritmaların ikili kodlu genetik algoritmalara göre daha hızlı çalıştığını göstermişlerdir.

Yedemale (2003), hazırladığı uygulama notunda; fırçasız DC motorların temel yapısı hakkında geniş bilgi vermiştir. Yazar çalışmasında, fırçasız motorların yapısı ve çalışma prensipleri, stator sargıları, rotor kalıcı mıknatısları, oluşan zıt-emk gerilimleri, hall sensörleri, hız-moment karakteristikleri, fırçasız DC motorların diğer motorlar ile karşılaştırılması, kapalı çevrim kontrolü, sensörsüz kontrolü, tipik uygulamaları ve motor teknik özellikleri konusunda bilgiler vermiştir.

Lee ve Ehsani (2003), tarafından yapılan çalışmada, fırçasız DC motor sürücülerin ileri simülasyon modelleri geliştirilmiştir. Geliştirilen modelde, PWM inverter elemanlarının akım ve gerilimleri gibi, hız ve momentin dinamik karakteristikleri etkili olarak gözlenebilmekte ve analiz edilmektedir. Bu yüzden, geliştirilen simülasyon modelinin fırçasız DC motor kontrol algoritmalarının geliştirilmesine kolaylık sağlayacağı ve hesaplama zamanı ile hafıza boyutlarını azaltacağı belirtilmektedir.

Singh ve Jain (2003), kalıcı mıknatıslı fırçasız DC motorlar için DSP tabanlı sayısal hız denetleyici isimli çalışmalarında, önerilen sürücü sistemin performans analizi için, sürücü sistemin matematiksel analizini yapmışlardır. Fırçasız DC motor sürücü sistemini, mümkün olan en az sayıda eleman ile gerçekleştirmişlerdir. Gerçekleştirdikleri sistem; bir güç devresi, kontrol donanımı ve yazılım kısımlarından oluşmaktadır. Gerçekleştirilen sürücü performansını, ilk kalkınma, sürekli çalışma ve yük değişimlerinde incelemişler ve uygulama için güvenilirliğini belirtmişlerdir.

Kim (2003), sıfıra yakın hızlardan nominal hıza kadar olan çalışma aralığında fırçasız DC motorların sensörsüz kontrolü isimli doktora çalışmasında; fırçasız DC motorlar için yeni bir sensörsüz kontrol yöntemi sunmuştur. Fırçasız DC motor sürücülerin çalışma prensibini inceleyerek, daha önce yapılan sensörsüz kontrol yöntemlerini vermiştir. Yeni önerilen sensörsüz kontrol yöntemi, motor hızından bağımsız ve akı dağılımına bağımlı olarak verilmiş, simülasyon ve deneysel uygulama ile doğrulanmıştır.

Yun ve arkadaşları (2003), genetik tabanlı bulanık 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 sonuçlar analiz edilmiştir.

Su ve McKeever (2004), çalışmalarında; fırçasız DC motorlar için düşük maliyetli bir pozisyon sensörsüz kontrol yöntemi sunmuşlardır. Üç fazlı bir motor için, motor faz gerilimlerinden sadece birinden dolaylı olarak alınan zıt emk yoluyla, rotor pozisyon bilgisini elde etmişlerdir. Zıt emk'nın hat üzerinde algılandığı noktaya göre bir filtre seçilmiş ve pozisyon algılama devresine basitlik getirilmiştir. Bu da; hız kontrolü için daha ucuz bir işlemci kullanılmasını sağlamıştır. Zıt emk'nın algılanmasının zor olduğu düşük hızlarda, hatasız pozisyon bilgisi elde etmek ve filtre nedeniyle oluşan ideal olmayan faz gecikmelerini düzeltmek için bir tablo verilmiştir. Yazarlar, bu yaklaşımın motorun çalışma aralığını ve verimini artırdığını, deneysel sonuçlarında bunu doğruladığını ifade etmişlerdir.

Xia ve arkadaşları (2004), genetik tabanlı bulanık denetleyicilerle fırçasız DC motorların hız kontrolü isimli çalışmalarında; genetik algoritmaları kullanarak bir bulanık denetleyicinin otomatik olarak tasarlandığı yeni bir yöntem sunmuşlar ve motor kontrolüne uygulamışlardır. Gerçekleştirilen pozisyon sensörlü kontrol sisteminde, iki kapalı çevrim kontrol döngüsü verilmektedir. İç kontrol çevrimi, motor momentini ayarlayan akım döngüsüdür. Dış kontrol çevrimi ise, parametreleri genetik algoritmalar ile çevrim dışı ayarlanan, bulanık mantık hız denetleyicidir. Önerilen yöntemin kararlı bir kontrol sağladığı ve diğer çalışmalara göre daha yüksek doğrulukta sonuçlar verdiği deneysel olarak gösterilmiştir.

Aydoğdu ve Akkaya (2005), fırçasız DC motorların pozisyon sensörsüz sayısal işaret işlemci (DSP) tabanlı bulanık kontrolü isimli çalışmalarında; trapezoidal zıt emk gerilimlerinden motor pozisyonunu belirlemiş ve hız kontrolü için kullanmışlardır. DSP ortamında gerçekleştirilen bulanık mantık kontrol algoritması ile sistem pratik olarak gerçekleştirilmiştir.

Qiao ve arkadaşları (2005), çalışmalarında; trafik akışı kontrolünde bulanık denetleyici yöntemini kullanmışlar ve bulanık denetleyicinin optimizasyonu için özel genetik algoritmalar geliştirmişlerdir. Gerçek kodlu genetik algoritmaların, ikili

kodlu genetik algoritmalara göre daha üstün olduğunu ve trafik kontrol sistemlerinde güvenle kullanılabileceğini belirtmişlerdir. Ayrıca, karmaşık ve zamanla değişen trafik akışı için, tek amaçlı bulanık denetleyicilerin yetersiz kaldıkları belirtilmiştir. Bunun için; trafik değişikliklerine adapte olabilen, çok amaçlı bir bulanık mantık kontrol algoritması önerilmiş ve bulanık mantık kontrol kuralları ve üyelik fonksiyonlarının genetik algoritmalar ile optimize edilmesi anlatılmıştır. Gerçekleştirilen sistemin geleneksel yöntemlere göre performansının daha iyi olduğu belirtilmiştir.

Literatürden de görüldüğü gibi fırçasız DC motorların kullanımı, kontrolünde elektronik komütasyon işlemine ihtiyaç duyması nedeniyle elektronik devre elemanlarının gelişimine paralel olarak yaygınlaşmaktadır. Literatür taramasında, fırçasız DC motorların sensörsüz hız kontrolünde genetik tabanlı bulanık denetleyicilerle ilgili çok az çalışmaya rastlanmıştır. Ancak, kullanımı hızla yaygınlaşan fırçasız DC motorların optimal kontrolü giderek önem kazanmaktadır.

Bu çalışmada, genetik algoritmalar kullanılarak optimal bulanık denetleyici için yeni ve uygulanabilir bir çözüm sunulmaktadır. Sistem pratik olarak DSP tabanlı gerçekleştirilerek sonuçlar irdelenmektedir.

3. MATERYAL VE METOT

3.1 Fırçasız DC Motorlar

Fırçasız DC (BLDC) motorlar, statoru üzerinde endüvi sargıları, rotoru üzerinde sabit mıknatısları bulunan ve zıt-emk dalga şekli trapezoidal olan senkron motorlardır. Yapısal olarak fırçasız DC motorlar, geleneksel sabit mıknatıslı DC (PMDC) motorların içten dışa terslenmiş versiyonudur.

Fırçasız DC motorların en önemli avantajları; yüksek verim, sessiz çalışma, fırça ve kolektör sistemlerinin kaldırılmasından dolayı bakım işleminin oldukça basit ve masrafsız olması, patlayıcı ortamlarda tehlike oluşturan elektriksel arkların oluşmaması, endüvi sargılarının statorda bulunması nedeniyle sargı sıcaklığının dışarıya kolayca aktarılması, rotor üzerinde hiçbir sargı bulunmadığından dolayı elektriksel kayıpların minimum olması olarak sayılabilir. Düşük güçlerde fırçasız DC motorlar, diğer motorlara göre daha iyi verim ve daha yüksek güç/hacim oranına sahip olduğundan, aynı motor boyutlarında daha fazla çıkış gücü sağlamaktadır.

Fırçasız DC motorların dezavantajları ise; kontrolünde DC anahtarlama için anlık rotor pozisyon bilgisi gerektirmesi, motorun çalıştırılmasında başlatma, yol alma ve sürekli çalışma durumları için farklı kontrol algoritmalarının uygulanması, rotor üzerindeki sabit mıknatısların zamanla mıknatıslık özelliklerinin zayıflaması olarak sayılabilir. Ayrıca, manyetik malzemelerdeki büyük gelişmelere rağmen, boyut-fiyat oranları göz önüne alındığında, sabit mıknatıslı uyartım sistemleri, genellikle 20 kW'ın altındaki küçük motorlar için daha uygun olmaktadır. Ancak, manyetik malzemeler ile ilgili çalışmalar, yakın gelecekte yüksek alanlı sabit mıknatısların geliştirilmesi ile birkaç MW'lara kadar sabit mıknatıslı motorların yapımının mümkün olacağını göstermektedir (Toliyat ve Gopalarathnam 2002).

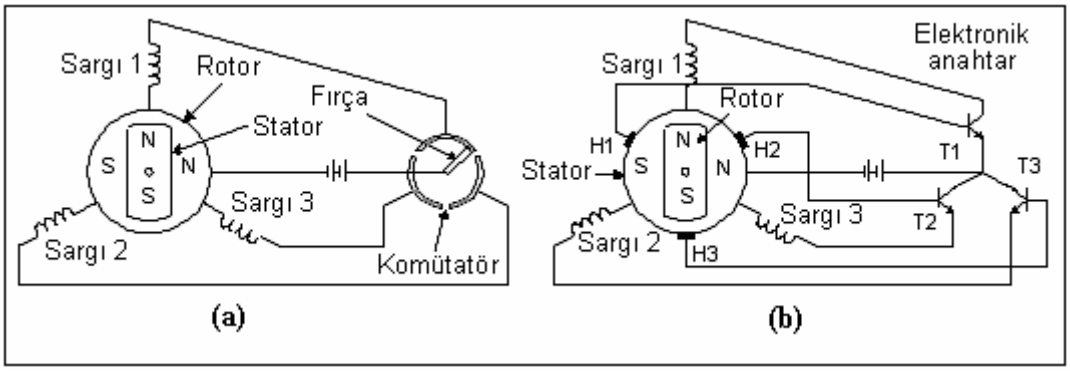
Yarı iletken anahtarlar ve DSP'lerdeki son gelişmeler, ayarlanabilir hızlı motor sürücülerinde devrim yapmıştır. Günümüzde elektrik motorlarının kontrolünde, karmaşık donanım yapılarından ziyade, yazılım tabanlı kontrol yapıları giderek artmaktadır. Bu da, fırçasız DC motorlar gibi özel elektrik motorlarının yaygınlaşmasına neden olmakta, daha güçlü kontrol yöntemlerinin uygulanmasına

imkan vermekte ve motor sürücü sistemlerinde performansı artırırken maliyette de oldukça büyük avantajlar sağlamaktadır. Ayrıca bu yönelimin en belirgin örneği, fırçasız DC motor sürücü sistemlerinde görülmektedir. Güç elektroniği teknolojisinin gelişmesine paralel olarak uygulamaların pek çoğunda fırçasız DC motor kullanılmaya başlanmıştır.

3.1.1 Mekanik yapısı ve tipleri

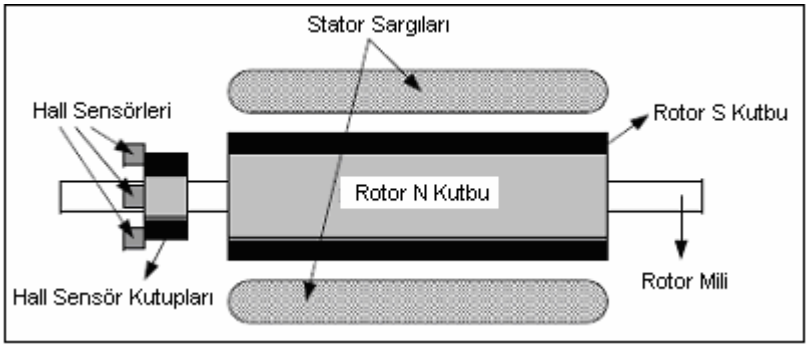
Fırçasız DC motorlar, üzerlerinde yoğun stator sargıları ve geniş sabit kutupları bulunan makinelerdir. Bu makinelerde, stator sargıları ve sabit mıknatıslı rotor kutupları, hava aralığında kare dalga formunda bir akı dağılımı elde edilebilecek biçimde yerleştirilir. Soğutma fanları gibi bazı hafif işlerde düşük performanslı küçük fırçasız DC motorlar kullanılır. Bunlar tek veya iki fazlı olarak yapılırlar. Diğer taraftan, MW mertebesindeki yüksek güçlü sürücüler için, çok fazlı motorlar tercih edilir. Çok fazlı motorlar, tek fazlı motorlara göre daha kararlı hareket ederler ve güç tüketimi bakımından daha tasarrufludurlar. Çok fazlı fırçasız DC motorlar on beş faza kadar üretilirler fakat, bu özel tasarım gerektirir ve en yaygını üç fazlı, dört fazlı ve beş fazlı motorlardır (Toliyat ve Gopalarathnam 2002).

Yapılan çalışmada, üç fazlı bir fırçasız DC motor kullanıldığından burada, konu olarak üç fazlı fırçasız DC motorlar ele alınmıştır. Şekil 3.1'de geleneksel sabit mıknatıslı DC motor ve fırçasız DC motor yapıları görülmektedir. Sabit mıknatıslı DC motor, komütatör ve fırça gibi mekanik anahtarlara sahiptir. Ancak fırçasız DC motorda mekanik anahtarların yerini elektronik anahtarlar almıştır (Kim 2003).



Şekil 3.1 (a) Sabit mıknatıslı DC motor (b) Fırçasız DC motor

Fırçasız DC motorlar temel yapı olarak, Şekil 3.2'de görüldüğü gibi üzerinde alan sargıları bulunan stator, sabit mıknatıs kutuplu rotor ve elektronik komütatör için rotor pozisyonunu ölçen pozisyon (Hall) sensörlerine sahiptir. Şekil 3.3'de görüldüğü gibi fırçasız DC motorun statoru, bir yüzü yalıtılmış çelik saçların preslenmesiyle elde edilmektedir. Asenkron motorların stator yapısına benzese de, temelde sargıların dağılımında farklı bir tarz vardır. Fırçasız DC motor sargıları, asenkron motorlarda olduğu gibi yıldız veya üçgen bağlanabilir. Fırçasız DC motorlarda stator oluk sayısı, rotor kutup sayısına, faz sayısına ve sargı şekline bağlı olarak seçilir. Genellikle kesirli bir oluk/kutup oranına sahip tasarım, cogging momentini minimize etmek için tercih edilir (Yedamale 2003, Hemati ve Leu 1992).



Şekil 3.2 Fırçasız DC motorun temel yapısı

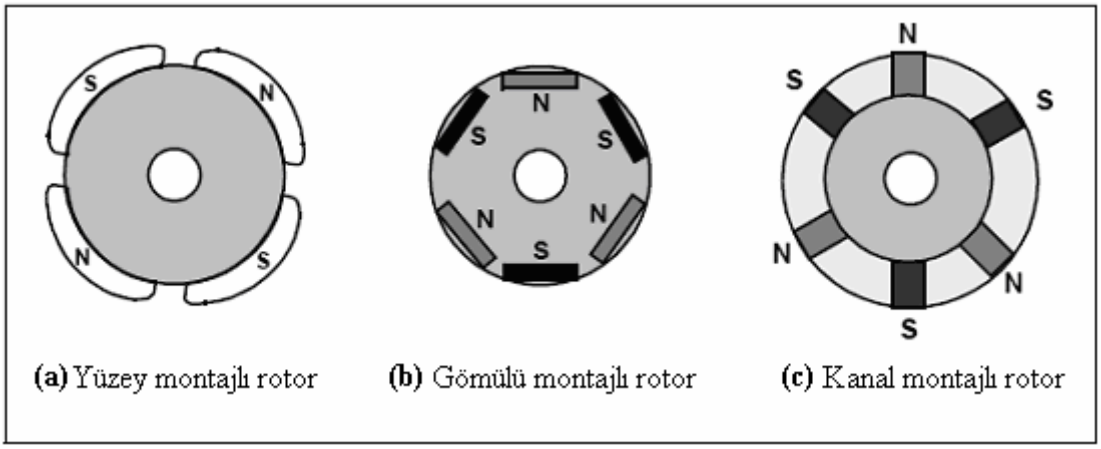


Şekil 3.3 Fırçasız DC motor statoru ve alan sargıları

3.1.1.1 Rotor çeşitleri

Fırçasız DC motorlarda, sabit mıknatıslı rotor kutupları rotor üzerine monte şekline göre, dikdörtgen kesitli veya dairenin bir parçası gibi eğimli olarak yapılırlar. Kutuplar, birbirleriyle bitişik ve stator sargılarıyla karşılıklı olarak rotor yüzeyine yapıştırılırlar. Rotor kutup sayısı, maksimum dönme hızıyla ters orantılıdır.

Fırçasız DC motorlar, Şekil 3.4'de görüldüğü gibi, mıknatısların rotor üzerine yerleştirilme şekline göre üçe ayrılırlar. Bunlar; sabit mıknatısları rotor yüzeyine monte edilmiş motorlar (Şekil 3.4.a), dikdörtgen kesitli sabit mıknatısları rotor içerisine gömülü monte edilen motorlar (Şekil 3.4.b) ve Şekil 3.4.c'de olduğu gibi dikdörtgen kesitli sabit mıknatısları rotor kanallarına yerleştirilen motorlardır (Yedamale 2003).



Şekil 3.4 Fırçasız DC motorların rotor çeşitleri

Şekil 3.4 (a)'da görüldüğü gibi, sabit mıknatısları rotor yüzeyine monte edilmiş motorlarda her bir mıknatıs, rotorun yüzeyine yerleştirilmiştir. Motor tasarımı açısından bu tip rotorları yapmak kolaydır ve mıknatıslar, stator kutuplarına daha yakın olduğu için cogging momenti daha küçüktür. Ancak yüksek hızlı çalışmada, yüzeye tutturulmuş sabit mıknatısların parçalanma ihtimali vardır. Bu motorlarda etkin hava boşluğu, rotor pozisyonu ile değiştiği için bir endüktans değişimi olduğu bilinmektedir.

Şekil 3.4 (b)'de görüldüğü gibi, dikdörtgen kesitli sabit mıknatısları rotorun içerisine monte edilmiş motorlarda her bir sabit mıknatıs, rotor nüvesinin iç tarafına yerleştirilmiştir. Sabit mıknatısı içerisine monte edilmiş rotorun kullanımı, yüzey monteli tip kadar yaygın değildir. Ancak, bu tip rotorlar, yüksek hızlı çalışmalar için uygundur. Genellikle bu tip motorlarda, rotor pozisyonu ile endüktans değişimi önemsenmeyecek kadar küçüktür.

Şekil 3.4 (c)'de görüldüğü gibi dikdörtgen kesitli sabit mıknatısları rotor kanallarına monte edilmiş motorlarda, her bir mıknatıs rotor yüzeyine açılan kanallara yerleştirilmiştir. Bu tip rotorlarda, önceki rotor tiplerinin iyi özellikleri sağlanmaya çalışılmıştır.

3.1.1.2 İç ve dış rotorlu fırçasız DC motorlar

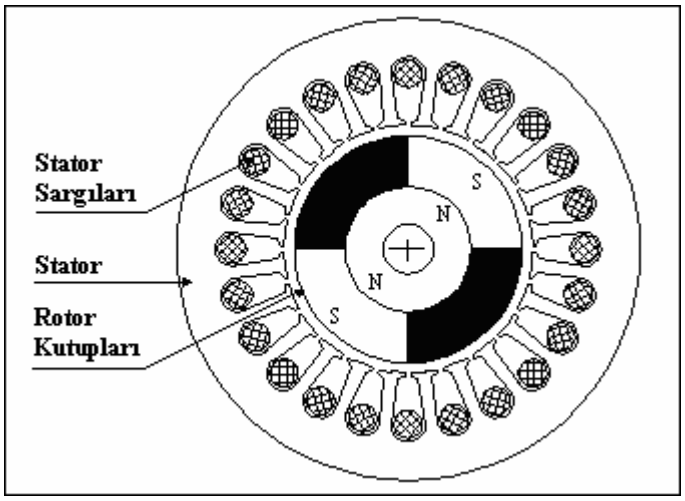
Fırçasız DC motorlar, rotorun içte veya dışta olmasına göre iç rotorlu fırçasız DC motorlar ve dış rotorlu fırçasız DC motorlar olarak da ikiye ayrılırlar.

İç rotorlu fırçasız DC motorlar, yapısal olarak senkron ve asenkron motorlara benzemektedirler. Dış tarafta bulunan stator, sabit endüvi sargılarını taşır. Rotorda ise mıknatıslar bulunur. Bu yapının en belirgin özelliği, eylemsizliğin düşük olması, bu nedenle de yüksek moment/atalet oranına sahip olmasıdır. İç rotorlu yapının dış rotorluya göre bir olumsuz yanı, özellikle yüksek hızlarda mıknatısların yerlerinden çıkma tehlikesidir. Bunu önlemek için rotorlar, yüksek elektriksel dirençli metal şeritler ile sarılırlar. Bir diğer olumsuz yanı ise, her ne kadar iç rotorlu yapıda statorun soğuması daha kolay olsa da dış rotorlu yapıda statorun seri üretimde sarılması ve işlenmesi daha kolay ve maliyeti daha azdır.

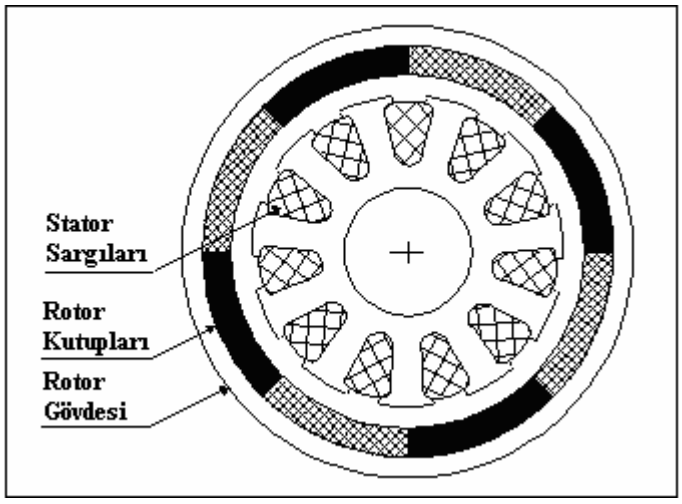
İç rotorlu fırçasız DC motorlar, genellikle endüstride servo motor ve kompresör uygulamalarında, özelliklede CNC tezgahlarında yüksek performanslı ana motor olarak kullanılırlar (Liu ve Cheng 1993). Şekil 3.5'te uygulamada sıkça karşılaşılan bir yapı görülmektedir.

Dış rotorlu yapı, fırçasız DC motor uygulamalarında en çok kullanılan yapıdır. Şekil 3.6'da görüldüğü gibi bu yapıda; endüvi sargılarını taşıyan stator içtedir. Mıknatısları taşıyan rotor ise dıştadır ve serbest olarak döner. Bu motorların

rotorlarında, iç rotorlu motorlara göre nispeten yüksek akı yoğunluğu gerekmediğinden, ferrit mıknatıslar daha fazla kullanım alanı bulmuştur. Stator ve rotor yapısı olarak bu motorlar, sabit mıknatıslı fırçalı doğru akım motorlarına çok benzemektedirler. Sargıların stator dış yüzeyinde olması, seri üretimde sarım işlemini önemli ölçüde kolaylaştırır. Bu tip motorlarda, iç rotorlu yapıda karşılaşılan yüksek hızlarda mıknatısların yerinden çıkma problemi yoktur. Ancak dış tarafta olduğu için mıknatısların kırılma problemi vardır.



Şekil 3.5 İç rotorlu fırçasız DC motorun mekanik yapısı



Şekil 3.6 Dış rotorlu fırçasız DC motorun mekanik yapısı

Dış rotorlu yapıda karşılaşılan bir diğer problem, rotordaki dengesizliğin etkisidir. Rotordaki küçük çapsal dengesizlikler, anma hızlarında bile, iç rotorlu yapıya göre nispeten büyük olan eylemsizliğin etkisi ile titreşimlere neden olur. Bu yapıdaki motorlar, yüksek eylemsizliklerinden dolayı ani yük değişimlerinde hız sabitliğini koruma özelliğine sahiptir. Bu yüzden, fan ve çamaşır makinesi uygulamalarında geniş kullanım alanı bulmuştur (Kim 2003).

3.1.1.3 Fırçasız DC motorlarda kullanılan sabit mıknatıslar

Sabit mıknatıslar, fırçasız DC motorların ana malzemeleridir. Fırçasız DC motorlarda kullanılan sabit mıknatıslar, B-H düzleminde geniş bir histerezis döngüsüne sahiptir. Son yirmi yılda sabit mıknatıs teknolojisi büyük ölçüde gelişmiştir. Günümüzde sabit mıknatıs malzemelerin pek çok tipleri mevcuttur. Bunlar Al-Ni-Co, ferrit (Seramik), toprak elementi samaryum-kobalt ve neodmiyum-iron-boron (NdFeB)'dir. Son yıllarda geliştirilen NdFeB mıknatıslar, en yüksek güç yoğunluğuna sahiptir. Ferrit mıknatıslar, ucuz maliyetinden dolayı en yaygın kullanılan mıknatıs çeşididir. Her mıknatıs çeşidi, fırçasız DC motorlarda farklı performans seviyelerine ve çeşitli kısıtlamalara yol açan farklı özelliklere sahiptir.

Sabit mıknatıs malzemelerin karakteristikleri Tablo 3.1'de karşılaştırılmaktadır. Ferrit mıknatıslar ucuzdur ancak çok daha düşük artık mıknatıslanma akı yoğunluğuna, artık mıknatısiyet yitirme kuvvetine ve güç üretimine sahiptir. Ayrıca ferrit mıknatısların kırılgan bir yapısı vardır. NdFeB mıknatıs en yüksek artık mıknatıslanmaya (B_r) ve artık mıknatıslık yitirme kuvvetine (H_c) sahiptir.

Bir motor için manyetik materyal seçiminde öncelikle dikkat edilecek husus, çalışma sıcaklığı aralığı, hacim başına moment ve mıknatısların çalışma ortam zorluklarıdır (Toliyat ve Gopalarathnam 2002). Yüksek güçlü kutuplar için doğal olarak geniş hacimli mıknatıslar gerekir. Sabit mıknatısların yüksek sıcaklığa maruz kalması, koersif gücünün ve akı yoğunluğunun bozulmasına neden olur. Bu yüzden yüksek çalışma sıcaklıklarında mıknatısların seçimi göz önüne alınmalıdır. Mıknatıslar, inverterden kaynaklanan hatalı kısa devre akımlarından dolayı

mıknatıslık özelliğini kaybedebilirler. Bu yüzden koruyucu elektronik akım koruma devreleri ile inverter akımı, güvenli sınırlarda tutulmalıdır.

Tablo 3.1 Sabit mıknatısların karakteristikleri

Al-Ni-Co	Yüksek B_r ve düşük H_c Mükemmel mekanik dayanım, büyük boyutlu sabit mıknatıslar için iyidir
Ferrit	Düşük B_r ve orta H_c Nispeten daha ucuz ve kolay kırılabilir
Nadir-toprak elemanı samaryum-kobalt	Yüksek B_r ve H_c Çok pahalı, küçük sıcaklık katsayısı
NdFeB	Yüksek B_r ve H_c Mükemmel performans, nispeten pahalı

3.1.1.4 Fırçasız DC motor kontrolünde kullanılan sensörler

Sensörler genel olarak, herhangi bir kaynaktan gelen işaret ya da uyarıyı alan ve bu işaretlere uygun bir sinyal üreten devre elemanlarıdır. Motor kontrolü açısından ise, denetlenen elektriksel ya da mekanik büyüklükten geri besleme almak ve kontrol devrelerinde işlenecek büyüklüğe dönüştürmek amacıyla kullanılan elektromanyetik, elektromekanik ya da optik elemanlardır. Motor kontrol uygulamalarında, rotorun açısal konumunu ve hızını algılamak için çoğunlukla enkoderler, hall sensörleri, resolver ve takogeneratörleri kullanılır.

Trapezoidal zıt emk'ya sahip fırçasız DC motorlarda rotor pozisyonu, genellikle hall sensörleri ile belirlenir. Hall sensörleri, silisyum temelli ve entegre sensör olmak üzere iki gruba ayrılır. Entegre sensörlerde elektronik arabirim devresi, silisyum sensörle aynı tabaka üzerinde imal edilir. Elektronik bir devre, zayıf olan hall sensör çıkışından az kayıplı olarak bozulmadan çıkış alınmasını sağlar ve çıkışı dijital olarak verir. Elektronik devre, bir eşik değeri içerir ve çıkışından iki durumlu bir sinyal elde edilir. Uygulanan manyetik alan, eşik değerin altında ise çıkış düşük, eşik değerin üstünde ise çıkış yüksek seviyededir.

3.1.2 Çalışma prensibi

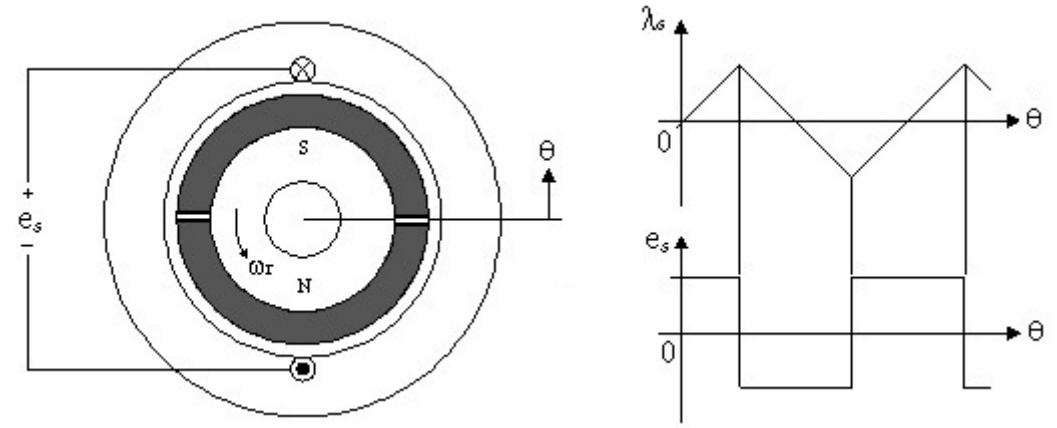
Fırçasız DC motor, sabit mıknatıslı bir rotora sahip ve elektronik komütasyon için rotor pozisyonu bilinmesi gereken senkron bir motordur. Bu motorlarda, yoğunlaştırılmış stator sargıları tarafından bir manyetik döner alan üretilmekte ve bu döner alan içine sabit mıknatıs kutuplara sahip rotor yerleştirilerek döner alan ile aynı hızda hareket etmesi sağlanmaktadır. Üretilen döner alan, rotor pozisyonuna göre anahtarlanan yarı iletken elemanların anahtarlanması ile oluşturulmaktadır. Asenkron motorlarda olduğu gibi, stator döner alanı ile rotor hızı arasında bir kayma yoktur. Fırçalı DC motorlarda kutup değişimi, komütatör ve fırçalarla mekanik bir şekilde yapılır. Mekanik komütatör, rotora sabitlendiği için anahtarlama anları, manyetik alanın değişen yönü ile otomatik olarak ayarlanır. Ancak fırçasız DC motorlarda kutup değişimi yarı iletken anahtarlarla elektronik olarak yapılır.

Fırçazsız DC motorlar, trapezoidal zıt emk'ya sahiptir ve sabit moment üretmek amacıyla kare dalga akımla çalışacak biçimde tasarlanırlar. Bu tip motorlar;

- Hava boşluğunda dikdörtgen dağılımlı mıknatıs akısı,
- Dikdörtgen akım şekli,
- Yoğunlaştırılmış stator sargıları,
- Sinüzoidal zıt emk'lı motorlara göre daha düşük üretim maliyeti,
- Sinüzoidal zıt emk'lı motorlara göre daha basit kontrol yapısı,
- Bir elektriksel çevrim için altı farklı komütasyon noktası,
- Komütasyon anlarında moment dalgalanması,

gibi özelliklere sahiptir. Bu tip fırçasız DC motorların uyartım akımı, kare dalga şeklindedir. Bu kare dalga akım, her faz sargısından bir periyot içerisinde 120° süreyle iki defa geçer. Trapezoidal zıt emk'lı fırçasız DC motorlar, sinüzoidal zıt emk'lı motorlarla karşılaştırıldığında, bazı önemli sistem basitleştirmeleri sağlar. Özellikle bir elektriksel çevrimde sadece altı komütasyon anı gerektiği için, rotor pozisyon sensörlerinin kararlılık gereksinimleri çok daha azdır.

Fırçasız DC motor sürücülerinde moment üretmek için, fırçasız DC motorun sargılarında oluşan zıt emk karakteristiklerini çok iyi anlamak gereklidir. Şekil 3.7'de, akı bağıntısı ve çeşitli pozisyonlarda meydana gelen tek bir bobinin zıt emk'sı görülmektedir.



Şekil 3.7 Tek bobinin zıt emk dalga şekli

Şekil 3.7 göz önüne alındığında, tek bir bobinin akı bağıntısı matematiksel olarak aşağıdaki gibi türetilebilir;

$$\lambda_s = (\pi r \ell) B_f \left(\frac{\theta}{\pi/2}\right) \qquad (-\pi/2 \le \theta \le \pi/2) \qquad (3.1)$$

Burada r; rotor yarıçapı, ℓ; rotor uzunluğu, B_f; sabit mıknatısın akı yoğunluğu, θ ise rotor pozisyonudur. Meydana gelen zıt-emk'nın büyüklüğü ise;

$$e_s = \frac{d\lambda_s}{dt} = \frac{d\lambda_s}{d\theta}\frac{d\theta}{dt} = \frac{\pi r \ell B_f \omega_r}{\pi/2} = 2B_f \ell r \omega_r \qquad (3.2)$$

olmaktadır. Burada ω_r, elektriksel olarak rotor açısal hızını (rad/sn) göstermektedir. Eş merkezli sargılarda meydana gelen toplam zıt emk'nın büyüklüğü aşağıdaki gibi yazılabilir.

$$E = 2N_s B_f \ell r \omega_r \qquad (3.3)$$

Burada N_s, bir faz sargısındaki iletken sayısıdır. Fırçasız DC motorlarda moment, enerjilenen stator fazlarıyla rotor kutuplarının aynı hizaya gelme eğilimi ile üretilir. Analitik olarak, fırçasız DC motorlardaki moment;

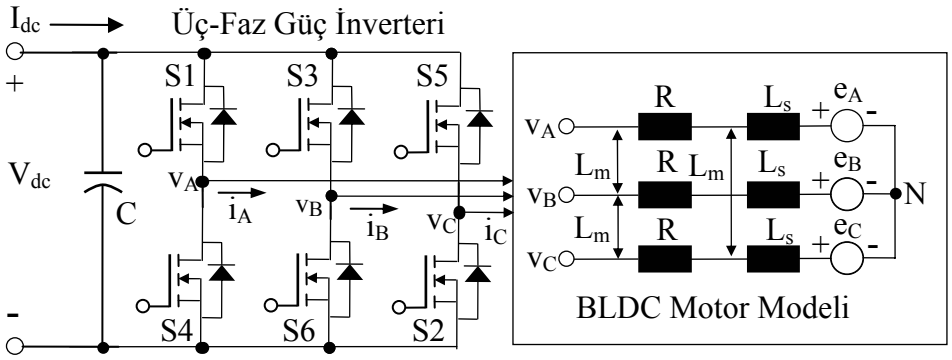
$$T_e = \frac{e_A i_A + e_B i_B + e_C i_C}{\omega_r} \tag{3.4}$$

şeklinde olmaktadır. Burada e_A, e_B, e_C; zıt emk değerlerini ve i_A, i_B, i_C; faz akımlarını göstermektedir. Mekanik hareket denklemlerinden rotor açısal hızının değişimi,

$$\frac{d}{dt}\omega_r = (\frac{p}{2}(T_e - T_L) - B\omega_r)/J \tag{3.5}$$

ifadesiyle verilebilir. Burada p motor kutup sayısını, T_L yük momentini, J atalet momentini ve B sürtünmeyi temsil etmektedir.

Denklem (3.4)'de görüldüğü gibi fırçasız DC motorlarda sabit moment üretmek için tahrik edilen fazlar, zıt emk'nın düz kısımları ile senkronize edilmelidir. Pratik sistemlerde üretilen moment, ideal bir şekilde düz değildir ve birkaç sebepten dolayı bir dalgalılığa sahiptir. Fırçasız DC motorların kontrolünde, motor sargılarının beslenmesi için inverter devresine ihtiyaç vardır. Şekil 3.8'de, inverter ve bir fırçasız DC motor eşdeğer devresi görülmektedir.



Şekil 3.8 İnverter ve Fırçasız DC motor eşdeğer devresi

Fırçasız DC motorlar ile sabit mıknatıslı DC motorlar arasındaki benzerliği göstermek amacıyla, bir faza ait zıt emk geriliminin genliği, denklem (3.3) referans alınarak aşağıdaki gibi tekrar yazılabilir.

$$E=k\phi\omega_r \tag{3.6}$$

Burada k her bir faz için bobin sarım sayısına bağlı katsayı, ϕ sabit mıknatıs akısı ve ω_r rotor hızıdır. Her bir 120°'lik periyot boyunca, mekaniğe çevrilen anlık elektriksel güç, seri iki fazın toplamından bulunur.

$$P_o=\omega_r T_e = 2\ E\ I \tag{3.7}$$

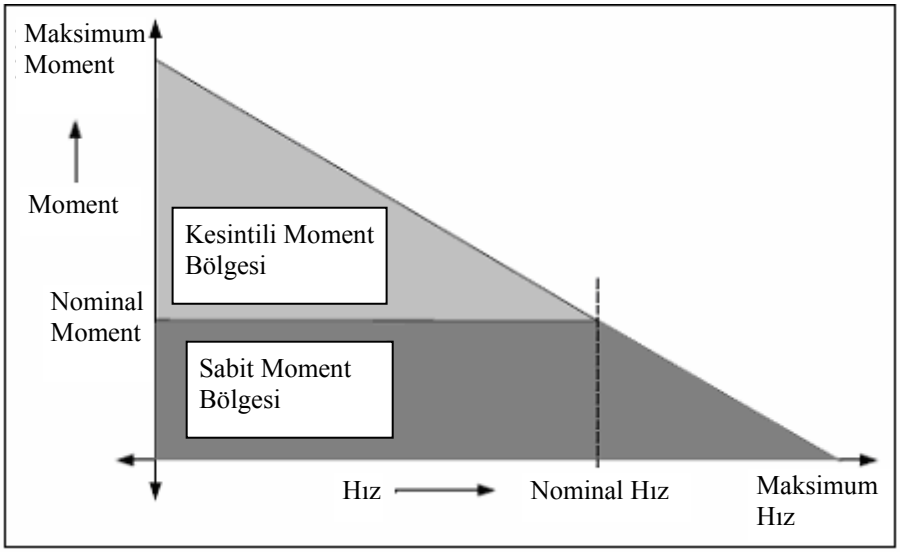
Burada, T_e çıkış momenti, I anlık faz akımıdır. Yukarıdaki (3.6) ve (3.7) denklemlerinden çıkış momenti yazılırsa,

$$T_e=2\ k\ \phi\ I = k_t\ I \tag{3.8}$$

olmaktadır. Burada k_t, moment sabitidir.

Geleneksel sabit mıknatıslı bir DC motorda moment-hız karakteristiği, endüvi reaksiyonu etkisinin önemli olduğu yüksek moment seviyeleri hariç lineerdir. "Fırçasız DC motor" terimi; fırçalı DC motordaki gibi lineer bir moment-hız karakteristiğine sahip a.c. motor, inverter ve rotor pozisyon sensörü bileşiminden oluşan sürücü sistemini ifade etmek için kullanılmaktadır Yani, fırçasız DC motorların moment-hız karakteristikleri ve performans denklemleri hemen hemen DC motorlarla aynıdır (Hendershot ve Miller 1994). Şekil 3.9'da bir fırçasız DC motorun hız/moment eğrisi görülmektedir.

Fırçasız DC motorlarda bir stator fazının moment dağılımı, faz akımı ve rotor pozisyonunun bir fonksiyonudur. Bir stator fazına, sabit bir doğru akım uygulanırsa ve rotor harici bir güç tarafından döndürülürse, sargı akımı ve manyetik akının birbirini etkilemesinden dolayı meydana gelen moment, mil pozisyonu ile periyodik olarak değişecektir. Bu karakteristik, motorun moment fonksiyonu veya statik moment/açı karakteristiği olarak bilinir.



Şekil 3.9 Fırçasız DC motorların hız/moment eğrisi

3.1.3 Matematiksel modeli

Fırçasız DC motor sürücülerin matematiksel analizleri ile ilgili değişik çalışmalar mevcuttur. Bu çalışmalarda fırçasız DC motor modeli; durum-uzayı denklemleri, fourier serileri ve d-q eksen modelleri şeklinde elde edilmiştir (Luk ve Lee 1994, Bonvin ve Perriard 2000, Lee ve Ehsani 2003, Singh ve Jain 2003). Fırçasız DC motorların zıt emk ve akım dalga şekillerinin sinüzoidal olmayan yapısından dolayı, d-q modelinin kullanımı elverişsiz olmaktadır. Ayrıca, bu parametreleri sinüzoidal kabul ederek model elde etmek anlamsızdır. Modelleme ve simülasyon çalışmalarında, faz değişimi yaklaşımını kullanmak daha kolay olmaktadır (Toliyat ve Gopalarathnam 2002).

Fırçasız DC motor sürücülerin matematiksel analizini yapabilmek için aşağıdaki kabullerin yapılması, kolaylık sağlamaktadır (Lee ve Ehsani 2003, Kim 2003).

- Motor doyumda değildir.
- Motor sargıları sabit bir dirence, öz endüktansa ve karşılıklı endüktansa sahiptir. Tüm fazların dirençleri ve endüktansları eşittir.
- Fazların hepsi aynı zıt emk şekline sahiptir.

- İnverterdeki güç yarıiletkenleri idealdir.
- Demir kayıpları ihmal edilebilir düzeydedir.

Yukarıdaki kabuller dikkate alınarak, Şekil 3.8 ile verilen bir fırçasız DC motorun gerilim denklemleri, matrisel formda,

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix} \quad (3.9)$$

$(L=L_s-L_m)$

ile ifade edilebilir. Burada v_A, v_B ve v_C faz gerilimlerini, e_A, e_B ve e_C zıt emk'ları, i_A, i_B ve i_C faz akımlarını, L_s öz endüktansı, L_m karşılıklı endüktansı göstermektedir.

Fırçasız DC motorların zıt emk dalga şekilleri, rotor pozisyonuna (θ) bağlı trapezoidal bir yapıdadır. Matematiksel olarak zıt emk değerleri;

$$\begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix} = E \begin{bmatrix} f_A(\theta) \\ f_B(\theta) \\ f_C(\theta) \end{bmatrix} \quad (3.10)$$

$$(E = k_e \omega_r)\,,\ (\frac{d}{dt}\theta = \omega_r)$$

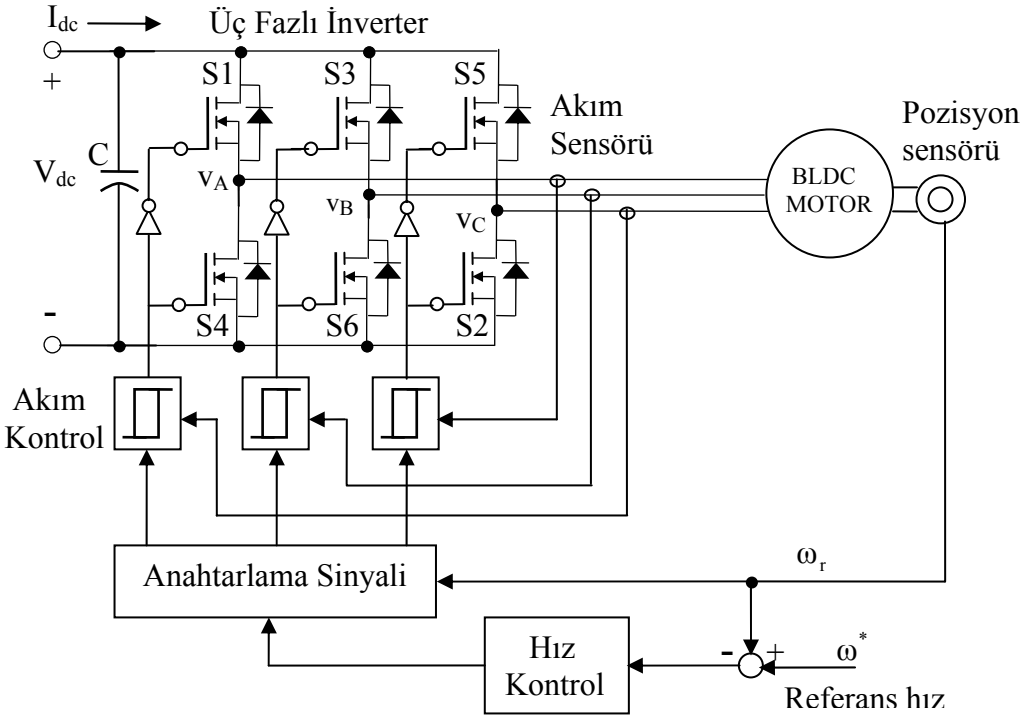
ile gösterilebilir. Burada k_e zıt emk sabiti, $f_A(\theta)$, $f_B(\theta)$ ve $f_C(\theta)$ sırasıyla A, B ve C fazlarına ait zıt emk dalga şeklini tanımlayan şekil fonksiyonlarıdır.

3.1.4 Fırçasız DC motor sürücüleri

Tipik bir fırçasız DC motor sürücü sistemi; fırçasız DC motor, inverter ve sayısal bir kontrol sisteminden oluşur. İnverter, bir köprü diyot vasıtasıyla şebekeden elde edilen bir DC güç kaynağına veya bir aküye bağlanır. Sayısal kontrolör, inverter vasıtasıyla fırçasız DC motorun her bir fazını sıra ile enerjilendirir. Sabit moment üretmek için bu enerjilendirme, rotor pozisyonu ile senkron yapılır. Bu nedenle pozisyon geri beslemesi için sistemde genellikle, fırçasız DC motorun içine

yerleştirilen mekanik bir pozisyon sensörü kullanılmaktadır. Şekil 3.10'da pozisyon geri beslemeli tipik bir fırçasız DC motor sürücü sistemi görülmektedir.

Hız kontrol birimi, mekanik pozisyonu okur ve hangi fazın anahtarlanacağına karar verir. İdeal olarak her bir stator fazı, darbeli kare dalga akımla tahrik edilir. Akım darbesinin büyüklüğü ve anahtarlama örneğinin zamanlaması, mekanik parametrelerle birlikte meydana gelen momentin büyüklüğü üzerinde doğrudan bir etkiye sahiptir.



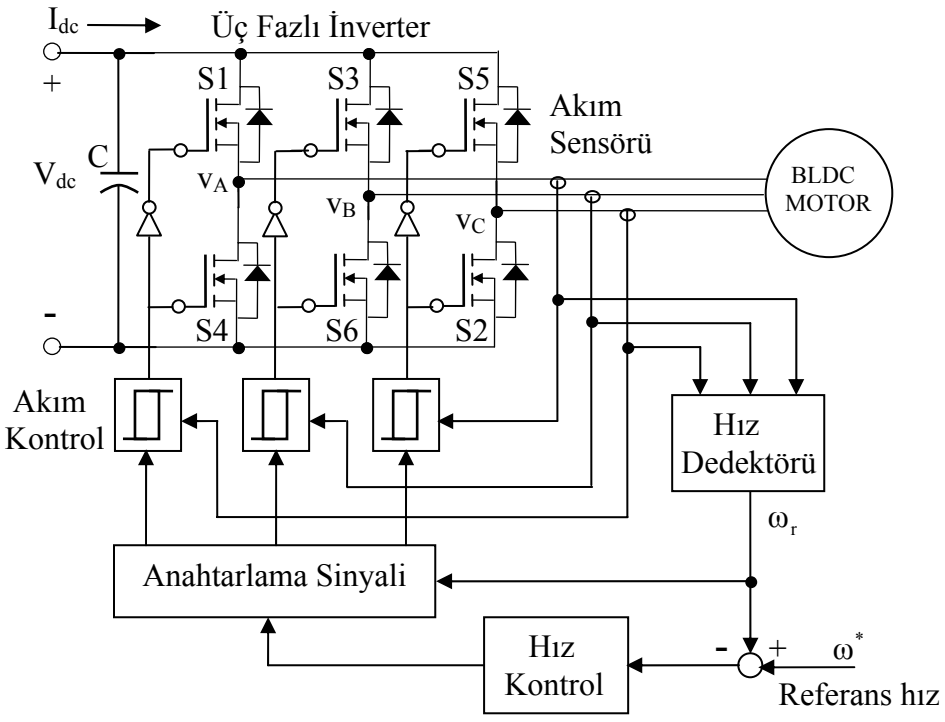
Şekil 3.10 Pozisyon geri beslemeli tipik bir fırçasız DC motor sürücü sistemi

Stator bobinleri, standart üç fazlı tam adımlı olarak yerleştirilir. Böylece fazın trapezoidal zıt emk dalgaları, elektriksel olarak 120° aralıklarla meydana gelir. Meydana gelen akım darbeleri, 120° süresince iletim durumunda, 60° süresince ise kesim durumundadır. Bu, her bir fazdan, elektriksel 360° periyodunun 2/3'ü süresince akım aktığı anlamına gelmektedir. Bu akımın 120°'si pozitif 120°'si negatif yönde akmaktadır. Maksimum ve sabit moment/amper'li motoru sürmek için hat akım darbelerinin, her bir fazın hat-nötr zıt emk gerilimleri ile senkron olması istenir.

3.1.5 Fırçasız DC motorların pozisyon sensörsüz kontrolü

Fırçasız DC motorun moment üretim prensibinden anlaşılacağı gibi sabit moment üretmek için, stator akımı uyartımlarının, rotor pozisyonu ile senkronize edilmesi gerektiği açıktır. Bu nedenle, fırçasız DC motorların kontrolü için rotor pozisyon bilgisi temeldir. Fırçasız DC motorlarda, elektriksel senkronizasyon için anahtarlanması gereken fazın belirlenmesinde, hall etkili pozisyon sensörleri kullanılır. Yüksek performanslı sürücüler için genellikle, daha yüksek çözünürlüğe sahip optik enkoderler veya resolverler kullanılır. Ancak bu mekanik pozisyon sensörleri, gerçek uygulamalarda bazı problemlere yol açar (Toliyat ve Gopalarathnam 2002). Şekil 3.11'de fırçasız DC motorların pozisyon sensörsüz kontrolü görülmektedir.

Mekanik pozisyon sensörleri ile ilgili ilk problem, sensörlerin pahalı olmasıdır. Rotor pozisyon sensörlerinin yüksek maliyeti, ev aletleri ve toptan üretim endüstrisi gibi maliyete duyarlı uygulamalar için bir sorun teşkil etmektedir.



Şekil 3.11 Fırçasız DC motorların pozisyon sensörsüz kontrolü

Mekanik pozisyon sensörleri ile ilgili ikinci problem, güvenilirliğin az olmasıdır. Eğer fırçasız DC motor, özellikle sıcak, soğuk veya nemli bir çevreye yerleştirilirse, pozisyon sensörleri çalışmayabilir veya sensörlerden alınan bilgi kaybolabilir ya da hatalı olabilir (Kim ve Youn 2002).

Mekanik pozisyon sensörlerinin üçüncü problemi, elektrik motorunun içinde veya etrafında pozisyon sensörlerinin yerleştirilebileceği bir yere ihtiyaç duyması yani ilave hacme sebep olmalarıdır. Bir örnek olarak klima kompresörlerinde kullanılan fırçasız DC motorlar, pozisyon sensörlerinin yerleştirilmesinin yapısal zorluğundan dolayı genellikle pozisyon sensörleri içermez. Ayrıca uzay araçları ve uçaklardaki gibi sınırlı alana sahip uygulamalarda mekanik sensörler, fırçasız DC motor sürücüsünün boyutunu büyütmekte ve uygulamaların gerçekleştirilmesine engel olmaktadır. Ayrıca mekanik pozisyon sensörleri, güç kablolarının yanında çalışırken bozuk veya hatalı sinyal üretebilmektedir. Bu yüzden harici pozisyon sensörlerinin kullanımı sınırlı alana sahip, maliyete duyarlı ve ayrıca güvenilirliğin çok önemli olduğu uygulamalarda tavsiye edilmemektedir. Pozisyon sensörlerinin kullanımının bu problemleri, araştırmacıların ve endüstrinin rotor pozisyon bilgisini elde etmek için alternatif metotları araştırmasına yol açmıştır (Su ve McKeever 2004).

Literatürde fırçasız DC motorlar için pek çok sensörsüz kontrol metodu görülmektedir. Kullanılan sensörsüz kontrol metotları öncelikle üç kategoride sınıflanabilir (Kim 2003);

1) Zıt emk temelli metotlar
2) Akı kestirim metodu
3) Önceki iki kategoriye girmeyen diğer özel metotlar.

Kaynaklarda pozisyon sensörsüz kontrol ile ilgili oldukça detaylı bilgiler verilmektedir. Bu çalışmada zıt-emk temelli yöntem kullanılmıştır.

3.2 Bulanık Denetleyiciler

Denetleyici tasarımının matematiksel modele ihtiyaç duyulmadan gerçekleştirilmesi ve lineer olmayan sistemlerde oldukça iyi sonuçlar vermesi nedeniyle, bulanık denetleyiciler, son yıllarda birçok uygulamada sıkça 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.

3.2.1 Denetleyici tasarımında temel ilkeler

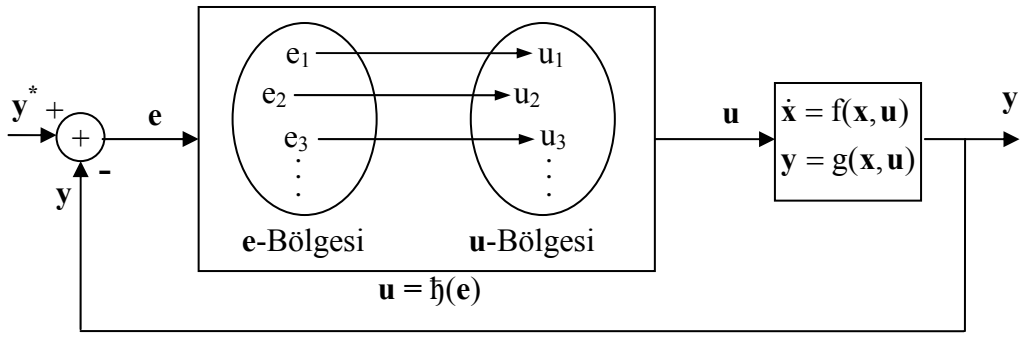
Lineer olmayan sistemlerin kontrolünde, yine lineer olmayan denetleyicilerin kullanılması gerekmektedir. Ancak, lineer olmayan denetleyicilerin tasarımı oldukça zor ve karmaşıktır. Bu yüzden, denetleyici tasarımında yeni arayışlar hızla devam etmekte ve bunun sonucunda yeni yöntemler ortaya konulmaktadır. Bu yöntemlerden bir tanesi, lineer olmayan kontrol problemini çözmek için bulanık denetleyicilerin kullanılmasıdır. Klasik denetleyiciler ile bulanık denetleyiciler arasındaki fark aşağıdaki gibi açıklanabilir. Durum uzayında, lineer olmayan bir dinamik sistem;

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= g(\mathbf{x}, \mathbf{u}) \end{aligned} \tag{3.11}$$

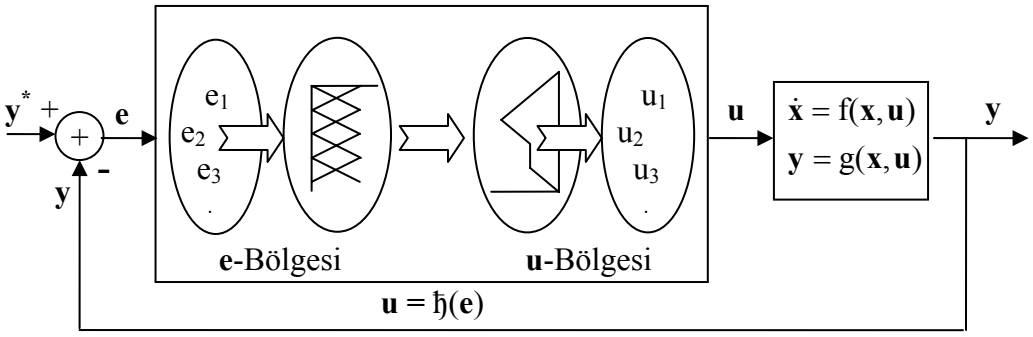
olarak verilsin. Burada; $\mathbf{x} \in R^n$ durum vektörünü, $\mathbf{u} \in R^r$ ve $\mathbf{y} \in R^m$ sırasıyla, sistem giriş ve çıkış vektörlerini ve r, m ve n ise tamsayıları göstermektedir. Ayrıca, $f : R^n \times R^r \rightarrow R^n$ ve $g : R^n \times R^r \rightarrow R^m$ şartlarında $f(.) \in R^n$ ile $g(.) \in R^m$ fonksiyonlarının dönüşümleri, düzgün ve yeterince iyidir. Denetleyici tasarımında temel amaç; geri besleme bilgileri kullanılarak arzu edilen çıkışı sağlayacak en uygun kontrol algoritmasının geliştirilmesidir. Şekil 3.12'de verilen sistemde giriş-çıkış ifadeleri cinsinden hata vektörü, $\mathbf{e} = \mathbf{y}^* - \mathbf{y}$ olarak tanımlanır. Burada; $\mathbf{y}^*$ arzu edilen çıkış vektörünü ve $\mathbf{y}$ ise sistemin gerçek çıkış vektörünü göstermektedir. Denklem (3.11) ile matematiksel olarak anlatılmak istenen; kararlılık sınırları içinde, hata tanım kümesinden ($\mathbf{e}$), sistem giriş kümesine ($\mathbf{u}$) en uygun dönüşümün bulunmasıdır. Matematiksel olarak bu dönüşüm $\mathbf{u} = \hbar(\mathbf{e})$ olarak gösterilebilir.

Klasik kontrol sistemine karşılık bulanık denetleyicilerde ise, Şekil 3.13'de görüldüğü gibi, bulanık dönüşüm ve mantıksal çıkarım işlemleri uygulanmaktadır.

Bulanık denetleyicilerde, dönüşüm işlemine bulanık çıkarım denilmektedir. Bulanık çıkarım işleminde ilk olarak keskin kümeler, bulanık kümelere dönüştürülmekte ve gerekli mantıksal çıkarım işlemleri yapılmaktadır. Daha sonra, tersine dönüşüm ile tekrar keskin kümeler elde edilmektedir. Bu olaya bulanıklaştırma ve durulaştırma denilmektedir. Ayrıca giriş ve çıkış değerlerinin normalize edilmesi gerekebilir. Klasik denetleyicilerde tanımlanan $\mathbf{u} = \hbar(\mathbf{e})$ ifadesi, bulanık denetleyicilerde üç farklı aşamada gerçekleştirilmektedir (Ashrafzadeh 1996).



Şekil 3.12 Klasik geri beslemeli kontrol sistemi

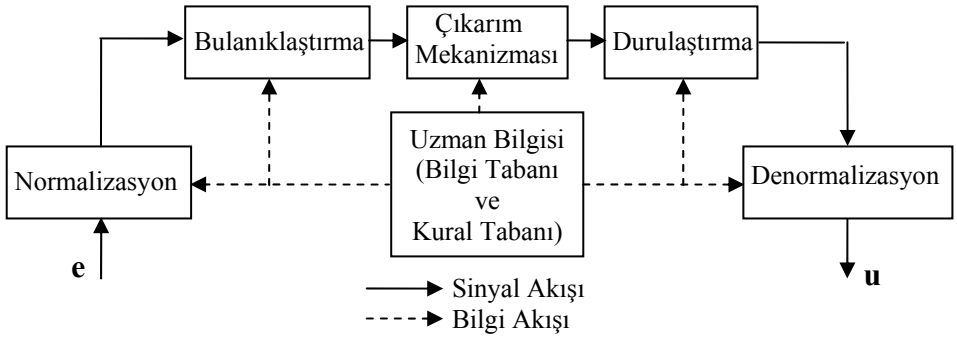


Şekil 3.13 Bulanık mantık kontrol sistemi

3.2.2 Bulanık denetleyicilerin temel yapısı

Şekil 3.14'de bir bulanık denetleyicinin temel yapısı görülmektedir. Şekilde görüldüğü gibi, bir bulanık denetleyici beş kısımdan oluşmaktadır. Bunlar;

- Giriş değerinin normalize edilmesi,
- Normalize edilen giriş değerinin bulanıklaştırılması,
- Kural tabanına göre çıkarım mekanizmasının çalıştırılması,
- Çıkış değerlerinin durulaştırılması,
- Çıkışın ters yönde normalize edilmesidir.



Şekil 3.14 Bulanık denetleyicinin temel yapısı

3.2.2.1 Normalizasyon parametreleri

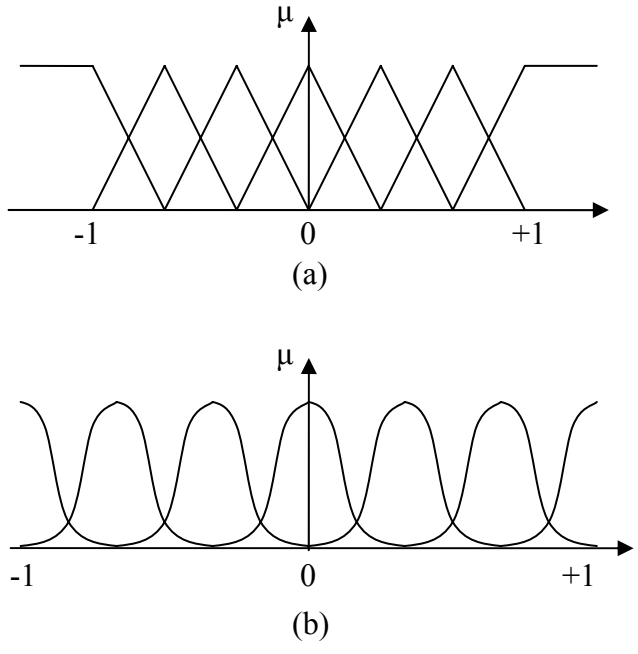
Şekil 3.13 ile verilen bir bulanık kontrol sisteminde, giriş ve çıkış vektörleri normalizasyon işlemine tabi tutularak, geri besleme, referans değer ve uç kontrol elemanı gibi fiziksel değişkenlerden bağımsız, üyelik fonksiyonları [-1, +1] aralığında çalışan bir yapıya dönüştürülür. Normalizasyon işlemleri, lineer bir karakteristiğe sahip olmasına rağmen, denetleyici performansında kritik öneme sahiptir. Her bir giriş ve çıkışa ayrı ayrı normalizasyon işlemi uygulanmalıdır.

3.2.2.2 Bulanıklaştırma ve durulaştırma

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 trapezoidal, üçgen, gausyen, quadratik ve cos fonksiyonu formunda değişik üyelik fonksiyonları tanımlanmaktadır. Şekil 3.15'de üçgen ve gausyen formda iki ayrı üyelik fonksiyonunun yapısı verilmektedir.

Burada her bir giriş değer uzayı, yedi farklı bulanık alt kümeye ayrılmıştır. Ayrıca, bulanık denetleyicinin çıkış vektörünün de, gerçek bir değer kümesi olması için, bir bulanık küme olan bulanık çıkarım işlemi sonuçlarının gerçek değerlere dönüştürülmesi gereklidir. Bu işleme, başlangıçta yapılan bulanıklaştırma işleminin tersi olan durulaştırma işlemi denir. Durulaştırma işlemlerinde maksimum üyelik, ağırlık merkezi, ağırlık ortalaması ve maksimum orta (mean-max) değeri gibi değişik yöntemler kullanılmaktadır (Elmas 2003).



Şekil 3.15 (a) Üçgen (b) Gausyen üyelik fonksiyonları

3.2.2.3 Çıkarım mekanizması

Şekil 3.13 ile verilen 2 girişli tek çıkışlı bulanık denetleyicide giriş vektörü **e**=(e_1,e_2) = ($e, \dot{e}$) ve çıkış değişkeni (u) olarak tanımlanmıştır. Burada; e_1 için e-hata ve e_2 için ise $\dot{e}$-hata değişimi giriş olarak alınmıştır. Giriş uzayından, çıkış uzayına dönüşüm için kullanılan bulanık fonksiyon (R_f) ise, bulanık denetleyici çıkışı için aşağıdaki bağıntı verilebilir.

$$u = (e \otimes \dot{e}) \circ R_f \tag{3.12}$$

Burada; $\otimes$ işareti kartezyen çarpım operatörünü, "o" sembolü ise (min-max gibi) bulanık çıkarım mekanizmasını göstermektedir. Bulanık çıkarım işlemleri için en çok kullanılan yöntemler; Max-Çarpım, Min-Max, Tsukamoto ve Takagi-Sugeno çıkarım yöntemleridir. Uygulamada en sık Min-Max çıkarım yöntemi kullanılmaktadır. Bulanık denetleyici için dilsel olarak tanımlanan kural tabanı, bir karar tablosu formunda ifade edilebilir. İki girişli ($e, \dot{e}$) ve tek çıkışlı (u) bulanık denetleyicinin karar tablosu Tablo 3.2'de görülmektedir. Bu tabloda; denetleyici girişleri ve çıkışı için, Negatif Büyük (NB), Negatif Orta (NO), Negatif Küçük (NK), Sıfır (S), Pozitif Küçük (PK), Pozitif Orta (PO) ve Pozitif Büyük (PB) sembolleri kullanılmıştır. Böyle bir tabloya, görünüşü açısından kayan tip karar tablosu veya kısaca kayan tip tablo denilmektedir.

Tablo 3.2 Kayan tip karar tablosu

ė \ e	**PB**	**PO**	**PK**	**S**	**NK**	**NO**	**NB**
NB	S	PK	PK	PO	PO	PB	PB
NO	NK	S	PK	PK	PO	PO	PB
NK	NK	NK	S	PK	PK	PO	PO
S	NO	NK	NK	S	PK	PK	PO
PK	NO	NO	NK	NK	S	PK	PK
PO	NB	NO	NO	NK	NK	S	PK
PB	NB	NB	NO	NO	NK	NK	S

3.2.3 Bulanık denetleyici tipleri

Bulanık denetleyiciler, kontrol sistemlerinde açık ve kapalı çevrim denetleyici olarak kullanılmaktadır. Uygulamada klasik kontrol teorisi ile bağlantılı olarak dört farklı tip bulanık denetleyici öne çıkmaktadır. Bunlar; P, PD, PI ve PID tip bulanık denetleyicilerdir (Tang ve ark. 2001, Reznik 1997).

- **P tip bulanık denetleyiciler**

  Geleneksel oransal denetleyici matematiksel olarak,

  $$u = K_P\, e \tag{3.13}$$

ile verilir. Bu denklem bulanık denetleyicilerde dilsel değişken (DD) olarak sembolik formda;

EĞER $e(n) = DD_e$ ise **O HALDE** $u(n) = DD_u$

şeklinde yazılır.

- **PD tip bulanık denetleyiciler**

  Geleneksel PD denetleyicinin çıkış ifadesi aşağıdaki gibi verilmektedir.

  $$u = K_P\, e + K_D\, \dot{e} \tag{3.14}$$

Burada; e ve $\dot{e}$ sırasıyla denetleyici girişine uygulanan hata ve hata değişimi, K_P ve K_D oransal ve türevsel kazanç sabitleridir. Denklem (3.14) ile verilen ifade, bulanık denetleyicilerde dilsel değişken (DD) olarak aşağıdaki gibi tanımlanır.

EĞER $e(n) = DD_e$ **ve** $\dot{e}(n) = DD_{\dot{e}}$ **ise O HALDE** $u(n) = DD_u$

Burada; DD_e, ifadesi e için dilsel değişken, $DD_{\dot{e}}$ ifadesi $\dot{e}$ için dilsel değişken, DD_u ifadesi u için dilsel değişken ve n ise işlem adımıdır. Bir denklemin yukarıdaki gibi dilsel olarak sembolik gösterimi, bulanık kurallar olarak bilinir ve bir bulanık kural kümesi, bir diferansiyel denklemin tüm dinamiklerini taklit eder.

- **PI tip bulanık denetleyiciler**

Geleneksel PI denetleyiciler matematiksel olarak aşağıdaki gibi tanımlanabilir. Buna göre çıkış;

$$u = K_P\, e + K_I \int e\, dt \tag{3.15}$$

olarak verilir. Burada; K_P ve K_I sırasıyla oransal ve integral kazanç sabitleridir. Eğer yukarıdaki integral denklemi bir diferansiyel denklem formuna dönüştürülürse, aşağıdaki gibi yazılabilir.

$$\dot{u} = K_P\, \dot{e} + K_I\, e \tag{3.16}$$

PD tip bulanık denetleyicilerde olduğu gibi PI tip bulanık denetleyiciler de aşağıdaki kurallar ile modellenir. Buna göre;

EĞER $e(n) = DD_e$ **ve** $\dot{e}(n) = DD_{\dot{e}}$ ise **O HALDE** $\dot{u}(n) = DD_{\dot{u}}$

olarak yazılır. Bu tipte, denetleyici çıkışı artan bir yapıdadır ve bu yüzden bulanık denetleyici dışında (u) kontrol çıkış değerinin üretilmesi için bir integratör gereklidir.

- **PID tip bulanık denetleyiciler**

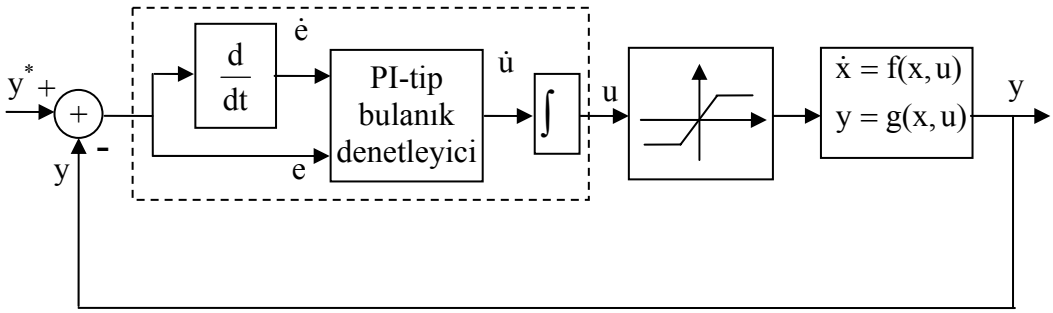
Geleneksel bir PID tip denetleyicinin matematiksel ifadesi;

$$u = K_P\, e + K_D\, \dot{e} + K_I \int e\, dt \tag{3.17}$$

şeklindedir. Böyle bir denklem bulanık denetleyici için sembolik formda;

EĞER $e(n) = DD_e$ **ve** $\dot{e}(n) = DD_{\dot{e}}$ **ve** $\Sigma e = DD_{\Sigma e}$ ise **O HALDE** $u(n) = DD_u$

şeklinde yazılır. Bu ifade PI ile PD tip bulanık denetleyicilerin birleştirilmesi demektir. Buna göre; PID tip denetleyici için kurallar tablo oluşturmak yerine, satır satır yazılarak elde edilir. PI tip bulanık denetleyici tasarımı daha kolay olması ve sürekli durum hatasını yok etmesi nedeniyle bu tez çalışmasında tercih edilmiştir. Bu yüzden, bundan sonra PI tip bulanık denetleyici yerine sadece bulanık denetleyici tanımı kullanılacaktır. Şekil 3.16'da, genel bir PI tip bulanık denetleyici görülmektedir.



Şekil 3.16 Kapalı çevrim kontrol sisteminde PI tip bulanık denetleyici

3.2.4 Bulanık denetleyici karakteristikleri

Bulanık denetleyicilerin önemini kavramak için, bu sistemlere farklı açılardan bakılmalıdır. Fonksiyonel açıdan bakıldığında, Şekil 3.16 ile verilen sistemin, bir statik transfer fonksiyonuna sahip olduğu açıktır. Otomatik olarak bu özellik, bulanık denetleyicinin giriş ve çıkışına bazı dinamik elemanların ilave edilebilmesi demektir. Görülüyor ki; kontrol sistemlerinde bulanık denetleyiciler statik bir karakteristiğe sahip iken, tüm denetleyicinin bir dinamik transfer fonksiyona sahip olduğu görülmektedir.

Denetleyicileri, lineerlik ve lineersizlik açısından da değerlendirmek gereklidir. Bulanık denetleyiciler, girişten çıkışa lineer olmayan bir dönüşüm gerçekleştirirler. Üstelik bulanık sistemler, özellikle de bulanık denetleyiciler, bir evrensel yaklaştırıcıdır. Şekil 3.16'daki bulanık denetleyici için bunun anlamı, herhangi bir lineer olmayan statik fonksiyonu, arzu edilen doğrulukta bir dereceye yaklaştırmasıdır. Bu özelliğinden dolayı bulanık denetleyiciler, lineer olmayan birçok sistemin kontrolünde oldukça etkilidir. İlave olarak bulanık denetleyiciler, yapısı gereği kayan tip tasarımı gibi değişik özelliklere sahiptir.

- **Bir statik transfer fonksiyonu olarak bulanık denetleyiciler**

Bulanık denetleyiciler, giriş-çıkış dönüşüm operatörleri gibi görülebilirler. Şekil 3.16 ile gösterilen bulanık denetleyicinin, bir statik transfer fonksiyonuna mı yoksa bir dinamik transfer fonksiyonuna mı sahip olduğu incelenebilir. Örnek olarak, PI tip bulanık denetleyicinin kural ifadesi tekrar yazılırsa;

EĞER $e(n) = DD_e$ **ve** $\dot{e}(n) = DD_{\dot{e}}$ ise **O HALDE** $\dot{u}(n) = DD_{\dot{u}}$

Bu ifade dolaylı olarak, sabit n değeri için denetleyici çıkışının bir önceki değere bağlı olmamasından dolayı, bulanık denetleyicilerde hiçbir dinamik etkinin olmadığını göstermektedir. Bu özellik, Şekil 3.16'da verilen PI tip bulanık denetleyicileri, statik transfer fonksiyonu içeren bir denetleyici yapmaktadır. Bu kavram, bulanık denetleyici dışında gerçekleştirilen türev ve integral gibi, zaman harici tüm operatörlerden elde edilebilir. Bu yüzden tüm denetleyicinin dinamik davranışı, giriş-çıkış sinyallerinin türev işlemi gibi ön filtreleme ya da integral işlemi gibi son filtreleme işlemleriyle oluşur. Bu, tüm denetleyicilerde $e \rightarrow u$, $(e, \dot{e}) \rightarrow \dot{u}$ dönüşümleri yapılırken oluşmaktadır.

- **Lineer olmayan bir eleman olarak bulanık denetleyiciler**

Bulanık denetleyiciler, genellikle lineer olmayan bir transfer fonksiyonuna sahiptir. Bu özellik, lineer olmayan kontrol uygulamaları için bulanık denetleyicileri etkili kılmaktadır. Örneğin, yukarıda PD tip bulanık denetleyici için tanımlanan kural ifadesi, lineer olmayan bir işlemdir ve bu şekilde oluşturulan kural tablosu da lineer olmayan diferansiyel denklem modeline karşılık gelmektedir. Yukarıdaki gibi kural tabanlı modellenebilen lineer olmayan diferansiyel denklemlere, nitel diferansiyel denklemler denir. Bulanık denetleyicilerde lineersizliğin sebebi potansiyel olarak kurallardan oluşurken, diğer yandan, bulanıklaştırma, bulanık çıkarım ve durulaştırma işlemleri de lineer olmayan bir doğaya sahiptirler. Ancak, belirtilmesi gereken nokta; üyelik fonksiyonları lineer olmayan bir yapı olmasına rağmen, yaygın görüşe göre sistem davranışının lineer olmamasının ana nedeni, çıkarım mekanizmasındaki kurallardan kaynaklanmaktadır (Ashrafzadeh ve ark. 1997). Bu nokta, üyelik fonksiyonları ve kurallar açıklandığında daha açık ortaya çıkacaktır.

- **Evrensel yaklaşımcı olarak bulanık denetleyiciler**

Bulanık denetleyicilerin bir evrensel yaklaşımcı olduğu ispatlanmıştır (Ashrafzadeh 1996). Daha önce bahsedildiği gibi, bir bulanık denetleyici aşağıdaki denklemde lineer olmayan bir dönüşüm olarak görülmektedir.

$$\mathbf{y} = f(\mathbf{x}) \tag{3.18}$$

Burada; **x** ve **y** sırasıyla giriş ve çıkış vektörleridir. Evrensel yaklaşım teoremi aşağıdaki gibi açıklanabilir.

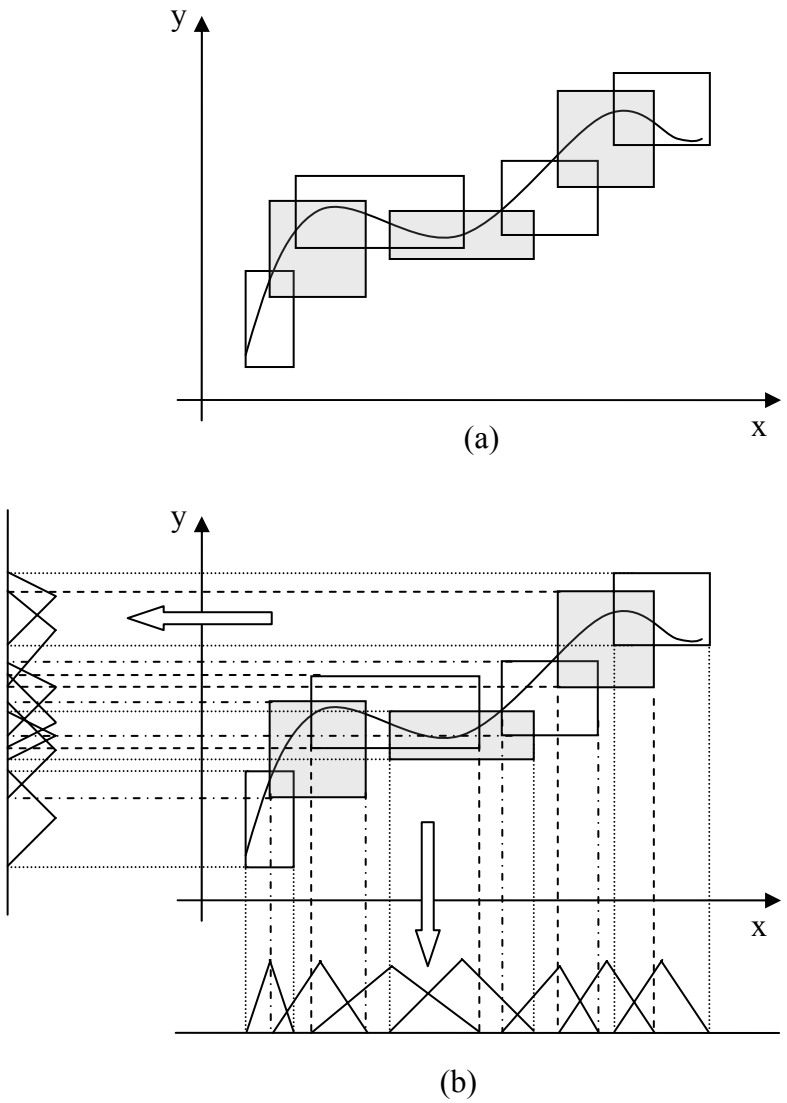
Bir $\mathbf{U} \in R^n$ kümesi üzerinde herhangi bir $F(\mathbf{x})$ gerçek sürekli fonksiyonu için aşağıdaki gibi verilen bir $f(\mathbf{x})$ bulanık sistemi vardır.

$$\mathrm{Sup}_{x \in U} \left| F(\mathbf{x}) - f(\mathbf{x}) \right| < \varepsilon$$

Burada; $F(\mathbf{x})$ ifadesi yaklaşılacak olan fonksiyonu, ε ise isteğe bağlı herhangi bir küçük pozitif sayıyı göstermektedir. Bulanık denetleyicilerde, üyelik fonksiyon sayılarının artırılması (dolayısıyla kuralların artırılması) ile, orijinal fonksiyona daha iyi yaklaşım sağlandığı görülmektedir. Bu işlem, belirli sayıdaki değerlerle bir sürekli fonksiyon elde etmeye benzer ve eldeki değer sayısı ne kadar fazla olursa gerçek eğriye o kadar fazla yaklaşılmaktadır. Ayrıca burada, belirli sayıdaki noktalardan bazı interpolasyon yöntemleri kullanılarak daha fazla noktanın elde edilmesi ve daha doğru yaklaşımın sağlanması da söz konusudur. Şekil 3.17 (a)'da lineer olmayan bir fonksiyona, bir bulanık sistemle nasıl yaklaşılacağı grafiksel olarak görülmektedir. Böylece etki alanlarından, giriş/çıkış uzayının üyelik fonksiyonları Şekil 3.17 (b)'deki gibi tanımlanabilir. Bu şekilde bulanık sistemin tasarım işlemi başlar ve kontrol kurallarının tanımlanması ile devam eder.

3.2.5 Bulanık denetleyicilerin tasarımı

Bulanık denetleyicilerin tasarımında, ilk olarak denetleyici için giriş ve çıkışların sayısına karar verilmelidir. Daha sonra giriş ve çıkış uzayları uygun sayıdaki bulanık kümelere ayrılmalıdır. Bulanık kümeler için en uygun üyelik şekilleri ve örtüşüm miktarı belirlenmelidir. Sonraki aşamada standart yöntemler ile uzman bilgisi, deneyimi veya önsezileri temelinde örnek bir karar tablosu oluşturulur. Bu noktada bulanık denetleyici, kapalı çevrim sisteme ilave edilir ve normalizasyon parametreleri, üyelik fonksiyon parametreleri ve bulanık kurallar değiştirilerek denetleyici ayarı yapılır.



Şekil 3.17 (a) Bulanık sistemde tipik bir statik fonksiyonun yaklaşımı (b) Lineer olmayan bir fonksiyon için üyelik fonksiyonlarını tanımlama işlemi

Bir bulanık denetleyicide, tasarım parametre sayısı konusunda bir fikir edinebilmek için z girişli, tek çıkışlı ve her değişken için m tane üyelik fonksiyonu olan bir sistem göz önüne alınır ve m tane üyelik fonksiyonunun her biri üç parametre ile ifade edilirse bu sistem, $3m(z+1)$ tane üyelik fonksiyon parametresi, m^z tane olası kural sayısı ve $(z+1)$ tane normalizasyon parametresi içerecek ve toplamda $3m(z+1)+m^z+(z+1)$ tane tasarım parametresine sahip olacaktır. Örneğin Şekil 3.16 ile gösterilen $z=2$ girişli, tek çıkışlı bir sistemde, eğer her değişken için $m=7$ üyelik fonksiyonu seçilirse, tasarım parametre sayısı yukarıdaki ifadeden 115 olarak

bulunur. Buradan sistem tasarımının ne kadar zor olduğu açıkça görülmektedir. Eğer sistematik bir metot yoksa, deneme yanılma yoluyla tüm parametrelerin ayarlanması oldukça zordur. Denetleyici performansı üzerine her bir tasarım parametresinin etkisini açıklamak için, her tasarım parametresi ayrı ayrı incelenmelidir.

3.2.6 Bulanık denetleyici tasarım parametreleri

Bulanık denetleyicilerin beş farklı adımdan oluştuğu daha önce ifade edilmişti. Burada, bu beş adımın tasarım parametreleri ve her parametrenin denetleyiciye etkisi verilecektir.

3.2.6.1 Normalizasyon parametrelerinin etkileri

Daha önce açıklandığı gibi, bulanık denetleyici tasarımında normalizasyon ve ters normalizasyon işlemlerinin yapılması gerekmektedir. Genel olarak Şekil 3.13 ile verilen z girişli tek çıkışlı bir bulanık denetleyicide giriş vektörü e=(e_1, e_2, e_3, …. e_z) olarak gösterilebilir. Bu durumda giriş vektörü için normalizasyon değerleri denklem (3.19) ile verilebilir.

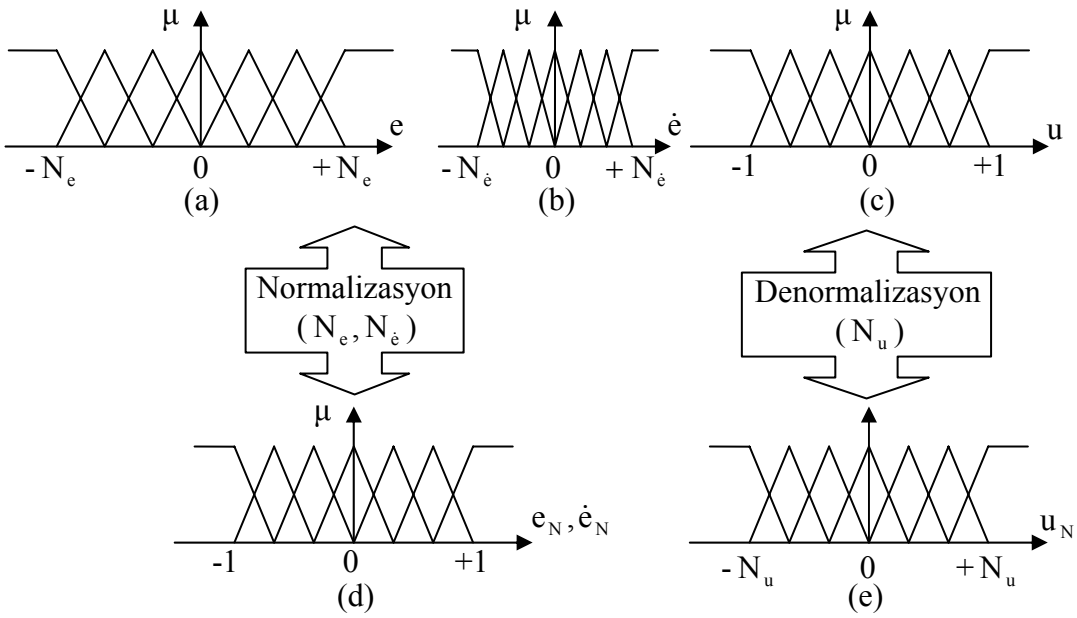
$$N_e = \begin{pmatrix} N_{e_1} & 0 & \cdots & 0 \\ 0 & N_{e_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & N_{e_z} \end{pmatrix} \tag{3.19}$$

Denklem (3.19)'a göre, iki girişli ve tek çıkışlı bir bulanık denetleyicide toplam 3 adet normalizasyon parametresinin tanımlanması gerekmektedir. Giriş değerleri **e**=(e_1,e_2) = $(e, \dot{e})$ ve çıkış değişkeni (u) olmak üzere normalizasyon parametreleri, e için N_e, $\dot{e}$ için $N_{\dot{e}}$ ve u için N_u olarak tanımlanabilir. Bu ifadeler kullanılarak giriş ve çıkış değerleri aşağıdaki gibi normalize edilebilir. Buna göre normalize edilmiş giriş değerleri;

$$e_N = e/N_e \qquad \dot{e}_N = \dot{e}/N_{\dot{e}} \qquad u_N = u/N_u \tag{3.20}$$

olmaktadır. Benzer şekilde bulanık denetleyicinin çıkış değeri de normalize edilmelidir. Buna göre normalize edilmiş çıkış değeri aşağıdaki şekilde bulunur.

Normalizasyon işlemleri, lineer işlemlerdir. Denetleyicide lineer ayarlama işlemi, ölçekleme faktörlerinin değiştirilmesi ile yapılır. Ölçeklendirme işlemi bu yönüyle, klasik lineer kontrol işleminde oransal (K_P) ve integral (K_I) sabitlerinin ayarlanmasına benzemektedir. Klasik kontrol sisteminde parametrelerin belirlenmesi, kararlılık açısından oldukça önemlidir. Üstelik, K_P ve K_I ile N_e ve $N_{\dot{e}}$ arasındaki bu benzerlik; bulanıklaştırma, çıkarım ve durulaştırma gibi bulanık denetleyicinin diğer kısımlarının, denetleyici derecesini artırarak lineersizliğe neden olduğunu göstermektedir. Bu işlem, geleneksel PI denetleyicilerin, bulanık denetleyicilerin bir özel hali olduğunu göstermektedir (Ashrafzadeh 1996). Şekil 3.18'de normalizasyon ve denormalizasyon faktörünün üyelik fonksiyonlarını nasıl değiştirdiği görülmektedir.

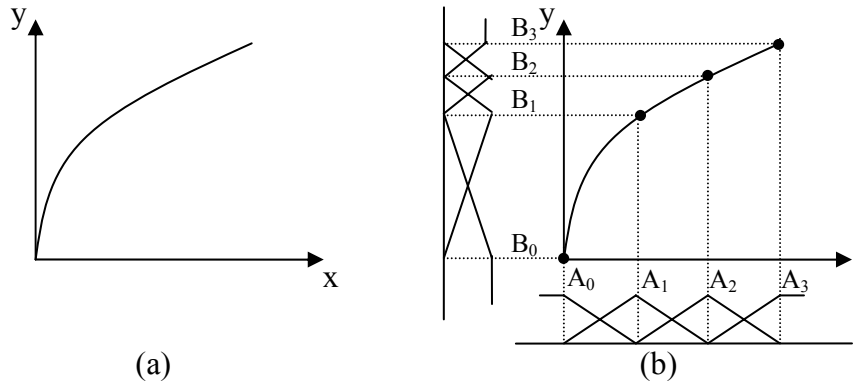


Şekil 3.18 Normalize edilmiş üyelik fonksiyonları

Normalizasyon işlemi, fiziksel sistemden alınan bilgilerin farklı tanım uzayında üyelik fonksiyonlarına dönüştürülmesinden ziyade, [-1,+1] gibi sabit bir aralıkta ölçeklendirilmesidir. Ölçeklendirme ile bulanık denetleyici genelleştirilmiş ve fiziksel sistemden bağımsız hale getirilmiş demektir. Üyelik fonksiyonları giriş domeni üzerinde sınırlı bir etkiye sahiptir. Halbuki denklem (3.20)'den görüldüğü gibi normalizasyon faktörleri genel etkiye sahiptir.

3.2.6.2 Üyelik fonksiyonları ve kuralların etkileri

Üyelik fonksiyonlarının ve kuralların göreceli olarak önemleri bir örnekle açıklanabilir. Bu amaçla Şekil 3.19 (a)'da verilen denetleyicinin tipik dönüşüm eğrisinin bilindiği kabul edilsin ve bir bulanık denetleyici ile bu fonksiyona yaklaşılmaya çalışılsın. Pratikte, sistemin gerçek davranışı tam olarak bilinmediği için, bu kabuller geçerli değildir. Ancak, bu kabuller üyelik fonksiyonları ve kuralların rolünü değerlendirmek için yapılmıştır.



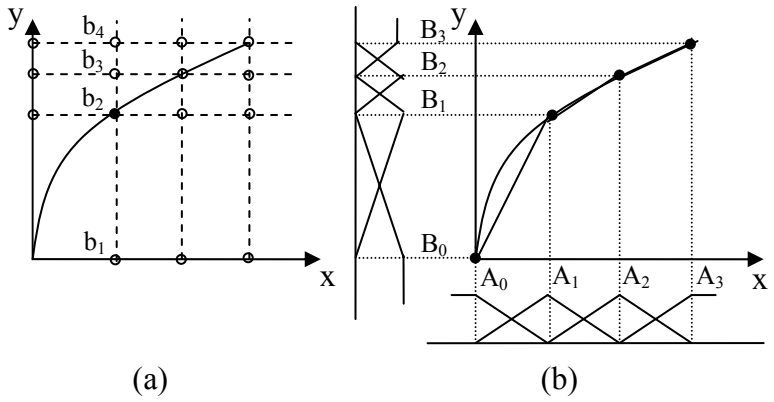
Şekil 3.19 (a) Tipik bir denetleyici dönüşüm eğrisi (b) Üyelik fonksiyonlarının tanımlanması

Bir bulanık denetleyici tasarımında gerekli dönüşümleri gerçekleştirmek için, normalizasyondan sonraki ilk adım, giriş-çıkış (örneğin x, y) uzayları için üyelik fonksiyonlarının tanımlanmasıdır. Giriş-çıkış fonksiyonlarının, Şekil 3.19 (b)'de gösterildiği gibi eğri üzerinde gelişigüzel dört noktanın seçilmesiyle elde edildiği kabul edilsin. Şekil üzerindeki oklar, uygun üyelik fonksiyonunun tanımlanmasında kullanılan ve fonksiyon üzerinde seçilen dört gelişigüzel noktanın izdüşümünü göstermektedir. Bu işlem halihazırda x-y uzayının bölünmesine neden olur. Şekil 3.20 (a)'da gösterildiği gibi bölünme, b_1, b_2, b_3 ve b_4 gibi bazı kesişim noktalarını içeren düzlemde tanımlanan bir çerçeve olarak görülebilir.

Bir sonraki adım, parçalanmış uzayda dönüşüm için gerekli kuralların tanımlanmasıdır. Bunun gibi artan fonksiyonlar için kurallar, aşağıdaki gibi kolayca tanımlanabilir.

EĞER x değeri A_0 düzlemine aitse, **O HALDE** y değeri de B_0 düzlemine aittir.
EĞER x değeri A_1 düzlemine aitse, **O HALDE** y değeri de B_1 düzlemine aittir.
EĞER x değeri A_2 düzlemine aitse, **O HALDE** y değeri de B_2 düzlemine aittir.
EĞER x değeri A_3 düzlemine aitse, **O HALDE** y değeri de B_3 düzlemine aittir.

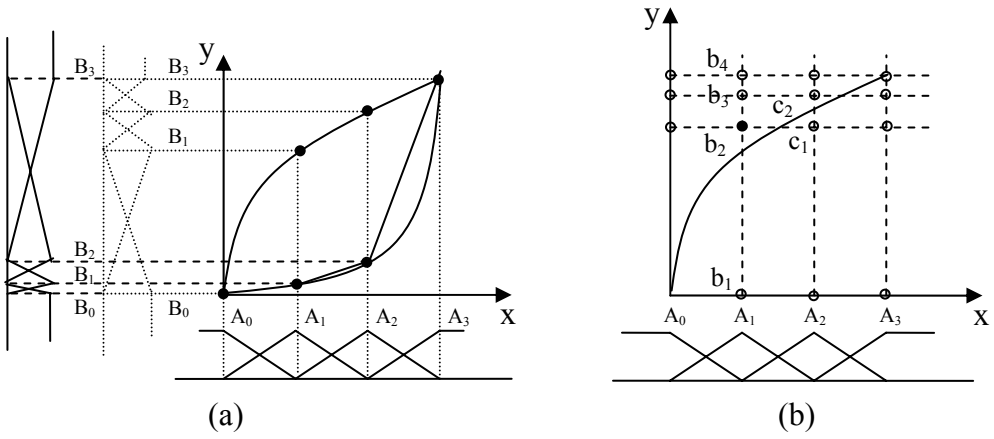
Şekil 3.20 (b)'den görüldüğü gibi yukarıda tanımlanan kuralların amacı; b_1, b_2, b_3, b_4 gibi noktalar arasından, sadece bir noktanın örneğin b_1 noktasının seçilmesidir. Bizim açımızdan, seçilen bu noktalar karakteristik noktalar olarak isimlendirilir çünkü, denetleyicinin yaklaşım fonksiyonunu karakterize ederler. Seçilen karakteristik noktalar birbirleri ile birleştirilerek bulanık modelin durulaştırılmasıyla yaklaşım fonksiyonu Şekil 3.20 (b)'deki gibi elde edilir.



Şekil 3.20 (a) Üyelik fonksiyonları ile çerçevenin tanımlanması (b) Bulanık model ile yaklaşım fonksiyonu

Bu açıdan bakıldığında, üyelik fonksiyonları ve kuralların amaçları şöyle verilebilir. Kurallar; örneğin artan veya azalan fonksiyonunun genel şeklini belirlerken, üyelik fonksiyonları; her bir artan veya azalan parçanın eğimini belirlemektedir. Üyelik fonksiyonlarının etkisi, daha çok yerel veya sınırlı görülürken, aynı kural kümesinde sistemin işlevini çarpıcı biçimde değiştirmektedir. Örneğin, x için yukarıdaki gibi aynı üyelik fonksiyonları, fakat y için farklı bir küme göz önüne alınsın. Benzer kurallar veya aynı dönüşümle, Şekil 3.21 (a)'da gösterildiği gibi kesikli ve düz çizgi halinde oldukça farklı sonuçlar elde edilecektir.

Bu işlem, bulanık modellemede üyelik fonksiyonlarının etkilerini göstermektedir. Buna göre; üyelik fonksiyonlarının birbirleriyle örtüşmesiyle, düşey eksen tamamen taranmaktadır. Bu durumda eğri, Şekil 3.21 (b)'de gösterilen b_1, b_2, b_3 ve b_4 noktalarından ziyade, b_1 b_4 düşey çizgisi üzerindeki seçilen herhangi bir noktayı kesmektedir. Aslında, daha fazla karakteristik nokta c_1 ve c_2 denilen noktalar arasında interpolasyonla elde edilerek birleştirilmiştir. Bu çözüm, bulanık denetleyici tasarımında avantaj sağlamaktadır (Ashrafzadeh ve ark. 1997).



Şekil 3.21 (a) Üyelik fonksiyonlarının dönüşüme etkisi (b) Daha çok kesişme durumunda bulanık dönüşüm

3.2.7 Geleneksel tasarımda karşılaşılan temel problemler

Bulanık sistemde en önemli kavram evrensel yaklaşım teoremidir. Bu teorem, kontrol mühendisliğinde bulanık sistemlerin pratik başarısı için bir tanımlama sağlar. 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.

- 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.

Sıralanan tüm problemleri çözmek için, problemin iç yüzüne bakılması gereklidir. Bir uzman tarafından tasarlanan bulanık denetleyici, kapalı çevrim sistemde, uzmanın sunduğu insan benzeri denetleyici rolünü gerçekleştirir. Eğer, bir şekilde insanda bulunan bilgi tabanını geliştirme işlemi modellenebilir ve benzetim yapılabilirse, büyük bir uzman bilgisi gerekmeden yukarıda verilen problem çözülecektir.

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 algoritmalardan oluşmaktadır.

Otomatik olarak bir sistemin tasarımını gerçekleştirmek için, denetleyicide bulunan optimizasyon problemi, uygulanan optimizasyon tekniğiyle çözülmelidir. 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, varolan 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.

3.3 Genetik Algoritmalar ile Optimizasyon

Genetik algoritmalar (GA), doğadaki biyolojik evrim temeline dayanan optimizasyon tekniğidir. Genetik algoritmalar ilk defa 1975 yılında Michigan üniversitesinde 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. Ancak, mühendislikte optimizasyon için genetik algoritmaların uygulanması oldukça yenidir (Ashrafzadeh 1996).

Genetik algoritmalar ile optimizasyon yapabilmek için, optimize edilecek olan parametreler tanımlanmalı ve bu parametreler optimizasyon işlemi için uygun biçimde kodlanmalıdır. Genetik algoritmalar, 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, algoritmada yürütülen matematiksel işlemler açısından oldukça önemlidir.

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.

3.3.1 Genetik algoritmalar

Genetik algoritmalar, 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).

Yukarıda belirtildiği gibi genetik algoritmalar, bireylerin populasyonu ile çalışır ve verilen bir probleme olası bir çözüm sunarlar. Her bir bireye, verilen problemi ne kadar iyi çözdüğünü gösteren uygunluk puanı verilir. Uygunluk puanı, bir kapalı çevrim kontrol sistemi için bir performans indeksi veya verilen bir tasarım için kuvvet/ağırlık oranı olabilir. Doğada, bu işlem bir organizmanın kaynaklar ve hatta çiftleşme için ne kadar etkili yarıştığına karşılıktır. Yüksek adaptasyon sağlayan uygun bireyler, oldukça büyük sayıda üreyeceklerdir. Kötü performanslı bireyler ise çok az üreyebilir veya hiç üreyemezler. Farklı soydaki iyi karakterli iki bireyin kombinasyonu, bazen ailelerine göre daha uygun bireylerin üremesini sağlayabilir. Böylece yeni bireyler, ortama giderek artan bir uygunluk sağlarlar.

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 populasyonunun rasgele 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.

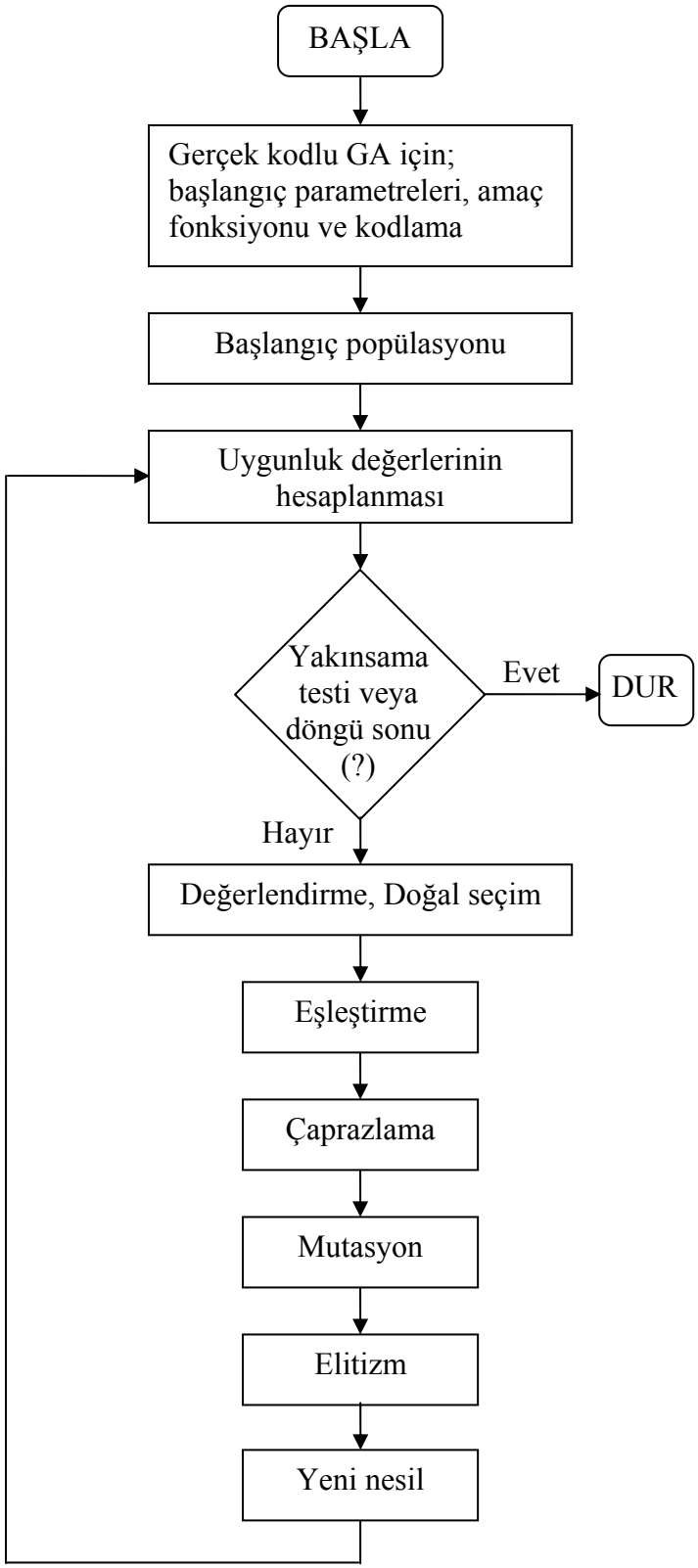
- Sürekli ve ayrık parametreleri optimize etmesi,
- Türevsel hesaplamalar gerektirmemesi,
- Amaç fonksiyonunu geniş bir spektrumda araştırması,
- Çok sayıda parametrelerle çalışma imkanı olması,
- Paralel PC 'ler kullanılarak çalıştırılabilmesi,
- Karmaşık amaç fonksiyonu parametrelerini, lokal minimum veya maksimumlara takılmadan optimize edebilmesi,
- Çoklu amaç fonksiyonları ile çalışabilmesi.

Her problemin çözümü için GA kullanmak iyi bir yol değildir. Birkaç parametreli 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.

3.3.2 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 Parametreli 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. Gerçek kodlu GA'nın akış diyagramı Şekil 3.22'de verilmiştir.



Şekil 3.22 Gerçek kodlu GA'nın akış diyagramı

3.3.2.1 Parametrelerin kodlanması ve amaç fonksiyonu

Genetik algoritmalar 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, $1xN_{par}$ boyutlu matris biçiminde aşağıdaki gibi yazılır.

$$\text{Kromozom} = [gen_1, gen_2, gen_3, \ldots\ldots\ldots gen_{Npar}]$$

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,

$$C = f(\text{kromozom}) = f(gen_1, gen_2, gen_3, \ldots\ldots\ldots gen_{Npar})$$

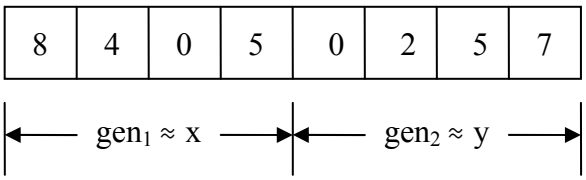
şeklinde ifade edilir. Örneğin iki parametreli bir optimizasyon için amaç fonksiyonu;

$$f(x, y) = x\sin(4x) + 1{,}1\,y\sin(2y) \qquad (3.21)$$

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

$$\text{Kromozom} = [gen_1, gen_2] \approx [x, y]$$

şeklinde ifade edilebilir. Bu problem, çok sayıda lokal minumuma 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 3.23'de görüldüğü gibi 8 adet on tabanlı sayıdan oluşacaktır. Şekilde verilen kromozom yapısında, parametrelerin gerçek değerleri, parametre sınırları içine normalize edilmelidir.



Şekil 3.23 Bir genetik algoritmada tipik kromozom dizgisi

3.3.2.2 Başlangıç popülasyonu

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 populasyon olarak tanımlanır. GA'nın çalışmaya başlayabilmesi için, N_{ipop} kromozomdan oluşan başlangıç popülasyonuna ihtiyaç vardır ve başlangıç populasyonu, N_{par} x N_{ipop} 'luk bir matris ile temsil edilir. GA'da başlangıç populasyonunun oluşturulması iki şekilde yapılır. Birincisinde; eğer programcı, geniş arama yüzeyinde lokal bir bölgede sonuç arayacaksa, o bölgeye yakın noktalardan oluşan bir başlangıç populasyonu tanımlayabilir. Bu, algoritmanın daha kısa sürede sonuca ulaşmasını sağlar ve birçok çalışmada, ince ayar (fine-tuning) için kullanılmaktadır. İkincisinde; başlangıç populasyonunda kromozomları oluşturan genler, denklem (3.22) ile verilen rasgele sayı üreteci ile elde edilir.

$$\text{İPOP} = (P_H - P_L) \text{ x Rasgele}\{N_{ipop},N_{par}\} + P_L \qquad (3.22)$$

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

3.3.2.3 Uygunluk değerlerinin hesaplanması

Parametrelerin bu şekilde değerleri belirlendikten sonra, amaç fonksiyonundan yararlanılarak uygunluk değerleri (C) hesaplanır. Denklem (3.21)'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 24x2 olmaktadır. Başlangıç popülasyonunun büyüklüğü, araştırma uzayının daha geniş seçilmesini sağlamaktadır. GA

döngüsünün sona erdirilebilmesi için, algoritma başlangıcında, bir yakınsama testi veya döngü sayısı tanımlanmalıdır.

3.3.2.4 Değerlendirme ve doğal seçim

Bir genetik algoritmanın her döngüsünde ilk adım, her bir kromozomun problemi ne kadar iyi çözebileceğini belirlemektir. Değerlendirme olarak bilinen bu adım, kullanılan kromozomların sadece bir açıklamasıdır. Değerlendirme, bir sonraki adımda ne kadar yeni çiftin üretilmesi gerektiğini belirlemede yardımcı olmaktadır.

GA'larda gelecek nesilde, hangi kromozomun yer alacağının belirlenmesi 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. Örneğin 24 adet kromozoma sahip Çizelge 3.1'deki populasyonda, ortalama uygunluk değeri 0.9039 ve en iyi kromozomun uygunluk değeri –16.26 olsun. Alt sıradan başlanarak, kromozomların 12 tanesi atılırsa, geri kalan kromozomların ortalama uygunluk değeri –4.27 olmaktadır. Her bir iterasyonda N_{pop}= 12 kromozom kullanılır ve bunun 6 tanesi N_{iyi}, 6 tanesi $N_{kötü}$ olarak tanımlanır (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:* ilk olarak kromozomların uygunluk değerleri amaç fonksiyonunda hesaplanır. Hesaplanan uygunluk değerleri, en küçükten en büyüğe doğru sıralanır. Çizelge 3.1 'den görüldüğü gibi 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.

Çizelge 3.1 Yirmi dört adet kromozomun sıralanması

No	x	y	Uygunluk Değerleri
1	9.0465	8.3097	-16.2555
2	9.1382	5.2693	-13.5290
3	7.6151	9.1032	-12.2231
4	2.7708	8.4617	-11.4863
5	8.9766	9.3469	-10.3505
6	5.9111	6.3163	-5.4305
7	4.1208	2.7271	-5.0959
8	2.7491	2.1896	-5.0251
9	3.1903	5.2970	-4.7452
10	9.0921	3.8350	-4.6841
11	0.6056	5.1942	-4.2932
12	4.1539	4.7773	-3.9545
13	8.4598	8.8471	-3.3370
14	7.2541	3.6534	-1.4709
15	3.8414	9.3044	-1.1517
16	8.6825	6.3264	-0.8886
17	1.2537	0. 4746	-0.7724
18	7.7020	7.6220	-0.6458
19	5.3730	2.3777	-0.0419
20	5.0071	5.8898	0.0394
21	0.9073	0.6684	0.2900
22	8.8857	9.8255	0.3581
23	2.6932	7.6649	0.4857
24	2.6614	3.8342	1.6448

N_{ipop} N_{iyi} $N_{kötü}$ N_{pop}

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.

3.3.2.5 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 $kromozom_{2i-1}$ ile $kromozom_{2i}$ i=1,2,3 için eşleştirilir. Bu örnekte $kromozom_1$ ile $kromozom_2$, $kromozom_3$ ile $kromozom_4$ ve $kromozom_5$ ile $kromozom_6$ eşleştirilmiştir. Bu metot, doğal bir yöntem olmamakla beraber yeni başlayanlar için kolay bir yoldur.

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

$$\text{Kromozom}=\text{Roundup}\{N_{iyi} \text{ x rasgele sayı}\} \tag{3.23}$$

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$_1$-kromozom$_5$, kromozom$_1$-kromozom$_2$ ve kromozom$_5$-kromozom$_3$ 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 (3.24) kullanılarak sıralamadaki yerlerine göre kromozomların olasılıkları P_n hesaplanır.

$$P_n = \frac{N_{iyi} - n + 1}{\sum_{n=1}^{N_{iyi}} n} = \frac{6 - n + 1}{1 + 2 + 3 + 4 + 5 + 6} = \frac{7 - n}{21} \qquad (3.24)$$

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. Benzer tarzda 6 tane rasgele sayı üretilerek eşleştirme gerçekleştirilir. Üretilen rasgele sayılar 0.1535, 0.6781, 0.0872, 0.1936, 0.7021 ve 0.3933'dir. Bu sayılara göre kromozom$_1$-kromozom$_3$, kromozom$_1$-kromozom$_1$ ve kromozom$_3$-kromozom$_2$ eşleştirilirler. Kromozomlar kendisi ile eşleştirilirse,

bu kromozomun gelecek nesilde daha etkin olarak yer aldığı söylenebilir. 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 aşağıdaki gibi hesaplanır.

$$C_n=f(\text{kromozom}_n)-f(\text{kromozom}_{Niyi+1}) \tag{3.25}$$

$$P_n = \left| \frac{C_n}{\sum_{p=1}^{N_{iyi}} C_p} \right| \tag{3.26}$$

Daha önce üretilen rasgele sayılar kullanılırsa kromozom_1-kromozom_3, kromozom_1-kromozom_1 ve kromozom_3-kromozom_1 eşleşmeleri görülür. Kromozomların eşleştirilmesinde, uygunluk değerlerine göre kromozomların seçilme ihtimalleri ağırlıklı rasgele eşleştirme metodu kullanılarak Çizelge 3.2'deki gibi elde edilmiştir.

Çizelge.3.2 Kromozomların uygunluk değerlerine göre seçilme ihtimalleri

n	$\mathbf{P_n}$	$\sum_{i=1}^{n} P_i$
1	0.2265	0.2265
2	0.1787	0.4052
3	0.1558	0.5611
4	0.1429	0.7040
5	0.1230	0.8269
6	0.0367	0.8637
7	0.0308	0.8945
8	0.0296	0.9241
9	0.0247	0.9488
10	0.0236	0.9724
11	0.0168	0.9892
12	0.0108	1.0000

Çizelge 3.2'den görüldüğü gibi, eşleştirme çoğunlukla yukarıdan aşağıya doğru yapılmaktadır. Çünkü sıralamanın altındaki kromozomların uygunluk değerleri küçüktür. Rasgele üretilen altı tane sayı çifti yandaki gibi sıralanabilir. (0.4679, 0.5344), (0.2872, 0.4985), (0.1783, 0.9554), (0.1537, 0.7483), (0.5717, 0.5546), (0.8024, 0.8907). Üretilen bu sayı çiftlerini kullanarak hangi kromozomların eşleştirileceğini bulabiliriz. Örneğin bu amaçla, birinci sayı çifti ele alınırsa, $0.4052 < 0.4679 \leq 0.5611$ olduğundan üçüncü anne kromozom ile, $0.4052 < 0.5344 \leq 0.5611$ olduğundan üçüncü baba kromozom eşleştirilmelidir. Benzer şekilde tüm eşleştirme işlemi aşağıdaki gibi gerçekleştirilir.

Anne=[3, 2, 1, 1, 4, 5] Baba=[3, 3, 10, 5, 3, 7]

3.3.2.6 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 aşağıdaki formül kullanılarak yeni nesiller elde edilir (Radcliff 1991).

$$P_{yeni}=\beta P_{an}+(1-\beta)P_{bn} \qquad (3.27)$$

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

İkinci nesil, β yerine 1-β konularak ve birinci neslin tümleyeni alınarak bulunur. β=1 ise, P_{an} baskın gelir ve P_{bn} ölür. β=0 ise P_{bn} baskın gelir ve P_{an} ölür. β=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 rasgele seçilebilir. Parametreler, farklı β 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.

$$P_{yeni1}=0.5P_{an}+0.5P_{bn} \quad (3.28)$$

$$P_{yeni2}=1.5\ P_{an}\ -0.5\ P_{bn} \quad (3.29)$$

$$P_{yeni3}=\ -0.5\ P_{an}\ +1.5\ P_{bn} \quad (3.30)$$

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

Diğer bir çaprazlama metodu olan "Heuristic çaprazlama" da ise β, 0-1 arasında değer alır ve yeni nesil aşağıdaki formüle göre elde edilir (Michalewicz 1994).

$$P_{yeni}=\beta(P_{an}-P_{bn})+P_{an} \quad (3.31)$$

Burada, her parametre için farklı β değeri üretilerek yeni nesiller elde edilir. Denklem (3.31)'e göre, izin verilen bazı değerlerin dışında üretilen yeni nesiller ihmal edilir. Algoritma, yeni β değeri üreterek işleme devam eder.

"Karma çaprazlama metodu" (blend crossover, BLX-α) bazı parametrelerin α değerlerini belirleyerek işleme başlar (Eshelman ve Shafter 1993). α 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.

$$\alpha=\text{roundup}\{\text{rasgele x } N_{par}\} \quad (3.32)$$

$$\text{Kromozom}_1=[P_{a1},P_{a2}.....P_{a\alpha}..... P_{aN_{par}}]$$

$$\text{Kromozom}_2=[P_{b1},P_{b2}..... ..P_{b\alpha}..... P_{bN_{par}}]$$

Burada a ve b indisleri sırasıyla anne ve baba kromozomlarını tanımlamak amacıyla kullanılmaktadır. Bu metotta, denklem (3.33) ve denklem (3.34) kullanılarak yeni nesiller elde edilir.

$$P_{yeni1}=P_{a\alpha}-\beta[P_{a\alpha}-P_{b\alpha}] \quad (3.33)$$

$$P_{yeni2}=P_{b\alpha}-\beta[P_{a\alpha}-P_{b\alpha}] \quad (3.34)$$

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

$$Nesil_1=[\ P_{a1},P_{a2}.....P_{yeni1}.....\ P_{bN_{par}}\ ]$$

$$Nesil_2=[\ P_{b1},P_{b2}.....P_{yeni2}\ .....\ P_{aN_{par}}\ ]$$

Eğer kromozomun birinci parametresi seçilirse, seçilen parametrenin sağındaki parametreler değiştirilir. Kromozomun en son parametresi seçilirse, seçilen parametrenin solundaki parametreler değiştirilir. Bu yöntemde β>1 olmadıkça sınır değerleri aşan herhangi bir parametre üretilmemektedir. Örneğin;

$Kromozom_2$=[5.2693, 9.1382] $Kromozom_3$=[9.1032, 7.6151]

ele alınsın. Rasgele sayı olarak üretilen α=1 için çaprazlama noktası olarak 1. parametre seçilmektedir. İkinci rasgele sayı olarak β=0.7147 üretilmektedir. Denklem (3.33) ve (3.34)'e göre yeni nesil aşağıdaki gibi bulunur.

$Nesil_3$=[5.2693-0.7147x5.2693+0.7147x9.1032,7.6151]=[**8.0094**,7.6151]

$Nesil_4$=[9.1032+0.7147x5.2693-0.7147x9.1032,9.1382]=[**6.3631**,9.1382]

Diğer kromozomlara da aynı kurallar uygulanarak yeni nesiller elde edilir.

3.3.2.7 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 algoritmada, 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. Örneğin; mutasyon olasılık değeri P_{mut}=0.04 olarak alınırsa, mutasyona uğrayacak parametre sayısı 0.04x24x2=2 olarak hesaplanır.

Çizelge 3.1'deki kromozom$_7$'nin birinci parametresi ile kromozom$_{22}$'nin ikinci parametresi mutasyona uğrarsa, kromozom$_7$'nin birinci parametresi ile kromozom$_{22}$'nin ikinci parametresi silinir ve yerine 0-10 arasında yeni bir rasgele sayı üretilerek konur.

Kromozom$_7$=[4.1208, 2.7271] => Kromozom$_7$=[8.6750, 2.7271]

Mutasyon süreci tamamlandıktan sonra, yeni popülasyonun sıralaması Çizelge 3.3'deki gibi olur. Algoritma, yedinci iterasyondan sonra minimum uygunluk değerini –18.5 olarak bulmaktadır.

Çizelge 3.3 Gerçek kodlu GA'nın ikinci nesilden sonraki sıralanışı (Haupt 1998)

No	**x**	**y**	**Uygunluk değerleri**
1	9.0465	8.3128	-16.2929
2	9.0465	8.3097	-16.2555
3	9.1382	5.2693	-13.5290
4	7.6151	9.1032	-12.2231
5	7.6151	9.1032	-12.2231
6	7.6151	9.1032	-2.2231
7	2.7708	8.4789	-11.6107
8	2.7708	8.4617	-11.4863
9	9.1382	8.0094	-11.0227
10	8.9766	9.3438	-10.4131
11	8.9766	9.3469	-10.3505
12	9.0465	7.9346	-9.8737
13	1.5034	9.0860	-6.6667
14	4.4224	9.3469	-5.6490
15	5.9111	6.3163	-5.4305
16	7.6151	6.3631	-5.1044
17	9.0921	4.2422	-5.0619
18	2.7491	2.1896	-5.0251
19	3.1903	5.2970	-4.7452
20	9.0921	3.8350	-4.6841
21	0.6956	5.1942	-4.2932
22	4.1539	4.7773	-3.9545
23	8.6750	2.7271	-3.4437
24	4.1208	3.1754	-2.6482

3.3.2.8 Elitizm

Elitizm, önceki populasyon içerisindeki en iyi uygunluk değerine sahip bireyin, olduğu gibi yeni populasyona taşınması işlemidir. En iyi birey, çaprazlama ve mutasyon gibi diğer genetik işlemlerden etkilenmeden, doğrudan yeni populasyona aktarılır. Bu işlem, GA'da sıkça uygulanan bir yöntemdir. Çalışma için yazılan algoritmada istenirse, seçme suretiyle elitizm işlemi yapılabilmektedir.

3.3.3 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.

Yer kürede yaşayan insanoğlunun hayatı, optimizasyon problemleri ve çözümleri ile doludur. Bununla ilgili birkaç örnek verilebilir. İşe giderken en iyi yolu kullanmak, bir projeyi yürütürken nereden başlanması gerektiğini belirlemek, bir şeyin uzunluğunu kısaltmak, ağırlığını azaltmak, bir ürünün cazibesini artırmak ve maliyetini düşürmek bir optimizasyon işlemidir.

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. Örneğin; bulunan kökün optimal bir çözüm olduğunu anlamak zordur. Çünkü bütün kökler, fonksiyonu sıfır yapmaktadır. 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. Ancak kesin hatlarıyla altı gruba ayrıldığı söylenemez. 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 kaşifler bu yolu kullanmışlardır.

2) *Tek ve çok parametreli 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 metodunda, 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. Pratikte değişkenlerin çokluğu nedeniyle problem, pek de basit değildir. En kısa yol, en hızlı yol olmayabilir. En hızlı yolu bulmak dinamik bir problemdir ve zamana, havanın durumuna, kazalara vb. bağlıdır.

4) *Sürekli ve ayrık parametreli 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 parametreli düşünülebilir. Ayrık parametreli optimizasyon,

kombinasyonel optimizasyon olarak da adlandırılır. Bir çizgide f(x)'in minimum değerini bulmaya çalışmak, sürekli parametreli optimizasyon olarak tanımlanır.

5) *Sınırlı ve sınırsız optimizasyon;* sınırlı optimizasyon, parametreleri bir tanım aralığında değerlendirir. Sınırsız optimizasyonda ise, parametreler her hangi bir değerde olabilir. Değişkenlerin sınırları kaldırılarak sınırlı parametreler sınırsız parametrelere çevrilirler. Çoğu nümerik optimizasyon rutinleri sınırsız parametrelerle çalışırlar. Örnek olarak f(x) fonksiyonu ele alınırsa ve sınırlarının $-1 \leq x \leq 1$ aralığında olduğu kabul edilirse bu fonksiyon, $x = \sin(t)$ tanımı kullanılarak sınırsız optimizasyona dönüştürülür. Burada t'nin değeri ne olursa olsun x değeri (-1,1) aralığında değişecektir. 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.

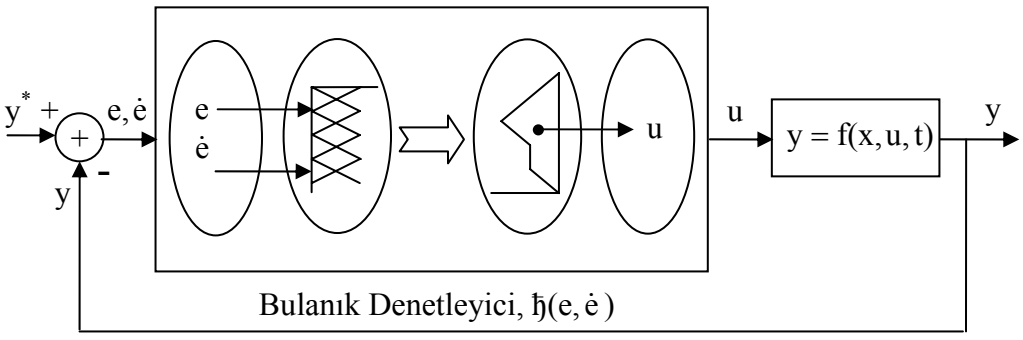
Yukarıdaki gruplandırmanın yanında optimizasyon metotlarını; deterministik metotlar ve istatistiksel metotlar olmak üzere iki ana gruba ayırmak mümkündür. Deterministik optimizasyon metotları, lokal maksimum veya minimuma yakınsayan algoritmalardır. Türevsel hesaplamalar veya türevsel yaklaşımlar deterministik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

Bulanık denetleyicilerin tasarımında karşılaşılan problemlerin çözümü ve optimal bir bulanık denetleyicinin tasarımı için, optimizasyon işlemine ihtiyaç vardır. Daha önceden belirtildiği gibi bulanık denetleyiciler, çok parametreli ve lineer olmayan bir yapıya sahiptir. Genetik algoritmalar, lineer olmayan çok parametreli ve çok modelli nesnel fonksiyonların optimizasyonları için oldukça uygundur. Bu özelliğinden dolayı genetik algoritmalar, bulanık denetleyicilerin tasarımında kullanılmıştır. Bulanık denetleyicilerin GA'lar ile tasarımı ve optimizasyonu için denetleyici parametre kümesi, genetik algoritmalarda kullanılmak üzere iki tabanlı bit dizisine veya on tabanlı gerçek kod yapısına dönüştürülmelidir.

Bir bulanık denetleyici tasarımında en uygun optimizasyon tekniğini seçmek için, tasarım probleminin en iyi şekilde tanımlanması gereklidir. İlk aşamada, kontrol sistemi için optimizasyon işleminde amaç fonksiyonu gibi görev yapacak bir performans indeksi tanımlanmalıdır. Bu performans ölçümü, özellikle geçici cevap için hatanın bir fonksiyonu olabilir. Örneğin, Şekil 3.24'de görülen sistem için performans indeksi aşağıdaki gibi tanımlanabilir.

$$J_{in} = \int_{t_0}^{t_f} Q(e)dt = \int_{t_0}^{t_f} Q(f(x,u,t))dt \qquad (3.35)$$

Burada; Q hatanın bir fonksiyonu, f sistem fonksiyonu, $u=\eta(e,\dot{e})$ denetleyici fonksiyonu ve x ise sistem durum vektörüdür.



Şekil 3.24 Kapalı çevrim sistemde bulanık denetleyici

Bir kontrol sisteminde optimizasyon işleminin amacı, amaç fonksiyonunu (performans indeksini) minimize edecek $\eta(e,\dot{e})$ denetleyici fonksiyonunun bilinmeyen parametre kümesini bulmaktır. Gradyan tabanlı optimizasyon tekniğinde, bilinmeyen parametreler için genellikle, $\eta(e,\dot{e})$ fonksiyonunun farklı tasarım parametreleri ile ilgili olarak amaç fonksiyonunun kısmi türevi hesaplanmalı ve daha sonra birbirine bağlı lineer olmayan bir diferansiyel denklem çözülmelidir. Bu çalışmada, bulanık denetleyicinin analitik ifadesi yani $\eta(e,\dot{e})$ fonksiyonu bilinmediğinden, problemi çözmek için optimizasyon işleminde gradyan tekniği, kullanılamaz. Üstelik, dikkate alınması gereken diğer faktörler vardır. Bunlardan birincisi; bulanık denetleyicinin dönüşüm fonksiyonu $\eta(e,\dot{e})$ lineer olmayan bir fonksiyondur. Bu özellik, Q ve özellikle $f(x,u,t)$ sistemi lineer olsa bile amaç fonksiyonunun lineer olmamasına neden olur. Bunun sonucunda, optimizasyon tekniğinin de lineer olmaması gereklidir. İkincisi; bulanık denetleyicilerin tasarımında parametre sayısı oldukça fazladır. Amaç fonksiyonu çok boyutlu bir uzay içerir ve daha önemlisi çok modelli geniş bir yüzey gibi görünebilir. Bu yüzden başlangıçta, çok parametreli bir fonksiyon gibi görünür. Bu açıdan bakıldığında, bu geniş çözüm uzayında global optimumların bulunması basit bir işlem değildir.

Gradyan tabanlı tekniğin aksine genetik algoritmalar, $\eta(e,\dot{e})$ denetleyici fonksiyonu için bir analitik ifadeye ihtiyaç duymazlar. Üstelik GA'larda amaç fonksiyonunun lineer olmaması ve çok modelli bir yapıya sahip olması problem oluşturmaz. Genetik algoritmalar türevlerle değil, sayı dizileri ile çalışırlar ve genellikle bu yapıları onları özellikle amaç fonksiyonunun çok parametreli ve lineer olmadığı durumlarda daha etkili yapar. Üstelik genetik algoritmalar, gradyan tabanlı teknikte olduğu gibi bir noktaya bakmaktan ziyade, noktaların popülasyonuna bakar. Bu özellik tüm arama uzayında daha iyi bir tanımlama sağlar. Gradyan tabanlı teknik, noktaya bağlı bir arama yaklaşımıdır ve bu yüzden eğer arama yüzeyi çok modelli bir yapıda ve başlangıç noktası global optimum noktaya yeterince yakın değilse, bir lokal noktada hataya düşme olasılığı daima vardır (Ashrafzadeh 1996, Tan ve Hu 1996).

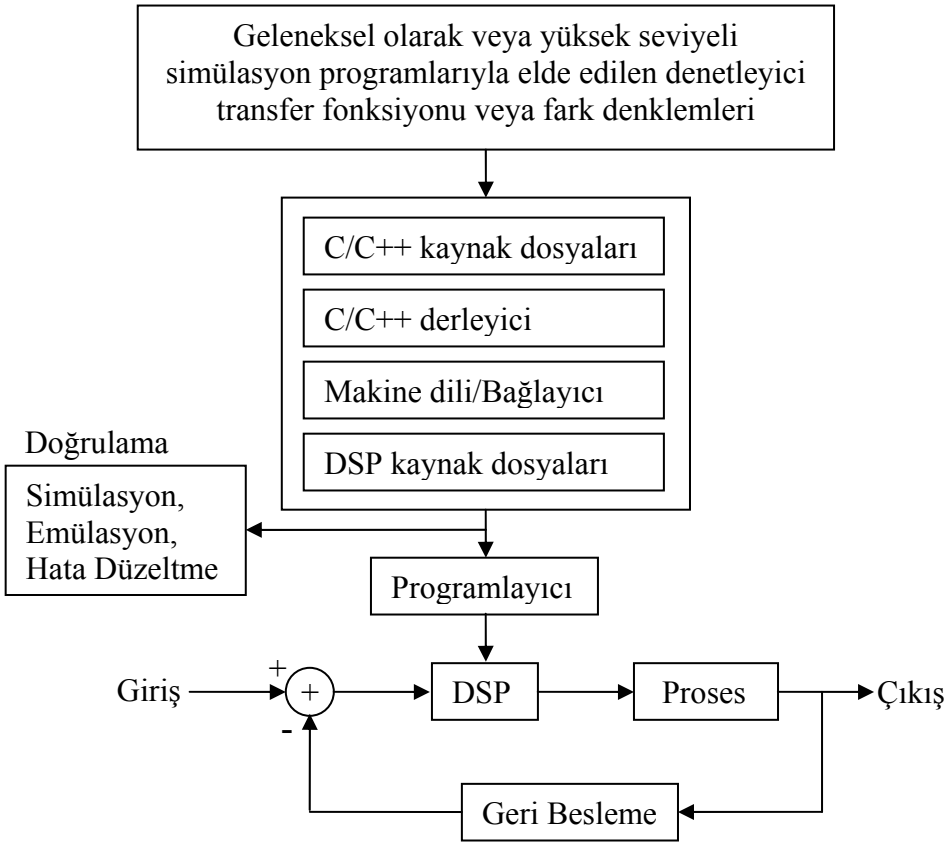
3.4 Sayısal İşaret İşlemciler (DSP) ve ADSP-21992

Uygulama devreleri için işlemci seçerken, o işlemcinin hangi amaçla üretildiğine dikkat edilmesi gerekir. Eğer uygulama, basit matematiksel hesaplama ve açık-kapalı kontrol gibi basit işlemler gerektiriyorsa, basit bir mikro-denetleyici yeterlidir. Büyük boyutlu veri saklama, veriler üzerinde işlem yapma, simülasyon çalışmaları ve yoğun matematiksel işlemler için ise en uygunu bir kişisel bilgisayardır. Ancak, gerçek zamanda bir analog sinyali örnekleyerek sayısal forma dönüştürüp, üzerinde yoğun matematiksel işlem yaptıktan sonra tekrar analog sinyale dönüştürmek gerekiyorsa bir DSP kullanılması gerekir. Dikkat edilmesi gereken bir diğer noktada, çok geniş ürün yelpazesine sahip olan DSP dünyasında, uygulama alanına göre en uygun DSP elemanının seçilmesidir. Son zamanlarda DSP'ler, yüksek örnekleme ve veri işleme özellikleriyle, birçok uygulamada karşımıza çıkmaktadır (Smith 2002).

DSP'lerin kontrol sistemlerinde kullanılması hızla yaygınlaşmaktadır. Analog teknikler kullanılarak tasarlanan ve analiz edilen geleneksel kontrol sistemlerinin yerini, tipik olarak bir bilgisayar programı veya programlanabilen sayısal sistemler almaktadır. Bir kontrol sisteminde geleneksel analog denetleyicilerin yerine DSP kullanılmasının bazı avantajları vardır. Bu avantajlar aşağıda sıralanmıştır.

- Yeni bir kontrol yöntemi, donanımı değiştirmeden kolayca uygulanabilmektedir.
- Kontrol sisteminde oluşan istenmeyen mekanik davranışlar, geliştirilen yazılımlar ile kolayca elimine edilebilmektedir.
- Denetleyici performansını iyileştirmek için performansı etkileyen faktörler, kolayca gözlenebilmekte ve yok edilebilmektedir.
- Daha az eleman kullanıldığından sistem daha küçük, daha açık, güç tüketimi daha az ve daha ucuz olmaktadır.
- Sistem daha güvenilir, kolay bakım yapılabilir ve test edilebilir olmaktadır.
- Sistem gürültüye karşı daha bağışıklıdır.
- Adaptif kontrol, optimal kontrol gibi karmaşık kontrol algoritmaları kolayca uygulanabilmektedir.

Bir kontrol sisteminde DSP'nin kullanılması ve denetleyici olarak programlanması işlemini gösteren yapı Şekil 3.25'de görülmektedir. DSP tabanlı bir kontrol sisteminin tasarım işleminde ilk olarak, ileri simülasyon programları ya da geleneksel yollarla denetleyici transfer fonksiyonu veya fark denklemleri elde edilir. Daha sonra bu denklemler, C/C++ gibi ileri programlama dilleri kullanılarak kaynak dosya formuna dönüştürülür. Bu dosyalar DSP için derlenir ve derlenen bu program yürütülmek üzere DSP'ye yüklenir. En son olarak programın doğruluğu denetlenir (Nekoogar ve Moriarty 2002).



Şekil 3.25 Kontrol sisteminde DSP kullanımı

Bu çalışmada, aşağıda verilen özelliklerinden dolayı ADSP-21992 sayısal işaret işlemcisi kullanılmıştır. Analog Devices firması, DSP tasarımında Süper-Harward işlemci mimarisini kullanarak oldukça yüksek performanslı elemanlar sunmuş ve ürünlerini ADSP olarak isimlendirmiştir (Kestler 2000). Bu tez çalışmasında kullanılan ADSP-21992 karışık sinyal (mixed-signal) DSP'ler, tek bir

entegre yapısı içerisinde sinyal işleme uygulamaları ve yeni nesil kontrol uygulamaları için yüksek performanslı çözümler sunmaktadır. ADSP-21992 sayısal işaret işlemcisi; 24 bit program ve 16 bit data RAM hafızaya sahip, 160 MHz çalışma frekanslı sabit noktalı (fixed point) bir DSP'dir. ADSP-21992 sayısal işaret işlemcinin temel özellikleri aşağıdaki gibi sıralanabilir.

- 160 MHz, ADSP-219x sabit noktalı DSP çekirdek,
- Dahili referans gerilimli 8 kanal, 14 bit, 20 MSPS ADC,
- 32 K program, 16K data RAM hafıza,
- 1 Mword'a kadar harici hafıza kapasitesi,
- 3 faz PWM üretim ünitesi,
- 2 adet özel PWM çıkışları,
- Artımsal enkoder arabirim ünitesi,
- 3 adet 32 bit genel amaçlı sayıcı, 1 watchdog sayıcı,
- 16 bit genel amaçlı I/O bayrakları,
- CAN, SPORT (senkron seri) ve SPI (seri çevre birim) arabirimleri,
- CROSSCORE^{TM} araç desteği mevcuttur.

Bu özellikler dikkate alındığında ADSP-21992 sayısal işaret işlemciler, birçok uygulamada kullanılabilecek kapasitedir. Bu uygulamaların bazıları aşağıda verilmiştir.

- Endüstriyel motor kontrolü,
- Kesintisiz güç kaynakları,
- Optik ağ kontrolü,
- Veri toplama sistemleri,
- Test ve ölçüm sistemleri,
- Akıllı sensörler,
- Robot kontrol uygulamaları.

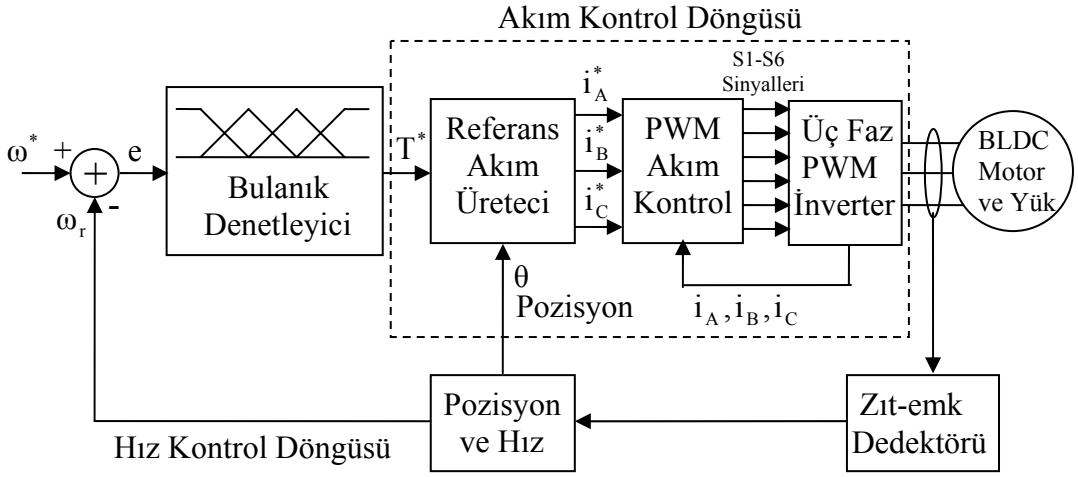
Yapılan çalışmada deneysel donanım için kullanılan ADSP-21992 sayısal işaret işlemcisi ve işlemcinin programlanması ile ilgili ayrıntılı bilgi, EK-A'da verilmiştir.

4. BULANIK DENETLEYİCİLERLE FIRÇASIZ DC MOTORLARIN SENSÖRSÜZ KONTROLÜ

Yapılan çalışmada ilk olarak, geleneksel yollarla tasarlanan bir bulanık denetleyici ile, fırçasız DC motorun sensörsüz kontrolü simüle edilmiş ve pratik olarak gerçekleştirilmiştir. Simülasyon çalışmasında, fırçasız DC motorların fark denklemleri formundaki sayısal modeli elde edilmiş ve bu model kullanılarak fırçasız DC motorun bulanık denetleyici ile kontrolü simüle edilmiştir. Simülasyon ile davranışı incelenen kontrol sistemi, ADSP-21992 sayısal işaret işlemcisi kullanılarak pratik olarak gerçekleştirilmiştir.

4.1 Bulanık Denetleyicili Fırçasız DC Motor Kontrol Sistemi

Gerçekleştirilen fırçasız DC motor kontrol sisteminin temel blok diyagramı Şekil 4.1'de görülmektedir. Blok diyagramından görüldüğü gibi, kontrol sisteminde hız kontrol ve akım kontrol olmak üzere iki ayrı döngü mevcuttur. Hız kontrol döngüsünün görevi; arzu edilen referans hız (ω^*) ile motorun gerçek hızı (ω_r) arasındaki hata değerini kullanarak, referans akım üreteci için referans moment (T^*) değerini üretmek ve motor hızını kontrol etmektir. Sistemde hız kontrol döngüsü için, iki girişli ve tek çıkışlı bir bulanık denetleyici kullanılmıştır. Akım kontrol döngüsünün görevi ise, referans akımlara göre motor akımını kontrol etmektir.



Şekil 4.1 Gerçekleştirilen fırçasız DC motor kontrol sisteminin temel blok diyagramı

Akım kontrol döngüsü içerisindeki referans akım üreteci, üç faz PWM akım kontrolü için referans akım değerlerini üretmektedir. Bir başka ifade ile, pratikte üç faz PWM inverter için doluluk oranını belirlemektedir. Bunun için referans akım üreteci, bulanık denetleyici tarafından üretilen referans moment değerini ve rotor pozisyonunu kullanmaktadır. Üç faz PWM akım kontrol biriminde, referans akımlar ile motorun gerçek akımları karşılaştırılmakta ve hata değerleri kullanılarak, inverter için anahtarlama sinyalleri üretilmektedir. Bu sinyaller ile üç faz PWM inverterde kullanılan 6 adet MOSFET kontrol edilmiştir. Fırçasız DC motorların sensörsüz kontrolünde, motor pozisyonunun belirlenmesi için zıt-emk dedektörü kullanılmıştır.

4.2 Fırçasız DC Motorların Simülasyon Modeli

Şekil 4.1'deki sistemin simüle edilebilmesi için, kontrol edilecek olan fırçasız DC motorun sayısal olarak modellenmesi gerekmektedir. Bu amaçla Bölüm 3'de ayrıntılı olarak verilen fırçasız DC motor denklemlerinden, nümerik yöntemler kullanılarak fark denklemleri formunda motorun sayısal modelleri elde edilmiştir. Sayısal modelleme işleminde dikkat edilmesi gereken önemli bazı noktalar vardır. Bunlardan birincisi; sayısal modelleme işleminde, örnekleme periyodunun (h) seçilmesidir. Örnekleme periyodu seçilirken modellenen sistemin tepki hızı dikkate alınmalıdır. Yani, sistemden yeterli sıklıkta örnek alınmalı ve bilgi kaybı olmamalıdır. İkinci önemli nokta; modelleme çalışmaları için mümkün olduğunca az kabullenme yapılmalı ve model elde edilirken tüm fiziksel parametreler dikkate alınmalıdır. Diğer önemli bir nokta ise; nümerik hesaplamalarda hassasiyete ve doğruluğa dikkat etmek gerekmektedir. Bu amaçla uygulamalarda çeşitli nümerik hesaplama yöntemleri kullanılmaktadır. Fırçasız DC motorların simülasyon modeli için, önceki bölümde verilen (3.9) denklemi gerekli matematiksel işlemlerden sonra, aşağıdaki gibi akım denklemi formunda yeniden yazılabilir.

$$\frac{d}{dt}\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \begin{bmatrix} 1/L & 0 & 0 \\ 0 & 1/L & 0 \\ 0 & 0 & 1/L \end{bmatrix}\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} - \begin{bmatrix} R/L & 0 & 0 \\ 0 & R/L & 0 \\ 0 & 0 & R/L \end{bmatrix}\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} - \begin{bmatrix} 1/L & 0 & 0 \\ 0 & 1/L & 0 \\ 0 & 0 & 1/L \end{bmatrix}\begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix}$$

$$(L = L_s - L_m) \qquad (4.1)$$

Burada v_A, v_B, v_C ; faz gerilimlerini, e_A, e_B, e_C ; zıt-emk'ları, i_A, i_B, i_C ; faz akımlarını, L_s ; öz endüktansı ve L_m; karşılıklı endüktansı göstermektedir. Ayrıca hareket denklemi ve rotor pozisyon ifadesi aşağıdaki gibi tekrar yazılabilir.

$$\frac{d}{dt}\omega_r = (\frac{p}{2}(T_e - T_L) - B\omega_r)/J \tag{4.2}$$

$$\frac{d}{dt}\theta = \omega_r \tag{4.3}$$

Simülasyon çalışması için (4.1), (4.2) ve (4.3) denklemlerinin nümerik çözümlerinin yapılması gereklidir. Bunun için bu çalışmada, dördüncü dereceden Runge-Kutta yöntemi kullanılmıştır.

4.2.1 Runge-Kutta yöntemi

Runge-Kutta yöntemi, diferansiyel denklemlerin nümerik çözümlerinde sıkça kullanılan bir yöntemdir. Runge-Kutta yönteminin diğer adı orta nokta metodudur. Runge-Kutta yönteminde temel yaklaşım; bir örnekleme periyodu (h) içinde, birden fazla noktada hesaplama yaparak daha doğru bir sonucun elde edilmesi ve böylece diferansiyel denklemlerin nümerik çözümlerinde hatanın adım adım azaltılmasıdır. Bir örnekleme periyodunda, kaç kez hesaplama yapılacağı Runge-Kutta yönteminin derecesi ile belirlenmektedir. Mühendislik çözümlerinde Runge-Kutta yöntemlerinden en sık kullanılanı, denklem (4.4) ile verilen dördüncü dereceden Runge-Kutta yöntemidir. Dördüncü dereceden Runge-Kutta denklemlerinde, her bir örnekleme periyodu süresince dört ayrı hesaplamanın yapılması gerekmektedir.

$$\begin{aligned}
k_1 &= f\,(x[n], y[n]) \\
k_2 &= f\,(x[n] + \frac{h}{2}, y[n] + \frac{k_1}{2}) \\
k_3 &= f\,(x[n] + \frac{h}{2}, y[n] + \frac{k_2}{2}) \\
k_4 &= f\,(x[n] + h, y[n] + k_3) \\
y[n+1] &= y[n] + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O(h^{4+1})
\end{aligned} \tag{4.4}$$

Burada, h; örnekleme periyodunu, x[n]; diferansiyel denklemin durum değişkenini, y[n]; bir önceki adımda hesaplanan çıkışı ve y[n+1] ise şu anki yeni hesaplanan çıkış değerini göstermektedir. Denklemde $O(h^{4+1})$ ile verilen ifade; dördüncü dereceden Runge-Kutta yöntemini ifade etmektedir (Rice 1983).

Denklem (4.1), (4.2) ve (4.3) ile verilen i_A, i_B, i_C, ω_r ve θ parametreleri, simülasyon için dördüncü dereceden Runge-Kutta yöntemi kullanılarak hesaplanabilir. Bunun için kurulan denklem sistemi, (4.5-4.9) denklemleriyle verilebilir. Denklem sisteminde; i_A için k_1, k_2, k_3, k_4; i_B için l_1, l_2, l_3, l_4; i_C için m_1, m_2, m_3, m_4; ω_r için q_1, q_2, q_3, q_4; ve θ için r_1, r_2, r_3, r_4 değişkenleri kullanılmıştır. Denklem (4.5) ile verilen ifade, örnekleme periyodunun (h) başlangıcında yapılan hesaplamayı, denklem (4.6) ve (4.7) ile verilen ifadeler periyodun orta (h/2) noktasında yapılan hesaplamaları ve denklem (4.8) ile verilen ifade ise periyodun (h) sonunda yapılan hesaplamayı ifade etmektedir. Denklem (4.9) ile parametrelerin yeni değerleri hesaplanmaktadır.

$$
\begin{aligned}
k_1 &= \frac{1}{L}(v_A - R\, i_A[n] - e_A) \\
l_1 &= \frac{1}{L}(v_B - R\, i_B[n] - e_B) \\
m_1 &= \frac{1}{L}(v_C - R\, i_C[n] - e_C) \\
q_1 &= \frac{1}{J}(\frac{p}{2}(T_e - T_L) - B\,\omega_r[n]) \\
r_1 &= \omega_r[n]
\end{aligned}
\qquad (4.5)
$$

$$
\begin{aligned}
k_2 &= \frac{1}{L}(v_A - R\,(i_A[n] + \frac{k_1}{2}) - e_A) \\
l_2 &= \frac{1}{L}(v_B - R\,(i_B[n] + \frac{l_1}{2}) - e_B) \\
m_2 &= \frac{1}{L}(v_C - R\,(i_C[n] + \frac{m_1}{2}) - e_C) \\
q_2 &= \frac{1}{J}(\frac{p}{2}(T_e - T_L) - B\,(\omega_r[n] + \frac{q_1}{2})) \\
r_2 &= \omega_r[n] + \frac{q_1}{2}
\end{aligned}
\qquad (4.6)
$$

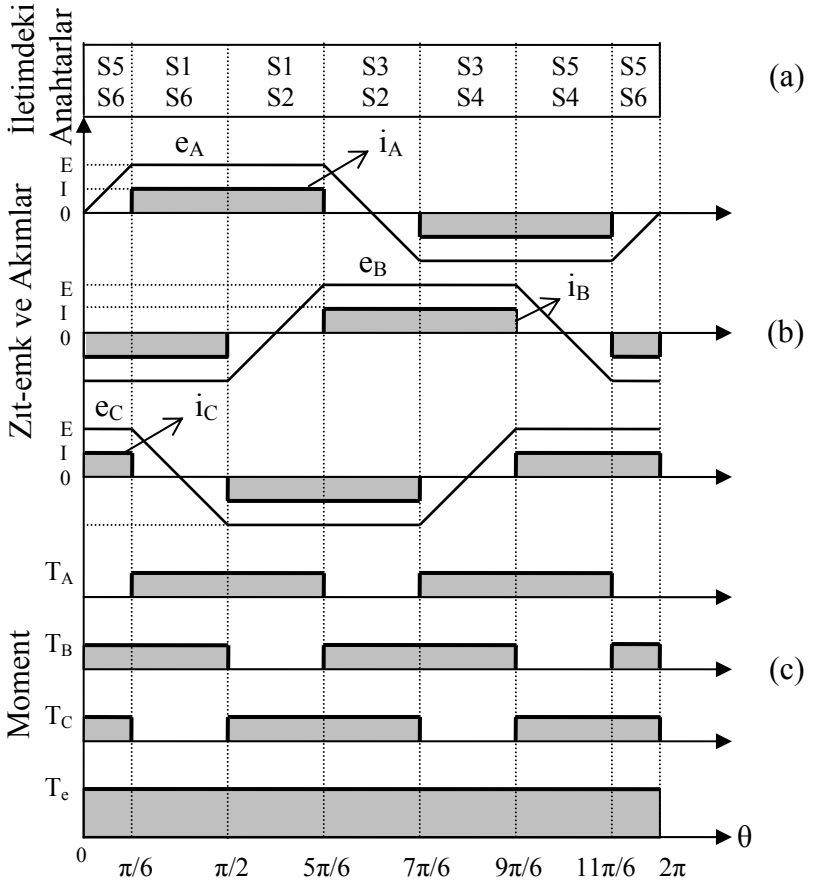
$$
\begin{aligned}
k_3 &= \frac{1}{L}(v_A - R\,(i_A[n] + \frac{k_2}{2}) - e_A) \\
l_3 &= \frac{1}{L}(v_B - R\,(i_B[n] + \frac{l_2}{2}) - e_B) \\
m_3 &= \frac{1}{L}(v_C - R\,(i_C[n] + \frac{m_2}{2}) - e_C) \\
q_3 &= \frac{1}{J}(\frac{p}{2}(T_e - T_L) - B\,(\omega_r[n] + \frac{q_2}{2})) \\
r_3 &= \omega_r[n] + \frac{q_2}{2}
\end{aligned}
\tag{4.7}
$$

$$
\begin{aligned}
k_4 &= \frac{1}{L}(v_A - R\,(i_A[n] + k_3) - e_A) \\
l_4 &= \frac{1}{L}(v_B - R\,(i_B[n] + l_3) - e_B) \\
m_4 &= \frac{1}{L}(v_C - R\,(i_C[n] + m_3) - e_C) \\
q_4 &= \frac{1}{J}(\frac{p}{2}(T_e - T_L) - B\,(\omega_r[n] + q_3)) \\
r_4 &= \omega_r[n] + q_3
\end{aligned}
\tag{4.8}
$$

$$
\begin{aligned}
i_A[n+1] &= i_A[n] + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
i_B[n+1] &= i_B[n] + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4) \\
i_C[n+1] &= i_C[n] + \frac{h}{6}(m_1 + 2m_2 + 2m_3 + m_4) \\
\omega_r[n+1] &= \omega_r[n] + \frac{h}{6}(q_1 + 2q_2 + 2q_3 + q_4) \\
\theta[n+1] &= \theta[n] + \frac{h}{6}(r_1 + 2r_2 + 2r_3 + r_4)
\end{aligned}
\tag{4.9}
$$

Yukarıda verilen denklem sisteminde, p; kutup sayısını ve n ise fark denklem sistemlerinde işlem adımlarını göstermektedir. Verilen denklemler ile fırçasız DC motorlarda faz akımları, rotor hızı ve pozisyonu elde edilmiştir.

Simülasyon çalışmasında, fırçasız DC motorlarda üretilen momentin bulunması için motor zıt emk gerilimlerinin hesaplanması gerekmektedir. Şekil 4.2'de, fırçasız DC motorların anahtarlama durumları, ideal zıt-emk faz gerilimleri, faz akımları ve moment dalga şekilleri görülmektedir.



Şekil 4.2 (a) İnverter anahtarlama durumları (b) üç faz zıt emk gerilim ve akımları (c) faz momentleri ve toplam moment

Fırçasız DC motorların zıt-emk dalga şekilleri, Şekil 4.2'den de görüldüğü gibi, rotor pozisyonuna (θ) bağlı trapezoidal bir yapıdadır. Matematiksel olarak zıt-emk gerilim değerleri;

$$\begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix} = E \begin{bmatrix} f_A(\theta) \\ f_B(\theta) \\ f_C(\theta) \end{bmatrix} \qquad (4.10)$$

$$(E = k_e \omega_r)$$

ile gösterilebilir. Burada; k_e, zıt-emk sabiti, $f_A(\theta)$, $f_B(\theta)$ ve $f_C(\theta)$ sırasıyla A, B ve C fazlarına ait zıt-emk dalga şekillerini tanımlayan şekil fonksiyonlarıdır. Yukarıda f(.) ile gösterilen şekil fonksiyonları, Şekil 4.2 göz önüne alınarak aşağıdaki gibi tanımlanabilir.

$$f_A(\theta) = \begin{cases} (6/\pi)\theta & (0 < \theta \le \pi/6) \\ 1 & (\pi/6 < \theta \le 5\pi/6) \\ -(6/\pi)\theta + 6 & (5\pi/6 < \theta \le 7\pi/6) \\ -1 & (7\pi/6 < \theta \le 11\pi/6) \\ (6/\pi)\theta - 12 & (11\pi/6 < \theta \le 2\pi) \end{cases} \quad (4.11)$$

$$f_B(\theta) = \begin{cases} -1 & (0 < \theta \le \pi/2) \\ (6/\pi)\theta - 4 & (\pi/2 < \theta \le 5\pi/6) \\ 1 & (5\pi/6 < \theta \le 9\pi/6) \\ -(6/\pi)\theta + 10 & (9\pi/6 < \theta \le 11\pi/6) \\ 1 & (11\pi/6 < \theta \le 2\pi) \end{cases} \quad (4.12)$$

$$f_C(\theta) = \begin{cases} 1 & (0 < \theta \le \pi/6) \\ -(6/\pi) + 2 & (\pi/6 < \theta \le \pi/2) \\ -1 & (\pi/2 < \theta \le 7\pi/6) \\ (6/\pi) - 8 & (7\pi/6 < \theta \le 9\pi/6) \\ 1 & (9\pi/6 < \theta \le 2\pi) \end{cases} \quad (4.13)$$

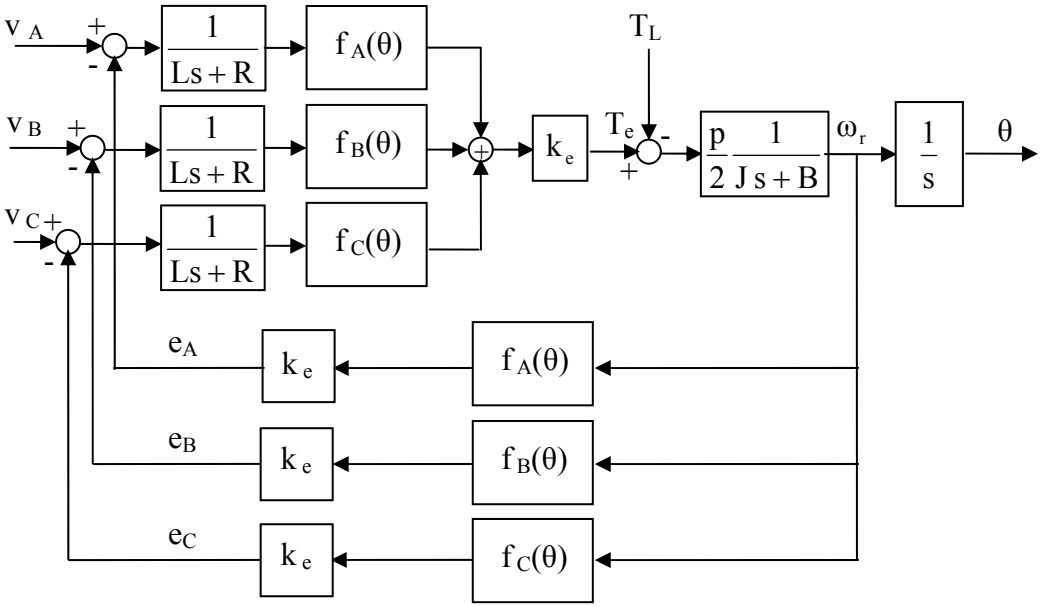
Fırçasız DC motor için moment ifadesi, (4.10)-(4.13) denklemleri kullanılarak, aşağıdaki gibi tekrar yazılabilir.

$$T_e = k_e (f_A(\theta)\, i_A + f_B(\theta)\, i_B + f_C(\theta)\, i_C) \quad (4.14)$$

Hesaplamalar sonucu elde edilen (4.5)-(4.14) denklemleri, fırçasız DC motor simülasyon çalışmalarında kullanılan matematiksel denklemlerdir. Elde edilen matematiksel ifadeler kullanılarak fırçasız DC motorların blok diyagramı çizilebilir. Bunun için motor faz akımları, laplas operatörü $(s = d/dt)$ kullanılarak aşağıdaki gibi tekrar yazılabilir.

$$\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \frac{1}{L\,s + R} \begin{bmatrix} v_A - e_A \\ v_B - e_B \\ v_C - e_C \end{bmatrix} \quad (4.15)$$

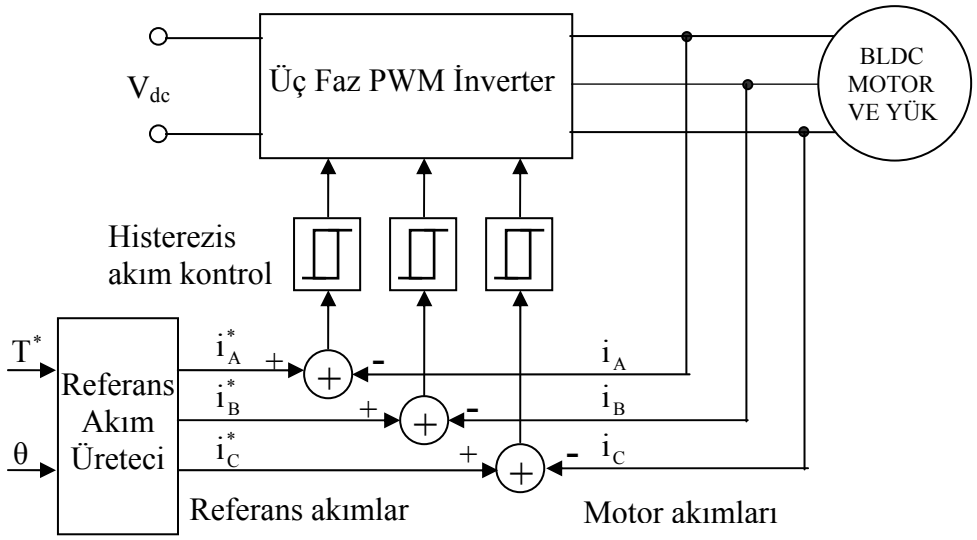
(4-10)-(4.15) denklemleri kullanılarak, bir fırçasız DC motorun blok diyagramı aşağıdaki gibi gösterilebilir.



Şekil 4.3 Fırçasız DC motorun blok diyagramı

4.3 Fırçasız DC Motor Sürücülerde Akım Kontrol Döngüsünün Modellenmesi

Fırçasız DC motorların kontrolünde, akım kontrol döngüsü olarak bilinen bir iç kontrol döngüsüne ihtiyaç vardır. Fırçasız DC motor sürücülerinin simülasyonu için, Şekil 4.4 ile verilen akım kontrol döngüsünün modellenmesi gerekmektedir.



Şekil 4.4 Fırçasız DC motor akım kontrol birimi

Şekil 4.4 ile verilen fırçasız DC motor akım kontrol döngüsü; referans akım üreteci, histerezis PWM akım kontrol birimi ve üç fazlı PWM inverterden oluşmaktadır.

4.3.1 Referans akım üreteci

Akım kontrol döngüsü içerisinde bulunan referans akım üretecinin girişine, hız kontrol döngüsü tarafından üretilen referans moment (T^*) ve pozisyon dedektöründen elde edilen rotor pozisyon bilgisi (θ) gelmektedir. Bu birimde ilk olarak, referans moment ifadesinden üretilecek olan referans akımların genliği,

$$I^* = T^*/k_t \tag{4.16}$$

denklemine göre hesaplanır. Burada k_t, motor moment sabitidir. Referans akım üreteci, referans motor faz akımlarını (i_A^*, i_B^*, i_C^*), rotor pozisyonuna (θ) bağlı olarak hesaplanan akım genliği (I^*) cinsinden Şekil 4.2'ye göre Tablo 4.1'deki gibi belirler. Üretilen bu akımlar, PWM akım kontrol birimine giriş olarak verilmektedir.

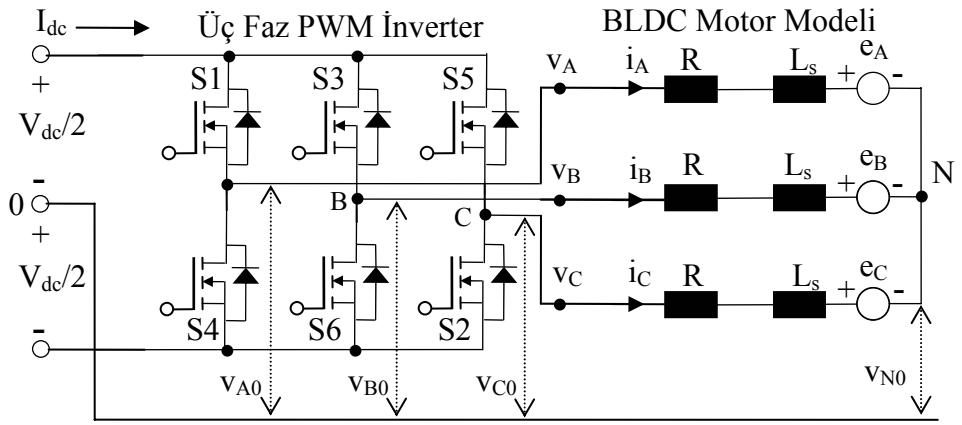
Tablo 4.1 Referans akımlar

Rotor pozisyonu (θ-Derece)	Referans akımlar (A)		
	i_A^*	i_B^*	i_C^*
0-30	0	$-I^*$	I^*
30-90	I^*	$-I^*$	0
90-150	I^*	0	$-I^*$
150-210	0	I^*	$-I^*$
210-270	$-I^*$	I^*	0
270-330	$-I^*$	0	I^*
330-360	0	$-I^*$	I^*

4.3.2 Üç faz PWM inverter ve histerezis akım kontrol modeli

Histerezis PWM akım kontrol devresinde, referans akım üretecinden gelen referans faz akımları ile, motorun gerçek faz akımları karşılaştırılarak akım hata değerleri elde edilmektedir. Bu değerler, bir histerezis bandından geçirilmekte ve Şekil 4.2 (b)'de verilen anahtarlama durumlarına uygun olarak üç faz PWM inverter anahtarlama sinyalleri ve motor faz gerilimleri üretilmektedir.

Akım kontrol döngüsü içerisinde yer alan üç fazlı PWM inverter ve fırçasız DC motor sisteminin eşdeğer devre şeması, Şekil 4.5'de verilmiştir. Şekilde verilen inverter sisteminde, yarı iletken MOSFET anahtarlar, Şekil 4.2 (b)'de verilen anahtarlama durumlarına göre düzenli bir şekilde iletime sokulmaktadır. Böylece motor sargılarına V_{dc} besleme gerilimi uygulanmaktadır. Şekilden görüldüğü gibi, her bir yarı iletken anahtar, elektriksel olarak 120° iletimde ve 60° kesimde kalmaktadır. Bunun sonucu, herhangi bir anda üç fazlı fırçasız DC motorun iki sargısı enerjili iken, üçüncü sargısı boşta kalmaktadır.



Şekil 4.5 Fırçasız DC motor ve üç faz PWM inverter devresi

Şekil 4.5'de görülen motor eşdeğer devresinde, her bir faz sargısı özdeş kabul edildiği için, üç fazlı sistem dengeli olmaktadır. Dengeli bir sistemde ise, denklem (4.17)'de verildiği gibi, faz gerilimleri, faz akımları ve zıt emk'larının toplamı sıfır olmaktadır. Eşdeğer devreden motor faz gerilimleri, denklem (4.18)'deki gibi yazılır.

$$\begin{aligned} i_A + i_B + i_C &= 0 \\ v_A + v_B + v_C &= 0 \\ e_A + e_B + e_C &= 0 \end{aligned} \tag{4.17}$$

$$\begin{aligned} v_A &= R\, i_A + L\frac{di_A}{dt} + e_A \\ v_B &= R\, i_B + L\frac{di_B}{dt} + e_B \\ v_C &= R\, i_C + L\frac{di_C}{dt} + e_C \end{aligned} \tag{4.18}$$

Benzer şekilde motor yıldız noktasının (N), DC kaynağın sanal orta noktasına göre gerilim farkına v_{N0} denilirse, inverter uç gerilimleri (v_{A0}, v_{B0}, v_{C0}) aşağıdaki gibi yazılabilir.

$$\begin{aligned} v_{A0} &= R\, i_A + L\frac{di_A}{dt} + e_A + v_{N0} \\ v_{B0} &= R\, i_B + L\frac{di_B}{dt} + e_B + v_{N0} \\ v_{C0} &= R\, i_C + L\frac{di_C}{dt} + e_C + v_{N0} \end{aligned} \tag{4.19}$$

Fırçasız DC motorların çalışmasında, iki fazın iletimde ve üçüncü fazın kesimde olduğu düşünülürse; sargı akımlarından biri sıfır, diğerleri ise aynı genlikte fakat birbirlerine zıt yönde olmaktadır. Örneğin faz akımları; $i_A = -i_B$ ve $i_C = 0$ olarak alınabilir. Bu özellik ve denklem (4.17) ile verilen ifadeler göz önüne alınır ve (4.19)'daki ifadeler alt alta toplanıp düzenlenirse,

$$v_{N0} = \frac{1}{3}(v_{A0} + v_{B0} + v_{C0}) \tag{4.20}$$

olarak elde edilir. (4.18) ve (4.19) denklemlerinden faz gerilimleri aşağıdaki gibi yazılabilir.

$$\begin{aligned} v_A &= v_{A0} - v_{N0} \\ v_B &= v_{B0} - v_{N0} \\ v_C &= v_{C0} - v_{N0} \end{aligned} \tag{4.21}$$

Denklem (4.21)'de (4.20) denklemindeki ifade yerine yazılarak faz gerilimleri, DC kaynağın sanal orta nokta gerilimleri cinsinden matris formunda aşağıdaki gibi elde edilir.

$$\begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} v_{A0} \\ v_{B0} \\ v_{C0} \end{bmatrix} \tag{4.22}$$

Akım kontrol döngüsünde, motor akımı hata değerleri (e_{ia}, e_{ib}, e_{ic}), üretilen referans faz akımlarından (i_A^*, i_B^*, i_C^*), gerçek motor faz akımları (i_A, i_B, i_C) çıkartılarak elde edilmiştir.

$$\begin{aligned} e_{ia} &= i_A^* - i_A \\ e_{ib} &= i_B^* - i_B \\ e_{ic} &= i_C^* - i_C \end{aligned} \tag{4.23}$$

Elde edilen akım hata değerleri, tanımlanan bir histerezis bandından ($\pm h_b$) geçirilir ve DC kaynağın sanal orta noktasına göre faz gerilimleri, V_{dc} besleme gerilimi cinsinden aşağıdaki gibi bulunur.

$$\begin{aligned} e_{ia} \geq h_b \;\; \text{ise} \;\; v_{A0} &= V_{dc}/2 \\ e_{ia} \leq h_b \;\; \text{ise} \;\; v_{A0} &= -V_{dc}/2 \end{aligned} \tag{4.24}$$

$$\begin{aligned} e_{ib} \geq h_b \;\; \text{ise} \;\; v_{B0} &= V_{dc}/2 \\ e_{ib} \leq h_b \;\; \text{ise} \;\; v_{B0} &= -V_{dc}/2 \end{aligned} \tag{4.25}$$

$$\begin{aligned} e_{ic} \geq h_b \;\; \text{ise} \;\; v_{C0} &= V_{dc}/2 \\ e_{ic} \leq h_b \;\; \text{ise} \;\; v_{C0} &= -V_{dc}/2 \end{aligned} \tag{4.26}$$

Denklem (4.24), (4.25) ve (4.26)'da verilen ifadeler denklem (4.22)'de kullanılarak motor faz gerilimleri, DC kaynak gerilimi cinsinden elde edilmiştir.

4.3.3 Akım kontrol döngüsü simülasyonu

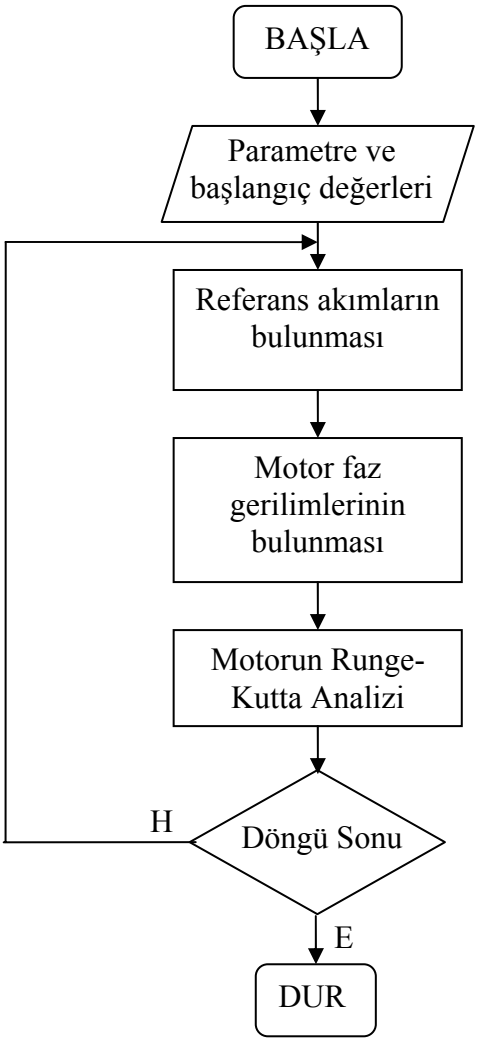
Elde edilen matematiksel ifadeler ve Tablo 4.2'de verilen motor ve inverter parametreleri kullanılarak, fırçasız DC motorun akım kontrol döngüsünün simülasyonu gerçekleştirilebilir.

Şekil 4.6'da akım kontrol döngüsü için simülasyon algoritması verilmiştir. Algoritmadan görüldüğü gibi, sistemde sadece akım denetimi yapılmaktadır. Bu nedenle giriş olarak referans akım genliği (I^*) uygulanmıştır. Giriş akımı ve rotor pozisyon bilgisi kullanılarak referans faz akımları elde edilmiştir. Bu akımlar ile gerçek motor akımları karşılaştırılarak akım hata değerleri bulunmuş ve yukarıda verilen denklemler kullanılarak motor faz gerilimleri elde edilmiştir. Bir sonraki

motor akımı, rotor pozisyonu ve rotor hızı, Runge-Kutta yöntemiyle elde edilmiş, daha sonra motor momenti ve zıt-emk değerleri hesaplanmıştır.

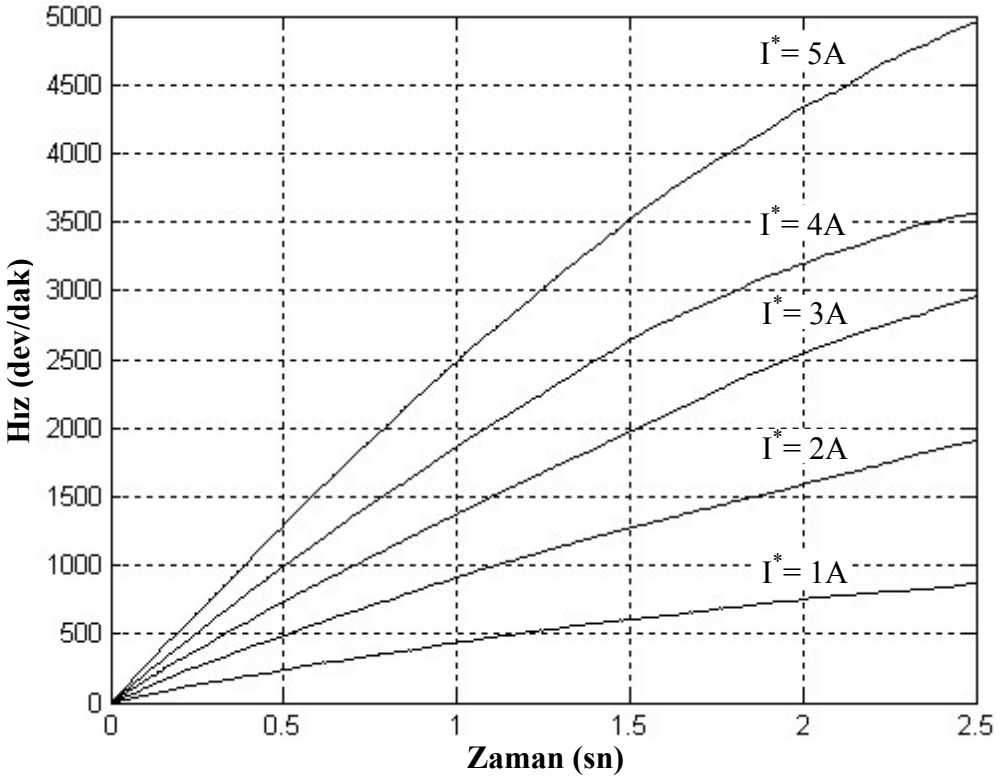
Tablo 4.2 Motor ve inverter parametreleri

Model	AMATEK-119003 Fırçasız DC motor
Güç	106 watt
Faz	üç fazlı
Sargı bağlantı şekli	Yıldız
Nominal hız	4228 dev/dak
Nominal akım	6.8 Amper
Stator eşdeğer direnci	R = 0.348 Ω
Stator eşdeğer indüktansı	L = 0.314 mH
Rotor atalet momenti	J = 0.0019 Ncm-s^2
Kutup sayısı	p = 8
Gerilim sabiti	k_e = 0.0419 V/rad/s
Moment sabiti	k_t = 4.19 Ncm/A
PWM frekansı	f_{PWM}=20 kHz
İnverter besleme gerilimi	V_{dc} = 24 volt
İnverter histerezis aralığı	h_{ba} = 0.5
İnverter akım sınırlayıcı	I_{base} = 20 A

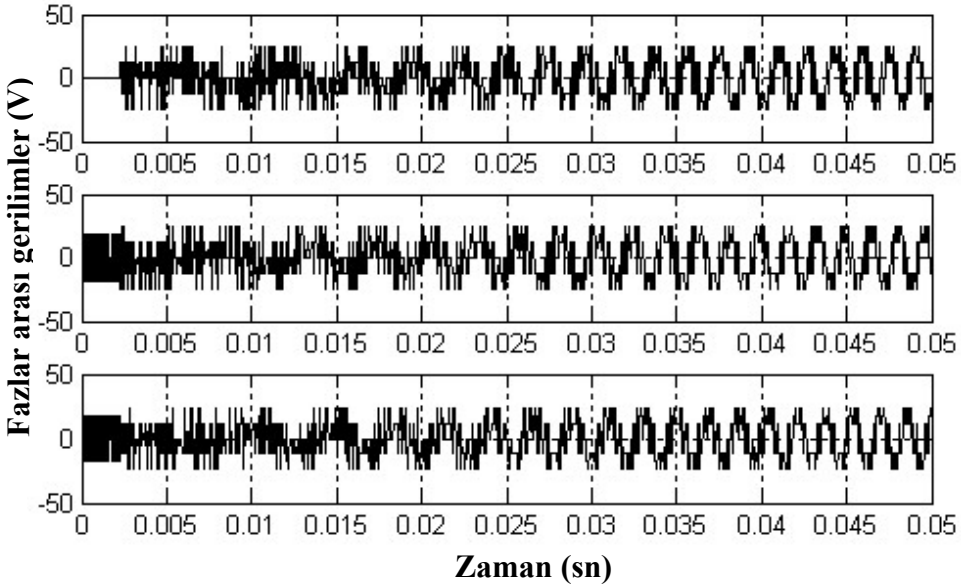


Şekil 4.6 Akım kontrol döngüsü için simülasyon algoritması

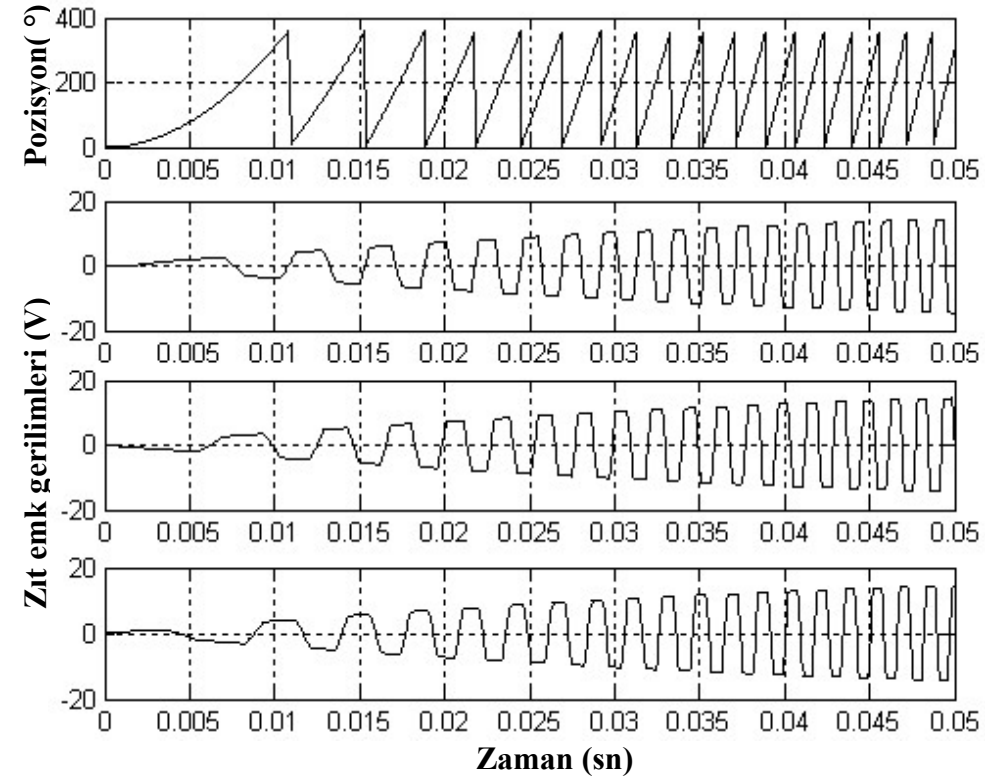
Verilen algoritmaya göre elde edilen simülasyon sonuçları aşağıdaki şekillerde verilmektedir. Şekil 4.7'de, değişik referans akım değerlerine karşılık motor hız cevap eğrileri verilmiştir. Simülasyon işleminde 20 kHz PWM frekansı için örnekleme periyodu h=50 mikrosaniye olarak alınmış ve motor boşta çalıştırılmıştır. Referans akım genliği olarak sırasıyla, I^*= 1A, 2A, 3A, 4A ve 5A değerleri uygulanmıştır. Aşağıdaki eğrilerden görüldüğü gibi, motorun kararlı bir hız noktasına ulaşma zamanı, referans akım genliği ile değişmektedir. I^*= 5A için simülasyonla elde edilen fazlar arası gerilimlerin değişimi Şekil 4.8'de, zıt-emk gerilimlerinin değişimi Şekil 4.9'da, faz akımlarının değişimi Şekil 4.10'da ve üretilen momentler Şekil 4.11'de ayrı ayrı verilmiştir.



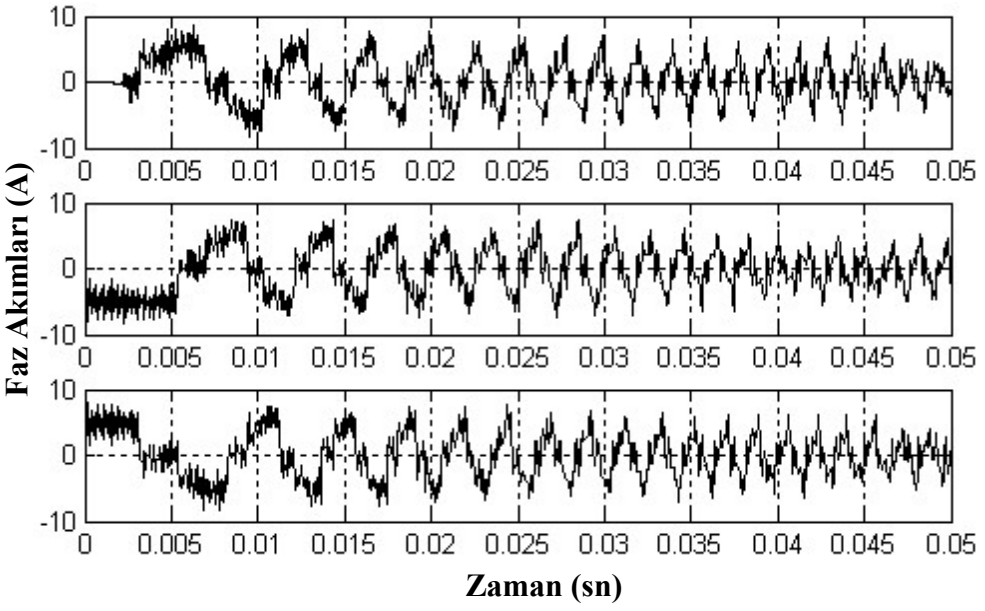
Şekil 4.7 I^*= 1A, 2A, 3A, 4A ve 5A değerlerine karşılık motor hız cevabı



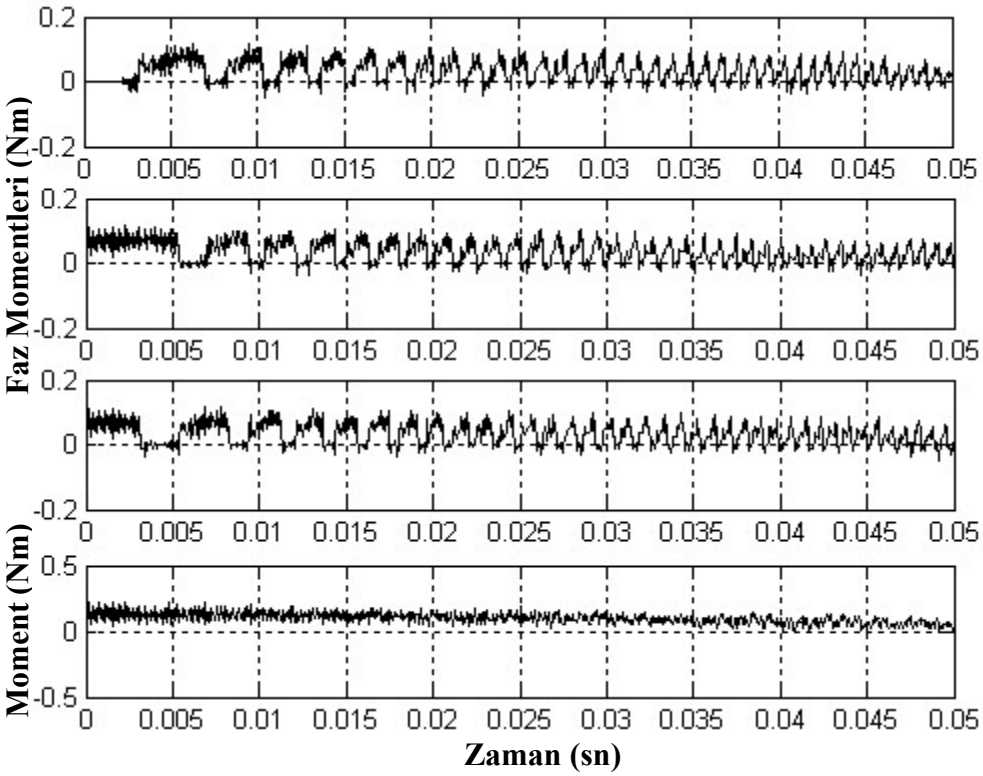
Şekil 4.8 I^*= 5A için fazlar arası gerilimler (volt)



Şekil 4.9 I^*= 5A için pozisyon ve zıt-emk gerilimleri



Şekil 4.10 I^*= 5A için motor faz akımları



Şekil 4.11 I^*= 5A için faz momentleri ve toplam moment

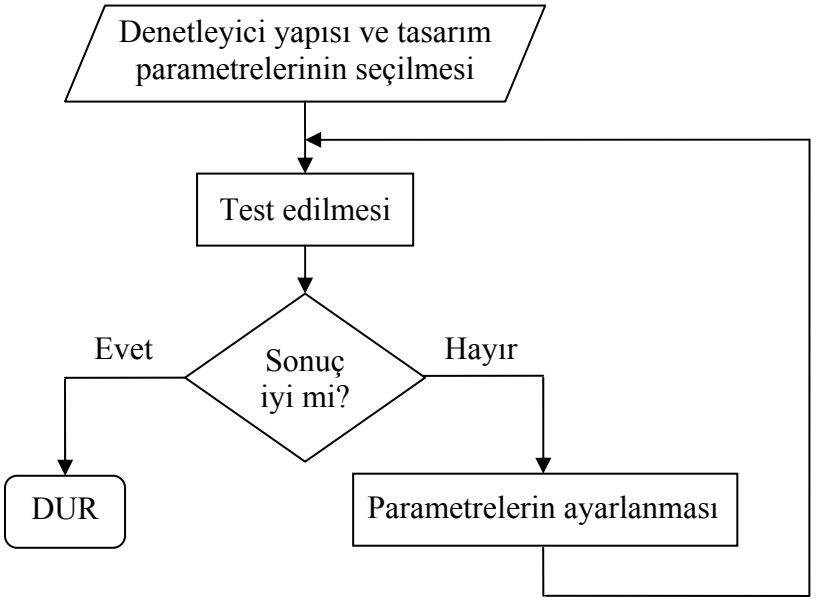
Fırçasız DC motor sürücülerinde, referans momenti (T^*) veya akımı (I^*) üreten ve hız kontrolü yapan dış kontrol döngüsüne ihtiyaç vardır. Bu nedenle simülasyon için dış kontrol döngüsünün tasarlanması gerekmektedir. Burada dış kontrol döngüsü için, geleneksel yollardan tasarlanan bir bulanık denetleyici kullanılmıştır.

4.4 Geleneksel Bulanık Denetleyicinin Tasarlanması

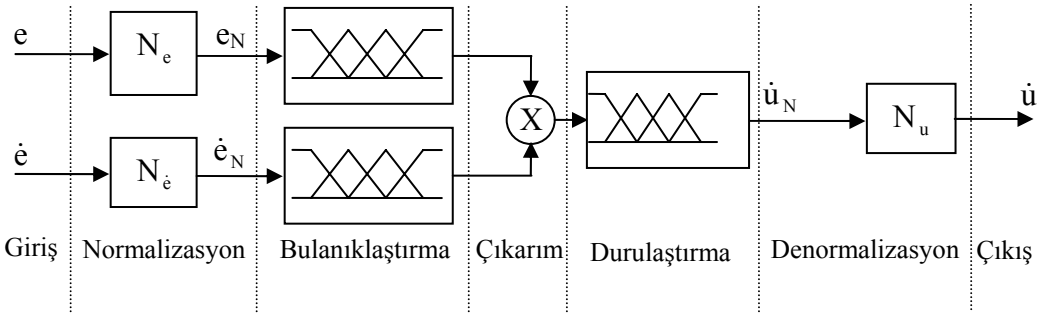
Bulanık denetleyicilerin, geleneksel yollarla tasarımında sistematik bir yaklaşım yoktur. Uzman kişinin bilgi ve becerisi, tasarımda en önemli etkendir. Geleneksel yollarla tasarımda en çok kullanılan yöntem, bulanık denetleyicinin kabaca tasarımını yaparak, deneme yanılma yoluyla denetleyici parametrelerinin daha iyi değerlere ayarlanmasıdır. Geleneksel olarak bir bulanık denetleyicinin tasarımı, diğer denetleyicilerin tasarımında olduğu gibi, denetleyici yapısı ve parametrelerinin belirlenmesiyle başlar. Bulanık denetleyici için üyelik fonksiyonları, kural tablosu ve çıkarım mekanizması belirlenir. Daha sonra kabaca

tasarlanan denetleyici test edilir ve test sonucuna göre denetleyici yapısı ve parametreleri değiştirilir. Şekil 4.12'de, bulanık denetleyicinin geleneksel tasarım algoritması verilmektedir.

Gerçekleştirilen iki girişli, tek çıkışlı bulanık denetleyici; bir çıkarım mekanizması, üç adet lineer olmayan üyelik fonksiyonları (iki bulanıklaştırma ve bir durulaştırma), bir lineer olmayan durulaştırma işlemi ve üç adet lineer ölçekleme (iki normalizasyon ve bir denormalizasyon) elemanlarından oluşmaktadır. Şekil 4.13'de, gerçekleştirilen PI tip bulanık denetleyicinin temel yapısı görülmektedir.



Şekil 4.12 Geleneksel bulanık denetleyici tasarım algoritması



Şekil 4.13 Temel bulanık denetleyici

Bulanık denetleyicinin geleneksel yollarla tasarımında, aşağıdaki adımlar takip edilmiştir.

- Giriş ve çıkışlar tanımlanmış ve kontrol edilecek parametreler ortaya konulmuştur.
- Giriş ve çıkışların seviyeleri belirlenmiş ve ölçekleme işlemi yapılmıştır.
- Dilsel olarak denetleyici karakteristiği tanımlanarak, kural tabanı oluşturulmuş ve çıkarım ünitesi tasarlanmıştır.
- Bulanık denetleyicilerde, keskin değer olan giriş değerlerinin bulanık değerlere, bulanık olan çıkış değerlerinin keskin değerlere dönüştürülmesi için uygun üyelik fonksiyonları seçilmiştir.

4.4.1 Giriş ve çıkışlar

PI tip geri beslemeli kontrol sistemlerinde, hata (e) ve hata değişiminin ($\dot{e}$) denetleyici girişi olarak kullanılması, evrensel bir yaklaşımdır. Gerçekleştirilen bulanık denetleyicide de, hata (e) ve hata değişimi ($\dot{e}$) giriş olarak kullanılmıştır. Hata hesabı, referans hız ile rotor hızı arasındaki fark alınarak aşağıdaki gibi bulunmaktadır.

$$e[n] = \omega^*[n] - \omega_r[n] \tag{4.27}$$

Burada e[n]; o anki hata değerini, $\omega^*[n]$; referans hızı ve $\omega_r[n]$ ise o anki gerçek motor hızını göstermektedir. Gerçek hız, motora bağlanan bir sensör yardımıyla veya bu çalışmada olduğu gibi sensör kullanılmadan zıt emk sıfır geçiş noktasını belirleme yoluyla dolaylı olarak elde edilmektedir. Diğer giriş olan hata değişimi ($\dot{e}$) ise;

$$\dot{e}[n] = e[n] - e[n-1] \tag{4.28}$$

ile bulunmaktadır. Bu denklemde görülen e[n-1] ifadesi, bir önceki hata değerini göstermektedir (Donescu ve ark. 1996).

Gerçekleştirilen PI tip bulanık denetleyici için çıkış, önceki bölümde denklem (3.15)'de verilmiştir. Denetleyici çıkışının hesaplanması için, bu ifadenin toplam biçiminde elde edilmesi gerekmektedir. Bu amaçla,

$$u[n] = u[n-1] + N_u \dot{u}[n] \qquad (4.29)$$

işleminin yapılması gerekmektedir. Burada; u[n] o anki denetleyici çıkış değerini, u[n-1] bir önceki hesaplanan çıkış değerini, N_u çıkış normalizasyon faktörünü ve $\dot{u}[n]$ ise o anki denetleyici çıkış değişimini yani durulaştırıcı çıkışını göstermektedir.

4.4.2 Normalizasyon faktörleri

Genel bir bulanık denetleyicinin tasarımı için, denetleyici giriş değerlerinin [+1,-1] aralığına ve çıkış değerinin de uç kontrol elemanına bağlı olarak arzu edilen değer aralığına normalize edilmesi gerekmektedir. Böylece, gerçekleştirilen bulanık denetleyici, genel bir yapıya dönüşmekte ve tüm çalışma uzayında genel bir çözüm sunmaktadır. Normalizasyon faktörlerinin ($N_e, N_{\dot{e}}, N_u$) seçiminde, sistem giriş ve çıkış seviyeleri önemli olmaktadır. Yani, bulanık denetleyiciye uygulanacak maksimum giriş değerleri ve çıkışta kullanılacak maksimum çıkış değeri göz önünde tutulmalıdır. Normalizasyon faktörleri bu değerleri aşmamalıdır. Bulanık denetleyici içerisinde lineer dönüşüm yapan normalizasyon faktörleri seçilirken, klasik PI tip denetleyicilerle benzeşim yapmak kolaylık sağlamaktadır. Tablo 4.3, yapılan benzeşim sonucu elde edilmiş ve ölçekleme parametrelerinin ayarlanmasında kolaylık sağlamıştır. Ayrıca Tablo 4.3'de, bu çalışma için deneme ve yanılma yoluyla elde edilen normalizasyon faktörleri verilmiştir. Simülasyon çalışmasında, bu değerler kullanılarak sistem çalıştırılmış ve sistemin davranışı gözlenmiştir.

Tablo 4.3 Bulanık denetleyicilerde ölçekleme faktörlerinin ayarlanması

Parametreler / **Hata Özelliği**	$N_{\dot{e}}$ **Artarken**	N_u **Artarken**	N_e **Artarken**
Sürekli durum hatası	Azalır	Azalır/Artar	Artar
Aşma/Osilasyon	Azalır	Azalır	Azalır
Cevap hızı	Artar	Artar	Azalır
Ayarlanmış Değerler	1	2.2	10

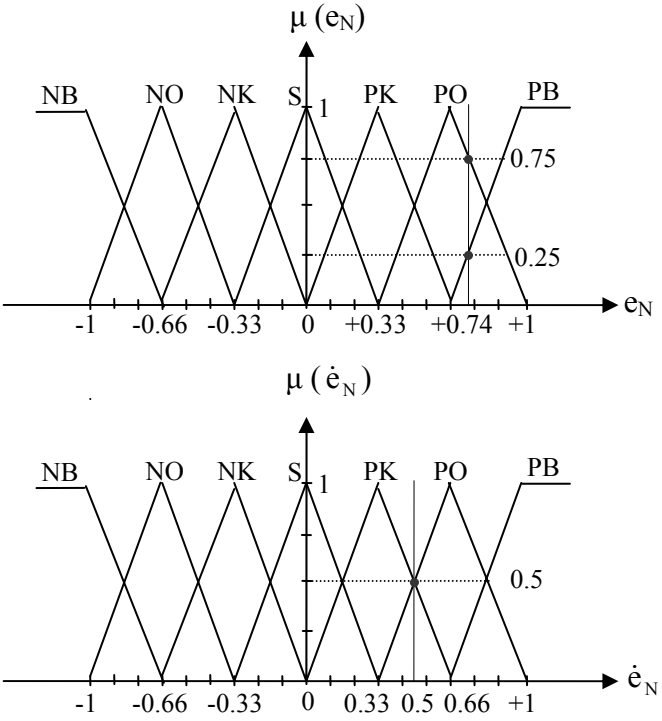
4.4.3 Bulanıklaştırma ve durulaştırma

Bulanıklaştırma ve durulaştırma, birbirinden bağımsız ve lineer olmayan bir işlemdir. Bulanık denetleyicilerde, denetleyici girişlerine keskin değerler olarak tanımlanan gerçek sinyal değerleri uygulanmaktadır. Halbuki bulanık denetleyiciler bulanık değerlerle çalışmaktadır. Bu nedenle, keskin giriş değerlerinin bulanık değerlere dönüştürülmesi gerekmektedir. Dönüşüm işlemi için üyelik fonksiyonu denilen dönüşüm fonksiyonu tanımlanmalıdır. Üyelik fonksiyonları, bulanıklaştırma ve durulaştırma işlemlerinde kullanılan bir ilişkiler kümesidir. Bu kümeler, keskin giriş değerlerini bulanık değerlere ve bulanık değerleri de keskin çıkış değerlerine dönüştürürler.

Gerçekleştirilen bulanık denetleyici sisteminde, hata ve hata değişimi olmak üzere iki, giriş ve çıkışın değişimi olmak üzere de bir çıkış mevcuttur. Denetleyicide, iki keskin giriş (hata-e, hata değişimi-$\dot{e}$) değerini bulanıklaştırmak için, iki ayrı üyelik fonksiyonu tanımlanmıştır. Ayrıca denetleyicide elde edilen bulanık çıkış değerlerinin, keskin çıkış (çıkışın değişimi-$\dot{u}$) değerine durulaştırılması gerekmektedir. Bu amaçla, bir üçgen üyelik fonksiyonu tanımlanmıştır.

Bulanıklaştırma işleminde, giriş değer uzayının kaç adet bulanık küme ile temsil edileceği öncelikle belirlenmelidir. Bunu belirlerken dikkat edilmesi gereken nokta, giriş uzayındaki tüm değerlerin bulanık kümelerde temsil edilmesi yani tüm giriş uzayını bulanık kümelerin kapsamasıdır. Mühendislik uygulamalarında motor hız kontrol sistemleri için genellikle yedi adet bulanık küme kullanılmaktadır (Lee ve Pang 1994). Benzer şekilde bu çalışmada da, giriş değerleri (hata-e, hata değişimi-$\dot{e}$) için yedi adet bulanık küme kullanılmıştır.

Şekil 4.14'de, gerçekleştirilen bulanık denetleyicinin giriş değerleri için tanımlanan üçgen üyelik fonksiyonları verilmiştir. Tanımlamada üçgen üyelik fonksiyonlarının, birbirleriyle %50 oranında örtüştüğü, birinin bittiği noktadan diğerinin başladığı ve her bir fonksiyonun tepe noktası konumunun komşu fonksiyonların alt noktası ile aynı olduğu kabul edilmiştir.



Şekil 4.14 Giriş değerleri için tanımlanan üyelik fonksiyonları

Şekil 4.14'de verilen e_N ve $\dot{e}_N$ ifadeleri, normalize edilmiş giriş değerleridir. Bu değerler, Tablo 4.3 ile verilen ve deneme yanılma yoluyla elde edilen ölçekleme faktörleri kullanılarak aşağıdaki gibi bulunmaktadır.

$$e_N = e/N_e = e/10 \tag{4.30}$$

$$\dot{e}_N = \dot{e}/N_{\dot{e}} = \dot{e}/1 \tag{4.31}$$

Şekil 4.14 ile verilen giriş üçgen üyelik fonksiyonları kullanılarak bulanıklaştırma işleminin nasıl yapıldığı bir örnekle açıklanabilir. Örneğin normalize edilmiş giriş değerleri; $e_N = 0.7425$ ve $\dot{e}_N = 0.5$ olsun. Şekil 4.14'de görüldüğü gibi hata girişi (e) için verilen $e_N = 0.7425$ değeri, üyelik fonksiyonunda PB ve PO bulanık kümeleriyle kesişmektedir. Bu kesişme noktalarının düşey eksendeki değerleri bize üyelik değerlerini vermektedir. Şekil 4.14'den aşağıdaki bulanık değerler elde edilmiştir.

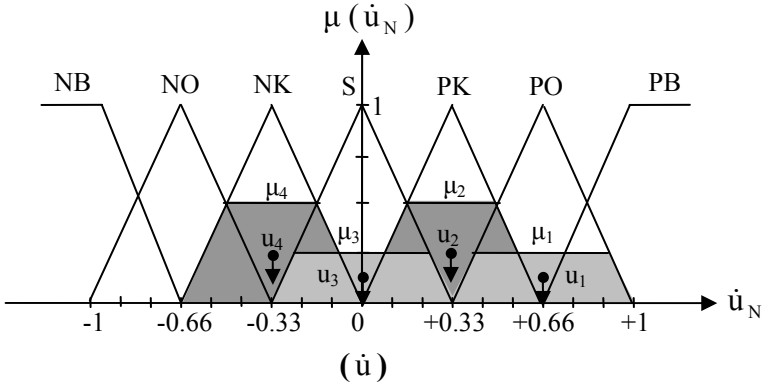
$$\mu_{PB}(e_N) = 0.25$$
$$\mu_{PO}(e_N) = 0.75 \qquad (4.32)$$

Benzer şekilde, hata değişim ($\dot{e}$) için verilen $\dot{e}_N = 0.5$ değeri, üyelik fonksiyonunda PO ve PK bulanık kümeleriyle kesişmektedir. Bu kesişme noktalarının düşey eksendeki değerleri bize üyelik değerlerini vermektedir. Buna göre Şekil 4.14'den aşağıdaki bulanık değerler elde edilmiştir. Elde edilen bulanık değerler, mantıksal çıkarım mekanizması için giriş değerleridir.

$$\mu_{PK}(\dot{e}_N) = 0.5$$
$$\mu_{PO}(\dot{e}_N) = 0.5 \qquad (4.33)$$

Sugeno tip bulanık denetleyicilerde, yapısı gereği durulaştırma işlemine ihtiyaç yoktur. Fakat, Mamdani tip bulanık denetleyicilerde, çıkarım mekanizmasından elde edilen değerler bulanık değerler olduğu için, durulaştırma işleminin yapılması zorunludur (Reznik 1997). Ayrıca, çıkış değerleri bulanık değerler olarak elde edileceğinden dolayı, bu değerler önceden (tasarım başlangıcında) bulanıklaştırılmış olmalıdır. Gerçekleştirilen sistemde, mantıksal çıkarım mekanizmasının çıkışı bulanık bir değerdir. Bu nedenle, durulaştırma işlemi yapılarak elde edilen bulanık değerlerin, girişte olduğu gibi keskin değerlere dönüştürülmesi gerekmektedir. Burada, durulaştırma işlemi için literatürde en sık kullanılan ağırlık merkezi yöntemi kullanılmıştır.

Durulaştırma işleminde de, bulanıklaştırma işleminde olduğu gibi bir üyelik fonksiyonu tanımlanmalıdır. Bulanıklaştırma işlemine benzer şekilde, durulaştırma işlemi için yedi adet bulanık kümeye sahip üyelik fonksiyonu Şekil 4.15'de görüldüğü gibi tanımlanmıştır. Gerçekleştirilen iki girişli, tek çıkışlı bulanık denetleyicide, denklem (4.32) ve (4.33) ile verilen bulanık giriş değerlerine karşılık, çıkarım mekanizmasından μ_1, μ_2, μ_3 ve μ_4 olarak isimlendirilen dört adet bulanık değer elde edilmiştir. Bu bulanık değerler, üyelik fonksiyonunda Şekil 4.15'deki gibi gösterilmiştir. Şekilde gösterilen u_i; i. kümenin ağırlık merkezini ve μ_i ise i. kümenin üyelik değerini ifade etmektedir.



Şekil 4.15 Çıkış değeri için tanımlanan üyelik fonksiyonu

Ağırlık merkezi durulaştırma yönteminin temeli, çıkarım mekanizmasından elde edilen ve örnek olarak Şekil 4.15'de gösterilen bulanık kümelerin, taralı kısımlarının toplam ağırlık merkezinin bulunmasıdır. Bu durum matematiksel olarak denklem (4.34)'de verilmektedir.

$$\dot{u}_N = \frac{\sum_{i=1}^{m} u_i * A(\mu_i)}{\sum_{i=1}^{m} A(\mu_i)} \qquad (4.34)$$

Denklemde verilen m; üyelik fonksiyonu küme sayısını ifade etmekte, u_i; üyelik fonksiyonunda i. bulanık kümenin ağırlık merkezini ve $A(\mu_i)$ ise; Şekil 4.15'de görüldüğü gibi i. bulanık kümenin taralı alanını ifade etmektedir. Taralı bölgenin alan değeri, geometri kurallarına göre kolayca hesaplanmaktadır. Gerçekleştirilen sistemde m=7 olarak alınmıştır.

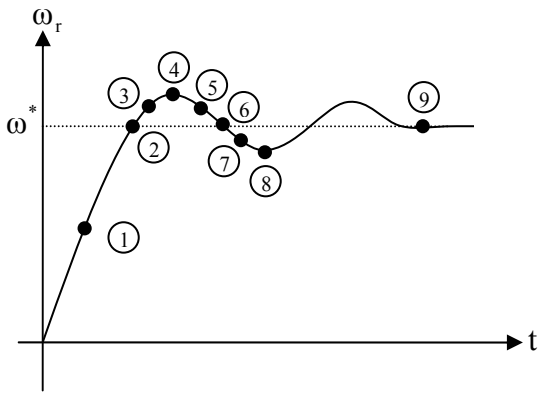
Denklem (4.29) ile verilen çıkış ifadesinde, denklem (4.34) ile elde edilen değer kullanılarak denetleyici çıkışı hesaplanabilir.

4.4.4 Çıkarım mekanizması

Bulanık denetleyicilerde çıkarım mekanizmasında, bulanıklaştırma biriminde elde edilen ve denklem (4.32) ve (4.33) ile verilen bulanık değerlerden, durulaştırma biriminde kullanılan ve Şekil 4.15 ile verilen bulanık çıkış değerlerine dönüşüm işlemi yapılmaktadır. Yapısı gereği bu işlem lineer olmayabilir. Bulanık çıkarım

işlemi, iki aşamada gerçekleştirilmektedir. Birinci aşama; bulanık denetim kurallarının yani kural tabanının elde edilmesidir. İkinci aşama ise; kural tabanı elde edildikten sonra, bulanık karar verme yönteminin yani bulanık çıkarım işleminin belirlenmesidir.

Geleneksel bulanık denetleyicilerde kural tabanı, tamamen uzman bilgisi ve tecrübesine dayalı olarak elde edilmektedir. Tasarımda kural tabanının elde edilmesine yardımcı olması bakımından, sistemin olası cevap eğrisi üzerinde dinamik işaret analizi yapılabilir (Elmas 2003). Bu analizde; girişlerin işaret değerlerine (P-Pozitif, N-Negatif, S-Sıfır) karşılık, çıkışın alması gereken işaret değerleri belirlenmiştir. Şekil 4.16'da olası sistem cevabı ve cevap eğrisi üzerinde işaret analizi yapılan değişik durumlar verilmiştir.



Şekil 4.16 Olası sistem çıkış eğrisi ve işaret analizi durumları

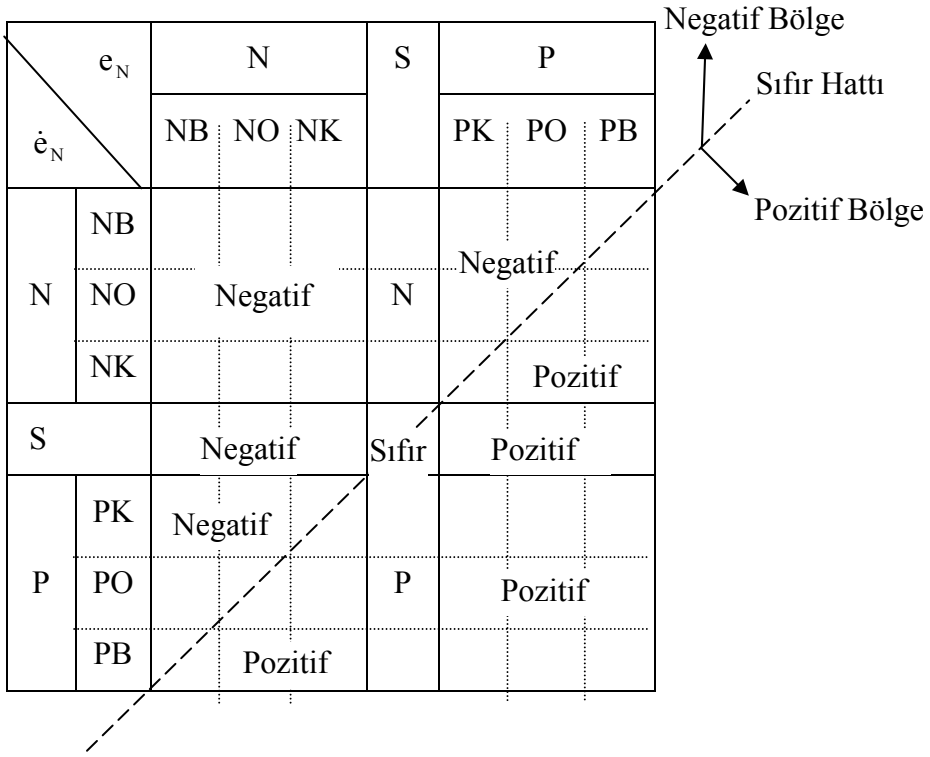
Şekil 4.16'daki cevap eğrisi üzerinde verilen durumlarda yapılan işaret analizlerinden, Tablo 4.4 ile verilen kural tabanı işaret değerleri elde edilmiştir.

Tablo 4.4 Kural tabanı işaret değerleri (P-Pozitif, N-Negatif, S-Sıfır)

Durumlar	1	2	3	4	5	6	7	8	9
e_N	P	S	N	N	N	S	P	P	S
$\dot{e}_N$	N	N	N	S	P	P	P	S	S
$\dot{u}_N$	P-N-S	N	N	N	P-N-S	P	P	P	S

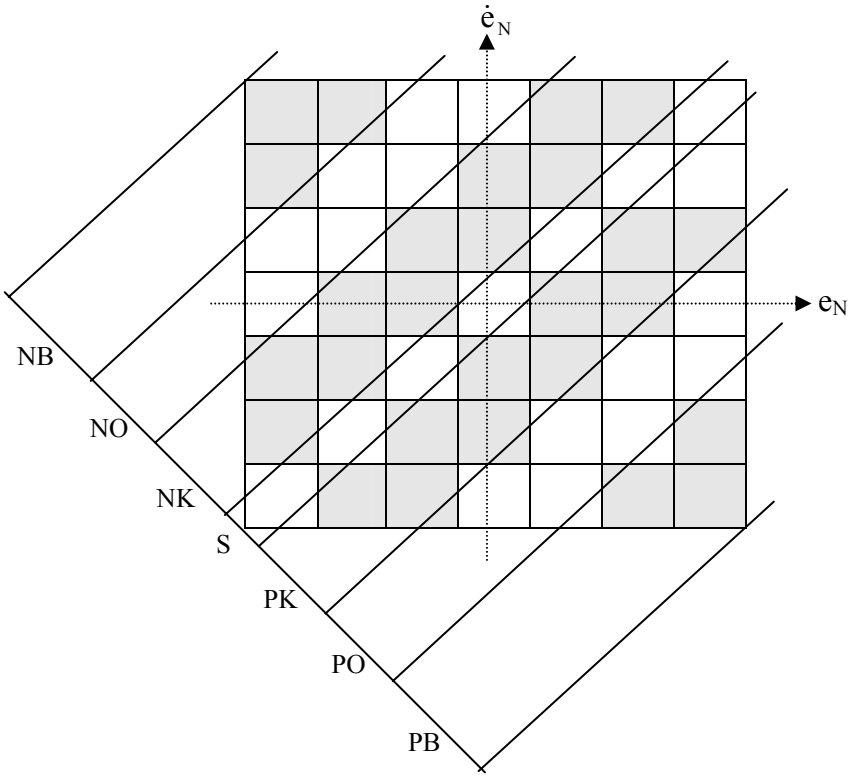
Tablo 4.4'den yararlanılarak gerçekleştirilen sisteme ait 7x7 boyutlu kural tablosu Tablo 4.5'de olduğu gibi oluşturabilir. Tablo 4.5'de görüldüğü gibi, bir sıfır hattı ile tablo ikiye bölünüp, negatif ve pozitif bölgeler oluşturulmuştur. Pozitif ve negatif bölgeler göz önüne alınarak, Tablo 4.6'da görüldüğü gibi sıfır hattına paralel doğrular çizilmek suretiyle, ara değerler elde edilebilir.

Tablo 4.5 Kural tablosu işaret dağılımı



Tablo 4.5 ile verilen kurallar incelendiği zaman, sıfır hattına göre simetrik bir yapıya sahip olduğu görülmekte ve kayan tip kural tablosu olarak isimlendirilmektedir. Bu yapıdaki kural tabloları, optimizasyon işlemlerinde, parametre sayısının azaltılması ve daha kolay optimizasyon yapılması bakımından kolaylık sağlamaktadır. Denetleyici kural tablosunun deneme yanılma yoluyla ayarlanması sonucu elde edilen yeni değerler Tablo 4.7'de verilmektedir.

Tablo 4.6 Geleneksel kural tablosu yapısı



Yukarıda oluşturulan kural tablosu, dilsel olarak da elde edilebilir. Bunun için 7x7=49 adet dilsel tanımlama yapılmalıdır. İki girişli ve tek çıkışlı PI tip bulanık denetleyicide, dilsel tanımlama aşağıdaki gibi yapılmıştır.

R_1 - EĞER e_N = NB ve $\dot{e}_N$ = NB ise $\dot{u}_N$ = NB

R_2 - EĞER e_N = NB ve $\dot{e}_N$ = NO ise $\dot{u}_N$ = NB

R_3 - EĞER e_N = NB ve $\dot{e}_N$ = NK ise $\dot{u}_N$ = NB

R_4 - EĞER e_N = NB ve $\dot{e}_N$ = S ise $\dot{u}_N$ = NO

R_5 - EĞER e_N = NB ve $\dot{e}_N$ = PK ise $\dot{u}_N$ = NO

R_6 - EĞER e_N = NB ve $\dot{e}_N$ = PO ise $\dot{u}_N$ = NK

R_7 - EĞER e_N = NB ve $\dot{e}_N$ = PB ise $\dot{u}_N$ = S

R_8 - EĞER e_N = NO ve $\dot{e}_N$ = NB ise $\dot{u}_N$ = NB

R_9 - EĞER e_N = NO ve $\dot{e}_N$ = NO ise $\dot{u}_N$ = NO

R_{10}- EĞER e_N = NO ve $\dot{e}_N$ = NK ise $\dot{u}_N$ = NO

……

R_{48} - EĞER e_N = PK ve $\dot{e}_N$ = PO ise $\dot{u}_N$ = PB

R_{49} - EĞER e_N = PB ve $\dot{e}_N$ = PB ise $\dot{u}_N$ = PB

Tablo 4.7 Deneme yanılma yoluyla ayarlanan kural tablosu

$\dot{e}_N$ \ e_N	NB	NO	NK	S	PK	PO	PB
NB	NB	NB	NO	NO	NK	NK	S
NO	NB	NO	NO	NK	NK	S	PK
NK	NO	NO	NK	NK	S	PK	PK
S	NO	NK	NK	S	PK	PK	PO
PK	NK	NK	S	PK	PK	PO	PO
PO	NK	S	PK	PK	PO	PO	PB
PB	S	PK	PK	PO	PO	PB	PB

Çıkarım mekanizmasında kural tabanı Tablo 4.7'de verildiği gibi elde edildikten sonra, bulanık çıkarım işleminin belirlenmesi gerekmektedir. Bu çalışmada, çıkarım işlemi için literatürde en sık kullanılan Min-Max çıkarım yöntemi kullanılmıştır.

Bulanık çıkarım, denklem (4.32) ve (4.33)'de verildiği gibi bulanıklaştırılan giriş değerlerinden, çıkış için durulaştırma işleminde kullanılmak üzere bulanık değerler seçilmesi işlemidir. Min-Max çıkarım işleminde bu seçme işlemi, en basit olarak minumum değerin seçilmesiyle yapılmaktadır. Bunun için aşağıdaki ifade kullanılmıştır.

$$\mu_i(\dot{u}_N) = \min\left(\mu(e_N), \mu(\dot{e}_N)\right) \tag{4.35}$$

Tablo 4.7'deki kurallara göre, Denklem (4.32) ve (4.33)'de verilen giriş değerleri için bulanık çıkış değeri, Min-Max çıkarım yöntemi kullanılarak aşağıdaki gibi elde edilmiştir. İki girişli sistemde denklem (4.36)'dan görüldüğü gibi, her girişin üyelik fonksiyonunda iki bulanık değere sahip olduğu düşünülürse, toplam dört adet bulanık çıkış elde edilmektedir. Denklem (4.36) ile elde edilen bulanık değerler, durulaştırma biriminde kullanılmaktadır.

$$\begin{aligned}
\mu_{PB}(\dot{u}_N) &= \mu_1(\dot{u}_N) = \min(\mu_{PB}(e_N), \mu_{PK}(\dot{e}_N)) = \min(0.25, 0.5) = 0.25 \\
\mu_{PB}(\dot{u}_N) &= \mu_2(\dot{u}_N) = \min(\mu_{PB}(e_N), \mu_{PO}(\dot{e}_N)) = \min(0.25, 0.5) = 0.25 \\
\mu_{PO}(\dot{u}_N) &= \mu_3(\dot{u}_N) = \min(\mu_{PO}(e_N), \mu_{PK}(\dot{e}_N)) = \min(0.75, 0.5) = 0.5 \\
\mu_{PB}(\dot{u}_N) &= \mu_4(\dot{u}_N) = \min(\mu_{PO}(e_N), \mu_{PO}(\dot{e}_N)) = \min(0.75, 0.5) = 0.5
\end{aligned} \tag{4.36}$$

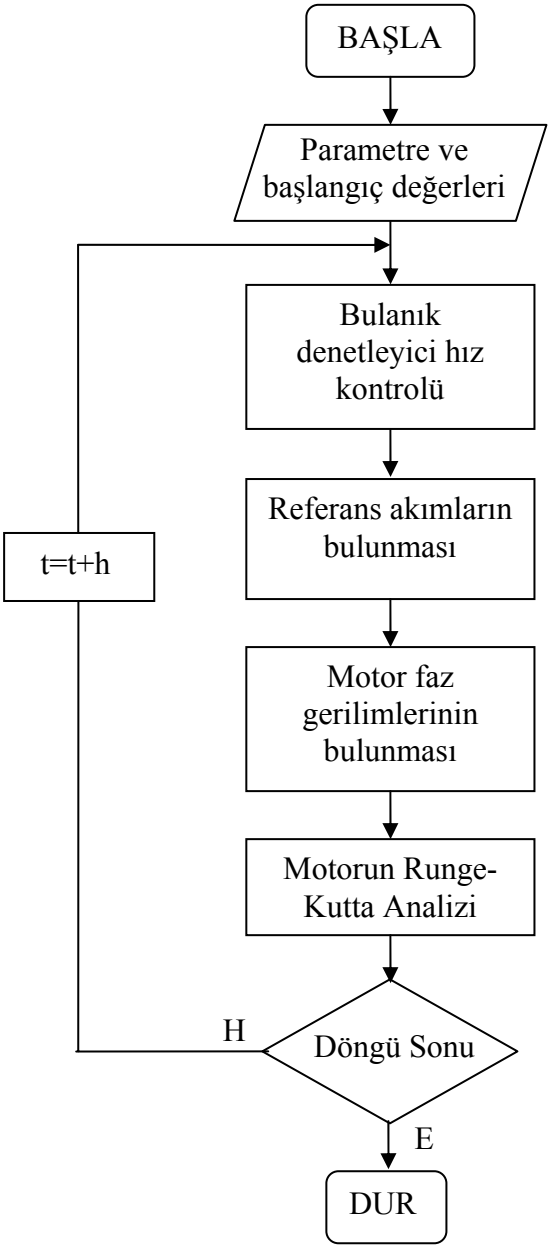
Deneme yanılma ile gerçekleştirilen sistemde, ayarlanan parametre sayısı fazla olduğu için oldukça fazla zaman almakta ve çok sıkıcı olmaktadır. Bunun yanında, sistemde kısmi olarak iyileşme olsa da, bu sonucun optimal çözüm olup olmadığı bilinmemektedir.

4.5 Simülasyon Sonuçları

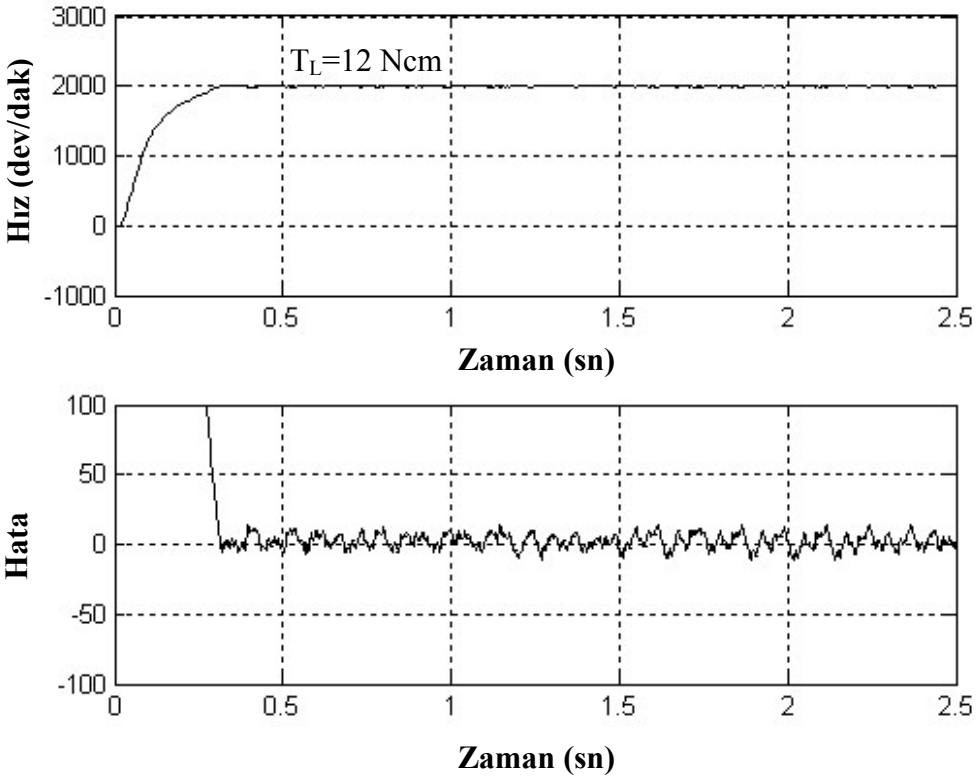
Fırçasız DC motor kontrol sistemi simülasyonunda, parametreleri Tablo 4.8'de verilen geleneksel bulanık denetleyici için Şekil 4.17 ile verilen kontrol algoritması kullanılmıştır. Algoritmada ilk olarak, sistem için gerekli başlangıç parametreleri tanımlanarak döngüye başlanmaktadır. Kontrol döngüsünde ilk olarak, referans akım veya momentin hesaplandığı hız kontrol döngüsü çalıştırılmakta, daha sonra referans motor akımları bulunmaktadır. Buna göre motor faz gerilimleri bulunarak motor analizi yapılmaktadır. Algoritmaya göre elde edilen simülasyon sonuçları, Şekil 4.18'de hız cevap eğrisi ve cevap hatası, Şekil 4.19'da motor faz akımları ve toplam akım, Şekil 4.20'de faz gerilimleri, Şekil 4.21'de faz momentleri ve toplam moment, Şekil 4.22'de pozisyon ve her bir fazın zıt emk gerilimleri şeklinde verilmiştir.

Tablo 4.8 Simülasyon parametreleri

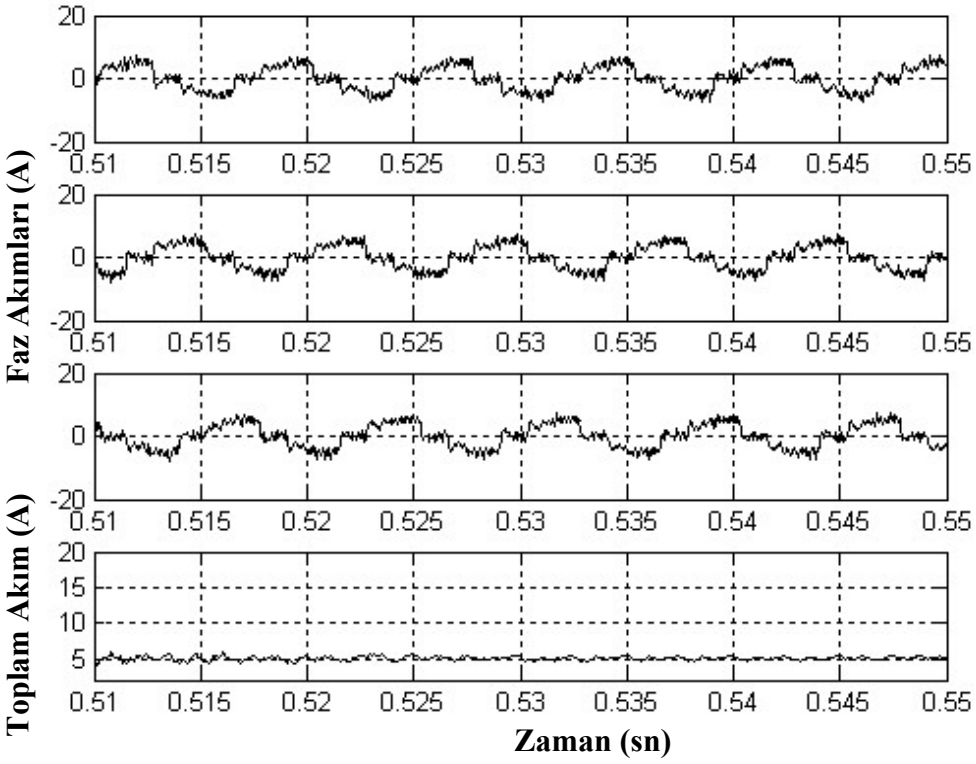
Simülasyon adı	Fırçasız DC motorların geleneksel Bulanık Denetleyici ile kontrolü
Motor ve inverter parametreleri	AMATEK-119003 Fırçasız DC motor, (Tablo 4.2'de verilmiştir)
Bulanık denetleyici parametreleri	Normalizasyon parametreleri: N_e=10, $N_{\dot{e}}$=1, N_u=2.2 Üyelik fonksiyonları: Şekil 4.14, Şekil 4.15 Kural tablosu: Tablo 4.7
Referans hız	ω^* = 2000 dev/dak
Örnekleme periyodu	h = 2.5 ms.
Motor yükü	T_L=12 Ncm (Yarı yük)



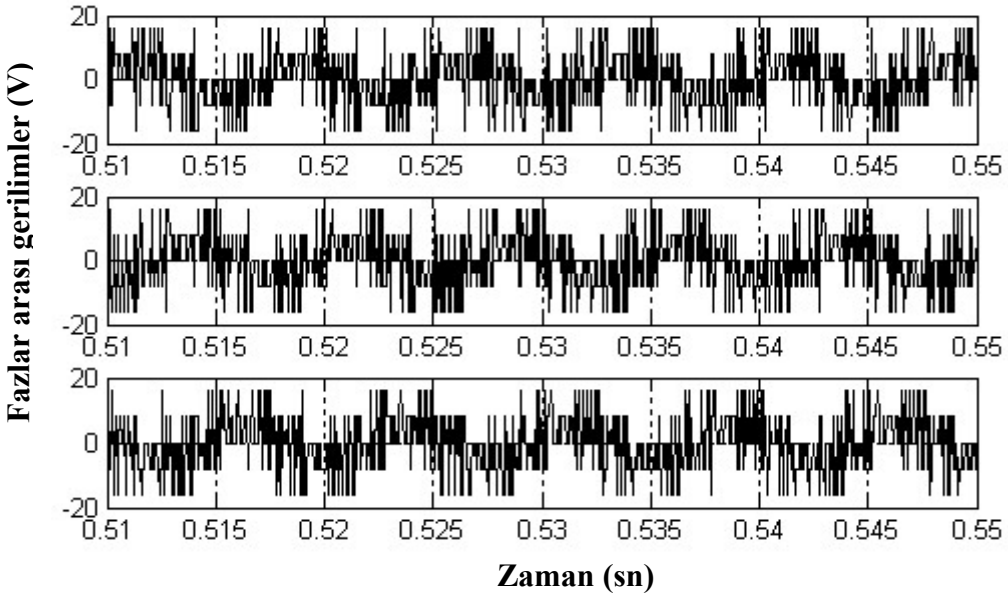
Şekil 4.17 Geleneksel bulanık denetleyici simülasyon algoritması



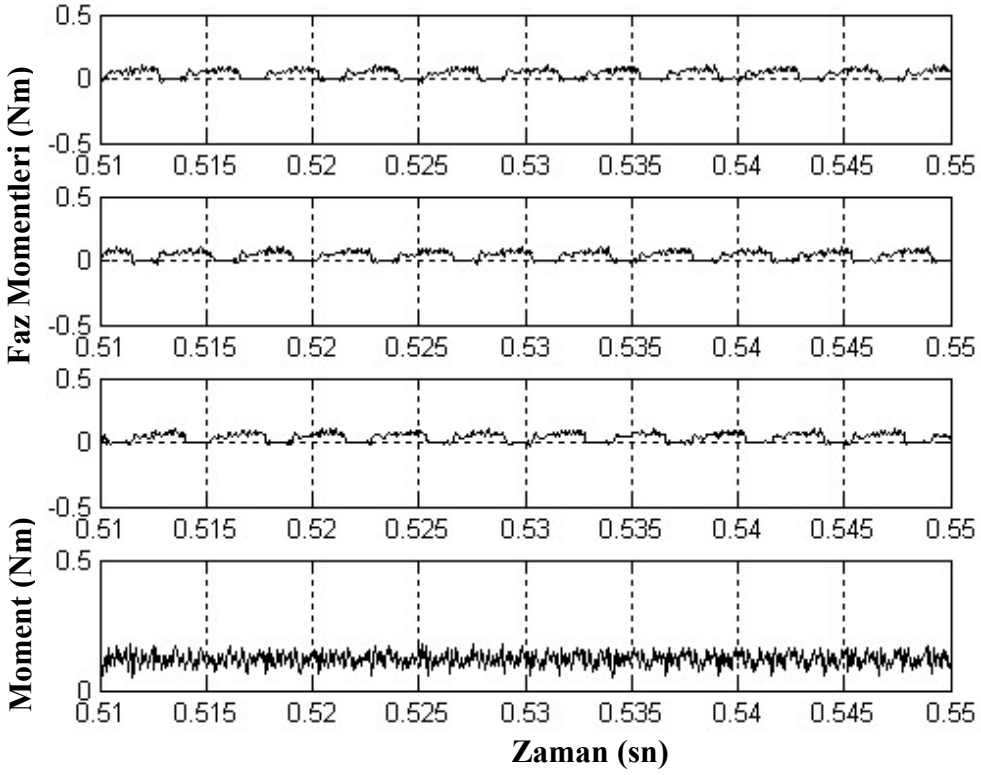
Şekil 4.18 Hız cevap eğrisi ve cevap hatası



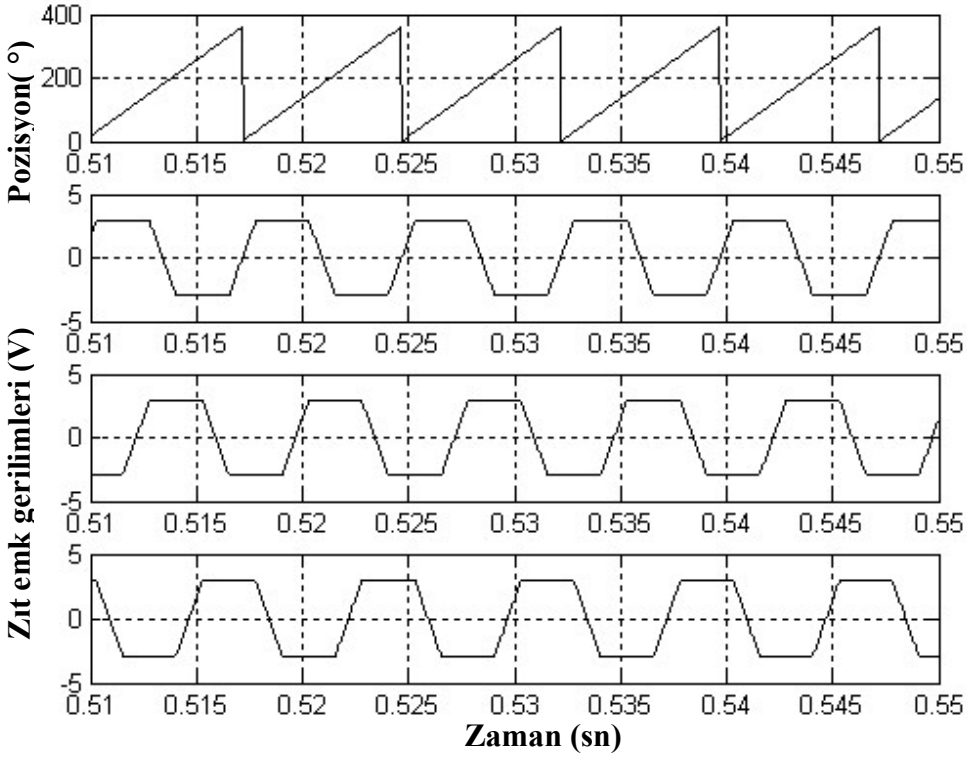
Şekil 4.19 Motor faz akımları ve toplam akım



Şekil 4.20 Motor faz gerilimleri



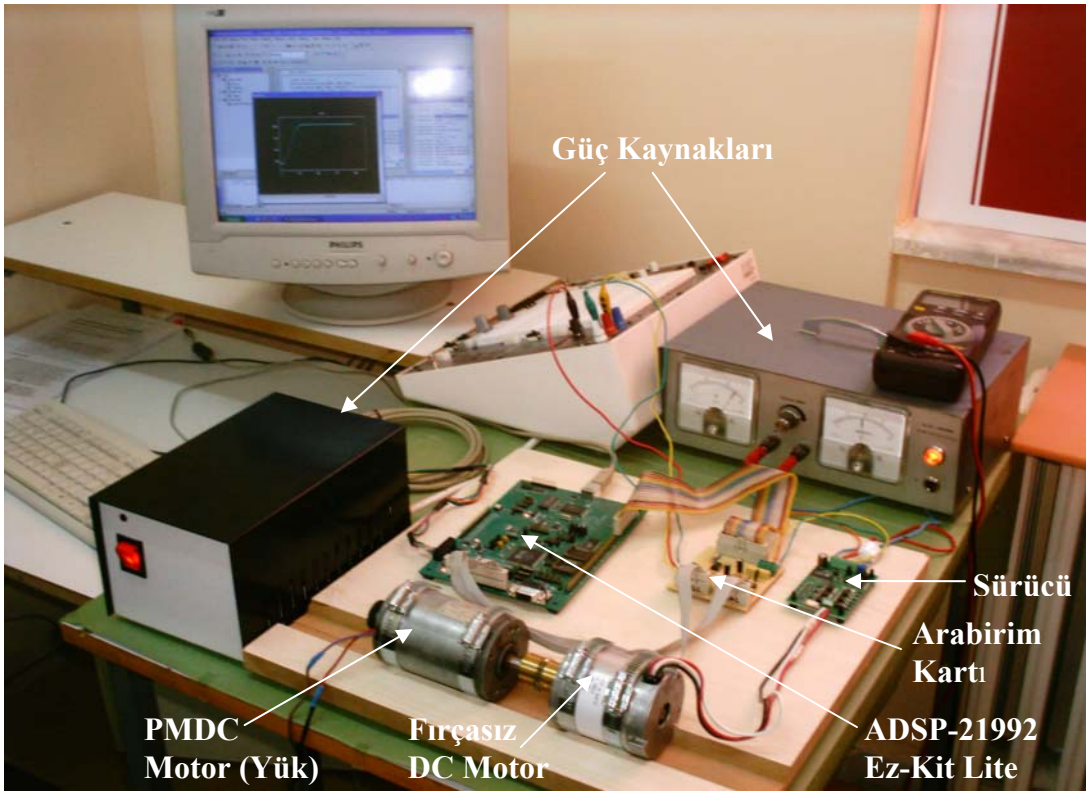
Şekil 4.21 Motor faz momentleri ve toplam moment



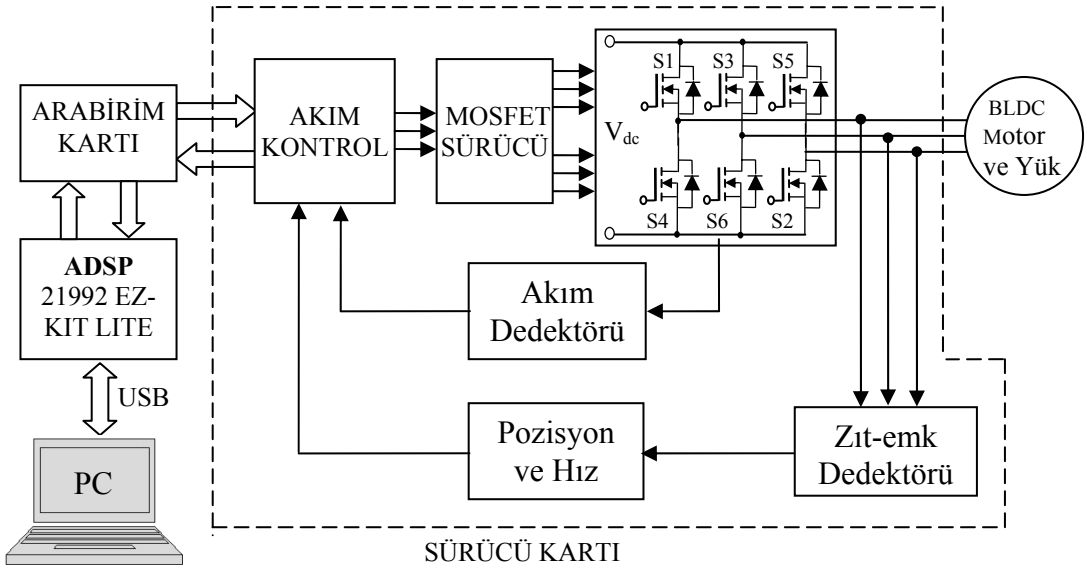
Şekil 4.22 Motor pozisyon ve zıt-emk gerilimleri

4.6 ADSP 21992 Tabanlı BLDC Motor Sensörsüz Kontrol Uygulama Devresi

Şekil 4.23.a'da, pratik olarak gerçekleştirilen ADSP-21992 tabanlı fırçasız DC motor sensörsüz kontrol sistemi ve Şekil 4.23.b'de de bu sistemin blok diyagramı görülmektedir. Şekilde görülen ADSP 21992 EZ-KIT LITE uygulama kartı, detaylı olarak EK A'da incelenmiştir. Fırçasız DC motor kontrolü ve diğer tüm işlemler için yazılan programlar, ADSP uygulama kartı üzerinde yürütülmektedir. Ayrıca, sistem çıkışı da bu kart üzerinden takip edilmektedir. Donanımda görülen arabirim kartında, ADSP uygulama kartı ile sürücü kartı arasında tampon görevi gören devreler mevcuttur. Sürücü kartında; üç faz PWM inverter için 6 adet MOSFET, MOSFET sürücü elemanı, zıt emk dedektörü, akım dedektörü ve akım kontrol elemanı bulunmaktadır.



(a)



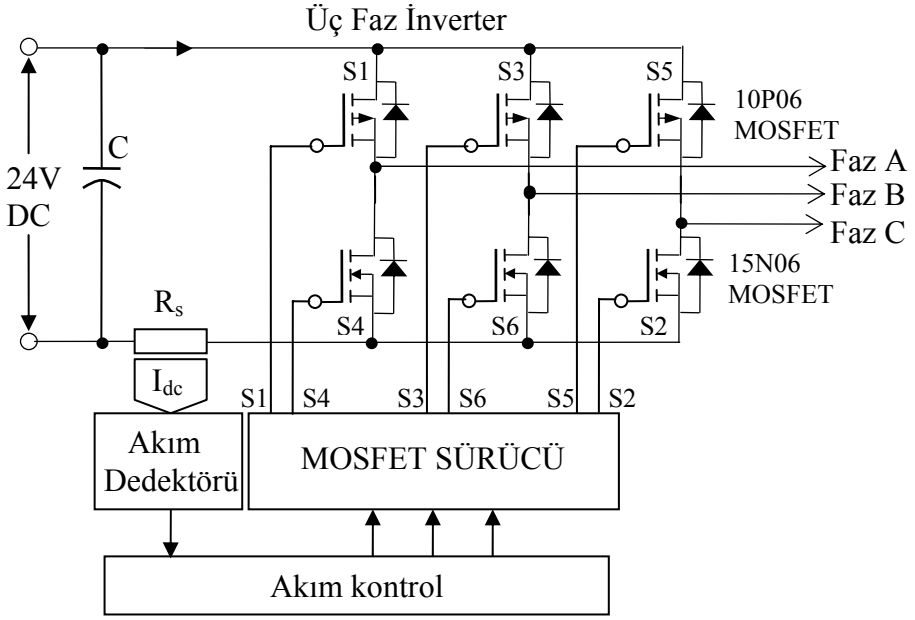
(b)

Şekil 4.23 Pratik olarak gerçekleştirilen sistemin a) fotoğrafı, b) blok diyagramı

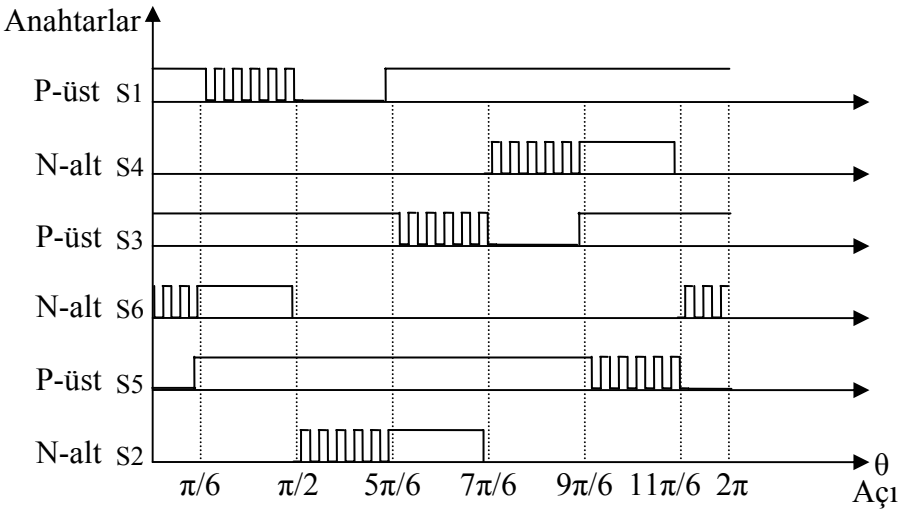
4.6.1 Üç faz PWM inverter

Üç fazlı fırçasız DC motorların çalıştırılmasında, üç faz PWM inverter kullanılması gerekmektedir. Bunun için uygun bir PWM frekansı belirlenmeli ve inverter devre elemanları bu frekansta çalışabilecek özellikte seçilmelidir. Bu çalışmada; PWM frekansı 20 kHz olarak seçilmiş ve Tablo 4.2'de verilen motor özelliğinden dolayı inverter besleme gerilimi V_{dc}=24 volt olarak belirlenmiştir. İnverter devresinde, 3 adet P kanallı 10P06 MOSFET ve 3 adet N kanallı 15N06 MOSFET kullanılmıştır. Devrede, P ve N kanallı MOSFET kullanılması kontrol açısından kolaylık sağlamaktadır. Şekil 4.24'de P ve N kanallı MOSFET ve sürücü elemanına sahip üç faz inverter yapısı verilmiştir.

Şekil 4.25'de, 120° elektriksel açıyla çalışan MOSFET'lerin anahtarlama düzenleri görülmektedir. Şekilden de anlaşılacağı gibi, herhangi bir anda sadece iki MOSFET iletimde, diğerleri kesimdedir. Ayrıca, PWM sinyallerinin iletimde olan MOSFET 'lere senkronize olarak uygulandığı görülmektedir. MOSFET'lerin ısınma problemleri, sürekli iletimde tutulmaları yerine %50 oranında iletimde, %50 oranında kesimde tutularak PWM ile çözülmüştür.



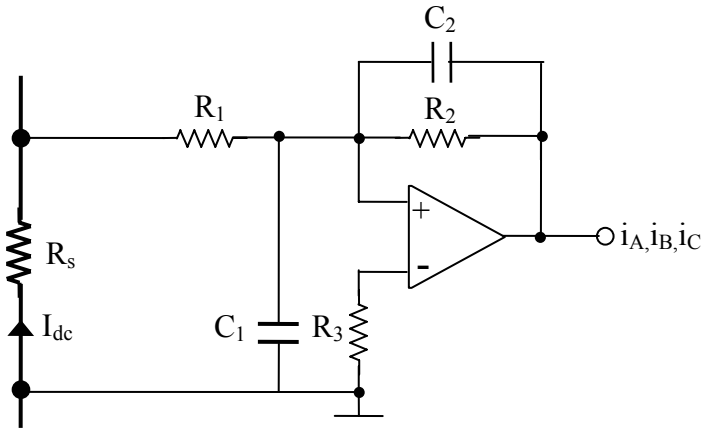
Şekil 4.24 Üç faz PWM inverter yapısı



Şekil 4.25 MOSFET anahtarlama durumları

4.6.2 Akım dedektörü

Fırçasız DC motor sürücülerin akım kontrol döngülerinde kullanılmak üzere motor faz akımlarının ölçülmesi gerekmektedir. Şekil 4.25'de verilen anahtarlama durumlarına göre, yıldız bağlı fırçasız bir DC motor için her bir sargının akımını ayrı ayrı ölçmek yerine, inverter hat akımını ölçmek yeterlidir. Akım kontrol ünitesinde, ölçülen bu akımın hangi faza veya sargıya ait olduğunu belirlemek mümkündür. İnverter hat akımını ölçmek için kullanılan devre Şekil 4.26'da verilmiştir. Hat akımı, inverter devresine seri bir direnç bağlanarak ölçülmektedir. Şekil 4.26'da verilen devrede, hat akımına karşılık gelen seri direnç gerilimi filtrelendikten sonra güçlendirilmekte ve 0-2 volt aralığında bir gerilim olarak çıkışa aktarılmaktadır.



Şekil 4.26 Akım ölçme devresi

Devre çıkış gerilimi, akım kontrol birimine uygulanarak sargı akımları elde edilmektedir. Referans akım ile ölçülen bu akım karşılaştırılarak, kontrol sinyalleri üretilmektedir. Akım kontrol ünitesinde ayrıca, akım koruma devresi de mevcuttur. Akım koruma devresinde, bir komparatör yardımıyla ölçülen hat akımı ile sınır akım karşılaştırılmakta ve gerektiğinde koruma amacıyla inverter devre dışı bırakılmaktadır.

4.6.3 Sensörsüz kontrol ve zıt emk dedektörü

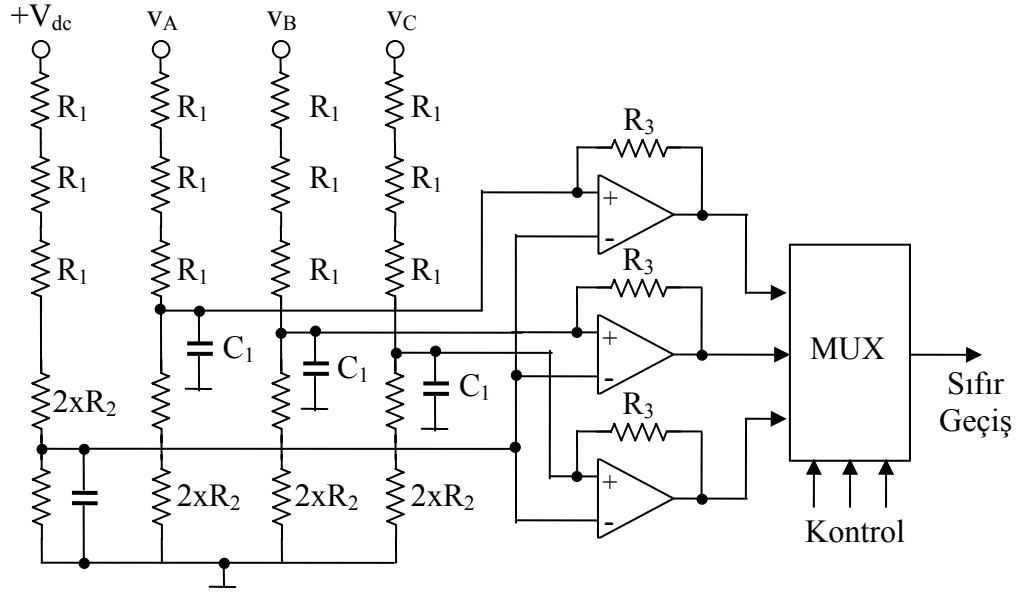
Yapısı gereği fırçasız DC motorların çalıştırılabilmesi için, rotor pozisyonunun bilinmesi ve bu pozisyona göre inverterde komütasyon sinyallerinin yani MOSFET anahtarlama sinyallerinin üretilmesi gerekmektedir. Pozisyon belirleme işleminde en basit yol, hall sensörlerinin kullanılmasıdır. Ancak bu çalışmada, dolaylı olarak rotor pozisyonu belirlenerek hall sensörleri iptal edilmiştir. Rotor pozisyonunun dolaylı olarak belirlenmesinde çeşitli yöntemler kullanılmaktadır. Bu çalışmada, rotorun dönme hareketi sonucu iletimde olmayan faz sargılarında oluşan zıt-emk gerilimlerinin sıfır geçiş noktaları tespit edilerek, rotor pozisyonu belirlenmiştir. Zıt-emk yöntemi, herhangi bir anda fırçasız dc motorun sadece iki fazının iletimde olması ve boşta kalan üçüncü fazdan zıt-emk geriliminin elde edilmesi temeline dayanmaktadır. Örneğin, A ve B fazlarının enerjili, C fazının ise boşta olduğunu veya S3 ve S4 anahtarlarının PWM frekansında iletimde, diğer anahtarların ise kesimde olduğu kabul edilsin. Bu durumda DC kaynağın orta noktasına göre faz gerilim ve akımları aşağıdaki gibi olacaktır.

$$\begin{aligned} &v_{A0} = -\frac{1}{2}V_{dc},\ v_{B0} = +\frac{1}{2}V_{dc} \\ &i_A = -i_B,\ i_C = 0,\ di_C = 0 \end{aligned} \tag{4.37}$$

Denklem (4.19)'da verilen ifade, denklem (4.37) şartlarında ele alınırsa, iletimde olmayan faz (C) gerilimi aşağıdaki gibi elde edilir.

$$v_{C0} = \frac{3}{2}e_C \tag{4.38}$$

Benzer şekilde A ve B fazları için de benzer ifadeler yazılabilir. Denklem (4.38), fırçasız DC motorlarda iletimde olmayan sargı gerilimini ölçmenin, o sargıya ait zıt emk gerilimini ölçmek olduğunu göstermektedir. Buradan hareketle, iletimde olmayan sargı gerilimi ölçülerek zıt emk gerilimi elde edilmekte, buradan da sıfır geçiş noktaları belirlenerek motor pozisyonu veya hızı bulunabilmektedir. Şekil 4.27'de zıt-emk gerilimlerini ölçen devre verilmiştir. Devreden görüldüğü gibi, faz gerilimleri R-C filtreler ile filtrelendikten sonra, iletimde olmayan faz, MUX yardımıyla seçilerek sıfır geçiş noktaları tespit edilmektedir.



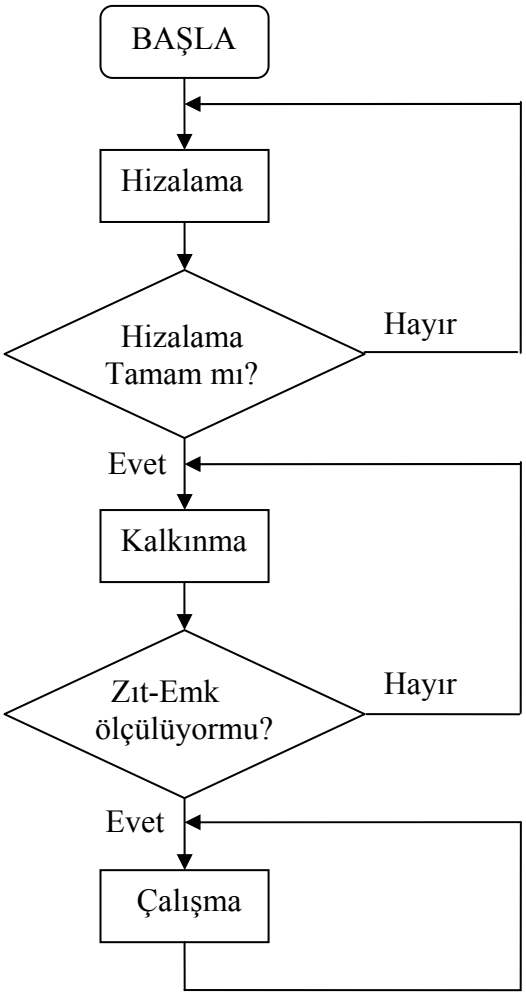
Şekil 4.27 Zıt-emk ölçme devresi

4.6.4 Sistemin çalışması

Fırçasız DC motorların sensörsüz kontrolünde, motorun özel yapısı nedeniyle, üç aşamalı bir kontrol algoritması uygulanmalıdır. Bu aşamalar;

- Hizalama (Rotoru bilinen bir pozisyona ayarlama)
- Kalkınma (Zıt emk yok, açık çevrim)
- Çalışma (Kapalı çevrim)

durumlarıdır Şekil 4.28'de fırçasız DC motorların üç aşamalı çalışma algoritması verilmiştir.



Şekil 4.28 Fırçasız DC motorların çalışma algoritması

Fırçasız DC motorlarda komütasyon sinyalleri, rotor pozisyonuna göre üretilmektedir. Bu nedenle motor çalıştırılmadan önce rotor, bilinen bir pozisyona konumlandırılmalıdır. Bu işleme hizalama denilmektedir. Hizalama işlemi, motorun elektriksel zaman sabitine bağlı olarak sürücü tarafından motorun iki fazına bir hizalama sinyali uygulanarak gerçekleştirilmektedir. Böylece rotor, bilinen bir pozisyona ayarlanmakta ve buna uygun komütasyon işlemi yapılmaktadır. Hizalama işlemi, kararlı bir kalkınma ve yüksek kalkınma momenti elde edebilmek için oldukça önemlidir. Fırçasız DC motorların sensörsüz kontrolünde, düşük hızlarda zıt emk gerilimlerinin çok küçük olmasından dolayı, sağlıklı bir ölçüm yapılamamakta ve pozisyon belirlenememektedir. Zıt emk gerilimlerinin sağlıklı bir biçimde ölçülebildiği motor hızına kadar açık çevrim kontrol uygulanmaktadır. Motorun mekanik ve elektriksel zaman sabitlerine göre, sıfır hızdan itibaren bir lineer

hızlanma sağlanarak ilk kalkınma gerçekleştirilmektedir. Bu aşamada, akım kontrol döngüsü ve akım koruma devresi çalışmaktadır. Gerçekleştirilen sistemde, motor yaklaşık 500 d/dk'ya kadar açık çevrim olarak çalışmaktadır. Fırçasız DC motorların normal çalışmasında, geri besleme sinyali olarak zıt emk gerilimlerinden elde edilen hız bilgisi kullanılmakta ve bulanık denetleyici ile kapalı çevrim kontrol yapılmaktadır.

Geleneksel yollardan tasarlanan ve simülasyonlarda kullanılan bulanık denetleyici, ADSP kod derleyici yardımıyla uygun biçimde ADSP-21992 elemanına kodlanarak, uygulama devresinde de kullanılmıştır. ADSP-21992 üzerinde bulunan dahili 0-2 V'luk girişlere sahip ADC ve uygulama kartında bulunan 0-2 V'luk çıkışa sahip seri DAC elemanlar sayesinde, sistemde bilgi giriş-çıkışı yapılmaktadır. Kapalı çevrim kontrol için, ADC'den alınan geri besleme sinyalleri denetleyicide kullanıldıktan sonra DAC yardımıyla çıkışa aktarılmaktadır. Arabirim kartında, diğer devrelerden alınan sinyaller, ADC için uygun değerlere dönüştürülmekte ve DAC'dan alınan sinyaller de diğer devreler için uygun değerlere çevrilmektedir.

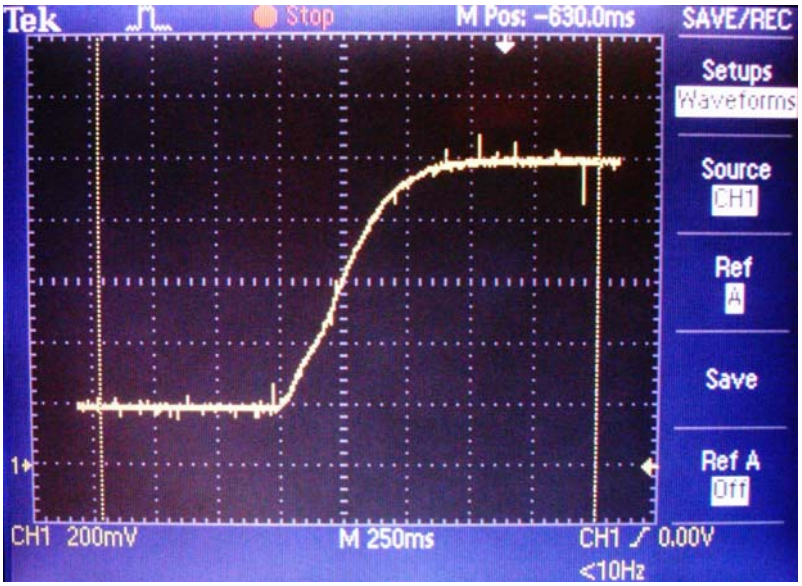
4.7 Deneysel sonuçlar

Uygulamada motor yarı yükte çalıştırılmakta ve 2000 devirde döndürülmektedir. Bulanık denetleyici kısmında geleneksel olarak ayarlanan ve Tablo 4.3'de verilen normalizasyon parametreleri, Şekil 4.14 ve 4.13 ile verilen üyelik fonksiyonları ve Tablo 4.7'de verilen nihai kural tablosu kullanılmıştır. Bu değerlere göre, Şekil 4.29'da motorun referans hıza yerleşme eğrisi görülmektedir. Şekil 4.30'da fazlar arası v_{AB}, v_{BC}, v_{CA} gerilimleri, Şekil 4.31'de A fazı gerilim ve akımı, Şekil 4.32'de v_{A0}, v_{B0}, v_{C0} faz gerilimleri ve zıt-emk dedektör çıkışı, Şekil 4.33'de üç faz alt anahtar (N-MOSFET) kapı sinyalleri ve Şekil 4.34'de ise A fazı alt ve üst anahtar (MOSFET) kapı sinyalleri birlikte görülmektedir.

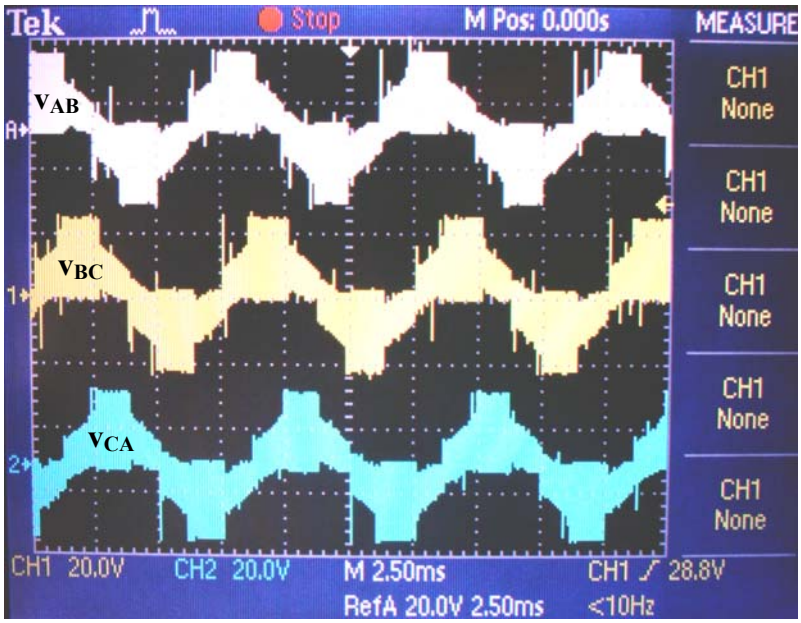
Şekil 4.32'de görülen kare dalga zıt-emk çıkış sinyalinin frekansı, motor hızıyla orantılıdır. Motor hızı;

$$n_r = \frac{40\, f_z}{p} \quad \text{(dev/dak)} \qquad (4.39)$$

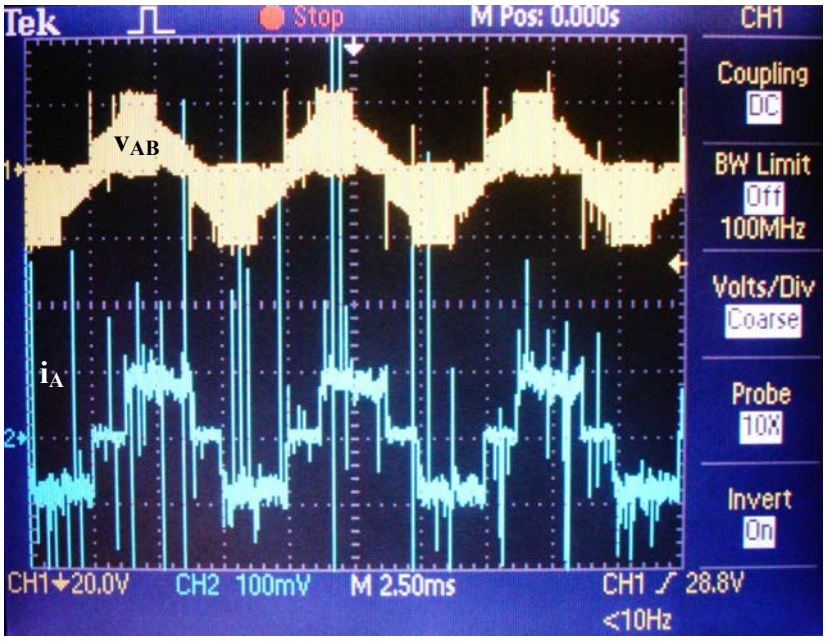
ifadesinden elde edilmektedir. Burada, n_r; dev/dak cinsinden motor hızı, f_z; zıt-emk çıkış sinyali frekansı ve p ise motor kutup sayısıdır. Şekil 4.32'den $f_z = 400$ Hz ve p=8 için n_r=2000 dev/dak bulunmaktadır.



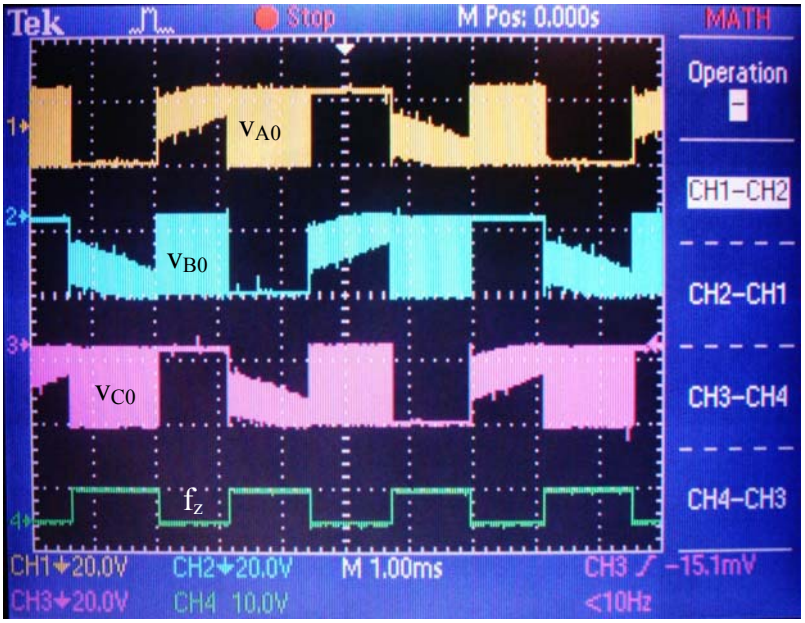
Şekil 4.29 Fırçasız DC motor hız-cevap eğrisi. Time/Div:250ms, Volt/Div:400 d/d



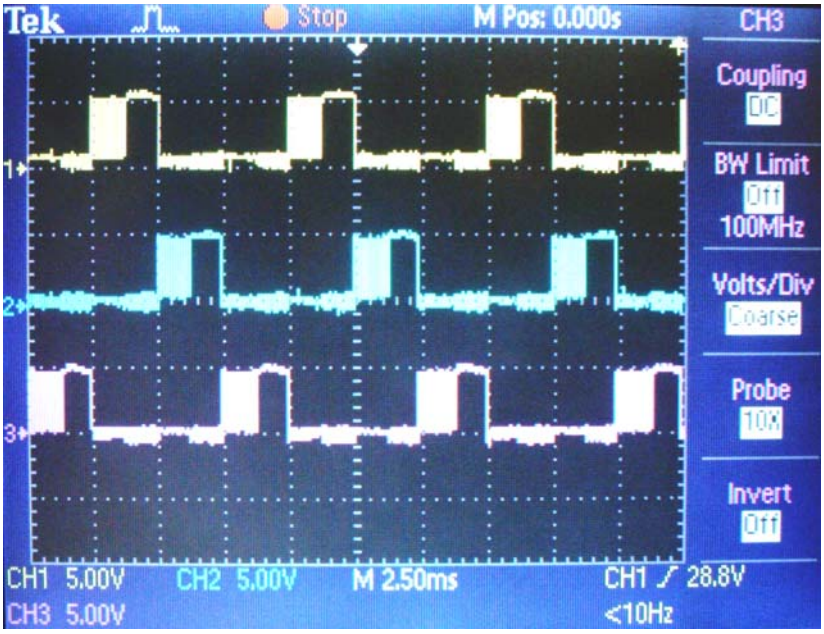
Şekil 4.30 Fırçasız DC motor fazlar arası v_{AB}, v_{BC}, v_{CA} gerilimleri. Time/Div:2.50ms, Volt/Div:20.0V



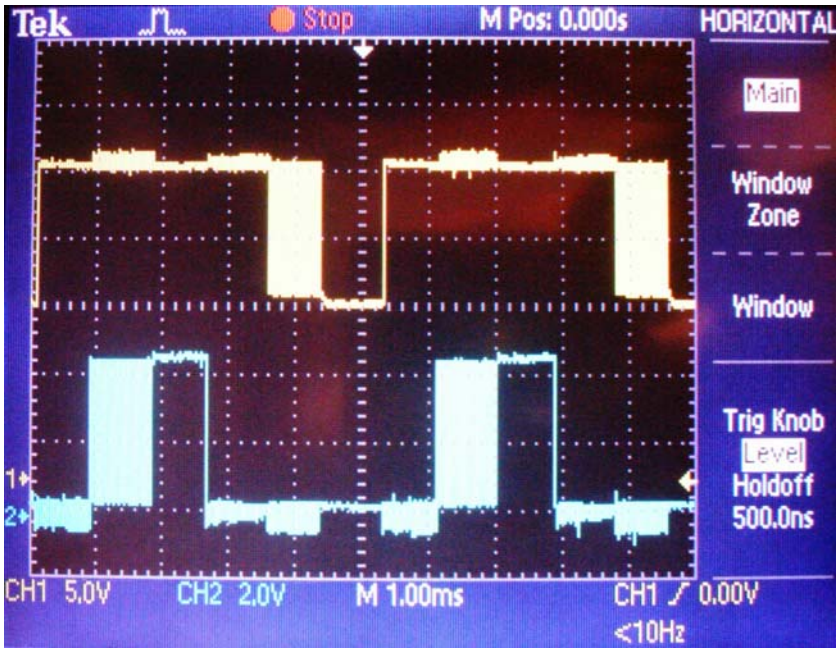
Şekil 4.31 Fırçasız DC motorun v_{AB} gerilimi ve i_A akımı. Time/Div:2.50ms, Volt/Div:20.0V(CH1), Amper/Div:2.00A(CH2)



Şekil 4.32 İnverter v_{A0}, v_{B0}, v_{C0} uç gerilimleri ve zıt-emk dedektör çıkışı. Time/Div:1.00ms, Volt/Div:20.0V(CH1,CH2,CH3), Volt/Div:10.0V(CH4)



Şekil 4.33 İnverterin alt kolundaki N kanal MOSFET'lerin kapı sinyalleri. Time/Div:2.50ms, Volt/Div:5.00V



Şekil 4.34 A fazı için inverter alt (N) ve üst (P) MOSFET kapı sinyalleri Time/Div:1ms, Volt/Div:5.0V(CH1), Volt/Div:2.0V(CH2)

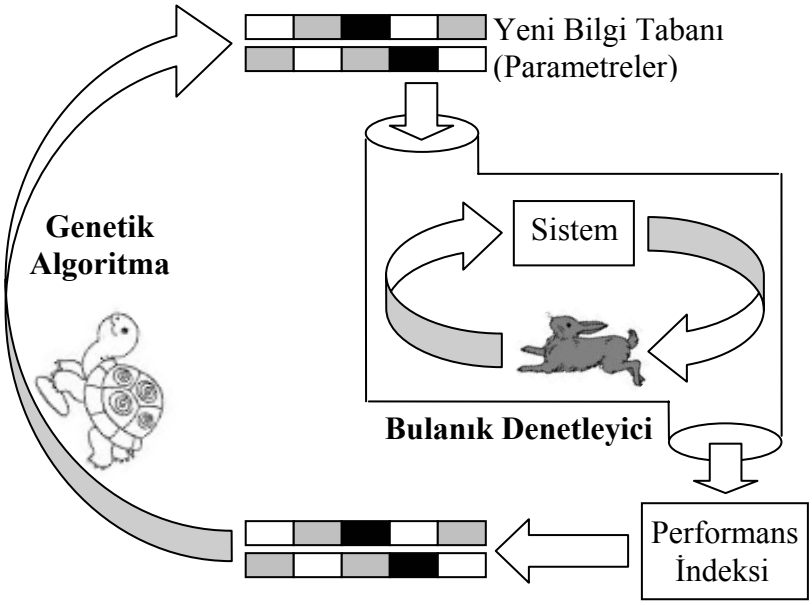
5. GENETİK TABANLI BULANIK DENETLEYİCİ İLE FIRÇASIZ DC MOTORLARIN SENSÖRSÜZ KONTROLÜ

Önceki bölümde gerçekleştirilen geleneksel bulanık denetleyicide, denetleyici parametreleri uzman bilgisi temelinde deneme ve yanılma yoluyla elde edilmiş ve denetleyicinin optimal çözüm sunacağı garanti edilememişti. Burada ise, optimal kontrol teorisine göre sistemin performans indeksi tanımlanmış, bu ifade genetik algoritmada uygunluk fonksiyonu biçimine sokularak optimizasyon amacıyla kullanılmıştır. Optimal bulanık denetleyici için tasarım parametreleri, genetik algoritmalar (GA) ile otomatik olarak elde edilmiştir.

5.1 Genetik Tabanlı Bulanık Denetleyicinin Tasarlanması

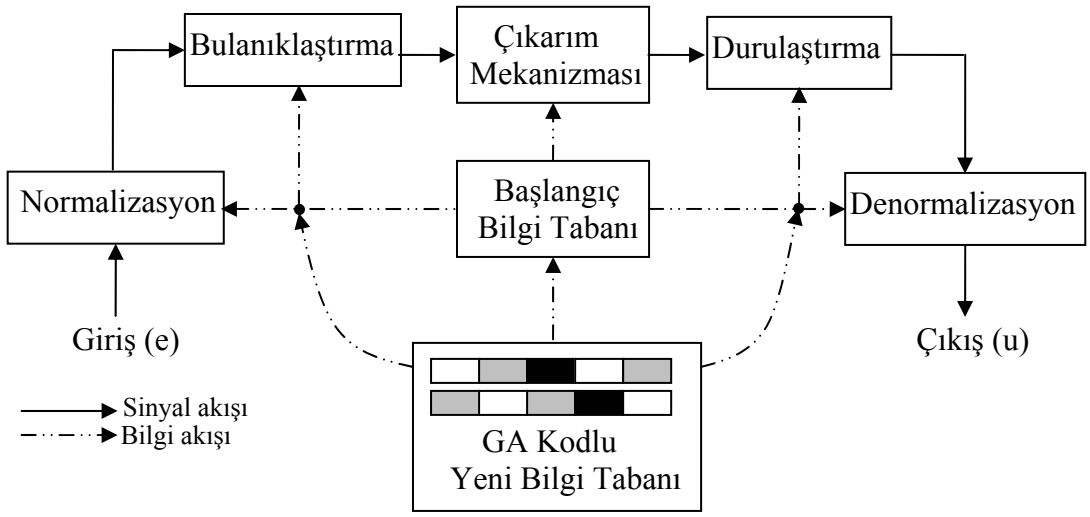
Hızla yaygınlaşan bulanık mantık sistemlerde, veri tabanı ve bulanık kural tabanının oluşturulmasında uzman bilgisine ihtiyaç vardır. Üstelik, mevcut uzman bilgisi, bulanık mantık sistem tasarımında parametrelerin en uygun değerlere ayarlanmasında yetersiz kalmakta ve tasarım işlemi uzun zaman almaktadır. Ancak son zamanlarda, bulanık mantık sistem tasarımında belirlenmesi gereken veri tabanı ve bulanık kural tabanları alanında uzman olan kişilere gerek duyulmadan sayısal bilgiler kullanılarak algoritmalarla otomatik olarak üretilebilmektedir (Qiao ve ark. 2005). Bunlar, öğrenebilen bulanık kurallar için gradyan yaklaşımı, bulanık kuralların oluşturulması ve optimizasyonu için yapay sinir ağlarının kullanılması, bulanık denetleyici parametrelerinin belirlenmesi ve optimizasyonu için genetik algoritmaların kullanılmasıdır (Xia ve ark. 2004, Herrera ve ark. 1998).

Bu çalışmada; genetik algoritmalar ile bulanık denetleyici parametreleri otomatik olarak belirlenmiş ve optimal bir bulanık denetleyici gerçekleştirilmiştir. Şekil 5.1'de, gerçekleştirilen genetik tabanlı bulanık denetleyicinin yapısı görülmektedir. Şekilden görüldüğü gibi; kontrol sistemi iki ana döngüden oluşmaktadır. Birincisi, fırçasız DC motoru kontrol eden bulanık denetleyici döngüsüdür. İkincisi ise, bulanık denetleyiciyi geri planda takip eden ve denetleyici parametrelerini ayarlayan genetik algoritma döngüsüdür.



Şekil 5.1 Genetik tabanlı bulanık denetleyici

Şekil 5.1'de görülen ve fırçasız DC motoru kontrol eden bulanık denetleyici döngüsü, önceki bölümde simüle edilmiş ve pratik olarak gerçekleştirilmiştir. Bu bölümde ise, önceki bölümde yapılan çalışmalara ilave olarak, bulanık denetleyici parametrelerini otomatik olarak ayarlayan bir genetik algoritma, sisteme dahil edilmiştir. Böylece, bulanık denetleyici algoritması yenilenerek, parametreleri otomatik olarak ayarlanabilen bir yapıya dönüştürülmüştür. Gerçekleştirilen genetik tabanlı bulanık denetleyicide; bulanık kurallar, normalizasyon parametreleri ve üyelik fonksiyonları sürekli yenilenerek, optimal bulanık denetleyici elde edilmiştir. Şekil 5.2'de genetik tabanlı bulanık denetleyici yapısı görülmektedir. Şekilde görülen sürekli çizgiler sinyal akışını, kesikli çizgiler ise bilgi akışını göstermektedir. Şekil 5.2'de verilen sistemde, başlangıçta bulanık denetleyici parametreleri isteğe göre GA'da rasgele üretilebildiği gibi, kullanıcı tarafından da belirlenebilmektedir. Bu parametreler kullanılarak sistem çalıştırılmakta ve sistemin ilk performans indeksi hesaplanmaktadır. Performans indeksini hesaplamada kullanılan ifade, genetik algoritmada uygunluk fonksiyonu olarak kullanılmıştır. Optimizasyon için kullanılan genetik algoritma döngüsü, uygunluk fonksiyonu veya iterasyon sayısına göre işlevini tamamladıktan sonra isteğe bağlı olarak devre dışı kalmaktadır. Bu durumda sistem, optimal bulanık denetleyici ile fırçasız DC motorların sensörsüz kontrolüne uygun olarak çalışmasına devam etmektedir.



Şekil 5.2 Bulanık denetleyici parametrelerinin GA ile yenilenmesi

Çalışmada, optimal bulanık denetleyicinin genetik algoritmalar yardımıyla tasarımı için, aşağıdaki adımlar takip edilmiştir.

1. Bulanık denetleyicinin tasarımı,
2. Performans indeksinin tanımlanması,
3. Optimize edilecek denetleyici parametrelerinin belirlenmesi ve optimizasyon yaklaşımları,
4. Denetleyici parametrelerinin genetik algoritmada kodlanması,
5. Denetleyicinin optimize edilmesi,
6. Sonucun değerlendirilmesi ve gerekirse optimizasyon döngüsüne devam edilmesi.

5.1.1 Temel bulanık denetleyici

Çalışma için geliştirilen ve optimizasyon amacıyla kullanılan genetik algoritma programında, genetik algoritmada kodlanan bulanık denetleyici parametreleri, başlangıçta rasgele üretilebildiği gibi tasarımcı kişi tarafından da girilebilmektedir. Bu nedenle, Bölüm 4'de geleneksel olarak deneme ve yanılma yoluyla tasarlanan bulanık denetleyici burada optimize edilecek temel denetleyici olarak alınmıştır. Optimizasyon işleminde başlangıç parametrelerinin girilmesiyle

optimizasyon yüzeyi oldukça sınırlanmakta ve işlem kolaylaşmaktadır. Sistem sınırlı da olsa genetik algoritmaların gerçek zaman uygulamalarına imkan vermektedir. Ayrıca geliştirilen sistem, performans indeksi temelinde çalıştığı için, sistem performansı hakkında fikir oluşabilmektedir.

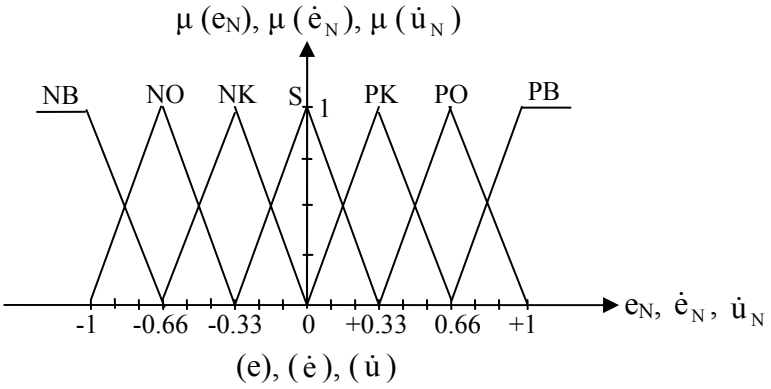
Temel bulanık denetleyici olarak, önceki bölümde tasarlanan bulanık denetleyici alınabilir. Buna göre bulanık denetleyici, 3 adet üyelik fonksiyonuna, 7x7 kural tablosuna, 3 adet normalizasyon parametrelerine, minimum-maksimum çıkarım mekanizmasına ve ağırlık merkezi durulaştırma yöntemine sahiptir. Daha önce tanımlanan dilsel değişkenler, bu bölüm içinde geçerlidir. Buna göre, normalizasyon parametreleri Tablo 5.1'de, 7x7 boyutlu kural tablosu Tablo 5.2'de ve giriş ve çıkış için üyelik fonksiyonları Şekil 5.3'de verilmiştir.

Tablo 5.1 Normalizasyon parametreleri

N_e	10
$N_{\dot{e}}$	1
N_u	2.2

Tablo 5.2 Başlangıç kural tablosu

$\dot{e}_N$ \ e_N	NB	NO	NK	S	PK	PO	PB
NB	NB	NB	NO	NO	NK	NK	S
NO	NB	NO	NO	NK	NK	S	PK
NK	NO	NO	NK	NK	S	PK	PK
S	NO	NK	NK	S	PK	PK	PO
PK	NK	NK	S	PK	PK	PO	PO
PO	NK	S	PK	PK	PO	PO	PB
PB	S	PK	PK	PO	PO	PB	PB



Şekil 5.3 Üyelik fonksiyonları

5.1.2 Performans indeksinin tanımlanması

Optimal kontrol teorisine göre, sistemi değerlendiren bir performans indeksi tanımlanmalıdır. Bu nedenle burada, sistemin performansını belirleyen bir performans indeksi tanımlanmıştır. Performans indeksi, ideal sistem performansı ile ölçülen gerçek sistem performansı değerlerinden elde edilen nicel bir değerdir. Optimizasyon işleminde bu değerin, belirlenen amacı sağlaması istenir. Bu nedenle, tüm optimizasyon problemlerinde bir amaç fonksiyonunun tanımlanması gerekir. Kontrol sistemi için tanımlanan performans indeksi, optimizasyon işlemlerinde amaç fonksiyonu olarak kullanılabilir. Ancak, kontrol sistemi için tanımlanan performans indeksinin sıfıra yakınsaması istenirse, optimizasyon işlemi için tanımlanan amaç fonksiyonunun da sıfıra yakınsaması gerekir. Kontrol sistemlerinde bazı durumlarda, birden fazla performans indeksinin sağlanması istenebilir. Bu durumda, optimizasyon işlemi için çoklu amaç fonksiyonu tanımlanmalıdır. Bunun için tanımlanan çoklu amaç fonksiyonu, hedeflenen birden fazla performans indeksinin bir lineer kombinasyonu olabilir.

Motor sürücü uygulamaları için performans indeksi tanımlanırken, sıfıra yakın sürekli durum hatası, küçük bir aşma ve hızlı geçici durum cevabına sahip olan birim basamak cevabı göz önüne alınmalıdır. Bu çalışmada, tüm bu özellikleri aynı anda sağlamak için, klasik kontrolde tanımlanan birden fazla performans indeksinin yerine, optimizasyon işleminde amaç fonksiyonu olarak da kullanılmak üzere, bir çoklu amaç fonksiyonu tanımlanmıştır (Ashrafzadeh 1996, Tang ve ark. 2001).

Çalışmada; bulanık denetleyiciye sahip fırçasız DC motor sürücü sisteminin hızlı geçici durum cevabını ölçmek için, aşağıdaki denklem kullanılmıştır.

$$J_1 = w_t \int_0^{t_1} |e| \, dt \tag{5.1}$$

Burada, J_1; geçici durum performans indeksi, w_t; çoklu amaç fonksiyonu içindeki ağırlık katsayısı ve e; hata değeridir. Sistemin sürekli durum hatasını ölçmek için denklem (5.2) kullanılmıştır.

$$J_2 = w_{ss} \int_{t_1}^{t} |e| \, t \, dt \tag{5.2}$$

Bu ifadede, J_2; sürekli durum performans indeksi, w_{ss}; çoklu amaç fonksiyonu içindeki ağırlık katsayısıdır. Sistemde, cevabın maksimum aşması ise aşağıdaki gibi ölçülmüştür.

$$J_3 = w_p \int_0^{t} \delta\left(\frac{d\omega_r}{dt}\right) |e| \, dt \tag{5.3}$$

Burada; J_3; maksimum aşma indeksi, w_p; aşma indeksinin çoklu amaç fonksiyonu içindeki ağırlık katsayısıdır. Ayrıca;

$$\int_{0^-}^{0^+} \delta\left(\frac{d\omega_r}{dt}\right) = \begin{cases} 1 & : \ d\omega_r/dt = 0 \\ 0 & : \ d\omega_r/dt \neq 0 \end{cases} \tag{5.4}$$

olarak verilmiştir. Burada, $\delta\left(\frac{d\omega_r}{dt}\right)$ ifadesi oluşan aşma noktasını ifade etmektedir. Çoklu amaç fonksiyonunu elde etmek için, yukarıda verilen ifadelerden bileşke bir performans indeksi tanımlanmıştır.

$$J_{in} = W \, J_i \tag{5.5}$$

Burada; denklemde verilen $J_i = [J_1 \ J_2 \ J_3]^T$ ifadesi, çoklu amaç fonksiyonunun performans vektörünü ve $W=[w_t \ w_{ss} \ w_p]$ ifadesi de, her bir kriterin ağırlık katsayısını göstermektedir. Bir sürücü sisteminin tasarımında w_t, w_{ss}, w_p ile verilen ağırlık

katsayılarının belirlenmesi, oldukça önemlidir. Literatürde, arzu edilen sistem cevap eğrilerinden ağırlık katsayıları aşağıdaki gibi verilmektedir (Ashrafzadeh, 1996).

$$w_t = 1 \quad w_{ss} = 1 \quad w_p = 6 \tag{5.6}$$

Denklem (5.1)-(5.5)'den, optimizasyon amacıyla kullanılan çoklu amaç fonksiyonu aşağıdaki gibi elde edilmiştir.

$$J_{in} = \int_0^{t_1} |e|\,dt + \int_{t_1}^{t} |e|\,t\,dt + 6\int_0^{t} \delta\left(\frac{d\omega_r}{dt}\right)|e|\,dt \tag{5.7}$$

Denklem (5.7) ile elde edilen çoklu amaç fonksiyonu, bulanık denetleyicinin optimizasyon işleminde amaç fonksiyonu olarak kullanılmıştır.

5.1.3 Optimize edilecek denetleyici parametreleri ve optimizasyon yaklaşımları

Yapılan çalışmada, fırçasız DC motorların sensörsüz kontrolü için iki girişli tek çıkışlı, 7x7 boyutlu kural tablosuna sahip, 3 adet üçgen üyelik fonksiyonu bulunan, minumum-maksimum çıkarım mekanizması ile çalışan ve durulaştırma biriminde ağırlık merkezi yöntemini kullanan bulanık denetleyici kullanılmıştır. Burada; ağırlık merkezi durulaştırma yöntemi, min-max çıkarım mekanizması ve üçgen üyelik fonksiyonu formu sabit kabul edilerek, geriye kalan diğer parametrelerin optimizasyonu yapılmıştır. Durum uzayında dönüşüm bölgelerini birbirinden tamamen ayırmak için, asimetrik üyelik fonksiyonları seçilmelidir (Arslan ve Kaya 2001). İki girişli tek çıkışlı bir denetleyici için, her bir değişkenin 7 üyelik fonksiyonu ile temsil edildiği kabul edilir ve her bir üçgen üyelik fonksiyonu üç parametre ile gösterilirse, üyelik fonksiyonlarını tanımlamak için toplam 63 parametre gerekir. Ayrıca 3 adet normalizasyon parametresi ve 7x7=49 adet kural tablosu hücresi belirlenmelidir. Bu şartlar altında optimize edilecek toplam parametre sayısı; 63+49+3=115 olmaktadır. Görüldüğü gibi arama yüzeyinde parametre sayısı çok fazladır (Ashrafzadeh 1996).

Bulanık denetleyici tasarımında kural tablosu ve üyelik fonksiyonlarının simetrik yapıları göze çarpmaktadır. Bu nedenle, gerçekleştirilen çalışmada, optimize

edilecek olan parametre sayısı uygun biçimde azaltılarak işlem kolaylaştırılmıştır. Optimizasyon işleminde bulanık denetleyici parametrelerin azaltılması, çok sayıdaki parametrenin optimizasyonuna göre kötü sonuçlar vermemektedir (Yun ve ark. 2003).

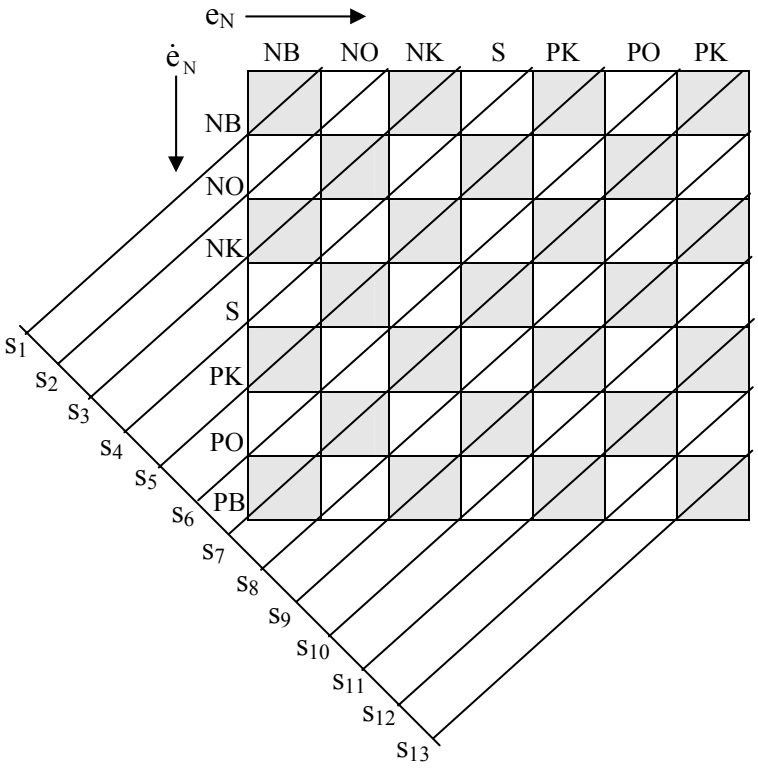
Bulanık denetleyicide, 3 adet normalizasyon parametresi ($N_e, N_{\dot{e}}, N_u$) bulunmaktadır. Bu parametrelerin optimizasyonu, denetleyici açısından oldukça önemlidir. Çünkü; bu parametreler PI tip denetleyicilerde kullanılan K_P ve K_I katsayılarına karşılık gelmektedir (Tan ve Hu 1996). GA ile normalizasyon parametrelerinin optimizasyonu için, [0-1000] değer aralığında bir arama yapılmıştır.

Kural tablosunun otomatik olarak oluşturulması veya yenilenmesi işleminde, kural tablosunu oluşturan 7x7=49 adet hücrenin belirlenmesi gerekmektedir. Yapılan çalışmalar incelendiği zaman, bulanık denetleyicilerin kural tablolarında bir simetri olduğu görülmektedir. Kolay anlaşılması açısından; NB= -3, NO= -2, NK= -1, S=0, PK=1, PO=2, PB=3 olarak numaralandırılırsa, Tablo 5.3'de verildiği gibi doğruların kestiği hücreler aynı ifadelere sahip olmaktadır. Bu nedenle, Tablo 5.3'de verilen s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13} parametrelerinin bulunması ile kural tablosu oluşturulmuş olmaktadır. Böylece kural tablosu parametresi 13'e indirilmiştir. Ayrıca, kural tablosundaki pozitif bölge ile negatif bölgenin simetrik olduğunu düşünülürse; $s_1 = -s_{13}$, $s_2 = -s_{12}$, $s_3 = -s_{11}$, $s_4 = -s_{10}$, $s_5 = -s_9$, $s_6 = -s_8$ olduğunu kabul edilebilir. Bu son adımla, parametre sayısı 7'ye kadar (s_1, s_2, s_3, s_4, s_5, s_6, s_7) indirilmiştir. GA'da bu parametreler {-3, -2, -1, 0, 1, 2, 3} sayı kümesinde arama yapılarak elde edilmiştir. Örneğin; arama sonucunda $s_2 = -2$ bulunursa, $s_{12} = -s_2 = 2$ olacaktır. Bu da; s_2 hücreleri NO, s_{12} hücreleri PO olacak demektir.

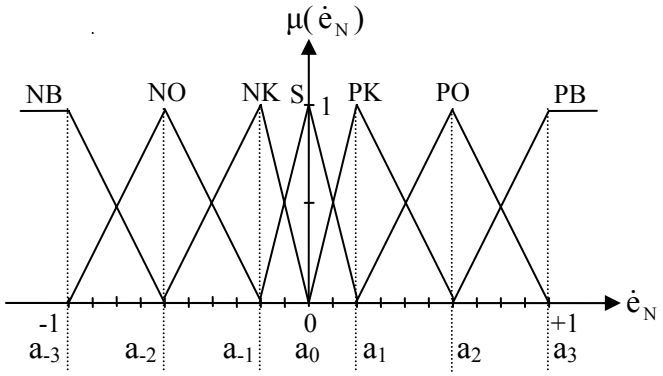
Gerçekleştirilen sistemde, üçgen üyelik fonksiyonları kullanılmıştır. Üçgen üyelik fonksiyonlarında belirlenmesi gereken en az üç parametre mevcuttur. Bunlar; üçgenin tepe noktası ve her bir kolun konumudur. Çalışmada Şekil 5.4'de görüldüğü gibi üçgenlerin %50 oranında örtüştüğü ve bu üçgenlerin tepe noktalarının aynı zamanda komşu üçgenlerin kollarının pozisyonlarına karşılık geldiği kabul edilmiştir. Bu durumda şekilden de görüldüğü gibi, 7 elemanlı bir üyelik fonksiyonunda, a_1, a_2, a_3, a_0, a_{-1}, a_{-2}, a_{-3} parametrelerinin bulunması gerekmektedir.

Şekilde görüldüğü gibi $a_3 = +1$, $a_{-3} = -1$ ve $a_0 = 0$ olmaktadır. Böylece bir üyelik fonksiyonunda bulunması gereken 4 parametre kalmaktadır. Eğer üyelik fonksiyonunun pozitif ve negatif kısımları simetrik ise; $a_1 = -a_{-1}$ ve $a_2 = -a_{-2}$ yazmak mümkündür. Böylece bir üyelik fonksiyonunda bulunması gereken parametre sayısı iki olmaktadır. Gerçekleştirilen sistemde üç üyelik fonksiyonu olduğu için bulunması gereken parametre sayısı toplam 6 tane olmaktadır. GA'da üyelik fonksiyonları için parametreler [0, 1] aralığında aranmıştır.

Tablo 5.3 Kayan tip kural tablosunun oluşturulması



Yukarıda irdelenen denetleyici parametreleri, GA'da kodlanarak optimum değerlerde elde edilmiştir. Görüldüğü gibi 115 parametreli sistem, yapılan kabuller doğrultusunda 3+7+6=16 parametreye kadar indirgenmiştir. Eğer daha esnek bir yapı arzulanırsa, parametre sayısı artmakta ve optimizasyon işlemi zorlaşmaktadır. Burada, yukarıda elde edilen 16 parametrenin optimizasyonu GA ile gerçekleştirilmiştir.



Şekil 5.4 Üçgen üyelik fonksiyonu parametreleri

GA ile bulanık denetleyicinin optimal tasarımında parametrelerin elde edilmesi için iki farklı yaklaşım ele alınmıştır. İlk yaklaşımda; optimal bulanık denetleyici için parametreler ardışık olarak belirlenmiştir. Bu yaklaşım, adaptif kontrol sistemlerinde uygulanan öz ayarlamalı sistemlere benzemektedir. İkinci yaklaşım ise; normalizasyon parametreleri, üyelik fonksiyonları ve kuralların eş zamanlı olarak optimize edildiği yöntemdir. Bulanık denetleyici parametreleri birbirinden tamamen bağımsız değildir (Wu and Liu 2000). Bu nedenle, optimizasyon işleminde, tüm parametrelerin aynı anda ayarlanması, en doğru çözümdür. Ancak yukarıda da belirtildiği gibi, bulanık denetleyici parametrelerinin çok olması, ikinci yaklaşımı oldukça zorlaştırmaktadır.

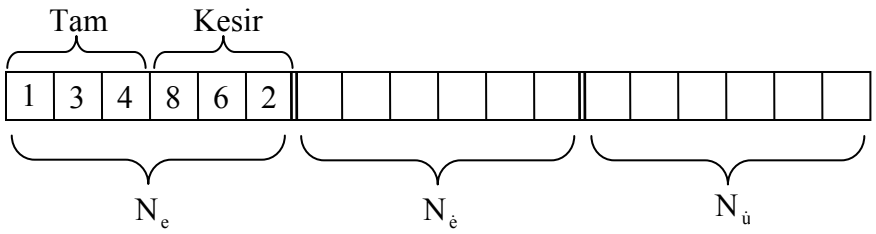
5.1.4 Denetleyici parametrelerinin genetik algoritmada kodlanması

Optimal bulanık denetleyicinin tasarımı için denetleyici parametreleri, GA'da arzu edilen çözüme göre kodlanmalıdır. Gerçekleştirilen çalışmada, daha önceden de belirtildiği gibi gerçek kodlu GA kullanılmıştır. Burada gerçek kodlu GA'da problemin kodlanması işlemi adım adım verilmektedir. Verilen adımlarda, üç farklı parametre ele alınmıştır. Çalışmada, optimizasyon işleminde kullanılan yaklaşıma göre, aşağıda tanımlanan kodlamalar kullanılmıştır.

Adım 1: Normalizasyon ve denormalizasyon parametrelerinin kodlanması

N_e, $N_{\dot{e}}$ ve N_u olarak gösterilen normalizasyon parametreleri, bulanık denetleyicilerde oldukça önemlidir. Çünkü karar tablosu ve üyelik fonksiyonları

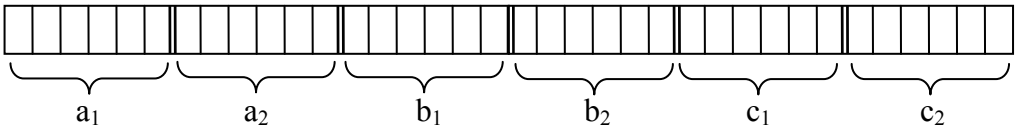
üzerinde büyük bir etkiye sahiptir. Aslında bu parametreler, giriş ve çıkış üyelik fonksiyonlarını, giriş-çıkış uzayında benzer biçimde değiştirir. Normalizasyon ve denormalizasyon işlemleri, doğrudan bulanık denetleyicilerin kararlılığına etki ederler. Gerçek kodlu GA'da, normalizasyon parametrelerinin kodlama biçimi Şekil 5.5'de verilmiştir. Burada, her bir normalizasyon parametresi için 3'ü tam 3'ü de virgülden sonra olmak üzere 6 basamak kullanılmıştır. Şekilden de görüldüğü gibi 3 parametre için 18 basamak kullanılmıştır. Ayrıca GA ile arama işleminde her parametre için değer aralığı tanımlanmıştır. Çalışmada normalizasyon parametrelerinin değer aralığı [0-1000] aralığıdır. Buna göre örneğin N_e=134,862 gibi bir değer olmaktadır.



Şekil 5.5 Normalizasyon parametrelerinin GA'da kodlanması

Adım 2: Üyelik fonksiyonlarının kodlanması

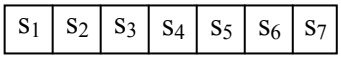
Normalizasyon ve denormalizasyon parametreleri ile karşılaştırıldığında üyelik fonksiyonları, denetleyici kazancının yerel olarak ayarlanmasından sorumludur. Durum uzayı açısından bakıldığında ise üyelik fonksiyonları, tanımlanan karakteristik noktalar çerçevesinde, durum uzayını bölmek yoluyla tanımlanırlar. Bu yüzden üyelik fonksiyonları, karakteristik noktalar kadar önemlidir. Giriş ve çıkış değişkenleri için karar tabloları benzer olmasına rağmen, farklı dönüşüm işlemleri sonucunda, her bir giriş ve çıkış için farklı üyelik fonksiyonu elde edilmektedir. Bu çalışmada yukarıda verildiği gibi, 3 üyelik fonksiyonu için toplam 6 parametre optimize edilmelidir. Bu parametreler a_1, a_2, b_1, b_2, c_1, c_2 ile ifade edilmiştir. Şekil 5.6'da bu 6 parametrenin gerçek kodlu GA'da kodlanması görülmektedir. Her bir parametre için kullanılan 6 basamaklı sayı ile, [-1, 1] çözüm aralığında yaklaşık 0.00001 çözünürlük elde edilmiştir.



Şekil 5.6 Üyelik fonksiyonu parametrelerinin GA'da kodlanması

Adım 3: Kural tablosunun kodlanması

Bulanık denetleyici kuralları, bulanık dönüşüm fonksiyonunun genel görünümü için önemlidir. Gerçekte durum uzayında kural tablosu, durum eğrisini kontrol eden, lineer olmayan bir dönüşümdür. 7x7 boyutlu kural tablosunda toplam 49 adet belirlenmesi gereken parametre olmasına rağmen, yukarıda belirtildiği gibi, kayan tip bir tablo oluşturularak parametre sayısı 7'ye kadar indirgenmiştir. Bunlar; s_1, s_2, s_3, s_4, s_5, s_6, s_7 olarak gösterilmiştir. Her bir parametrenin alabileceği değer kümesi {-3,-2,-1,0,1,2,3} olduğu için, gerçek kodlu GA'da tek haneli rakamla ifade edilmiştir. Karar tablosunun GA'da kodlanması Şekil 5.7'de görülmektedir.



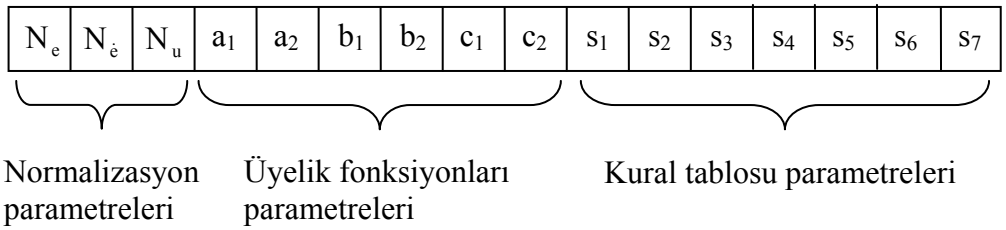
Şekil 5.7 Karar tablosunun GA'da kodlanması

Adım 4: Tüm denetleyicinin kodlanması

Bir bulanık denetleyicinin tasarımı, genellikle denetleyici kısımlarının sırayla ayarlanmasıyla gerçekleştirilir. Bu ardışık tasarım işleminde ilk olarak, üyelik fonksiyonları ve karar tablosunun sabit olduğu kabul edilerek herhangi bir optimizasyon tekniği ile normalizasyon parametreleri bulunur ve bu parametreler sabitlenir. Daha sonra kural tablosu optimizasyon işlemine tabi tutulur. Son olarak, elde edilen normalizasyon parametreleri ve yeni kural tablosu kullanılarak optimizasyon tekniği ile yeni üyelik fonksiyonları bulunur.

Bulanık denetleyici kısımları, birbirinden tamamen bağımsız parçalar değildir. Yani, bir parametrenin değiştirilmesi, diğer parametreleri etkilemektedir. Bu yüzden, ardışık tasarım işlemi ile gerçekleştirilen çözüm gerçek optimal çözüm olmayabilir. Bundan anlaşıldığı gibi, denetleyici parametrelerinin farklı kümeleri arasında bir etkileşim vardır. Örneğin, normalizasyon parametrelerinin değiştirilmesi

ile, tüm üyelik fonksiyonları değişmektedir. Böylece, aynı giriş değeri için farklı üyelik fonksiyonları elde edilmektedir. Elbette bu, denetleyici çıkışı için yeni bir değer demektir. Hatta bu gösteriyor ki, aslında bir bulanık denetleyicinin tüm tasarım parametrelerinin etkileşimi denetleyici başarısı için önemlidir ve bu nedenle prensipte parametrelerin optimal ayarı ve denetleyici parametrelerinin optimizasyon işlemi en iyi şekilde aynı zamanda yapılmalıdır. Bunu gerçekleştirmek için, bulanık denetleyicinin tüm tasarım parametreleri GA'da yeniden kodlanmalıdır. Bunun için en basit yol, yukarıda ayrı ayrı verilen kodların birleştirilmesidir. Şekil 5.8'de bir bulanık denetleyicinin tüm bilgilerini içeren kodlama biçimi görülmektedir.



Şekil 5.8 Tüm denetleyicinin GA'da kodlanması

Adım 5: Amaç fonksiyonunun kodlanması

Genetik algoritmalarda, her bir birey, problemin olası bir çözümünü gösterir ve bireylerin yani, çözüm sonuçlarının değerlendirilmesi için özel amaç fonksiyonu gereklidir. Böylece, her bir kromozom veya çözüm için, çözüm kalitesini gösteren bir sayısal değer hesaplanmaktadır. Optimizasyon işleminde, kapalı çevrim sistemin performans indeksi, GA'da amaç fonksiyonu olarak kullanılabilmektedir. Denklem (5.7) ile verilen performans indeksi, çalışmada GA için minimize edilecek çoklu amaç fonksiyonu olarak kullanılmıştır. Denklemden de görüldüğü gibi; geçici durum, sürekli durum ve maksimum aşma durumunda oluşan hata (e), amaç fonksiyonu için temel değişken olarak kullanılmıştır.

5.2 Genetik Tabanlı Bulanık Denetleyici ile Fırçasız DC Motorların Sensörsüz Kontrolü

Burada, genetik tabanlı bulanık denetleyici ile özellikleri Tablo 4.2'de verilen fırçasız DC motorun sensörsüz kontrolü, simüle edilmiş ve pratik olarak gerçekleştirilmiştir. Önce simülasyon çalışmaları yapılarak, ardışık optimizasyon

yaklaşımı ve eş zamanlı optimizasyon yaklaşımlarında problem çözülmüştür. Her bir yaklaşımda elde edilen optimal parametreler, uygulama devresinde kullanılarak sistem davranışı incelenmiştir. Aşağıda her iki yaklaşım için ayrı ayrı sonuçlar verilmiştir. Uygulama devresi olarak, Bölüm 4'de kullanılan ve Şekil 4.23 ile verilen ADSP-21992 tabanlı sistem kullanılmıştır.

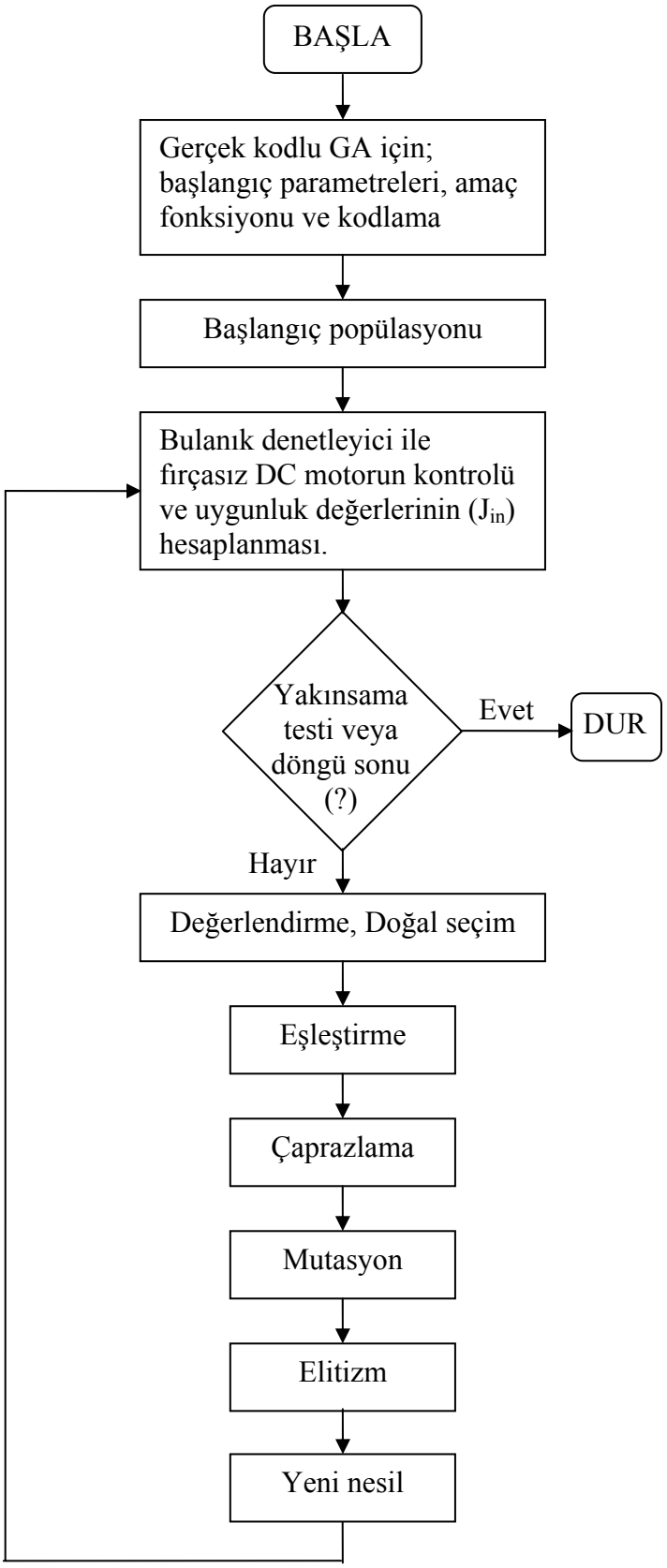
5.2.1 Gerçek kodlu GA algoritması

Bulanık denetleyici optimizasyonunda, optimize edilmesi gereken parametre sayısı fazla olduğu için gerçek kodlu GA'lar daha uygun olur. Ayrıca gerçek kodlu GA'da, parametrelerin hassasiyeti daha fazla olmakta ve hafızada daha az yer kaplamaktadır. Yazılan gerçek kodlu GA algoritması, oldukça esnek bir yapıya sahiptir. Şekil 5.9'da, bu çalışmada yürütülen ve bulanık denetleyiciyi optimize eden gerçek kodlu GA algoritmasının akış şeması görülmektedir. Algoritmada, gerçek kodlu GA için başlangıç populasyonu, kullanıcı tarafından verilebildiği gibi, tercihe göre rasgelede üretilebilmektedir. Başlangıç parametrelerinin belirlenmesi, GA ile gerçekleştirilen ardışık optimizasyon işleminde ve kabaca tasarlanan sistemlerin optimal çözümlerinde kolaylık sağlamakta, optimal çözümlere ve gerçek zaman uygulamalarına imkan vermektedir. Ayrıca yapılan programda, çözüm parametreleri sınırlandırılarak, belirli bir aralıkta hızlı arama yapılabilmekte ve algoritmada her parametrenin duyarlılığı ayarlanabilmektedir. Bu özellik sayesinde, parametre tiplerine göre kodlama yapılabilmektedir. Böylece bazı parametreler tam sayı, bazıları ise kesirli sayı olarak kodlanabilmektedir. Çalışmada kullanılan gerçek kodlu GA'da; mutasyon olasılığı, çaprazlama olasılığı ve populasyon sayısı değiştirilebilmektedir. Ayrıca algoritmada, elitizm özelliği ve optimizasyon işleminin sonlandırılması için nesil sayısı ve uygunluk fonksiyonu sınırı tanımlanabilmektedir.

Gerçekleştirilen algoritmada, başlangıç populasyonunun oluşturulmasında;

$$N_{ipop}=(P_H-P_L)\ x\ Rasgele\{P_{ipop}\}+P_L \qquad (5.8)$$

ifadesi kullanılmıştır. Burada, P_H; parametrelerin üst sınır değeri, P_L; parametrelerin alt sınır değeri, Rasgele $\{P_{ipop}\}$; ise ±0.5 arasında rasgele üretilen bir sayıdır.



Şekil 5.9 Gerçek kodlu GA tabanlı bulanık denetleyici akış şeması

Şekil 5.9 ile verilen akış şemasından görüldüğü gibi, GA'da uygunluk değerleri, populasyon içerisindeki her bir birey için bulanık denetleyici ile fırçasız DC motorların kontrolü bir defa yürütülerek performans indeksinden elde edilmiştir.

Gerçek kodlu GA'da uygunluk fonksiyonu olarak denklem (5.7)'de verilen performans indeksi kullanılmış ve genetik algoritma ile performans indeksi minimize edilmeye çalışılmıştır. Çalışmada optimizasyon amacıyla yazılan gerçek kodlu GA programının tüm özellikleri Tablo 5.4'de toplu olarak verilmiştir.

Tablo 5.4 Gerçek kodlu GA'nın özellikleri

Parametre sayısı	Ayarlanabilir
Parametre değer aralığı	Ayarlanabilir
Parametre duyarlılığı	Ayarlanabilir
Başlangıç populasyonu	İsteğe göre (0-Rasgele, 1-Özel) üretilir
Mutasyon olasılığı	Ayarlanabilir (Tipik olarak < 0.1)
Çaprazlama olasılığı	Ayarlanabilir (Tipik olarak 1'e yakın)
Populasyon sayısı	Ayarlanabilir (20)
Elitizm	Seçilebilir (1-Evet, 0-Hayır)
Durdurma kriteri	Tanımlanabilir
Nesil sayısı	Ayarlanabilir
Doğal seçim yöntemi	Rulet Tekerleği
Başlangıç populasyonu	$N_{ipop} = (P_H - P_L)$ x Rasgele$\{P_{ipop}\} + P_L$
Uygunluk (Amaç) fonksiyonu	$J_{in} = \int_0^{t_1} \lvert e \rvert dt + \int_{t_1}^{t} \lvert e \rvert t\, dt + 6 \int_0^{t} \delta\left(\frac{d\omega_r}{dt}\right) \lvert e \rvert dt$
Eşleştirme	Rulet tekeri ağırlık yöntemi
Çaprazlama	Karıştırma yöntemi

Gerçek kodlu GA için doğal seçim işlemi, Rulet tekeri yöntemiyle yapılmıştır. Buna göre, her iterasyonda kromozomların toplam uygunluk değeri hesaplanır. Her bir kromozomun uygunluk değeri, toplam uygunluk değerine bölünerek, her kromozom için 0-1 arasında değişen seçim ihtimalleri bulunur. Daha sonra kümülatif ihtimaller hesaplanır. Populasyon sayısı kadar 0-1 arasında "rasgele"

sayılar üretilir. Üretilen rasgele sayılar, birinci kromozomun kümülatif seçim ihtimalinden küçük ise birinci kromozom seçilir. Büyük ise, ikinci kromozom veya diğerlerinin kümülatif ihtimalleriyle karşılaştırılarak, hangisinden büyükse o kromozom seçilir. Böylece, Rulet tekeri seçim metodu gerçekleştirilmiş olur. Eşleştirme işleminde Rulet tekeri ağırlık yöntemi kullanılmıştır. Çaprazlama işlemi ise karıştırma yöntemi ile yapılmıştır. Çaprazlanan bireyler mutasyon ihtimaline göre mutasyona uğratılıp, yeni nesil elde edilmektedir. Ayrıca en iyi bireyin korunması için algoritmaya, elitizm özelliği eklenmiştir.

5.2.2 Simülasyon Sonuçları

5.2.2.1 Ardışık optimizasyon yaklaşımı simülasyon sonuçları

Ardışık optimizasyon yaklaşımında ilk olarak, normalizasyon parametreleri optimize edilmiştir. Daha sonra bu normalizasyon parametreleri kullanılarak kural tablosu yenilenmiştir. Son olarak yenilenen kural tablosu ve optimal normalizasyon parametreleri kullanılarak üyelik fonksiyonları optimize edilmiştir. Ardışık optimizasyon yaklaşımında kullanılan gerçek kodlu GA parametreleri Tablo 5.5'de verilmiştir.

Tablo 5.5 Gerçek kodlu GA parametreleri

Parametre sayısı	3 (N_e , N_u , $N_{\dot{e}}$)
Parametre değer aralığı	[0 0 0] - [1000 1000 1000]
Parametre duyarlılığı	0.0001
Başlangıç populasyonu	0-Rasgele
Mutasyon olasılığı	0.05
Çaprazlama olasılığı	0.8
Populasyon sayısı	20
Elitizm	1-Evet
Durdurma kriteri	J_{in}<0.1
Nesil sayısı	100

Adım 1: Normalizasyon parametrelerinin optimizasyonu

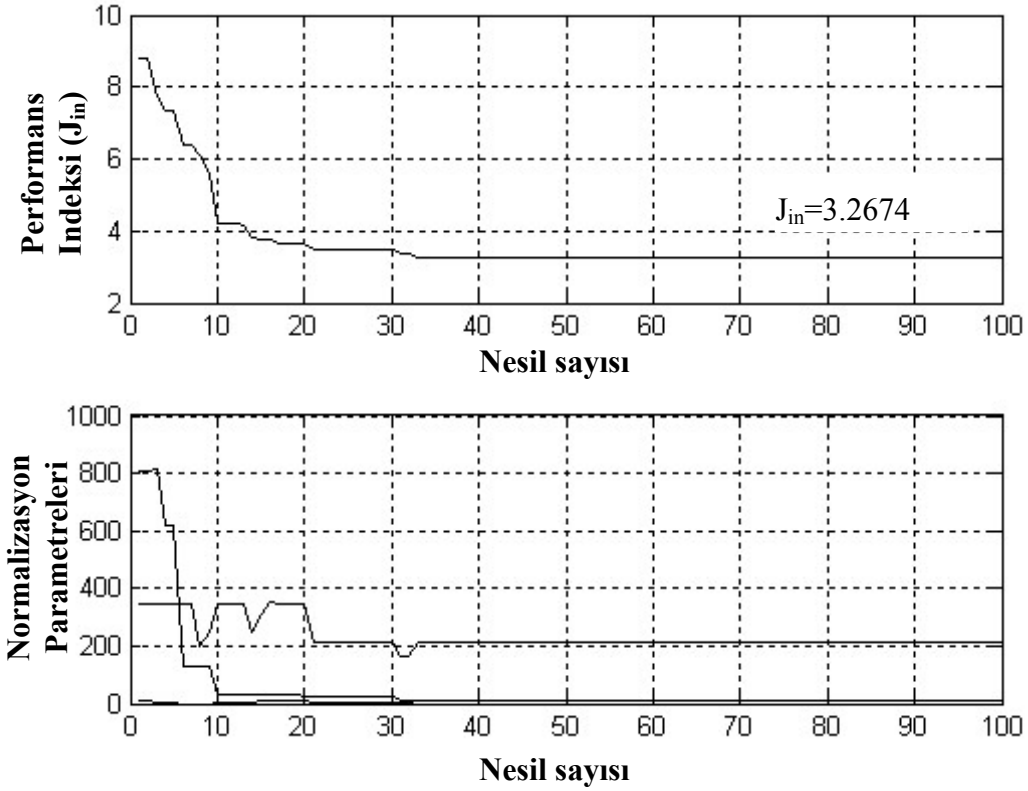
Bu adımda, genetik algoritmada Denklem (5.7)'de verilen performans indeksi kullanılarak en iyi normalizasyon parametreleri bulunmaya çalışılmıştır. Başlangıçta, bulanık denetleyici yapısında Tablo 5.2 ile verilen standart kural tablosu ve Şekil 5.3 ile verilen başlangıç üyelik fonksiyonları kullanılmıştır. GA ile elde edilen optimum normalizasyon parametreleri ve yakınsanan performans indeksi değeri, Tablo 5.6'da listelenmiştir. Şekil 5.10'da, optimizasyon işlemiyle elde edilen performans indeksi ve parametrelerin optimal değerlere yerleşme eğrileri görülmektedir. Şekilden görüldüğü gibi, optimizasyon işlemi sadece 33 nesil sonra yakınsanmaktadır. Tasarım parametre sayısı düşük olduğundan, arama uzayının tüm noktalarını tarayabilmek amacıyla mutasyon oranı yüksek seçilmiştir.

Tablo 5.6 Ardışık GA ile elde edilen optimal normalizasyon parametreleri

N_e	214.116
$N_{\dot{e}}$	9.965
N_u	11.048
J_{in} (Performans indeksi)	3.2674

Adım 2: Kural tablosunun optimizasyonu

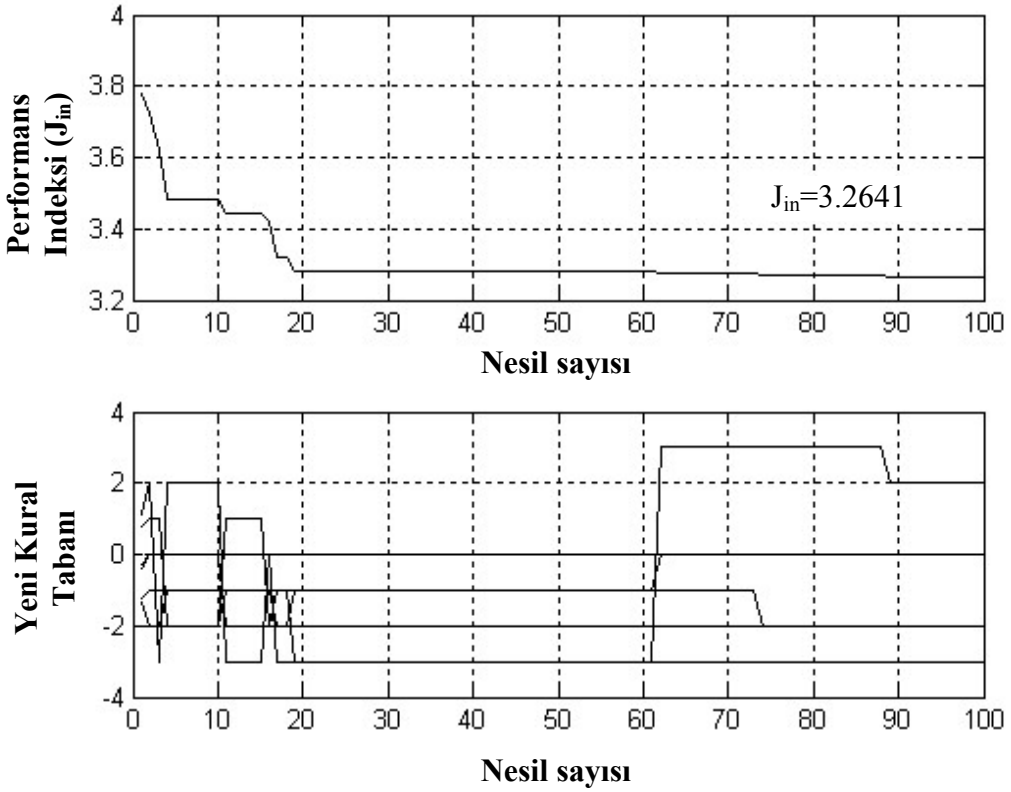
İkinci adımda ise, yukarıda elde edilen optimal normalizasyon parametreleri ve Şekil 5.3 ile verilen başlangıç üyelik fonksiyonları kullanılarak, GA ile bulanık denetleyicinin kural tablosu yeniden elde edilmiştir. Tablo 5.7'de, yaklaşık 88 nesil sonra elde edilen yeni bulanık mantık kural tablosu görülmektedir. Tabloda görüldüğü gibi, bulanık kurallar tamamen yenilenmiş ve geleneksel bulanık denetleyicide öngörülen yaklaşımın yeterli olmadığı anlaşılmıştır. Şekil 5.11'de, GA ile yeni kural tablosu elde edilirken oluşan sistem performans indeksi değişimi ve parametre yerleşme eğrileri verilmiştir. Sistemin yeni performans indeksi J_{in}=3.2641 olarak bulunmuştur. Önceki adımda elde edilen performans indeksine göre daha küçük ve sistem daha iyidir.



Şekil 5.10 Performans indeksi ve Normalizasyon parametrelerinin yerleşme eğrileri

Tablo 5.7 Ardışık GA ile elde edilen kural tablosu

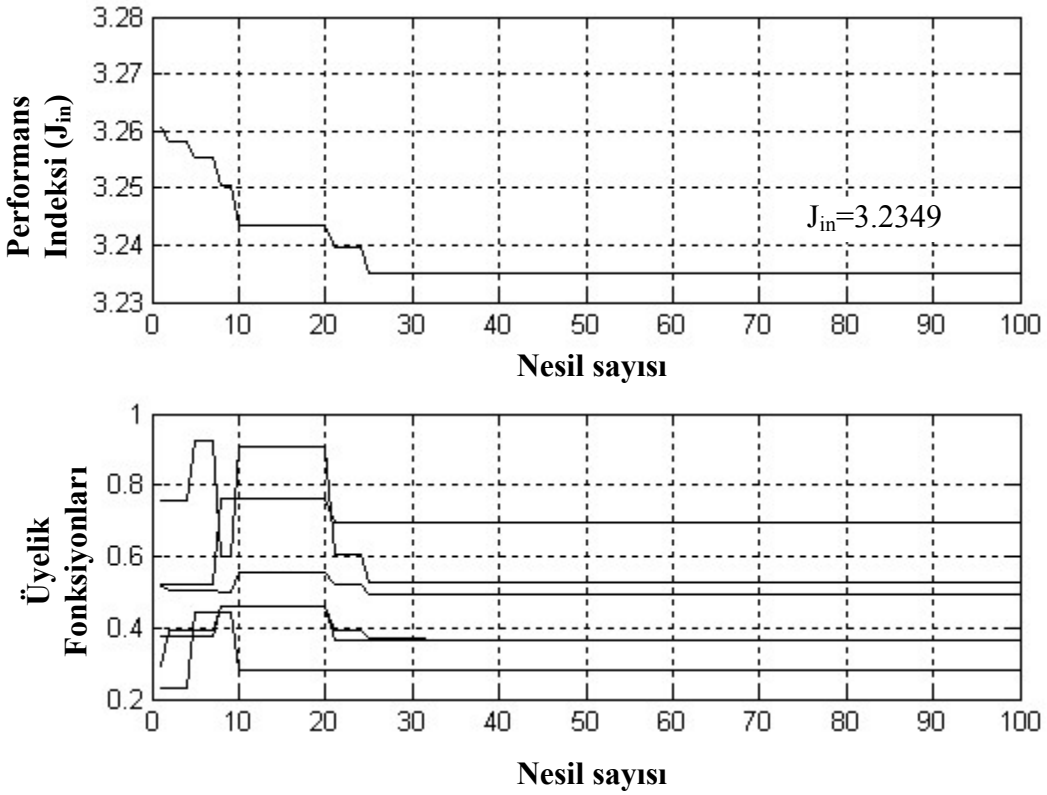
$\dot{e}_N$ \ e_N	NB	NO	NK	S	PK	PO	PB
NB	NO	PO	NB	NO	S	NO	S
NO	PO	NB	NO	S	NO	S	PO
NK	NB	NO	S	NO	S	PO	S
S	NO	S	NO	S	PO	S	PO
PK	S	NO	S	PO	S	PO	PB
PO	NO	S	PO	S	PO	PB	NO
PB	S	PO	S	PO	PB	NO	PO



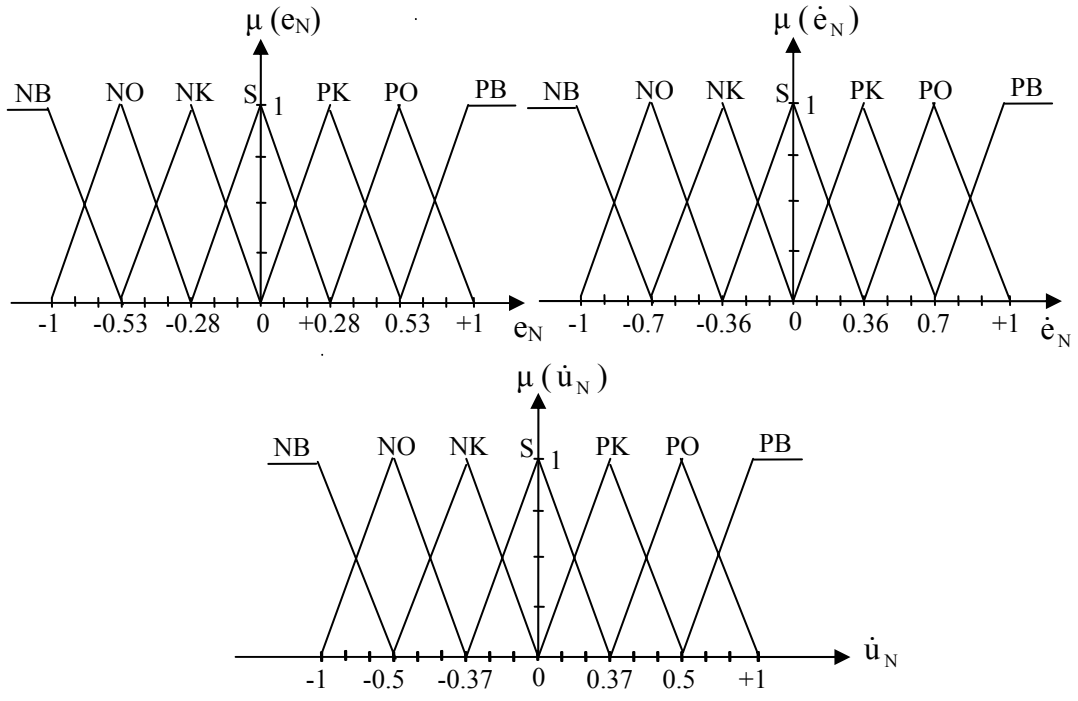
Şekil 5.11 Performans indeksi ve yeni kural tablosu değerleri

Adım 3: Üyelik fonksiyonlarının optimizasyonu

Üçüncü adımda, bulanık denetleyicinin üçgen üyelik fonksiyonları, yukarıda elde edilen normalizasyon parametreleri ve yeni kural tablosu kullanılarak GA ile yeniden elde edilmiştir. Şekil 5.12'de, kontrol sistemi performans indeksinin değişim eğrisi ve üyelik fonksiyonu parametrelerinin optimal değerlere yerleşme eğrileri görülmektedir. Optimizasyon işlemi, 28 nesil sonra yakınsamaktadır. Yapılan optimizasyon işlemi sonunda kontrol sisteminin yeni performans indeksi J_{in}=3.2349 olarak bulunmuştur. Bu aynı zamanda gerçek kodlu GA ile gerçekleştirilen ardışık optimizasyon için sonuç performans indeksidir. Sonuç performans indeksinin, önceki adımlarda elde edilen performans indekslerinden daha küçük olduğu görülmektedir. Yani, bu adımda kullanılan denetleyici parametreleri kontrol sisteminde daha iyi sonuç vermektedir. Şekil 5.13'de, elde edilen yeni üyelik fonksiyonları görülmektedir.



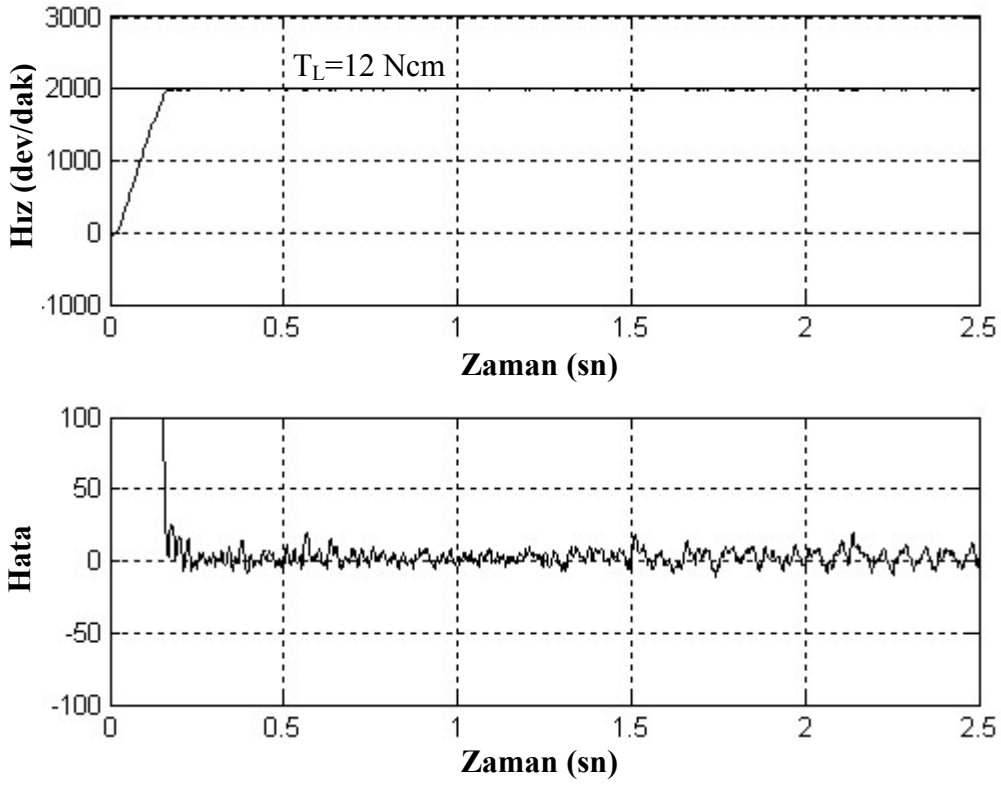
Şekil 5.12 Performans indeksi ve yeni üyelik fonksiyonu parametreleri



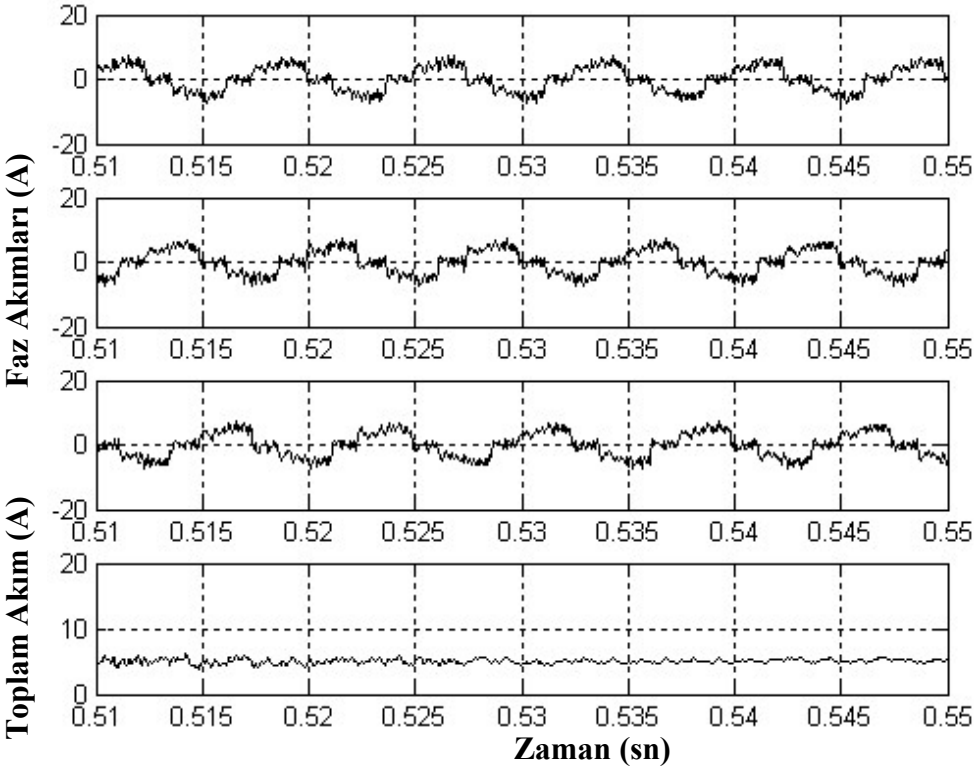
Şekil 5.13 GA ile elde edilen yeni üyelik fonksiyonları

Adım 4: Motor cevap eğrileri

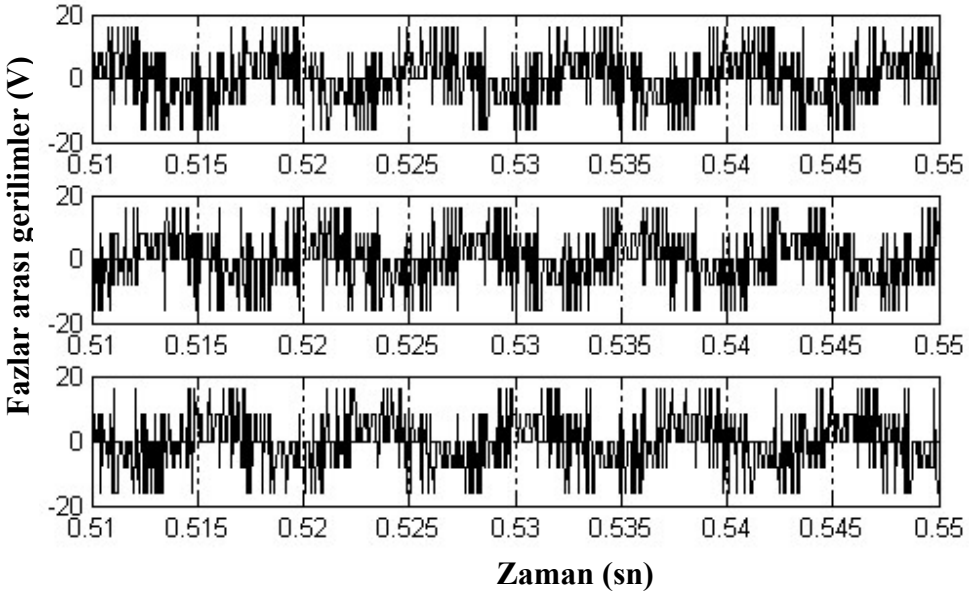
Bulanık denetleyici için GA ile elde edilen optimal değerler kullanılarak, fırçasız DC motorun hız kontrolü yapılmıştır. Ardışık optimizasyon işleminde, her adım için performans indeksinin gittikçe azaldığı yani sistemin daha iyi kontrol edildiği görülmektedir. Ardışık optimizasyon yaklaşımı sonucu elde edilen parametre değerlerinde motor cevap eğrileri aşağıda verilmiştir. Buna göre; Şekil 5.14'de hız cevap eğrisi ve cevap hatası, Şekil 5.15'de motor faz akımları ve toplam akım, Şekil 5.16'da faz gerilimleri, Şekil 5.17'de faz momentleri ve toplam moment, Şekil 5.18'de pozisyon ve her bir fazın zıt emk gerilimleri verilmiştir. Ayrıca Şekil 5.19'da, GA ile ardışık ve geleneksel olarak tasarlanan bulanık denetleyicinin hız-cevap eğrileri karşılaştırma için birlikte verilmiştir. Şekil 5.20'de ise, iki durum için büyütülmüş hata eğrileri verilmiştir.



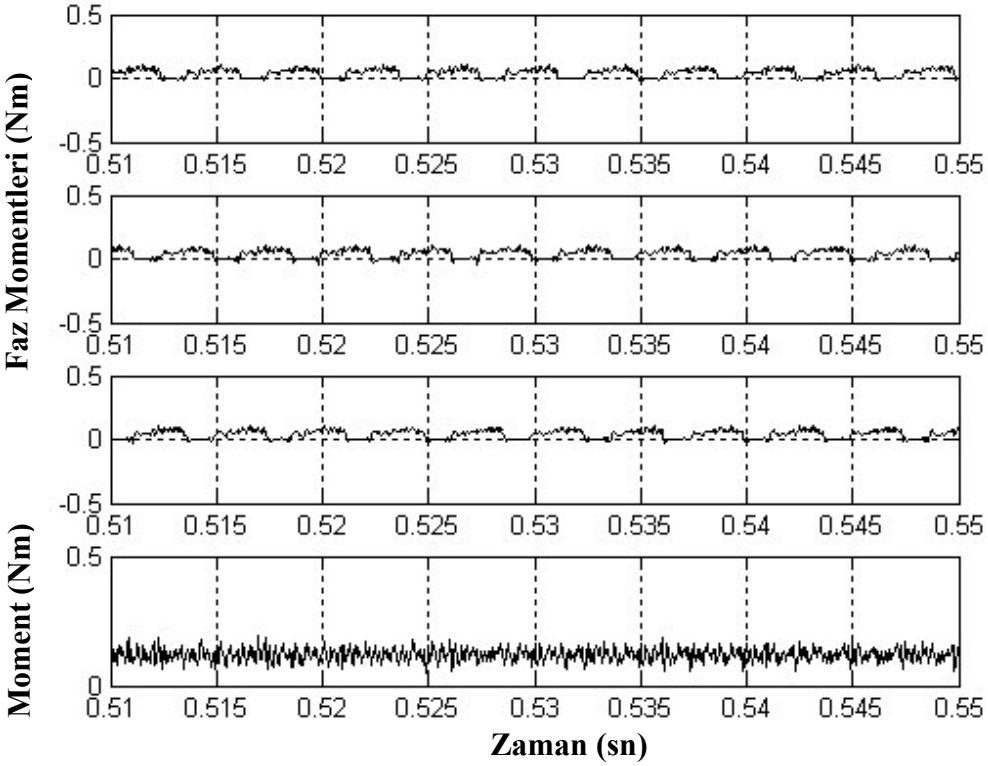
Şekil 5.14 Hız-cevap eğrisi ve cevap hatası



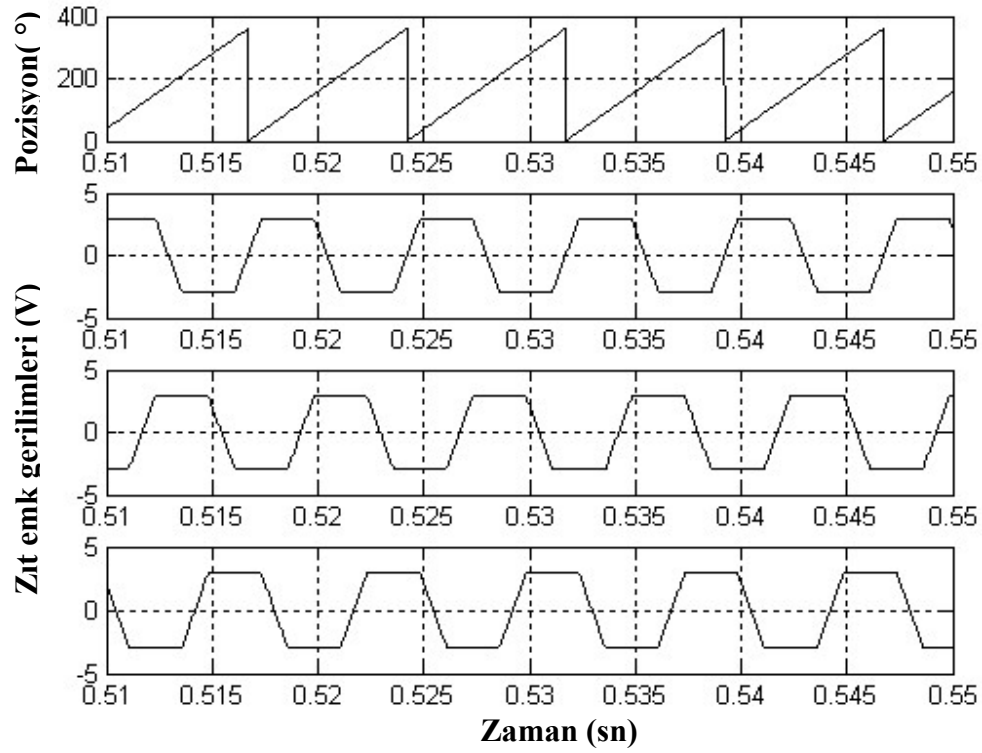
Şekil 5.15 Motor faz akımları ve toplam akım



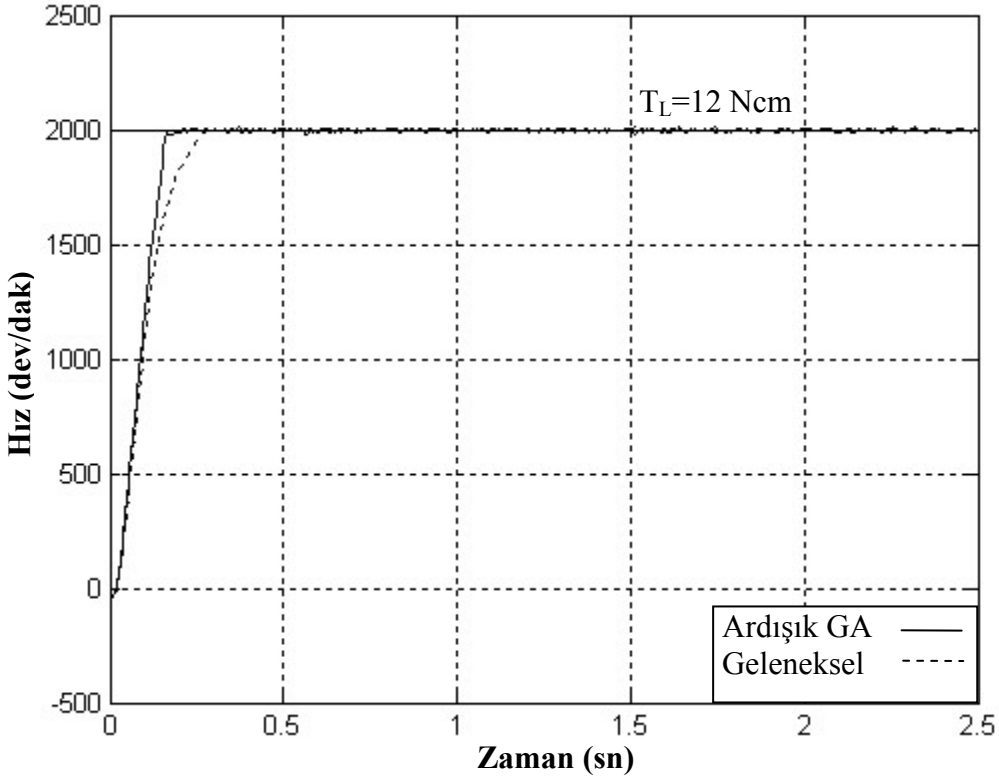
Şekil 5.16 Motor faz gerilimleri



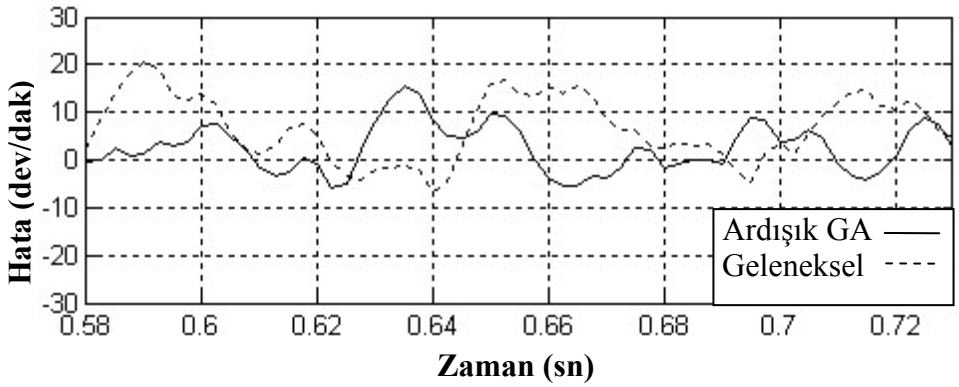
Şekil 5.17 Motor faz momentleri ve toplam moment



Şekil 5.18 Motor pozisyon ve zıt-emk gerilimleri



Şekil 5.19 GA ile ardışık tasarlanan ve geleneksel olarak tasarlanan bulanık denetleyicinin hız cevap eğrileri



Şekil 5.20 GA ile ardışık tasarlanan ve geleneksel olarak tasarlanan bulanık denetleyicinin büyütülmüş hata eğrileri

Simülasyon sonuçlarından da görüldüğü gibi, bulanık denetleyicilerin tasarımında ardışık GA yaklaşımı yeterince iyi sonuç vermektedir. Ayrıca, bulunan yeni parametre değerlerinin performans açısından değerlendirilmesi yapılırsa; J_{in}=3.2349 olarak bulunmaktadır.

5.2.2.2 Eş zamanlı optimizasyon yaklaşımı simülasyon sonuçları

Bulanık denetleyici tasarımında kullanılan ardışık optimizasyon yönteminde, denetleyici parametreleri birbirlerinden tamamen bağımsız olmadığı için tam bir optimal çözüm bulunamayabilir. Bu yüzden, denetleyici parametrelerinin aynı anda optimizasyonu daha doğru bir yaklaşımdır. Bu nedenle eş zamanlı optimizasyon için, genetik algoritmada normalizasyon parametreleri, üyelik fonksiyonları ve karar tablosu birlikte tek bir kod içerisinde ifade edilmiştir. Böylece, kontrol yüzeyinde uygun normalizasyon parametreleri belirlenirken, aynı zamanda kontrol yüzeyini en iyi şekilde kapsayan kural tablosu da elde edilmekte ve üyelik fonksiyonları tarama yüzeyini optimal biçimde bölümlere ayırmaktadır.

Bu çalışmada, özellikleri Tablo 5.8'de tanımlanan gerçek kodlu genetik algoritma kullanılmıştır. Tablodan görüldüğü gibi, optimize edilecek 16 parametre için bir arama aralığı tanımlanmış ve parametrelerin duyarlılığı 0.0001 olacak şekilde ayarlanmıştır. Optimize edilecek parametre sayısı fazla olduğu için, mutasyon ve çaprazlama olasılıkları ardışık GA'ya göre daha büyük seçilmiştir. Programı sonlandıracak maksimum nesil sayısı da büyük seçilmiştir.

Tablo 5.8 Eş zamanlı optimizasyon için Gerçek kodlu GA parametreleri

Parametre sayısı	16
Parametre değer aralığı	[1000 1000 1000 7 7 7 7 7 7 7 1.4 2.9 1.4 2.9 1.4 2.9] [0 0 0 1 1 1 1 1 1 1 0.1 1.5 0.1 1.5 0.1 1.5]
Parametre duyarlılığı	0.0001
Başlangıç populasyonu	0-Rasgele
Mutasyon olasılığı	0.1
Çaprazlama olasılığı	1
Populasyon sayısı	20
Elitizm	1-Evet
Durdurma kriteri	$J_{in}<0.1$
Nesil sayısı	1000

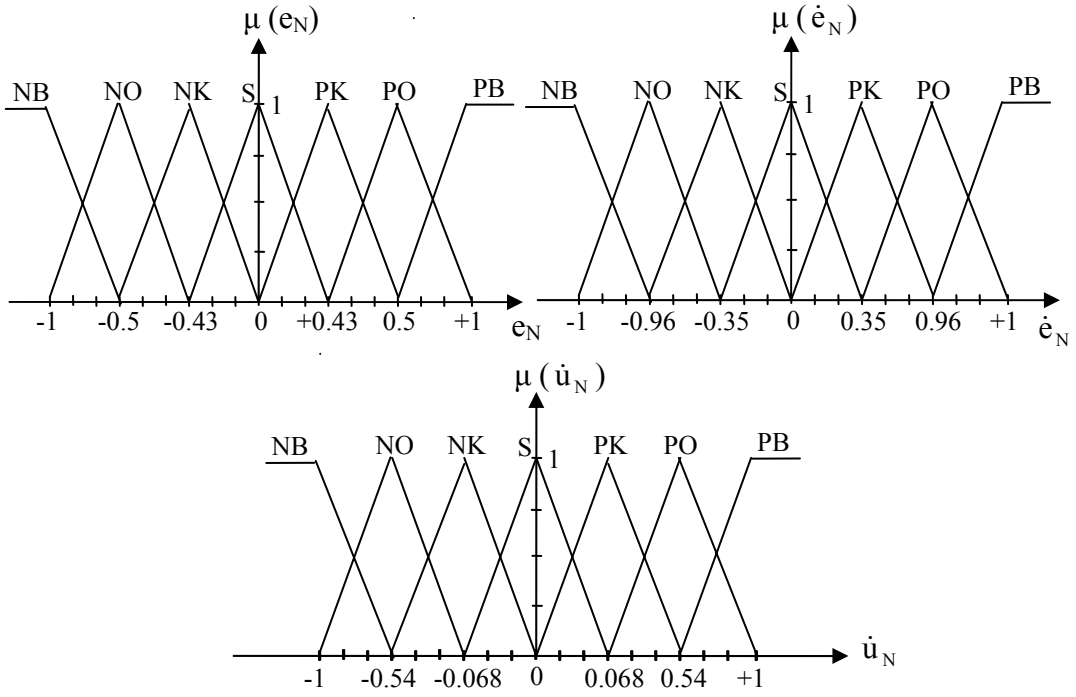
Algoritma yürütüldükten sonra elde edilen optimal normalizasyon parametreleri, kural tablosu ve üyelik fonksiyonları sırasıyla Tablo 5.9, Tablo 5.10 ve Şekil 5.21'de verilmiştir. Bulanık denetleyicili sistemin performans indeksinin yerleşme eğrileri Şekil 5.22'de görülmektedir. Elde edilen optimal değerlerde kontrol sistemi performans indeksi J_{in}=3.2327 değerine yakınsamıştır.

Tablo 5.9 Eş zamanlı optimizasyonda GA ile elde edilen optimal normalizasyon parametreleri

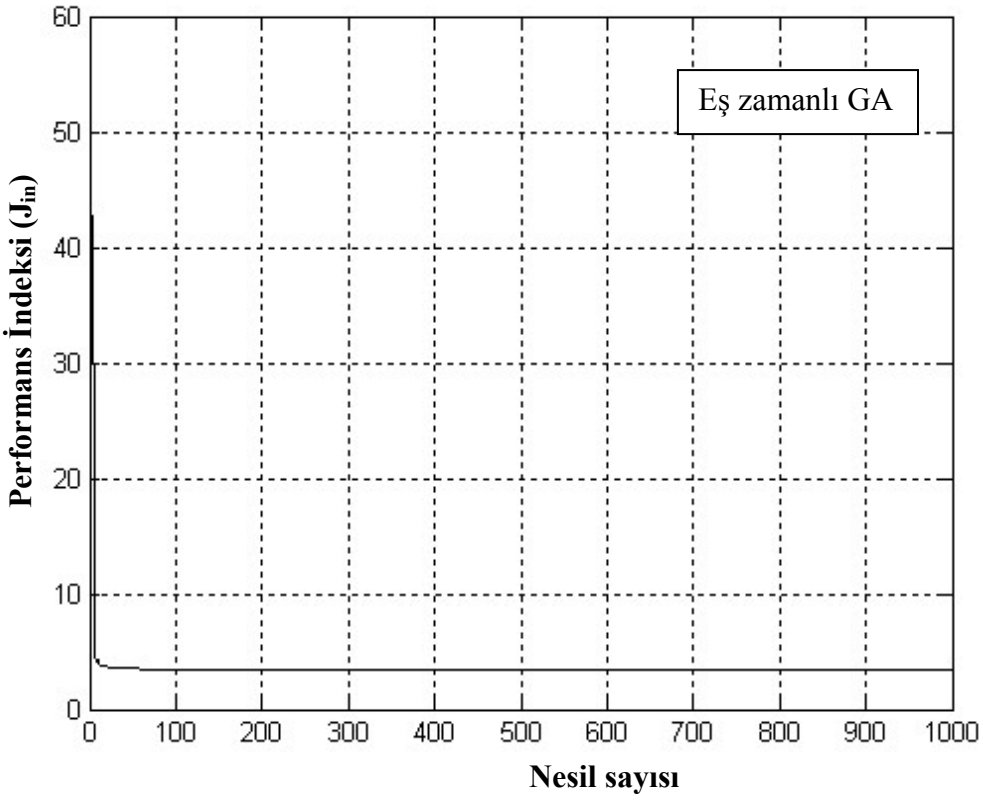
N_e	123.034
$N_{\dot{e}}$	2.816
N_u	3.283
J_{in} (Performans indeksi)	3.2327

Tablo 5.10 Eş zamanlı GA ile elde edilen kural tablosu

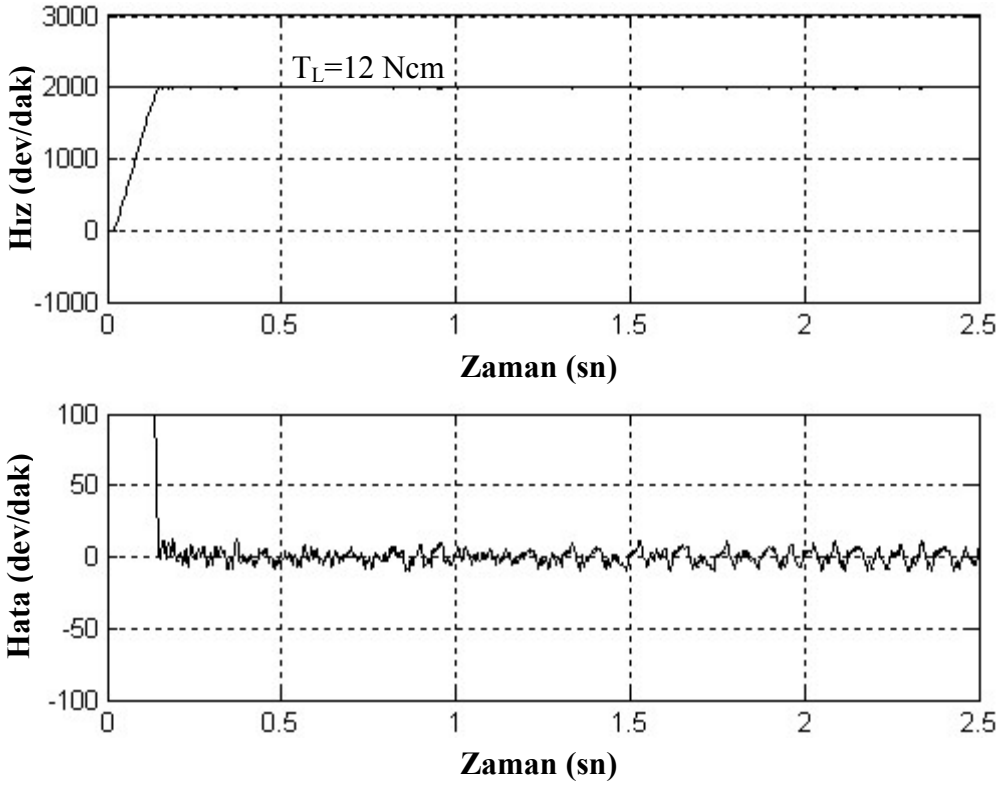
$\dot{e}_N$ \ e_N	NB	NO	NK	S	PK	PO	PB
NB	NB	PO	NK	NB	NB	NO	NK
NO	PO	NK	NB	NB	NO	NK	PO
NK	NK	NB	NB	NO	NK	PO	PB
S	NB	NB	NO	NK	PO	PB	PB
PK	NB	NO	NK	PO	PB	PB	PK
PO	NO	NK	PO	PB	PB	PK	NO
PB	NK	PO	PB	PB	PK	NO	PB



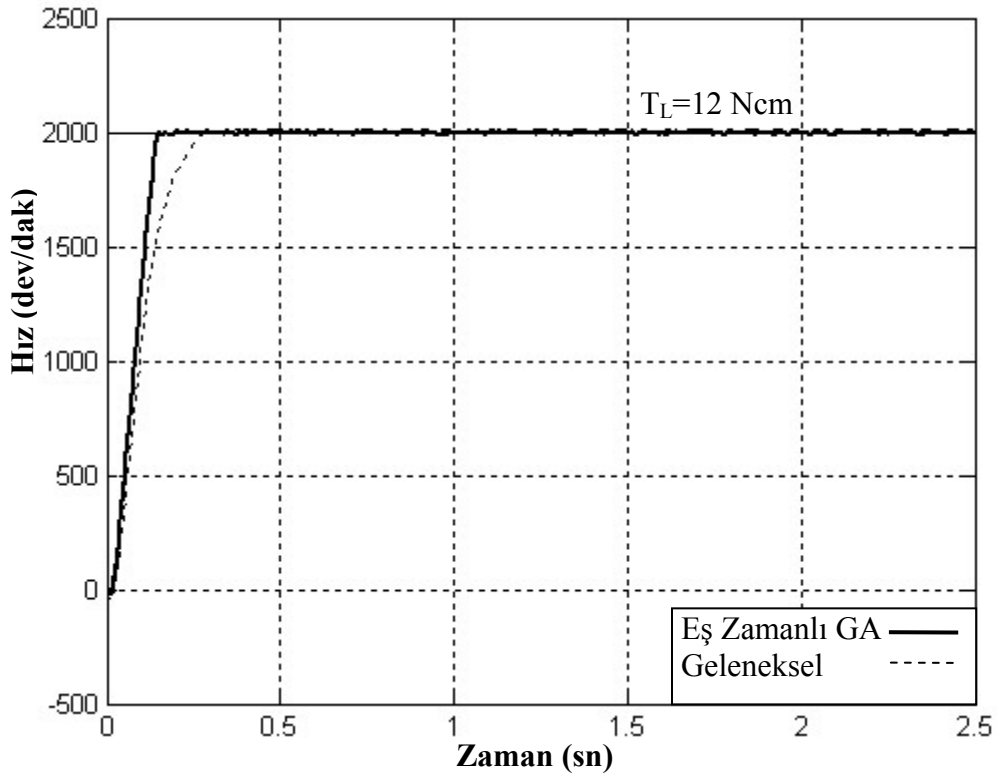
Şekil 5.21 Eş zamanlı optimizasyon sonucu bulunan üyelik fonksiyonları



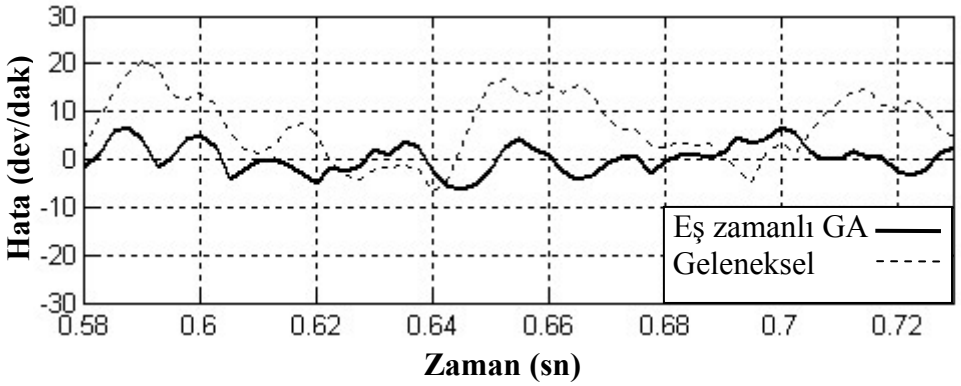
Şekil 5.22 Eş zamanlı optimizasyonda performans indeksinin yerleşme eğrisi



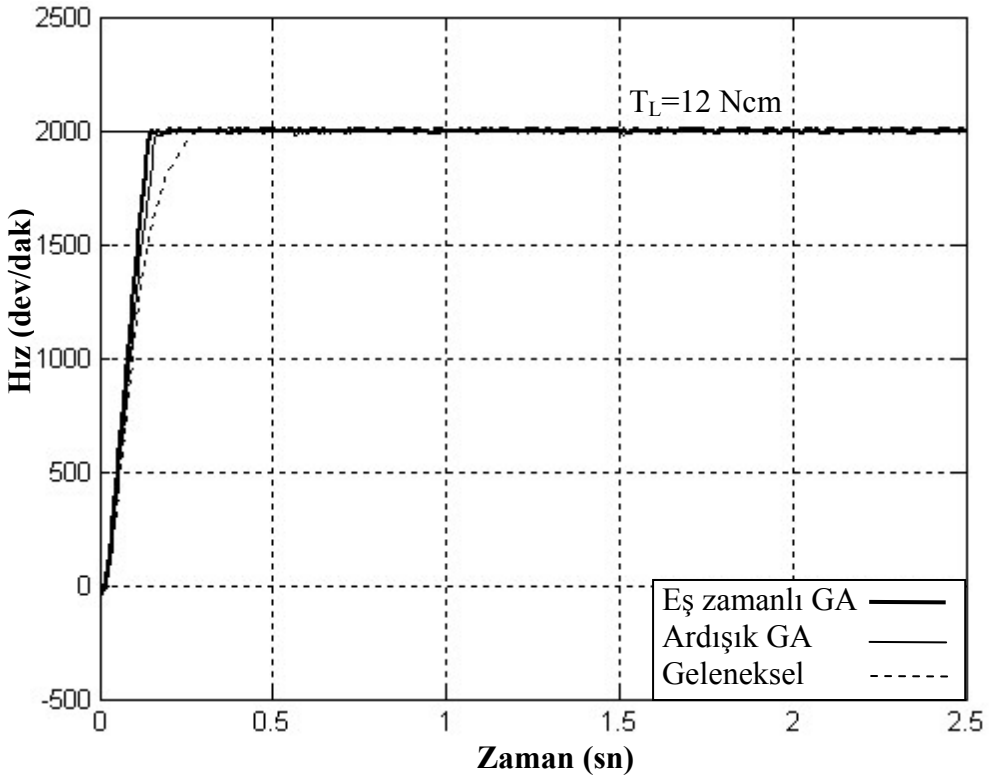
Şekil 5.23 Hız-cevap eğrisi ve cevap hatası



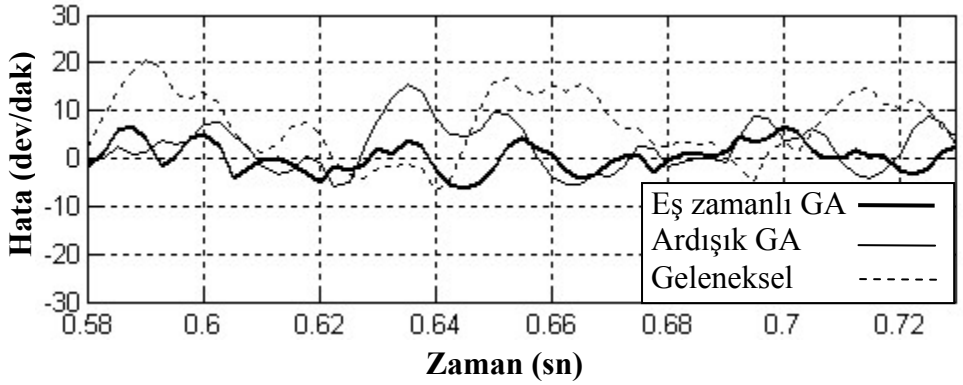
Şekil 5.24 Eş zamanlı GA ve geleneksel bulanık denetleyicinin hız-cevap eğrileri



Şekil 5.25 Eş zamanlı GA ile ardışık tasarlanan ve geleneksel olarak tasarlanan bulanık denetleyicinin büyütülmüş hata eğrileri



Şekil 5.26 Eş zamanlı GA, Ardışık GA ve geleneksel yöntemle tasarlanan bulanık denetleyicinin hız-cevap eğrileri



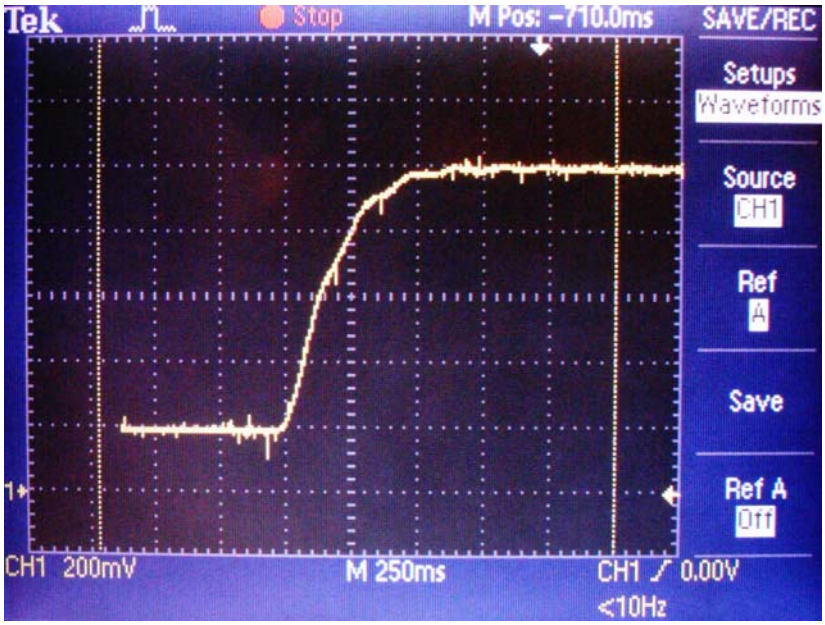
Şekil 5.27 Eş zamanlı GA, ardışık GA ve geleneksel yöntemle tasarlanan bulanık denetleyicilerin büyütülmüş hata eğrileri

5.2.3 Deneysel Sonuçlar

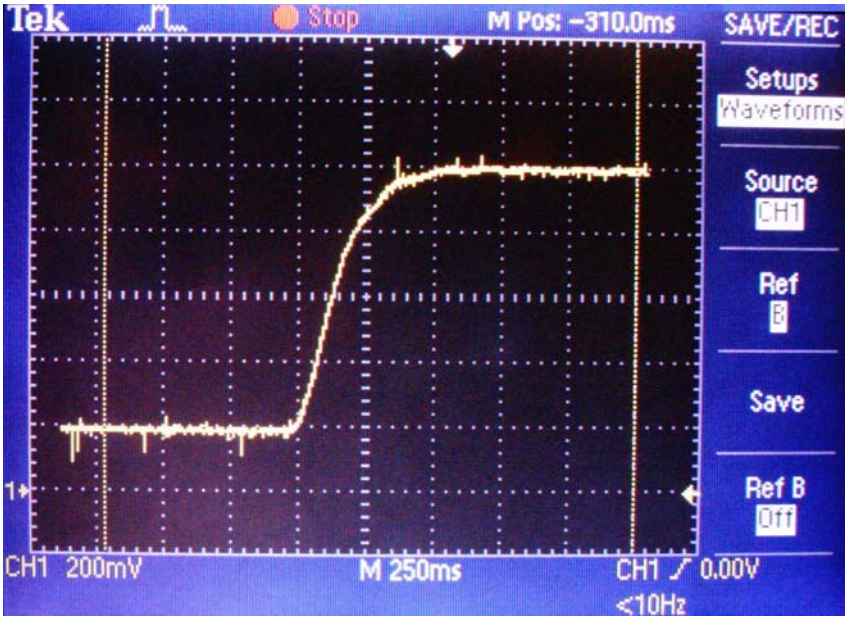
Uygulama devresi olarak, burada da Şekil 4.23 ile verilen Bölüm 4'deki ADSP-21992 tabanlı sistem kullanılmıştır. Simülasyon çalışmalarında elde edilen optimal denetleyici parametreleri kullanılarak, fırçasız DC motorun sensörsüz kontrolü gerçekleştirilmiştir.

Ardışık tasarım ve eş zamanlı tasarım için ayrı ayrı uygulama yapılmıştır. İlk olarak, ardışık GA optimizasyon yönteminde Tablo 5.6 ve 5.7 ile Şekil 5.13'de verilen optimal parametreler için sistem cevabı Şekil 3.28'deki gibi elde edilmiştir. Bulanık denetleyicinin hız-cevap eğrisinden görüldüğü gibi yerleşme zamanı yaklaşık 600 ms'dir. İkinci olarak, eş zamanlı GA ile optimizasyon yönteminde Tablo 5.9 ve 5.10 ile Şekil 5.21'de verilen optimal parametreler için sistem cevabı Şekil 5.29'daki gibi olmaktadır. Şekilden görüldüğü gibi, eş zamanlı GA ile optimize edilen bulanık denetleyicinin referans hıza yerleşme zamanı yaklaşık 500 ms'dir. Şekil 5.30'da ise eş zamanlı GA ve ardışık GA ile optimize edilen bulanık denetleyici hız cevap eğrileri birlikte verilmiştir. Burada, simülasyon işlemlerine benzer şekilde eş zamanlı GA yönteminin biraz daha iyi olduğu görülmektedir.

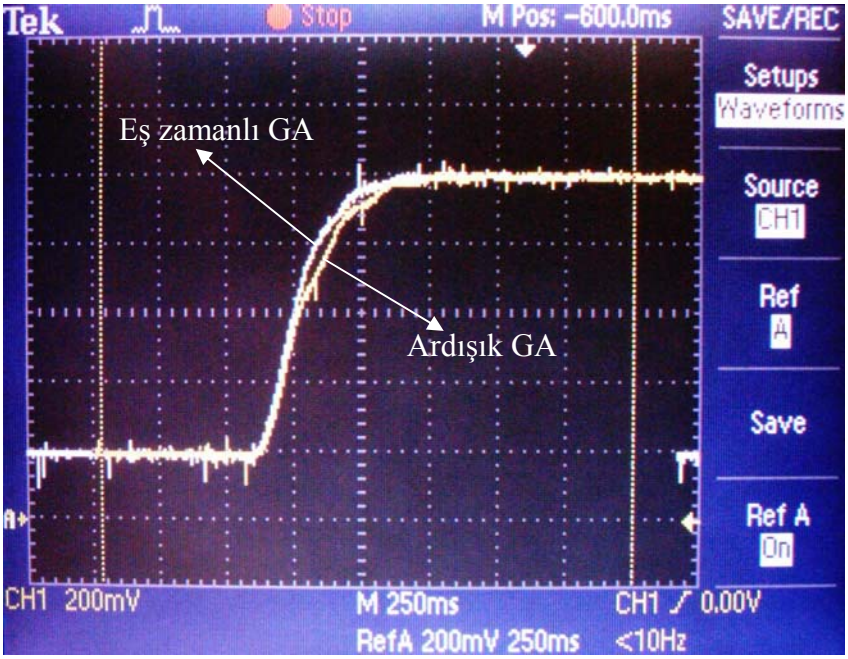
Şekil 5.31'de, ardışık GA ile tasarlanan bulanık denetleyici ve geleneksel olarak tasarlanan bulanık denetleyici hız-cevap eğrileri ve Şekil 5.32'de, eş zamanlı GA ile tasarlanan bulanık denetleyici ile geleneksel olarak tasarlanan bulanık denetleyici için hız cevap eğrileri birlikte verilmiştir. Burada, her iki optimizasyon yaklaşımının geleneksel yönteme göre daha iyi bir kontrol sağladığı görülmektedir.



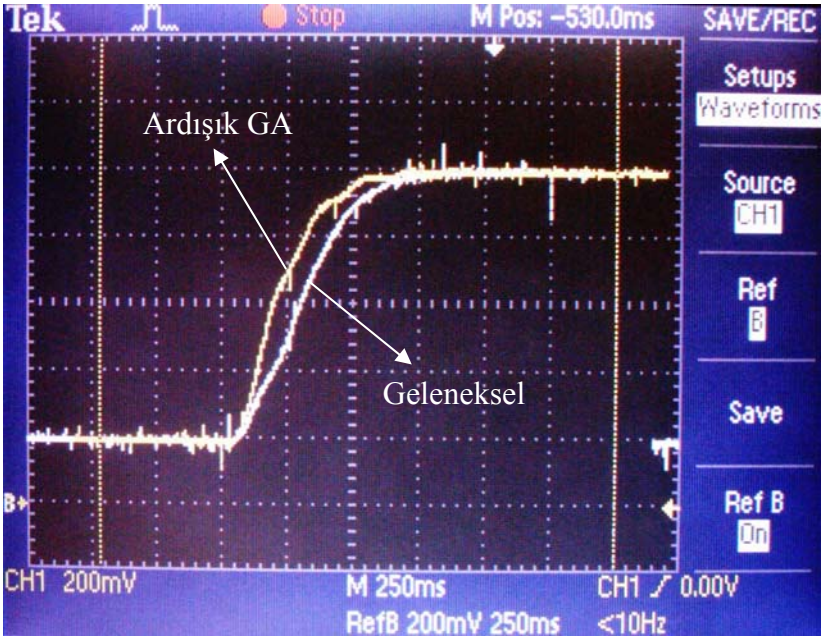
Şekil 5.28 Ardışık GA ile tasarlanan bulanık denetleyici hız-cevap eğrisi
Time/Div:250ms, Volt/Div:400 d/d



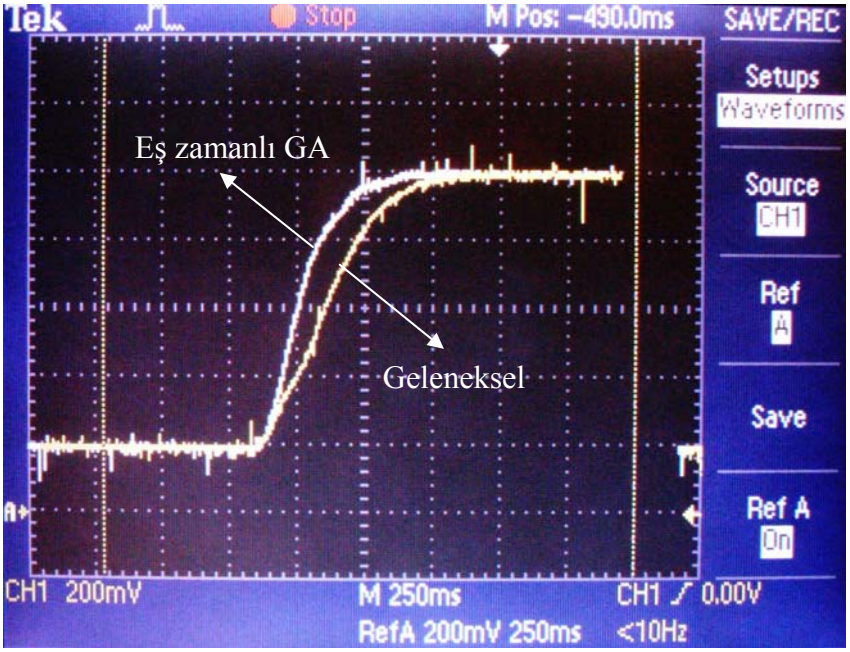
Şekil 5.29 Eş zamanlı GA ile tasarlanan bulanık denetleyici hız-cevap eğrisi
Time/Div:250ms, Volt/Div:400 d/d



Şekil 5.30 Eş zamanlı GA ve ardışık bulanık denetleyici hız-cevap eğrileri
Time/Div:250ms, Volt/Div:400 d/d



Şekil 5.31 Ardışık GA ve geleneksel bulanık denetleyici hız-cevap eğrileri
Time/Div:250ms, Volt/Div:400 d/d



Şekil 5.32 Eş zamanlı GA ve geleneksel bulanık denetleyici hız-cevap eğrileri
Time/Div:250ms, Volt/Div:400 d/d

6. SONUÇ VE ÖNERİLER

6.1 Sonuçlar

Bu çalışmada, kullanımı hızla yaygınlaşan özel elektrik motorlarından fırçasız DC motorların, genetik tabanlı bulanık denetleyici ile optimal kontrolü simüle edilmiş ve işlem gücü yüksek olan bir DSP kullanılarak sensörsüz hız kontrolü pratik olarak gerçekleştirilmiştir.

Simülasyon çalışmasında, fırçasız DC motorların modellenmesi işleminde, literatürde verilen sinüsoidal zıt emk fonksiyonunun yerine, motor yapısına daha uygun olan zıt emk trapezoidal şekil fonksiyonu tanımlanmış ve böylece daha doğru bir motor modeli elde edilmiştir. Simülasyon işleminde diferansiyel motor denklemleri dördüncü dereceden Runge-Kutta nümerik integrasyon metodu ile hesaplanmıştır. Böylece simülasyonda, hesaplama hataları ve nümerik kararsızlıklar azaltılmıştır. Ayrıca, fırçasız DC motorları sürmek için kullanılan inverter yapısı incelenmiş ve matematiksel olarak modellenmiştir.

Fırçasız DC motor kontrol sisteminde, ilk olarak bulanık denetleyici tasarımı, geleneksel deneme ve yanılma yöntemi kullanılarak gerçekleştirilmiştir. Bu yöntemde, tasarım işlemi uzun zaman almakla beraber, denetleyici performansı hakkında bir kriter belirlenememiş ve optimal kontrol garanti edilememiştir. Sadece görsel olarak sistem cevap eğrisi gözlemlenebilmiştir.

Bulanık denetleyicinin optimal tasarımında optimizasyon işlemi için gerçek kodlu bir genetik algoritma başarıyla uygulanmıştır. Bulanık denetleyicinin GA ile optimizasyonu için, sistem performansını ölçen ve GA'da uygunluk değeri olarak kullanılan çok parametreli bir performans indeksi (J_{in}) tanımlanmıştır. Tanımlanan performans indeksi, çoklu amaç fonksiyonu formuna sokularak GA ile minimize edilmiştir. Bulanık denetleyicinin GA ile optimizasyonu için iki ayrı yaklaşım kullanılmıştır. İlki; GA ile önce normalizasyon, daha sonra kural tablosu ve son olarak üyelik fonksiyonlarının optimize edildiği ardışık optimizasyon yöntemidir. İkincisi; tüm bulanık denetleyici parametrelerinin aynı anda optimize edildiği eş zamanlı optimizasyon yaklaşımıdır.

Simülasyon sonuçlarına göre; ardışık optimizasyon yönteminde, GA ile denetleyicinin 3 adet normalizasyon parametresinin optimizasyonu sonucunda performans indeksi J_{in}=3.2674 değerine düşmüş ve bu işlem yaklaşık 35 nesil sonra gerçekleşmiştir. Ardışık optimizasyon yönteminin ikinci adımında kural tablosu GA ile yeniden belirlenmiştir. Bu amaçla 7x7=49 adet hücreye sahip olan kural tablosu, kayan tip bir tabloya dönüştürülerek optimizasyon parametre sayısı 7'ye kadar indirgenmiştir. GA ile bu parametrelerin optimizasyonu sonucu, yeni kural tablosu elde edilmiş ve yaklaşık 90 nesil sonra performans indeksi J_{in}=3.2641 değerine düşmüştür. Ardışık optimizasyon yönteminin son adımında, GA ile bulanık denetleyicinin üyelik fonksiyonları optimize edilmiştir. İki girişli tek çıkışlı olan denetleyicide üç adet üyelik fonksiyonu ve her birinde 7'şer adet üçgen bulanık küme bulunmaktadır. Her bir bulanık küme üç adet parametre ile temsil edilirse, optimize edilmesi gereken parametre sayısı oldukça büyük çıkmaktadır. Bu nedenle ilk olarak simetri özellikleri ve kümelerin örtüşme özellikleri kullanılarak GA ile optimize edilecek parametre sayısı 6'ya kadar indirgenmiştir. Bu altı parametrenin GA ile optimizasyonu sonucu, yeni üyelik fonksiyonları elde edilmiş ve performans indeksi yaklaşık 30 nesil sonra J_{in}=3.2349 değerine kadar düşmüştür. Ardışık optimizasyon işlemi toplamda, yaklaşık 150 nesilde gerçekleştirilmiştir. Ardışık optimizasyon sonucu elde edilen optimal bulanık denetleyici parametrelerinde oldukça iyi sistem cevap eğrileri elde edilmiştir. Ardışık GA ile optimizasyon işlemiyle, geleneksel yönteme göre sistemin geçici ve sürekli rejim cevabında gözle görülür bir iyileşme sağlanmıştır.

Eş zamanlı optimizasyon yönteminde ise, normalizasyon parametreleri, kural tablosu ve üyelik fonksiyonu parametreleri aynı anda eş zamanlı olarak GA ile optimize edilmiştir. Optimizasyon işlemi sonucunda sistemin performans indeksi J_{in}=3.2327 değerine kadar düşmüştür. Bu değer, yapılan çalışmada elde edilen en iyi optimizasyon değeridir. Eş zamanlı GA ile optimizasyon işlemi yaklaşık 850 nesil sonra gerçekleşmiştir. Bu yöntem ile bulanık denetleyicinin geçici ve sürekli rejim cevabında geleneksel yönteme ve ardışık GA'ya göre daha iyi sonuçlar elde edilmiştir. Ancak, Ardışık optimizasyon yönteminde toplam 150 nesilde tüm denetleyici optimize edilirken, tüm parametrelerin aynı anda optimize edildiği eş zamanlı yaklaşımda yaklaşık 850 nesilde optimizasyon gerçekleşmiştir. Buradan

görülmektedir ki, ardışık optimizasyon daha az zaman almaktadır. Ayrıca, sistem cevap eğrileri incelendiğinde, ardışık optimizasyon yönteminin yeterince iyi sonuç verdiği görülmektedir. Ardışık optimizasyon işleminde yukarıda da verildiği gibi kural tablosunun optimizasyonu ile performans indeksinde çok az bir iyileşme sağlanmıştır. Standart kural tablosu kullanılırsa ardışık optimizasyon işlemi 65 nesil sonra yakınsamaktadır.

Deneysel çalışmada, ADSP-21992 işlemci kullanılarak fırçasız DC motorların sensörsüz kontrolü başarılı bir şekilde gerçekleştirilmiştir. Sistemde hız sensörü kullanılmadan zıt emk sinyalleri yardımıyla motor hızı dolaylı olarak belirlenmiş ve geri besleme olarak denetleyiciye uygulanmıştır. Uygulamada fırçasız DC motor pozisyon sensörleri elimine edildiği için, sensörlerden dolayı oluşan olumsuzluklar giderilmiştir. Denetleyici olarak, simülasyon çalışmasında gerçekleştirilen geleneksel denetleyici, ardışık GA ve eş zamanlı GA ile optimize edilen bulanık denetleyiciler kullanılmıştır. Simülasyon işlemlerinde kullanılan denetleyici algoritmaları pratik uygulama için DSP işlemcisine uygun yapıya dönüştürülmüş ve gerçek zaman uygulama yapılarak, fırçasız DC motorların bulanık denetleyici ile sensörsüz kontrolünde yeni bir çözüm ortaya konulmuştur.

Deneysel sonuçlar incelendiği zaman, simülasyon sonuçları ile uyum içinde olduğu, simülasyon işlemlerinden elde edilen optimal denetleyicilerin uygulamada başarıyla kullanılabileceği görülmüştür.

Hızla sanayileşen ülkemizde, teknolojik ürünlerin büyük bir kısmının yurtdışından ithal edildiği düşünülürse, simülasyon ortamında yürütülen ve pratik olarak gerçekleştirilen bu çalışma, ilgili konularda çalışan kişi ve firmalara yardımcı olacak ve ülke ekonomisine katkı sağlayacaktır.

6.2 Öneriler

Gerçekleştirilen sistem, oldukça genel bir yaklaşım içermesinden dolayı, fırçasız DC motorların sensörsüz kontrolünün yanında, bulanık denetleyicilerin kullanıldığı lineer olmayan diğer sistemlerde de etkin olarak kullanılabilir yapıdadır.

Yapılan bu çalışmada elde edilen sonuçlar göz önüne alındığında, GA ile, diğer optimizasyon problemlerinin çözümünü etkili bir şekilde gerçekleştirmek mümkündür. Özellikle, GA ile çok parametreli ve çoklu amaç fonksiyonuna sahip sistemlerde etkili çözüm elde edilebilmektedir. Optimize edilecek olan parametre sayısı fazla ise, gerçek kodlu GA'ları kullanmak avantaj sağlamaktadır.

Bu çalışmada, simülasyon ortamında gerçekleştirilen optimizasyon işleminin, gerçek zaman uygulamalarında yapılabilirliği araştırılmalıdır. Bunun için, GA'ların özel yapılara sokularak işlem süresinin azaltılması düşünülebilir. Ayrıca, çalışmada seçimi kullanıcıya bırakılan denetleyici yapısıyla ilgili parametreler de GA ile elde edilerek, bulanık denetleyici tasarımı tam otomatik bir yapıya sokulabilir.

Fırçasız DC motorların sensörsüz kontrol uygulamalarında yapay sinir ağları gibi diğer yapay zeka yöntemleri, adaptif kontrol yöntemleri ve optimal kontrol yöntemleri uygulanabilir.

Diğer optimizasyon yöntemleri kullanılarak bulanık denetleyiciler optimize edilebilir ve karşılaştırma yapılabilir.

Gerçekleştirilen simülasyon algoritmaları ve uygulama devreleri, eğitim amaçlı kullanılabilir ve görsel özelliği sayesinde kolayca anlaşılabilir.

Çalışmada kullanılan Runge-Kutta yöntemi, diğer benzer nümerik problemlerin çözümünde kullanılabilir.

GA'ları gerçek zaman uygulamalarda, yakın optimal çözüm (near optimal solution) amacıyla kullanmak mümkündür.

7. KAYNAKLAR

Adewuya, A.A. 1996. New methods in Genetic Search with Real Valued Chromosomes. M.S. Thesis, Massachusetts Institute of Technology, Cambridge.

Ahn, T., Kwon, Y., Kang, H. 2000. Drive of Induction Motors using a Pseudo-On-Line Fuzzy-PID Controller Based on Genetic Algorithm. Transaction on Control, Automation and System Engineering 2: 85-91.

Angeline, P.J. 1995. Evolution revolution: An introduction to the special track on genetic and evolutionary programming. IEEE Expert Intelligent Systems and their Applications 10: 6-10.

Arslan, A., Kaya, M. 2001. Determination of Fuzzy Logic Membership Functions using Genetic Algorithms. Elsevier Fuzzy Sets and Systems 118: 297-306.

Ashrafzadeh, F. 1996. Genetic Based Auto-Design of Fuzzy Controllers. PhD. Thesis, The University of Galgary, Alberta.

Ashrafzadeh, F., Nowicki E.P., Mohammadian, M., Salmon, J.C. 1997. Optimal Approximation of Nonlinear Functions by Fuzzy Systems. IEEE Canadian Conf. on Electrical and Computer Engineering, pp. 781-784 Newfoundland, May 25-28.

Aydoğdu Ö., Akkaya R. 2005. DSP Based Fuzzy Control of a Brushless DC Motor Without Position and Speed Sensors. Proc. of 4th Int. Advanced Technologies Symposium, pp.182-187 Konya, September 28-30.

Bonvin, F., Perriard, Y. 2000. BLDC Motor Control in Multiple dq Axes. Paper no.475 Proc. of the IEE Int. Conf. on Power Electronics and Variable Speed Drives, London, September 18-19.

Chen, S.M., Chen Y.C. 2002. Automatically Constructing Membership Functions and Generating Fuzzy Rules using Genetic Algorithms. An International Journal Cybernetics and Ssytems 33: 841-862.

Çunkaş, M., Akkaya R. 2002. İkili ve gerçek kodlu genetik algoritmaların karşılaştırılması. S.Ü. Müh. Mim. Fak. Dergisi 7.2: 11-17.

Çunkaş M. 2004. Elektrik motorlarında genetik algoritma ile tasarım optimizasyonu. Doktora Tezi, S.Ü. Fen Bilimleri Enstitüsü, Konya.

Davis, L. 1991. The Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York.

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., pp. 689-694. Baveno, June 23-27.

Eshelman, L.J., Shafter D.J. 1993. Real-coded genetic algorithms and interval-schemata. Foundations of Genetic Algorithms 2. Whitley D.L. (ed), pp.187-203, Morgan Kaufman, San Mateo, CA.

Elmas, Ç. 2003. Bulanık Mantık Denetleyiciler, Seçkin Yayıncılık, Ankara.

Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, New York.

Goldberg, D.E. 1993. Making genetic algorithms fly: A lesson from the Wright brothers. Adv. Technol. 2:1-8.

Haupt Randly L., Haupt Sue E. 1998. Practical Genetic Algorithms. A Willey-Interscience Publication, USA.

Hemati, N., Leu, M.C. 1992. A Complete Model Characterization of Brushless DC Motors. IEEE Transactions on Industry Applications 28.1: 172-180.

Hendershot, J.R., Miller, T.J.E., 1994. Design of Brushless Permanent-Magnet Motor, Oxford Univ. Pres Inc. New York.

Herrera, F., Lozano, M., Verdegay, J.L. 1998. A Learning Process for Fuzzy Control Rules using Genetic Algorithms. Elsevier Fuzzy Sets and Systems 100: 143-158.

Holland, J.H. 1975. Adaptation in Natural and Artificial Systems, The University of Michigan Press. Ann Arbor.

Kestler, W., 2000. Mixed Signal and DSP Design Techniques. Published by Analog Devices, Inc. USA.

Kim, K.H., Youn, M.J. 2002. DSP Based High Speed Sensorless Control for a Brushless DC Motor using a DC Link Voltage Control. Electric Power Components and Systems 30: 889-906.

Kim T. 2003. Sensorless Control of the BLDC Motors from Near-Zero to Full Speed. PhD. Thesis, Texas A&M University, Texas.

Kinzel, J., Klawonn, F., Kruse, R. 1994. Modifications of Genetic Algorithms for Designing and Optimizing Fuzzy Controllers. Proc. First IEEE Conf. on Evolutionary Computation, pp.28-33, Orlando, June 27-29.

Lee, C.K., Pang, W.H. 1994. A Brushless DC Motor Speed Control System using Fuzzy Rules. Paper no.399, Proc. of the 5th. IEE Int. Conf. on Power Electronics and Variable Speed Drives, London, October 26-28.

Lee B., Ehsani M. 2003. Advanced Simulation Model for Brushless DC Motor Drives. Electric Power Component and Systems 31: 841-868.

Liu, T.H., Cheng, C.P., 1993. Controller Design for A Sensorless Permanent Magnet Synchronous Drive System. IEE Electric Power Applications 140: 369-378.

Luk, P.C.K., Lee, C.K. 1994. Efficient Modelling for a Brushless DC Motor Drive. Proc. of the 20th. Int. Conf. on IEEE Industrial Electronics Society, Vol.1, pp.188-191 Bologna, September 5-9.

Matsui, N. 1993. Sensorless Operation of Brushless DC Motor Drives. Proc. of the 19th. Annual Conference of IEEE Industrial Electronics Society, Vol.2, pp.739-744 Hawaii, November 15-19.

Matsui, N., Shigyo, M. 1992. Brushless DC Motor Control without Position and Speed Sensors. IEEE Transactions on Industry Applications 28.1: 120-127.

Michalewicz, Z. 1994. Genetic Algorithms + Data structures = Evolution Programs. 2nd ed. Springer-Verlag, New York.

Nekoogar, F., Moriarty, G. 1998. Digital Control Using Digital Signal Processing, Prentice Hall, USA.

Pierre, D.A. 1992. Optimization in McGraw-Hill Encyclopedia of Science and Technology, McGraw-Hill, New York.

Qiau J., Xuan H., Jiang J. 2005. Improvement of Genetic Algorithm and Its Applicaiton in Optimization of fuzzy Traffic Control Algorithm. Springer-Verlag AAIM LNCS 3521: 132-141.

Radcliff, N.J. 1991. Formal analysis and random respectful recombination. In Proc. of Fourth Int. Conf. on Genetic Algorithms, Morgan Kauffman, San Mateo, CA.

Reznik, L. 1997. Fuzzy Controllers, Newnes, Boston, USA.

Rice, J.R. 1983. Numerical Methods, Software and Analysis, McGraw-Hill, New York.

Shao, J. 2003. Direct Back EMF Detection Method for Sensorless Brushless DC (BLDC) Motor Drives. M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Sing B., Jain K. 2003. Implementation of DSP Based Digital Speed Controller for Permanent Magnet Brushless DC Motor. IE(I) Journal-EL. 84: 16-21.

Smith, S.W. 2002, The Scientist & Engineer's Guide to Digital Signal Processing, California Technical Publishing, San Diego, California.

Su G.J., McKeever J.W. 2004. Low-Cost Sensorless Control of Brushless DC Motors With Improved Speed Range. IEEE Transactions on Power Electronics 19: 296-302.

Tan, G.V., Hu, X. 1996. On Designing Fuzzy Controllers using Genetic Algorithms. Proc. of 5th IEEE Int. Conf. on Fuzzy Systems, Vol.2, pp.905-911. New Orleans, September 8-11.

Tang, K.S., Man, K.F., Chen, G., Kwong, S. 2001. An Optimal Fuzzy Controller. IEEE Transactions on Industrial Electronics 48.4: 757-765.

Toliyat H.A., Gopalarathnam T. 2002. AC Machines Controlled as DC Machines (Brushless DC Machines/Electronics). The Power Electronic Handbook. Skvarenina T. L. (ed) pp.78-100, CRC Press LLC, New York.

Wander, G., Paul, P. ve John, W. 2000. Application of Genetic Algorithm to the Online Tuning of Electric Drive Speed Controllers. IEEE Transaction on Industrial Electronics 47.1: 217-219.

Wright, A., 1991. Genetic algorithms for real parameter optimization. Foundations of Genetic algorithms 2. Rawlins G.J.E. (ed), pp. 205-218, Morgan Kaufman, San Mateo, CA.

Wu, C.J., Liu, G.Y. 2000. A Genetic Approach for Simultaneous Design of Membership Functions and Fuzzy Control Rules. Journal of Intelligent and Robotic Systems 28: 195-211.

Xia C., Guo P., Shi T., Wang M. 2004. Speed Control of Brushless DC Motor using Genetic Algorithm Based Fuzzy Controller. Proc. of Int. Conf. on Intelligent Mechatronics and Automation, pp.460-464, Chengdu, August 26-31.

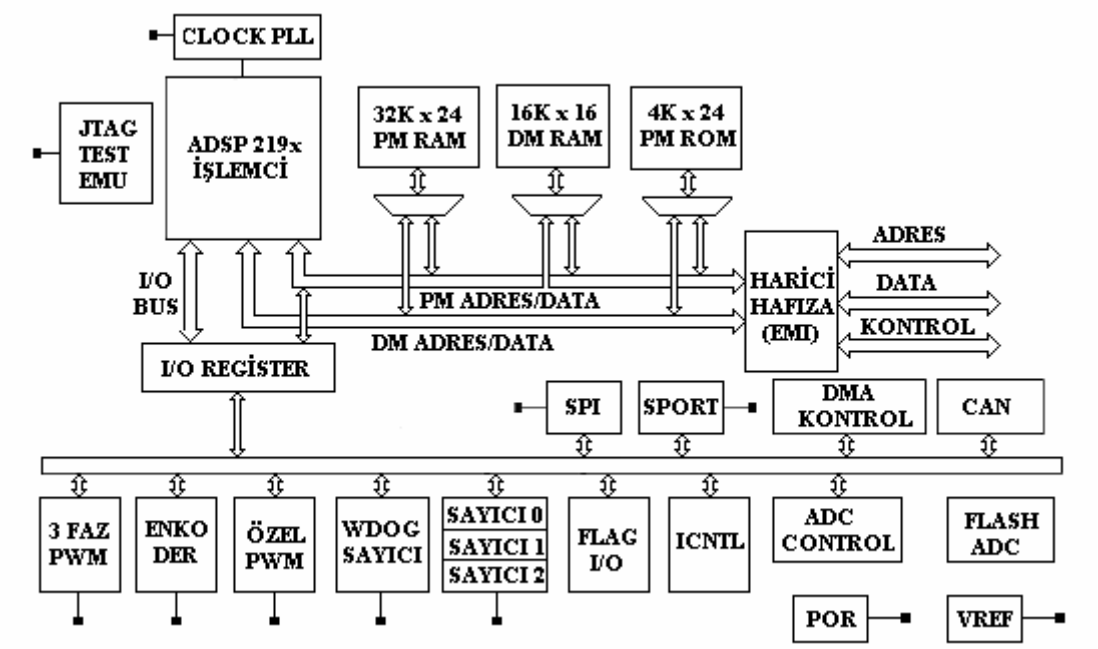
Yedamale, P. 2003. Brushless DC (BLDC) Motor Fundamentals, Microchip Technology Inc. App. Note, USA.

Yun Y., Gen M., Seo S. 2003. Various Hybrid Methods Based on Genetic Algorithm with Fuzzy Logic Controller. Journal of Intelligent Manufacturing 14: 401-419.

Zadeh, L.A. 1965. Fuzzy Sets. Elsevier Information and Control 8.3: 338-353.

EK A: ADSP-21992 İŞLEMCİLER İLE PROJE GELİŞTİRME

ADSP-21992 sayısal işaret işlemci mimarisi Şekil EK-A1'de verilmiştir. İşlemci üç adet bağımsız hesaplama ünitesine sahiptir. Bunlar; aritmetik-lojik birim (ALU), çarpma-akümülatör (MAC) ve kaydırma (shifter) birimleridir. ADSP-21992'nin hafıza haritası Tablo EK-A1'de verilmiştir.



Şekil EK-A1 ADSP-21992 mimarisi

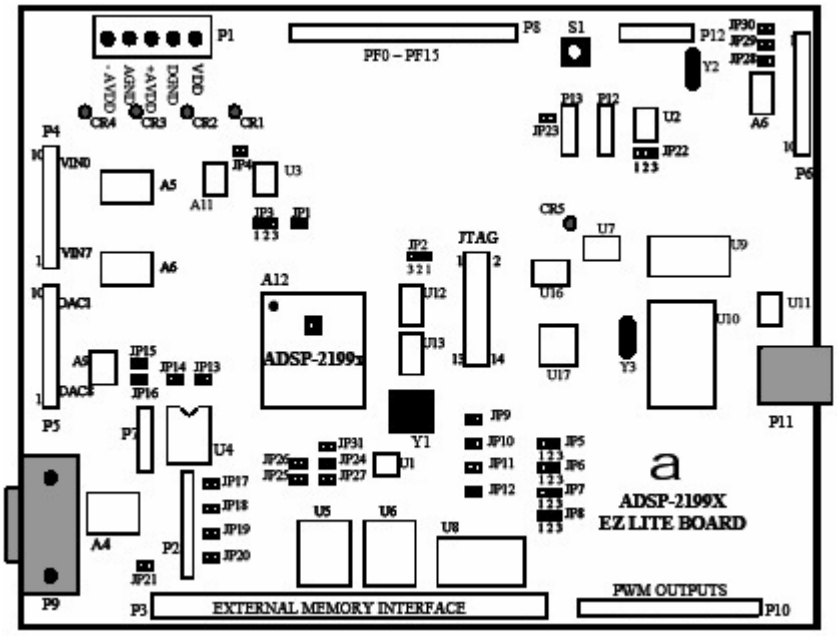
Tablo EK-A1 ADSP-21992 bellek haritası

	Başlangıç Adresi	**Bitiş Adresi**	**Sayfa**	**İçerik**
Dahili Bellek	0x00 0000	0x00 7FFF	0	32K x 24-Bit Program Hafıza
	0x00 8000	0x00 BFFF	0	16K x 16-Bit Data Hafıza
	0x00 C000	0x00 FFFF	0	Boş (16K)
Harici Bellek	0x01 0000	0x3F FFFF	1-63	Harici Hafıza (Bank 0)
	0x40 0000	0x7F FFFF	64-127	Harici Hafıza (Bank 1)
	0x80 0000	0xBF FFFF	128-191	Harici Hafıza (Bank 2)
	0xC0 0000	0xFE FFFF	192-254	Harici Hafıza (Bank 3)
Boş	0xFF 0000	0xFF 0FFF	255	4K x 24-Bit Boot ROM (Dahili)
	0xFF 1000	0xFF FFFF	255	Boş (Dahili)
I/O Bellek	0x00 0000	0x00 1FFF	0-7	16-Bit Dahili I/O Hafıza
	0x00 2000	0x03 FFFF	8-255	16-Bit Harici I/O Hafıza

Şekil EK-A1'den görüldüğü gibi, işlemcide etkili data transferi için; program hafızası adres yolu (PMA), program hafızası data yolu (PMD), data hafızası adres yolu (DMA), data hafızası data yolu (DMD), doğrudan hafıza erişim adres yolu ve doğrudan hafıza erişim data yolu kullanılır (Analog Devices 2001).

ADSP-21992 Ez-Kit Lite geliştirme kartı

ADSP-21992 DSP'nin tüm özelliklerini kolayca kullanabilmek için Analog Devices tarafından ADSP-21992 Ez-Kit Lite deney kartı üretilmiştir. Şekil EK-A2 de ADSP-21992 Ez-Kit Lite geliştirme kartının genel yapısı görülmektedir.



Şekil EK-A2 ADSP-21992 Ez-Kit Lite geliştirme kartı

ADSP-21992 Ez-Kit Lite kartının temel özellikleri aşağıdaki gibi sıralanabilir.

- Üzerinde ADSP-21992 160 MHz sabit noktalı DSP entegresi,
- SPI üzerinde 8 kanal 12 bit AD5328BRU DAC,
- ±2.5 ve ±3.3 volt gerilim regülatörleri,
- DSP osilatör devresi,
- USB, PWM, genel amaçlı I/O, RS-232 UART ve harici bellek arabirimleri,
- CAN ve enkoder arabirim devreleri,

- 512K x 8 Flash bellek, harici 64K x 16 SRAM hafıza,
- Analog sinyal devreleri,
- JTAG arabirimi için 14 uçlu emülatör bağlantısı,
- Giriş-çıkış bağlantı soketleri,
- Konfigürasyon anahtarları.

ADSP-21992 Ez-Kit Lite kartı, kullanıcıya ait özel çalıştırma kodlarını yüklemek için flash bellek devresine sahiptir. Anahtar konfigürasyonları ile seçilen çalıştırma kipi sayesinde, geliştirme kartı bilgisayar olmadan tek başına çalıştırılabilmektedir. Ek olarak, harici bellek arabirimi (EMI) ile bellek kapasitesi artırılmaktadır. Sistemde, DSP gerilimi 2.5 volt ve çevre birimler 3.3 volt ile çalıştırılabilmektedir. ADSP-21992 Ez-Kit Lite deney kartında DSP'ye yeniden başlat (reset) üretmek için birkaç yol vardır. Bunlar; açılış (power on reset-POR), yeniden başlat düğmesi, harici yeniden başlat, yazılım ile yeniden başlat işlemleridir.

VisualDSP++ görsel yazılım programı

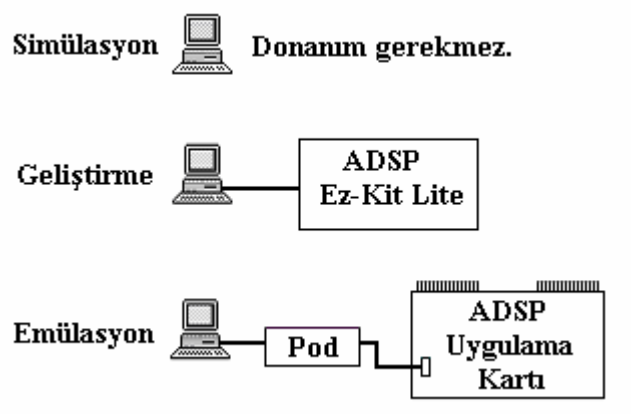
VisualDSP++, ADSP sayısal işaret işlemcilerin etkili bir şekilde programlanması, simülasyon ile denenmesi ve yazılım hatalarının düzeltilmesi için gereken tüm araçlara sahip görsel bir yazılım programıdır. VisualDSP++ yazılım programı ile; makine dilinde veya C/C++ dilinde program yazma, yazılan tüm programları derleme, program içerisinde hata ayıklama, DSP üzerindeki kaydedicileri gerçek zamanda takip edebilme ve grafik çizebilme işlemleri görsel özellikleri kullanarak kolayca yapılmaktadır. VisualDSP++ programı; Windows XP, Windows 2000, Windows NT ve Windows 98 işletim sistemlerinde Blackfin DSP, TigerSHARC DSP, SHARC DSP, ADSP-218x DSP, ADSP-219x DSP ve ADSP-2199x DSP ailesinin tamamını desteklemektedir. VisualDSP++ programı aşağıda verilen özelliklere sahiptir.

- Geliştirilmiş program yazma kapasitesi sayesinde C/C++ veya makine dili ortamlarında kaynak dosya oluşturma veya değişiklik yapmak mümkündür.
- Esnek bir proje tasarımı sunmaktadır. Görsel öğeler kullanılarak yeni proje oluşturmak, eski projelerde değişiklik yapmak mümkündür.

- Kod geliştirme araçlarına erişim kolaydır. VisualDSP++ bünyesinde; C/C++ derleyici, VIDL derleyici, makine dili (assembler), bağlayıcı (linker), dallandırıcı (splitter) ve yükleyici (loader) bulunmaktadır.
- VisualDSP++ programı Kernel (VDK) desteği sunar. VDK penceresinde kernel nesnelerine kolayca erişim ve yeni nesne oluşturmak mümkündür.
- Esnek bir proje derleme seçeneğine sahiptir. Dosya veya proje seviyesinde derleme sunmaktadır. Derleme işleminin aşamaları gözlenebilmektedir. Derleme hata verirse, hata noktalarını göstermektedir.
- Program yazma işlemi C, C++ veya makine dillerinden biriyle veya karışık olarak yapılabilir. C dilinde yazılan program makine dilinde görülebilir.
- Programda hata ayıklama oldukça kolay ve etkilidir. Program adım adım yürütülerek veya kısım kısım çalıştırılarak hata tespiti yapılabilmektedir. Register, yığın ve hafıza bölgeleri gözlenebilmektedir.
- Data dizileri grafiksel olarak çizdirilebilmekte, kesme üreterek giriş ve çıkışlar gözlenmektedir.

VisualDSP++ programında proje geliştirme aşamaları

Tipik bir DSP yazılım projesi üç aşamadan meydana gelir. Bunlar; simülasyon, geliştirme (evaluation) ve emülasyondur. Şekil EK-A3'de proje geliştirme aşamaları görülmektedir.



Şekil EK-A3 Proje geliştirme aşamaları

Yukarıdaki şekilde görüldüğü gibi simülasyon işleminde, herhangi bir DSP donanımına ihtiyaç yoktur. Bu aşamada donanım mühendisi yeni DSP donanımını tasarlarken, yazılım mühendisi DSP yazılımı için simülasyon ortamında çalışma yapabilir. Simülasyon işlemi, kullanılacak hedef sistemdeki DSP'ye ait bellek ve I/O özelliklerini tamamen modeller. Simülasyon işleminde program yazılıp çalıştırılabilir ve hata ayıklaması yapılabilir. Geliştirme işleminde, Ez-Kit Lite geliştirme modülleri kullanılır. Yapılan programlar derlenip gerçek DSP üzerine yüklenir. Gerçek zamanda sistem davranışı gözlenebilir. Emülasyon aşamasında, geliştirilen donanım ile bir JTAG emülatör vasıtasıyla iletişim sağlanır ve programlar DSP üzerine yüklenir. DSP performansı gözlenebilir.

Program geliştirme adımları

VisualDSP++ programının ilk açılış çalışma sayfası Şekil EK-A4'de görülmektedir. VisualDSP++ ortamında program geliştirmek için aşağıdaki adımlar takip edilmelidir.

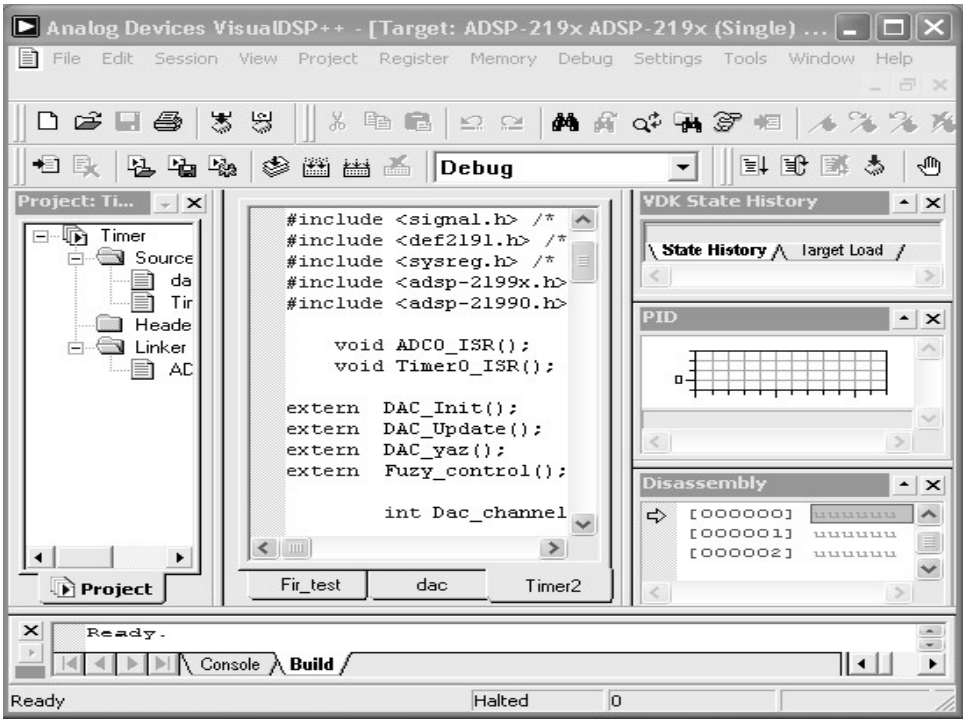
1. VisualDSP++ ortamında yapılan her şey bir projenin parçasıdır. Bu nedenle başlangıçta yeni proje oluşturulmalıdır (Şekil EK-A5). Proje dosyalarında (.DPJ) kaynak dosya listeleri, proje özellikleri gibi program yapı bilgileri bulunur.

2. Proje özelliklerinin belirlenmesi için projede kullanılacak DSP ve proje özellikleri "Project option" iletişim kutusundan seçilmelidir (Şekil EK-A6).

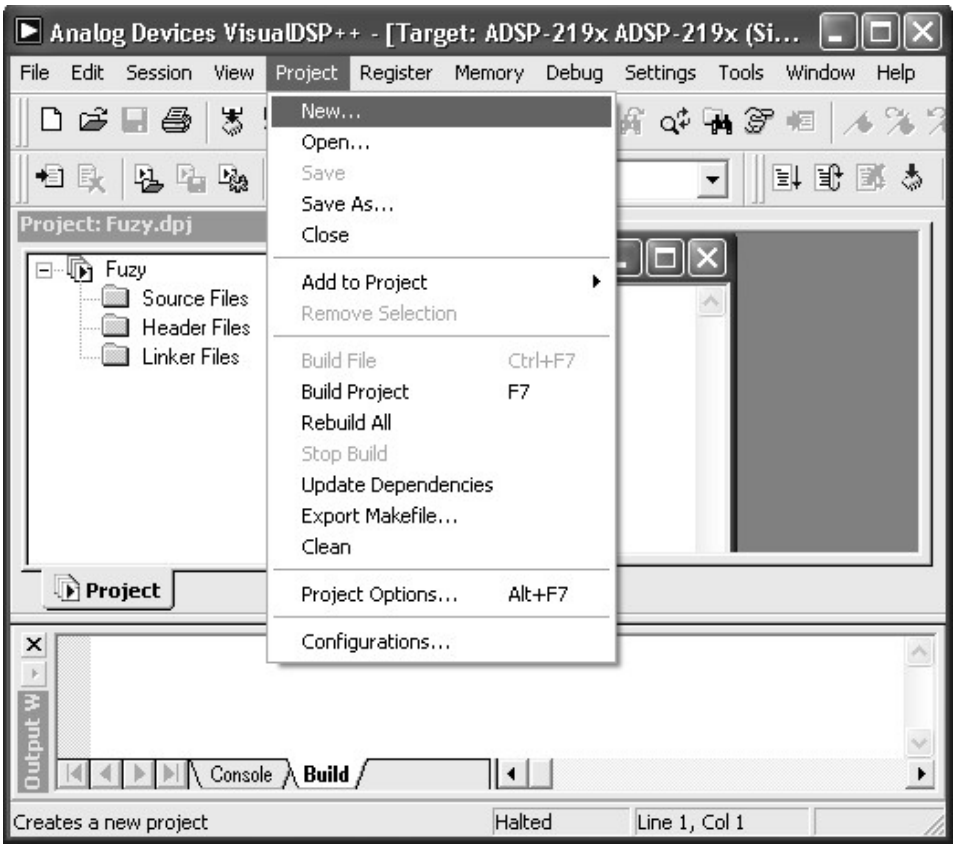
3. Şekil EK-A7'de görüldüğü gibi projeye önceden yazılmış kaynak dosyalar ilave edilir ya da C, C++ veya makine dillerinde yeni kaynak dosyalar yazılır. VisualDSP++ editör penceresi yeni kaynak dosya oluşturmak için kullanılır.

4. Proje oluşturulduktan sonra, ADSP-21992 makine dilinde derlenmelidir. Şekil EK-A8'de derleme işlemi görülmektedir. Ayrıca burada hata ayıklaması yapılabilir.

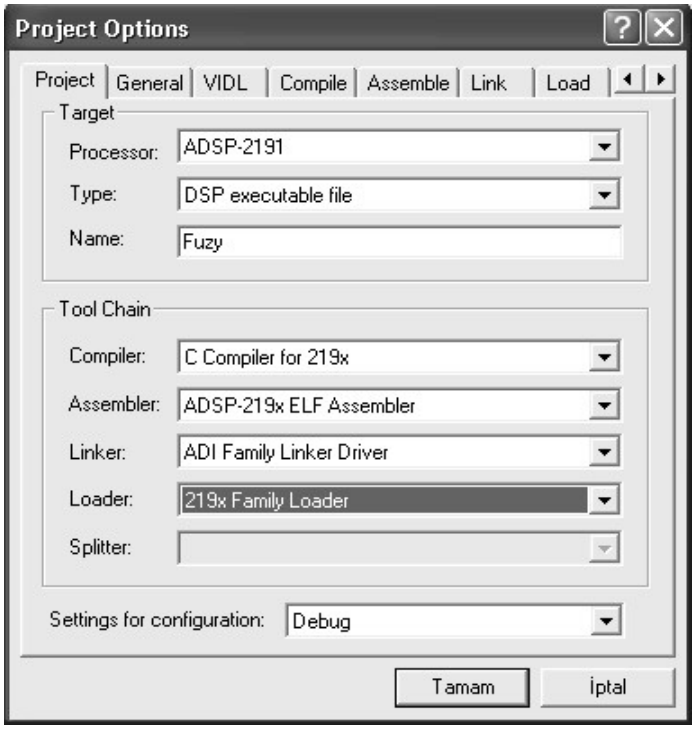
5. Derlenen program yürütülmek üzere ADSP'ye yüklenir ve program çalıştırılır. (Şekil EK-A9)



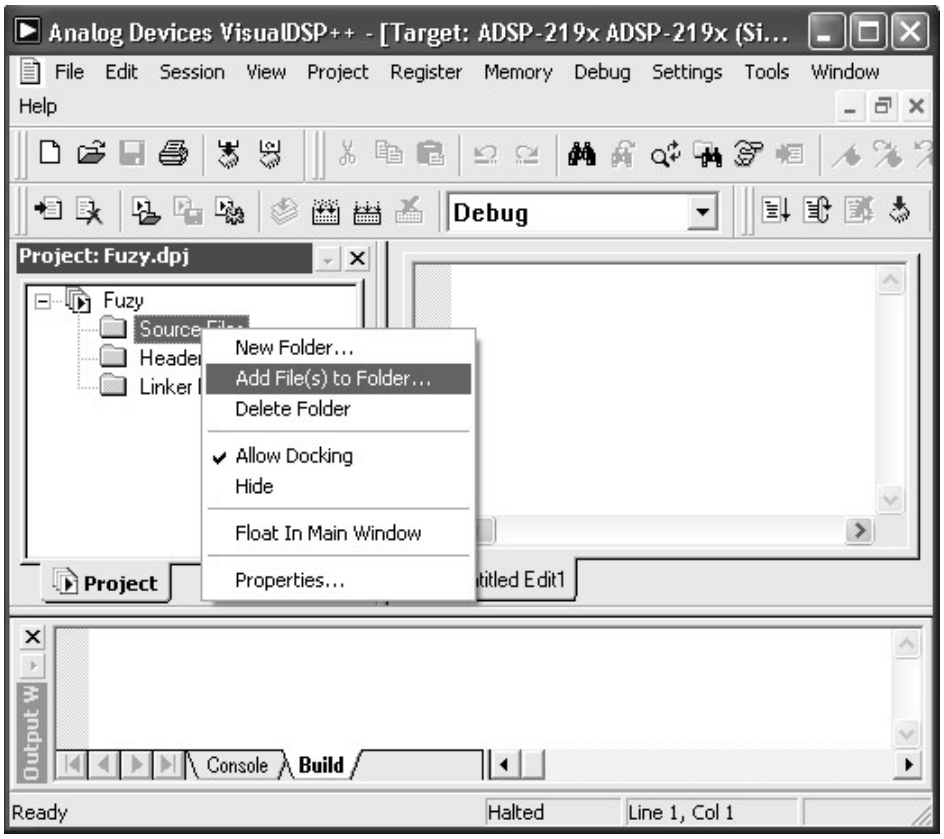
Şekil EK-A4 VisualDSP++ yazılım programı çalışma sayfası



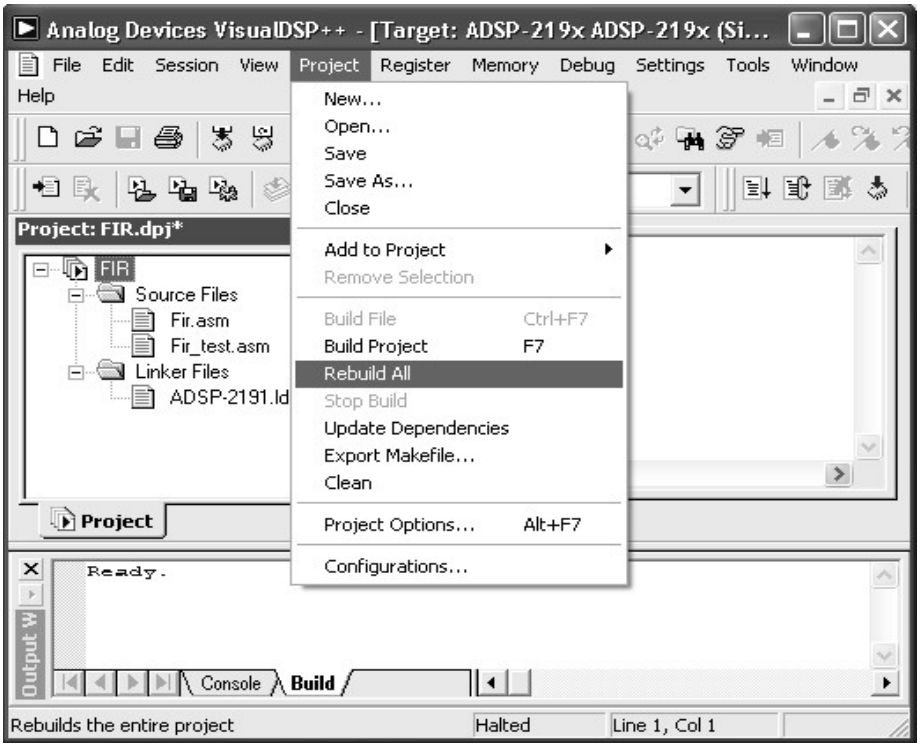
Şekil EK-A5 Yeni proje oluşturma işlemi



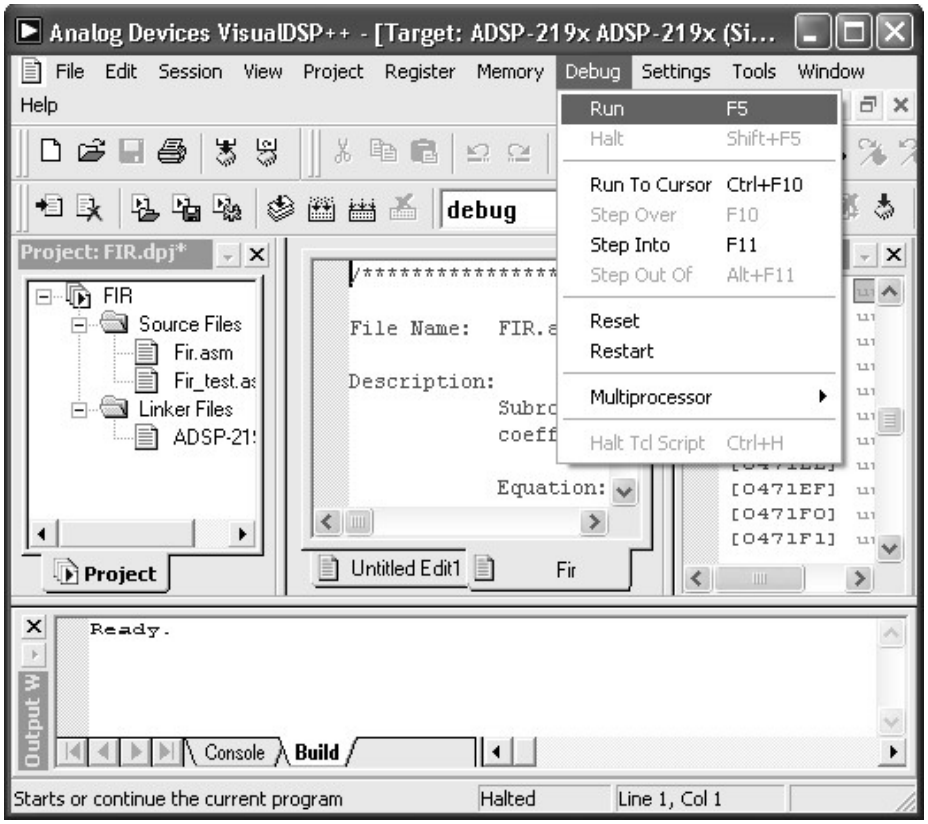
Şekil EK-A6 Proje özelliklerinin belirlenmesi



Şekil EK-A7 Projeye kaynak dosyaların ilave edilmesi



Şekil EK-A8 Projenin derlenmesi



Şekil EK-A9 Projenin çalıştırılması

EK B: BULANIK DENETLEYİCİYİ GENETİK ALGORİTMA İLE OPTİMİZE EDEN SİMÜLASYON PROGRAMI

Bulanık denetleyiciyi genetik algoritma ile optimize etmek amacıyla MATLAB ortamında bir program geliştirilmiştir. Bu program alt programlar ile beraber çalışmaktadır. Bu alt programlar; optimizasyon programı, performans indeksinin yani uygunluk değerlerinin hesaplandığı kapalı çevrim BLDC motor bulanık denetleyici programıdır. Aşağıda, parametrelerin tanımlandığı programın giriş kısmı verilmiştir.

```
% Program adı : GA_Fuzzy_BLDC_24.m
% Programcı   : ÖA
% Tarih       : 4 Ekim 2005
% Yenileme    :
% Amacı       : BLDC motorun GA tabanlı Optimal Bulanık Kontrolü

clear                   % Hafıza temizlenir
rand('state',0)         % Random sayı üreteci kurulur.


% Kullanıcı aşağıya optimizasyon için gerekli parametreleri girer

NUM_TRAITS=3;                   % Değişken sayısı
HIGHTRAIT=[30 30 30];           % Değişken üst sınırları
LOWTRAIT=[0 0 0];               % Değişken alt sınırları
SIG_FIGS=[6 6 6];               % Değişken için gen sayısı
DECIMAL=[2 2 2];                % Değişken tam kısım
MUTAT_PROB=0.05;                % Mutasyon olasılığı (tipik <0.1)
CROSS_PROB=0.8;                 % Çaprazlama olasılığı (tipik yaklaşık 0.8)
SELF_ENTERED=0;                 % "0": rasgele başlangıç populasyonu
                                % "1": kullanıcı tar. Girilen başlangıç populasyonu
POP_SIZE=20;                    % Populasyon içindeki birey sayısı
ELITISM=1;                      % Elitizm ON/OFF, 1/0
DELTA=1000;                     % Durdurma için minimum nesil sayısı

EPSILON = 0.01;                 % Durdurma için
                                % minimum uygunluk değeri
MAX_GENERATION=1000;            % Döngü sayısı
                                % Durdurma için nesil sayısı


% Başlangıç populasyonunun oluşturulması

popcount=1;                     % Başlangıçta nesil sayısı
                                % 1 olarak atanır.
if SELF_ENTERED == 0            % 10 tabanında rasgele başlangıç populasyonu
                                % oluşturuluyor

for pop_member = 1:POP_SIZE
    for current_trait = 1:NUM_TRAITS,
                trait(current_trait,pop_member,popcount)=...
```

```
                (rand-(1/2))*(HIGHTRAIT(current_trait)-LOWTRAIT(current_trait))+...
                (1/2)*(HIGHTRAIT(current_trait)+LOWTRAIT(current_trait));
    end
end
else

   for pop_member = 1:POP_SIZE
      for current_trait = 1:NUM_TRAITS,
         trait(current_trait,pop_member,popcount)=0;
      end
   end
end


% Kromozom uzunluklarının hesaplanması
% Değişkenlerin kromozom içindeki yerlerinin bulunması

CHROM_LENGTH=sum(SIG_FIGS)+NUM_TRAITS;

                                    % Kromozom uzunlu, işaret sayısı da
                                    % eklenerek bulunur.

TRAIT_START(1)=1;                   % İlk kromozom işaret
                                    % sayısından başlanır.

for current_trait=1:NUM_TRAITS,     % Diğer kromozomların işaret
                                    % sayısı yerleri tespit edilir.


TRAIT_START(current_trait+1)=...
  TRAIT_START(current_trait)+SIG_FIGS(current_trait)+1;

end
```

EK C: BULANIK DENETLEYİCİ DSP YAZILIM KODU

Bulanık denetleyici ile fırçasız DC motoru kontrol eden bu program, C dilinde yazılarak, derleyici ile ADSP-21992 makine dilinde derlenmiştir. Programda, ADC elemanlardan alınan motor hız bilgisi bulanık denetleyiciye uygulanarak, denetim için gerekli referans akım üretilmektedir. Referans akım bilgisi kullanılarak motor için kontrol sinyalleri belirlenmektedir. Burada sadece ana program verilmiştir. Alt programlar ve başlık dosyalar verilmemiştir.

```
/*********************************************************************
BLDC motorun bulanık denetleyici ile sensörsüz kontrolü:
ADSP-21992 EZ-KIT
Tarih 06/01/2006
Tasarım Omer Aydogdu - OA */
*********************************************************************/
#include <signal.h> /* Interrupts */
#include <def2191.h> /* MMRs */
#include <sysreg.h> /* Intrinsics */
#include <adsp-2199x.h> /* Intrinsics */
#include <adsp-21990.h> /* Intrinsics */
#include "main.h"

        void ADC0_ISR();
        void Timer0_ISR();
extern  DAC_Init();
extern  DAC_Update();
extern  DAC_yaz();
extern  Fuzzy_control();
int Dac_channel_0;

main()
{       int temp;
/* Sistem registerlerinin ayarlanması*****************************************************/
        sysreg_write(sysreg_IOPG, Clock_and_System_Control_Page);
        temp = io_space_read(SYSCR);
        temp |= 0x0011;                         /* Map IVT To Page 0 */
        io_space_write (SYSCR, temp);
/* Tüm Interruptların resetlenmesi  ***********************************/
        disable_interrupts();
        sysreg_write(sysreg_IRPTL, 0x0000);
        sysreg_write(sysreg_ICNTL, 0x0000);
        sysreg_write(sysreg_IMASK, 0x0000);
/*Timer0_ISR için 0.interrupt, ADC0_ISR için 1. interrupt **********************/
        sysreg_write(sysreg_IOPG, Interrupt_Controller_Page);
        io_space_write(IPR5, 0xFFF0);
        io_space_write(IPR3, 0x1FFF);
        temp=sysreg_read(sysreg_IMASK);
        temp |= 0x0030;
        sysreg_write(sysreg_IMASK, temp);
/* Timer Sayfasına git – Initialize Timer0 *************************/
        sysreg_write(sysreg_IOPG, Timer_Page);
        io_space_write(T_CFGR0, 0x001D);
```

```
/* Timer0 Period Register (High Word) *************************/
        io_space_write(T_PRDH0, PRDHigh);
/* Timer0 Period Register (Low Word) *************************/
        io_space_write(T_PRDL0, PRDLow);
/* Timer0 Width Register (High Word) *************************/
        io_space_write(T_WHR0, WidthH);
/* Timer0 Width Register (Low Word) *************************/
        io_space_write(T_WLR0, WidthL);
/* Enable Timer0 ********************************************/
        io_space_write(T_GSR0, 0x0100);


        DAC_Init();

/*ADC'nin ayarı*********************************************************/
                sysreg_write(sysreg_IOPG, ADC_Page);
                io_space_write(ADC_CTRL, 0x0207);           /*for softconversation start*/
                //io_space_write(ADC_CTRL, 0x0207);
/* İnterrupt atamaları *************************************************/
                interrupt(SIG_INT4, Timer0_ISR);
                interrupt(SIG_INT5, ADC0_ISR);
/* Globally Enable Interrupts ******************************************/
                enable_interrupts();
                while(1);                   /* wait for interrupts */

}                       /* end of main */


        void Timer0_ISR()
        {
                sysreg_write(sysreg_IOPG, Timer_Page);      /* Go To Timer I/O Page */
                io_space_write(T_GSR0, 0x1);            /* Clear TMR0 Interrupt Latch Bit */
                sysreg_write(sysreg_IOPG, ADC_Page); /*ADC softconversation start*/
                io_space_write(ADC_SOFTCONVST, 0x1); /*Set ADC conversation Bit */
        }               /* end Timer0_ISR **/
                void ADC0_ISR()
        {       //ADC Oku
                sysreg_write(sysreg_IOPG, ADC_Page);        //ADC sayfasına git
                Dac_channel_0 = io_space_read(ADC_DATA0);//ADC0 bilgisini oku
                io_space_write(ADC_STAT, 0x0100);   //ADC interrupt sil
                Fuzzy_control();                    //kontrol out üret
                DAC_Update();                       //out kontrol işaretini çıkışa aktar
        }
```