T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BİR FIRÇASIZ DOĞRU AKIM MOTORUNUN MODELLENMESİ VE KONTROLÜ

Mert AKSÖZEK

DANIŞMAN
Dr. Öğr. Üyesi Nevra BAYHAN

Elektrik- Elektronik Mühendisliği Anabilim Dalı

Elektrik-Elektronik Mühendisliği, Tezli Yüksek Lisans Programı

Aralık, 2024

TEZ KABUL VE ONAYI

Mert AKSÖZEK tarafından, **Dr. Öğr. Üyesi Nevra BAYHAN** danışmanlığında hazırlanan **"BİR FIRÇASIZ DOĞRU AKIM MOTORUNUN MODELLENMESİ VE KONTROLÜ"** başlıklı bu çalışma, jürimiz tarafından **26/12/2024** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Dr. Öğr. Üyesi Nevra BAYHAN İstanbul Üniversitesi-Cerrahpaşa Elektrik-Elektronik Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Doç. Dr. Rana ORTAÇ KABAOĞLU İstanbul Üniversitesi-Cerrahpaşa Elektrik-Elektronik Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Dr. Öğr. Üyesi Fatih ADIGÜZEL Yıldız Teknik Üniversitesi Kontrol ve Otomasyon Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret

Hayatım boyunca bana maddi ve manevi desteğini esirgemeyen annem, babam ve kardeşime ithaf ediyorum…

BÜTÇE DESTEKLERİ

BİR FIRÇASIZ DOĞRU AKIM MOTORUNUN MODELLENMESİ VE KONTROLÜ

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Tez çalışmamın her aşamasında bilgisi, tecrübeleri ve fikirleriyle bana yol gösteren değerli danışman hocam Sayın Dr. Ögr. Üyesi Nevra BAYHAN'a, ve desteklerinden dolayı Arş. Gör. Dr. Yılmaz Seryar ARIKUŞU'na en içten teşekkürlerimi sunarım.

Ayrıca bana her zaman destek olan aileme ve arkadaşlarıma çok teşekkür ederim.

Aralık 2024 Mert AKSÖZEK

İÇİNDEKİLER

Sayfa No

TEZ KABUL VE ONAYI ..... **ii**
BEYAN ..... **iii**
BÜTÇE DESTEKLERİ ..... **v**
TEŞEKKÜR ..... **vi**
İÇİNDEKİLER ..... **vii**
ŞEKİL LİSTESİ ..... **ix**
TABLO LİSTESİ ..... **xiii**
SİMGE VE KISALTMA LİSTESİ ..... **xv**
ÖZET ..... **xviii**
ABSTRACT ..... **xx**
1. GİRİŞ **1**
2.KAVRAMSAL ÇERÇEVE **5**
2.1. ELEKTRİK MOTORLARI 5
2.1.2. Elektrik Motorlarının Kontrol Uygulamalarında Kullanım Yerleri 5
2.1.3. Elektrik Motorlarının Çalışma Prensibi 6
2.2. DOĞRU AKIM (DC) MOTORLAR 6
2.2.1. Seri Bağlı DC Motor 7
2.2.2. Şönt Bağlı DC Motor 7
2.2.3. Bileşik (Compound) DC Motor 7
2.2.4. Kalıcı Mıknatıslı DC Motor 7
2.2.5. Fırçasız DC Motorlar 8
2.2.6. DC Motorların Özelliklerinin Karşılaştırılması 9
2.3. DC MOTORUN MODELLENMESİ 10
2.4. FIRÇASIZ DC MOTORUN MATLAB SİMÜLASYON ORTAMINDA MODELLENMESİ 11
2.5. FIRÇASIZ DC MOTORUN TRANSFER FONKSİYONUN OLUŞTURULMASI 14
2.6. FIRÇASIZ DC MOTORUN KONTROLÜ 16
2.6.1. PID Kontrol 17
2.6.2. PI Kontrol 18
2.6.3. Bulanık Mantık Kontrol (Fuzzy Logic Control) 19
3. YÖNTEM **22**

3.1. METASEZGİSEL OPTİMİZASYON ALGORİTMALARI 22
3.1.1. Genetik Algoritma (GA) 23
3.1.2. Parçacık Sürü Optimizasyon Algoritması (PSO) 26
3.1.3. Yapay Arı Kolonisi Optimizasyon Algoritması (ABC) 29
3.1.4. Çiçek Tozlaşma Optimizasyon Algoritması (FPA) 32
3.1.5. Gri Kurt Optimizasyon Algoritması (GWO) 35
3.1.6. Yarasa Optimizasyonu Algoritması (BA) 38
3.1.7. Yusufcuk Algoritması(DA) 40
3.1.8. Atom Arama Optimizasyon Algoritması (ASO) 44
3.1.9. Su Döngüsü Algoritması (WCA) 47
3.2. KLASİK KONTROL YÖNTEMİ 51
3.2.1. Ziegler Nichols Yöntemi 51
3.2.2. Ziegler Nichols Yönteminin Fırçasız DC Motora Uygulanması 52
3.3. METASEZGİSEL ALGORİTMALARIN FIRÇASIZ DC MOTORA UYGULANMASI 53
3.3.1. Metasezgisel Algoritmalarla Optimize Edilmiş PID Kontrolörün Fırçasız DC Motor Modeline Uygulanması 55
3.3.2. Metasezgisel Algoritmalarla Optimize Edilmiş PI Kontrolörün Fırçasız DC Motor Modeline Uygulanması 55
3.3.3. Metasezgisel Algoritmalarla Optimize Edilmiş PI Tabanlı FGPI Kontrolörün Fırçasız DC Motor Modeline Uygulanması 56
4. BULGULAR 61
4.1. PID KONTROLÖR İÇİN SONUÇLAR 61
4.2. PI KONTROLÖR İÇİN SONUÇLAR 74
4.3. BULANIK MANTIK PI (FGPI) KONTROLÖR İÇİN SONUÇLAR 86
5. TARTIŞMA 99
6. SONUÇ VE ÖNERİLER 102
KAYNAKLAR 105
İNTİHAL RAPORU İLK SAYFASI 110
ETİK KURUL İZİN YAZISI 111
KURUM İZNİ YAZILARI 112
PATENT HAKKI İZNİ 113
ÖZGEÇMİŞ 114

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Örnek bir elektrik motoru 5

Şekil 2.2: Fırçasız DC elektrik motorunun temel **e**lektrik devre şeması [33] 10

Şekil 2.3: DC motor blok diyagramı [34] 10

Şekil 2.4: DC motorun MATLAB Simulink ortamında modeli [35] 12

Şekil 2.5: MATLAB ortamında sistem tanımlama arayüz şeması [30,35] 14

Şekil 2.6: MATLAB ortamında sistem tanımlama doğruluk değerleri [35] 15

Şekil 2.7: MATLAB ortamında motorun sistem tanımlama transfer fonksiyonu [35] 16

Şekil 2.8: PID kontrol blok diyagramı [14, 16] 17

Şekil 2.9: PI kontrol blok diyagramı [14, 16] 18

Sekil 2.10: Birim basamak giriş uygulanmış bir sistemin çıkış eğrisi ve performans kriterleri [16,17] 19

Sekil 2.11: Bulanık mantık kontrolör temel çalışma yapısı[68] 19

Şekil 2.12: MATLAB ortamında bulanık mantık çıkarımı [15, 35] 20

Şekil 2.13: MATLAB ortamında üyelik fonksiyonu grafiği [15,35] 21

Şekil 3.1: Tez çalışmasında kullanılan metasezgisel optimizasyon algoritmaları şeması 23

Şekil 3.2: Gen dizilimi 23

Şekil 3.3: Genetik algoritma akış diyagramı [19] 25

Şekil 3.4: Kuş sürüsü 26

Şekil 3.5. PSO akış diyagramı [20] 28

Şekil 3.6: Arı sürüsü 29

Şekil 3.7 ABC algoritması akış diyagramı [21] 31

Şekil 3.8: Çicek tozlaşması [50] 32

Şekil 3.9: FPA akış diyagramı [22,50] 34

Şekil 3.10: Gri kurt ve hiyerarşisi 35

Şekil 3.11: GWO akış diyagramı [23] 37

Şekil 3.12: Yarasanın avını bulması....38

Şekil 3.13: Yarasa algoritması akış diyagramı [24,54]....40

Şekil 3.14: Yusufcukların birbirleriyle hizalanması....41

Şekil 3.15: Yusufçuk algoritması akış diyagramı [25, 57]....44

Şekil 3.16: Atomların birbirleriyle olan etkileşimi....45

Şekil 3.17: Atom arama algoritması akış diyagramı [26,58]....47

Şekil 3.18: Yağmur damlacıkların nehire düşmesi....48

Şekil 3.19: Su döngüsü algoritması akış diyagramı [27,60]....51

Şekil 3.20: Ziegler Nichols yönteminin fırçasız DC motor sistemine uygulanması [64]....52

Şekil 3.21: Metasezgisel algoritmalarla optimize edilmiş kontrolörlerin uygulandığı bir fırçasız DC motor modelinin sistem blok diyagramı [48,65]....54

Şekil 3.22: Optimize edilmiş FG-PI kontrolörlü fırçasız dc motor sisteminin blok diyagramı [15, 36]....57

Şekil 3.23.a: Motor hız hatası giriş üyelik fonksiyonlarının aralığı[35]....58

Şekil 3.23.b: Motor hızınn değişim hatası üyelik fonksiyonlarının aralığı[35]....58

Şekil 3.23.c: Motor çıkışındaki ΔKp üyelik fonksiyonlarının aralığı[35]....59

Şekil 3.23.d: Motor çıkışındaki ΔKi üyelik fonksiyonlarının aralığı[35]....59

Şekil 4.1: Optimize edilmiş PID kontrolörler uygulanmış bir fırçasız DC motor sisteminin (0-4) saniye aralığında cevap eğrileri....61

Şekil 4.2: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin pozitif bölgedeki yükselme zamanları (%10-%90 kriterine göre)....63

Şekil 4.3: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin negatif bölgedeki yükselme zamanları (%10-%90 kriterine göre)....64

Şekil 4.4: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin pozitif bölgedeki yerleşme zamanları (%2 bandına göre)....65

Şekil 4.5: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin negatif bölgedeki yerleşme zamanları (%2 bandına göre)....66

Şekil 4.6: Metasezgisel algoritmaların optimize edilmiş PID kontrolöre sahip sisteme ilişkin pozitif bölgedeki yüzde aşım değerleri....67

Şekil 4.7: Metasezgisel algoritmaların optimize edilmiş PID kontrolöre sahip sisteme ilişkin negatif bölgedeki yüzde aşım değerleri....68

Şekil 4.8: Ziegler-Nichols kontrol yöntemi ile optimize edilmiş PID kontrolöre ilişkin pozitif ve negatif bölgelerdeki yerleşme zamanları (%2), yükselme zamanları (%10-%90) ve yüzde aşım değerleri 69

Şekil 4.9: DA,WCA,GWO,GA,PSO metasezgisel algoritmalarının PID kontrolör yakınsama eğrileri..... 72

Şekil 4.10: ASO, ABC, BA, FPA metasezgisel algoritmalarının PID kontrolör yakınsama eğrileri..... 73

Şekil 4.11: Optimize edilmiş PI kontrolörler uygulanmış bir fırçasız DC motor sisteminin (0-4) saniye aralığında cevap eğrileri..... 74

Şekil 4.12: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yükselme zamanları (%10-%90 kriterine göre) 76

Şekil 4.13: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin negatif bölgedeki yükselme zamanları (%10-%90 kriterine göre)..... 77

Şekil 4.14: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yerleşme zamanları (%2 bandına göre)..... 78

Şekil 4.15: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin negatif bölgedeki yerleşme zamanları (%2 bandına göre) 79

Şekil 4.16: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yüzde aşım değerleri..... 80

Şekil 4.17: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin negatif bölgedeki yüzde aşım değerleri 81

Şekil 4.18: Ziegler-Nichols kontrol yöntemi ile optimize edilmiş PI kontrolöre ilişkin pozitif ve negatif bölgelerdeki yerleşme zamanları (%2), yükselme zamanları (%10-%90) ve yüzde aşım değerleri 82

Şekil 4.19: DA,WCA,GWO,GA,BA metasezgisel algoritmalarının PI kontrolör yakınsama eğrileri..... 84

Şekil 4.20: ASO, ABC, PSO, FPA metasezgisel algoritmalarının PI kontrolör yakınsama eğrileri..... 85

Şekil 4.21: Optimize edilmiş FGPI kontrolörler uygulanmış bir fırçasız DC motor sisteminin 0-2 saniye aralığında cevap eğrileri..... 86

Şekil 4.22: Optimize edilmiş FGPI kontrolörler uygulanmış bir fırçasız DC motor sisteminin 2-4 sn aralığında cevap eğrileri 87

Şekil 4.23: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yükselme zamanları (%10-%90 kriterine göre)..... 89

Şekil 4.24: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin negatif bölgedeki yükselme zamanları (%10-%90 kriterine göre) 90

Şekil 4.25: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yerleşme zamanları (%2 bandına göre) 91

Şekil 4.26: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin negatif bölgedeki yerleşme zamanları (%2 bandına göre) 92

Şekil 4.27: Metasezgisel algoritmaların optimize edilmiş FGPI kontrolöre ilişki pozitif bölgedeki aşım değerleri 93

Şekil 4.28: Metasezgisel algoritmaların optimize edilmiş FGPI kontrolöre ilişkin negatif bölgedeki aşım değerleri 94

Şekil 4.29: Ziegler-Nichols yöntemi ile optimize edilmiş FGPI kontrolörün uygulandığı sisteme ilişkin pozitif bölgedeki yerleşme zamanı (%2), yükselme zamanı (%10-%90) ve yüzde aşım değerleri 95

Şekil 4.30: Ziegler-Nichols yöntemi ile optimize edilmiş FGPI kontrolörün uygulandığı sisteme ilişkin negatif bölgedeki yerleşme zamanı (%2), yükselme zamanı (%10-%90) ve yüzde aşım değerleri 96

TABLO LİSTESİ

Sayfa No

Tablo 2.1. DC motorlarının özelliklerinin karşılaştırılması [1-5] 9

Tablo 2.2: Fırçasız DC motorun seçilen parametre değerleri [31,32] 11

Tablo 2.3: MATLAB ortamında bulanık mantık kural örneği [30, 32] 21

Tablo 3.1: Ziegler Nichols kazanç parametreleri hesaplama formülleri [13, 63] 52

Tablo 3.2: Ziegler-Nichols yöntemi ile bulunan PI ve PID kontrolör parametreleri 53

Tablo 3.3: Metasezgisel algoritmalarda kullanılan başlangıç değerleri 55

Tablo 3.4: Metasezgisel algoritmalar optimize edilmiş PID kontrolör parametre değerleri 55

Tablo 3.5: Metasezgisel algoritmalar optimize edilmiş PI kontrolör parametre değerleri 56

Tablo 3.6: Bulanık mantık kontrol yönteminin çıkışındaki $\Delta Kp(t)$ kontrol parametresine ilişkili kurallar 60

Tablo 3.7: Bulanık mantık kontrol yönteminin çıkışındaki $\Delta Ki(t)$ kontrol parametresine ilişkili kurallar 60

Tablo 4.1: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PID kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin pozitif bölge (0-2 sn arası) performans kriter değerleri 70

Tablo 4.2: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PID kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin negatif bölge (2-4 sn arası) performans kriter değerleri 70

Tablo 4.3: PID kontrolöre sahip sistem için metasezgisel algoritmaların optimizasyon çalışma süreleri 73

Tablo 4.4: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PI kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin pozitif bölge (0-2 sn arası) performans kriter değerleri 83

Tablo 4.5: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PI kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin negatif bölge (2-4 sn arası) performans kriter değerleri 83

Tablo 4.7: PI kontrolöre sahip sistem için metasezgisel algoritmaların optimizasyon çalışma süreleri 85

Tablo 4.8: Metasezgisel algoritmaları ve Ziegler Nichols kontrol yöntemi ile optimize edilmiş FGPI kontrolöre ilişkin pozitif bölge değerleri (0-2 sn) 97

Tablo 4.9: Metasezgisel algoritmaları ve Ziegler Nichols kontrol yöntemi ile optimize edilmiş FGPI kontrolöre ilişkin negatif bölge değerleri (2-4 sn) 97

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

Kp : Oransal Kazanç
Ki : İntegral Kazanç
Kd : Türevsel Kazanç
t : Zaman
U(t) : Kontrolör çıkış sinyali
E(t) : Sistemin hata sinyali
R(t) : Kontrolör referans giriş sinyali
e : Sistem hata değeri
Y(t) : Sistemin zamana bağlı çıkış değeri
Ra : Motor sargılarında oluşan elektriksel direnç
La : Motorun manyetik alanda oluşturan bobin değeri
Vb : Motorun dönmesiyle oluşturulan ters elektromanyetik kuvvet
Tm : Motorun ürettiği mekanik torkαα
wm : Motor rotorunun dönme hız değeri
Va : Motor gerilim değeri
Bm : Motorun sürtünme katsayısı
Kb : Motorun geri besleme emf sabiti
Ki : Motorun elektromekanik kuvvet sabiti
J : Motor ataleti
Ls : Stator faz endüktansı
Rs : Stator faz direnci
P : Motor kutup çifti
λ : Mıknatıslar tarafından oluşturulan akı bağlantısı
tr : Yükselme zamanı
ty :Yerleşme zamanı
N : Popülasyon sayısı
SN : Toplam kaynak sayısı
$\mathbf{X_k^i}$: i numaralı çözümün k iterasyonundaki konum vektörü
$\mathbf{V_k^i}$: i numaralı çözümün k iterasyonundaki hız vektörü
$\mathbf{X_k}$: k'inci iterasyon değerindeki vektörü
$\mathbf{V_k}$: k'inci iterasyon değerindeki hız vektörü
$\mathbf{X_i}$: i numaralı çözümün konum vektörü
$\mathbf{V_k}$: i numaralı çözümün hız vektörü
g : Küresel optimum en iyi çözüm değeri
$\mathbf{X_{ust}}$: Üst sınır değer
$\mathbf{X_{alt}}$: Alt sınır değer
c1 : Parçacık sosyal öğrenme katsayısı
c2 : Parçacık bireysel öğrenme katsayısı
p : Çicek üretim katsayısı
D : Uzaklık değeri
g : Küresel optimum en iyi çözüm değeri
dim : Çözüm uzayının boyut sayısı
$\mathbf{F_i}$: En iyi yiyecek kaynağına çözümü
$\mathbf{Z_i}$: Düşmandan kaçma çözümü

S_i : Ayrışma kuvveti
B_i : Hizalama kuvveti
C_i : Uyum kuvveti
c : 0 ile 2 arasında değer alan rastgale sayı değişken çarpanı
s : Adım büyüklüğü çarpanı
k : İterasyon değeri
q_1 : Ayrılma ağırlık katsayısı
k_1 : Eylemsizlik ağırlığı katsayısı
z_2 : Sıralanma ağırlık katsayısı
f_1 : Besin faktörü ağırlık katsayısı
r_1 : 0 ile 1 arası sayı değeri
Q_i : i numaralı çözümün k frekans vektörü
σ : Levy uçuş katsayısı
M_i^k : i numaralı çözümün k iterasyonundaki atom kütlesi vektörü
c_3 : Su damlacıkların geniş adım değerini veren sabit çarpan
u_1 : Denize yakın arama katsayısı
Tu : Kritik periyot
Ku : Kritik kazanç parametresi

Kısaltmalar	Açıklama
PID	Oransal-İntegral-Türev Kontrolör
PD	Oransal-Türev-Kontrolör
PI	Oransal-İntegral Kontrolör
FOPID	Kesirli Mertebeden Orantılı İntegral Türev
IAE	Mutlak Hata İntegrali
ITAE	Zaman Ağırlıklı Mutlak Hata İntegrali
ISE	Karesel Hata İntegrali
RMSE	Kök Ortalama Karesel Hata
GA	Genetik Algoritma
DA	Yusufcuk Algoritması
PSO	Parçacık Sürü Optimizasyonu
BA	Yarasa Algoritması
ASO	Atom Arama Algoritması
GWO	Gri Kurt Algoritması
FPA	Çicek Tozlaşma Algoritması
ABC	Yapay Arı Kolonisi Algoritması
WCA	Su Döngüsü Algoritması
CSA	Guguk Kuşu Arama Algoritması
GSA	Yerçekimi Arama Algoritması
MFO	Güve-Alev Optimizasyonu
WOA	Balina Optimizasyon Algoritması
SSA	Salp Sürü Algoritması
SCA	Sinüs-Kosinüs Algoritması
DC	Doğru Akım
AC	Alternatif Akım
EMF	Elektromotor kuvvet

ZN : Ziegler Nichols Yöntemi
TF : Transfer Fonksiyonu
FGPI : Bulanık Mantık-PI Kontrolör
FPD : Bulanık Orantı Türev Denetleyici
FPID : Bulanık Orantı Türev İntegral Denetleyici
FPE : Son Tahmin Hatası

ÖZET

YÜKSEK LİSANS TEZİ

BİR FIRÇASIZ DOĞRU AKIM MOTORUNUN MODELLENMESİ VE KONTROLÜ

Mert AKSÖZEK

İstanbul Üniversitesi-Cerrahpaşa
Lisansüstü Eğitim Enstitüsü
Elektrik- Elektronik Mühendisliği Anabilim Dalı
Elektrik-Elektronik Mühendisliği, Tezli Yüksek Lisans Programı

Danışman: Dr. Öğr. Üyesi Nevra BAYHAN

Tez çalışmasında bir fırçasız doğru akım (DC) motor modeline oransal-integral(PI), oransal-integral-türev (PID) ve bulanık mantık-PI (FGPI) kontrolörler uygulanarak elde edilen sistemin performans analizi yapılmıştır. Bu amaçla öncelikle seçilen uygun parametre değerleriyle bu fırçasız doğru akım motorunun MATLAB Simulink ortamında sistem modeli oluşturulmuştur. Daha sonra klasik Ziegler-Nichols yöntemi ve ayrıca yusufçuk optimizasyon algoritması (DA), su döngüsü algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa optimizasyon algoritması (BA) ve çiçek tozlaşma optimizasyon algoritması (FPA) gibi dokuz tane farklı metasezgisel optimizasyon algoritmaları ile PI, PID ve bulanık mantık-PI (FGPI) kontrolörlerin parametrelerinin optimizasyonu yapılmış ve fırçasız DC motor modelimize uygulanarak sistemin çıkış sinyalleri yükselme süreleri, yerleşme süreleri ve yüzde aşım değerlerine göre kıyaslanmıştır. Ayrıca kullanılan dokuz adet metasezgisel optimizasyon yöntemlerinin amaç fonksiyonlarının yakınsama eğrileri ve çalışma hızları da birbirleriyle karşılaştırılmıştır.

Optimize edilmiş PI ve FGPI kontrolörler uygulanmış fırçasız DC motor sistemimizin birim basamak giriş altında pozitif ve negatif bölgelerdeki performans kriterleri karşılaştırıldığında FGPI denetleyicilerin PI kontrolörlere göre yüzde aşımları azaltırken, yükselme sürelerini ise arttırdığı görülmüştür. GWO ve PSO ile optimize edilmiş PID ve FGPI kontrolörler fırçasız DC motor modelimize uygulandığında, hesaplanan yerleşme sürelerinin diğer metasezgisel algoritmalarınkinden ve Ziegler-Nichols yöntemininkinden daha küçük

olduğu görülmüştür. Hız kontrolü fırçasız DC motorlar için önemli olduğundan, en küçük yerleşme sürelerini veren GWO ve PSO algoritmaları bu konuda önem kazanmaktadır. Metasezgisel algoritmalarla optimize edilmiş PID, PI ve FGPI için aşım yüzdeleri dikkate alındığında motor modelimizde PID kontrolör kullanmanın diğer iki kontrolöre göre daha az yüzdesel aşımlar ve genelde daha küçük yükselme süreleri elde etmeyi sağladığı görülmüştür. Metasezgisel algoritmaların hem PID hem de PI kontrolörler için yakınsama eğrileri karşılaştırıldığında genelde GWO algoritmasının daha düşük hata değeri verdiği anlaşılmıştır. Metasezgisel algoritmaların hem PID hem de PI kontrolörler için çalışma süreleri kıyaslandığında ise FPA algoritmasının optimizasyon sonuçlarına daha hızlı ulaştığı görülmüştür.
Aralık 2024 , 135 sayfa.

Anahtar kelimeler: Fırçasız DC motor, Metasezgisel optimizasyon yöntemleri, PID, PI, bulanık mantık PI kontrolörleri

ABSTRACT

M.Sc. THESIS

MODELING AND CONTROL OF A BRUSHLESS DIRECT CURRENT MOTOR

Mert AKSÖZEK

İstanbul University-Cerrahpaşa
Institute of Graduate Studies
Department of Electrical and Electronics Engineering
Electrical and Electronics Engineering, Master's Thesis

Supervisor: Assist. Prof. Dr. Nevra BAYHAN

In this thesis, the performance analysis of the obtained system was performed by applying proportional-integral (PI), proportional-integral-derivative (PID) and fuzzy logic-PI (FGPI) controllers to a brushless direct current (DC) motor model. For this purpose, firstly, the system model of this brushless direct current motor was created in MATLAB Simulink environment with the appropriate parameter values selected. Then, the parameters of PI, PID and fuzzy logic-PI (FGPI) controllers were optimized with nine different metaheuristic optimization algorithms such as classical Ziegler-Nichols method and dragonfly optimization algorithm (DA), water cycle algorithm (WCA), particle swarm optimization (PSO), genetic algorithm (GA), gray wolf optimization (GWO), artificial bee colony (ABC), atom search optimization (ASO), bat optimization algorithm (BA) and flower pollination optimization algorithm (FPA) and applied to our brushless DC motor model and compared according to the rise times, settling times and percent overshoot values of the system output signals. In addition, the convergence curves and operating speeds of the objective functions of the nine metaheuristic optimization methods used are compared with each other.

When the performance criteria of our brushless DC motor system with optimized PI and FGPI controllers are compared in positive and negative regions under unit step input, it is seen that FGPI controllers reduce percentage overshoots and increase rise times compared to PI controllers. When PID and FGPI controllers optimized with GWO and PSO are applied to our brushless DC motor model, it is seen that the calculated settling times are smaller than those of other metaheuristic algorithms and Ziegler-Nichols method. Since speed control is important for brushless DC motors, GWO and PSO algorithms that give the smallest settling times gain importance in this regard. When the overshoot percentages for PID, PI and FGPI optimized with metaheuristic algorithms are taken into consideration, it is seen that using PID controller in our motor model provides less percentage overshoots and generally smaller rise times compared to the other two controllers. When the convergence curves of metaheuristic algorithms are compared for both PID and PI controllers, it is understood that the GWO algorithm generally gives lower error values. When the running times of metaheuristic algorithms are compared for both PID and PI controllers, it is seen that the FPA algorithm reaches the optimization results faster.

December 2024, 135 pages.

Keywords: Brushless DC motor, Metaheuristic optimization methods, PID, PI, Fuzzy logic- PI controllers

1. GİRİŞ

Elektrik motorları mekanik enerjiyi elektrik enerjisine dönüştürürler. Elektrik motorları alternatif akım (AC) ve doğru akım (DC) elektrik motorları olarak ikiye ayrılır. DC motorlar da kendi arasında fırçalı DC motorlar ve fırçasız DC motorlar olmak üzere ikiye ayrılır. Fırçalı DC motor çeşitleri seri bağlı DC motor, şönt bağlı DC motor, bileşik DC motor ve kalıcı mıknatıslı DC motorlardır [1-3].

Elektrik motorları, günümüzde sanayi tesisleri, otomasyon, ulaşım, ev aletleri, otomotiv, taşımacılık, yenilebilir enerji, medikal cihazlar gibi pek çok farklı alanlarında kullanılır. İhtiyaçlara göre elektrik motorlarının hız, pozisyon ve tork kontrolü özelliklerinden yararlanılır [4,5].

Elektrik motorlarının tarihsel gelişim sürecine bakıldığında özellikle 18. Yüzyılın sonlarında Aleksander Volta ve Luigi Galvani, elektrik akımını oluşturun elektrokimyasal tepkimeler keşfine bağlı olarak bu motorların gelişim süreci başlamıştır [6]. 1820 yılında Hans Christian Ørsted elektrik akımının oluşturduğu manyetik alanı bularak elektrik ve manyetizma arasındaki ilişkiyi ortaya koymuştur [7]. 1831 yılında Michael Faraday yaptığı çalışmayla elektromanyetik indüksiyonunu bularak makine teorisinin gelişimine öncülük etmiştir [8]. İlk döner elektrik makinası 1834 yılında Moritz Hermann Von Jacobi tarafından üretilmiştir. 20. yüzyılın başlarında ise Nicola Tesla'nın alternatif akımı (AC) keşfetmesi ile AC elektrik motorları geliştirilmeye başlanmıştır [9].

Elektrik makinalarının fabrikalarda kullanımı, sanayi devriminin olmasıyla 19.yüzyılın sonlarında artmıştır. Elektrik makinalarının kullanımın yaygınlaşması, otomasyon ve kontrol uygulamalarının gelişimini sağlamıştır [10]. 1911 yılında Elmer Sperry tarafından jiroskop tabanlı dengeleme sistemleri üzerine yaptığı çalışmayla modern kontrol teorisinin gelişimine katkıda bulunmuştur [11]. Bu gelişmeden sonra 1922 yılında mühendis Minorsky gemilerin otomatik dümenin kurulumu için kontrol modeli geliştirmiştir [12]. 1942 yılında Ziegler-Nichols tarafından PID ve PI kontrol için temel bir kontrolör parametrelerini ayarlama yöntemi geliştirilmiştir. [13]. Bu yöntem PI ve PID kontrolörlerin yanı sıra bulanık mantık-PI (FGPI) gibi hibrit kontrolörlere de uyarlanabilmektedir [14,15]. Günümüzde parametre değerleri ayarlanarak optimize edilmiş PI, PID ve FGPI gibi farklı yapıdaki pek çok kontrolörlerin uygulandığı değişik sistem modellerinin performanslarını karşılaştırmak için yüzde aşım,

yerleşme zamanı ve yükselme zamanı kriterlerini inceleyen literatürde birçok bilimsel çalışmalar mevcuttur [16,17].

Tasarım kolaylığı ve düşük maliyetleri dolayısıyla klasik PI ve PID kontrolörler sıklıkla kullanılmasına rağmen; tam istenilen performansı sağlayamamaları gibi durumlarda başta doğadan esinlenen metasezgisel optimizasyon algoritmaları gibi pek çok yapay zeka yöntemleri daha hassas kontrolörler tasarlamak için literatürde yaygın olarak kullanılmaktadır. Bir örnek olarak Gökçe ve diğ.'nin bir çalışmasında DC motor hız kontrol problemi Ziegler-Nichols (ZN) ve Parçacık Sürüsü Optimizasyonu (PSO) olmak üzere iki farklı yaklaşım kullanılarak çözülmüştür [17, 18]. 1975 yılında John Holland tarafından doğal seçim ve genetik temelli bir optimizasyon tekniği olan genetik algoritma (GA) bulunmuştur [19]. Ardından 1995 yılında Kennedy ve Eberhart'ın yapmış olduğu çalışmalarla parçacık sürü optimizasyonu (PSO) algoritmasını ortaya çıkarmışlardır [20]. Bu yöntemler günümüze kadar birçok sisteme uygulanmıştır. Karaboğa, D. tarafından 2005 yılında arıların yiyecek arama davranışlarına göre oluşturulmuş yapay arı kolonisi (ABC) algoritması bir başka metasezgisel optimizasyon algoritması olarak geliştirilmiştir [21]. Yang X.S. tarafından 2012 yılında çiçeklerin tozlaşma sürecini model alarak oluşturulmuş çiçek tozlaşma algoritması (FPA) geliştirilmiştir [22]. Mirjalili S, ve diğ., doğadaki gri kurtların liderlik hiyerarşisi ve avlanma mekanizmasından esinlenerek gri kurt optimizasyon (GWO) algoritmasını oluşturmuşlardır [23]. Yarasaların yankı konumlama davranışına dayanan bir başka metasezgisel algoritma yöntemi olan yarasa algoritması (BA), Yang X.S. tarafından geliştirilmiştir [24]. Mirjalili S, tarafından sürü özelliğine bağlı oluşturulan doğadaki yusufçukların statik ve dinamik sürü davranışları dikkate alınarak yusufçuk algoritması (DA) geliştirilmiştir [25]. Zhao W, ve diğ., atomların doğal hareketlerini ve birbirleri ile olan etkileşimlerini inceleyen atom arama optimizasyon algoritmasını (ASO) önermişlerdir [26]. Atom arama algoritması atomların çekim kuvvetleri üzerine kuruludur. Eskander H. ve diğ., su döngüsü sürecinin gözlemlenmesine ve nehirlerin gerçek dünyada denize nasıl aktığına dayanarak suyun doğadaki akışından esinlenilerek su döngüsü algoritmasını (WCA) geliştirmişlerdir [27].

Tez çalışmasında sistem olarak DC motorlar arasından fırçasız doğru akım (DC) motor seçilmiştir. Fırçasız DC motorun bakım gereksiniminin düşük olması, yüksek performans ve yüksek verimliliğe sahip olması diğer DC motorlara göre avantaj sağlamaktadır [28,29]. Literatürden [30-32]'deki motor parametre ayarları kullanılarak seçilen fırçasız DC motorun modellenmesi, MATLAB Simulink ortamında gerçekleştirilmiştir. Simulink ortamında gerçek zamana bağlı elde edilen veriler kullanılarak sistem tanımlama (system identification toolbox)

kullanılarak fırçasız DC motorun transfer fonksiyon modeli oluşturulmuştur. Bu tezin amacı yusufçuk optimizasyonu (DA), su döngüsü optimizasyon algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa optimizasyon algoritması (BA) ve çiçek tozlaşma optimizasyon algoritması (FPA) gibi metasezgisel optimizasyon yöntemleri ile parametreleri ayarlanmış ve optimize edilmiş PID, PI ve PI optimizasyon parametrelerinin performansını daha da iyileştirmek amacıyla kullanılan bulanık mantık-PI (FGPI) kontrolörlerin Matlab Simulink ortamında modeli oluşturulan bir fırçasız DC motor modeline uygulanarak elde edilen sistem cevaplarının yerleşme süreleri, yükselme süreleri ve yüzde aşım gibi zaman tanım bölgesi kriterleri açısından performanslarının karşılaştırılıp bu sistem için en optimum olan metasezgisel algoritmaların tespit edilmesidir.

Bu tez çalışması giriş, kavramsal çerçeve, yöntem, bulgular, tartışma ve sonuç kısımlarından oluşmuştur. Tezin ikinci bölümü olan kavramsal çerçeve kısmında elektrik motorları hakkında genel bilgiler anlatılarak literatürden parametre değerleri seçilen bir fırçasız DC motorun simülasyon ortamında sistem modelinin elde edilmesi ele alınmıştır. Ayrıca bu bölümde modeli oluşturulan bu motoru kontrol etmek için tezde kullanılacak olan PI, PID ve bulanık mantık kontrolörler incelenmiştir. Tezin üçüncü bölümü olan yöntem kısmında, bu tezde kullanılacak olan metasezgisel optimizasyon algoritmalardan yusufçuk optimizasyonu (DA), su döngüsü optimizasyon algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa optimizasyon algoritması (BA) ve çiçek tozlaşma optimizasyon algoritması (FPA) ve klasik kontrol yöntemi olan Ziegler-Nichols (ZN) hakkında gerekli bilgiler, algoritmaların sözde (pseudo) kodları ve akış diyagramları anlatılmıştır. Metasezgisel algoritmalar 3 farklı uygunluk fonksiyonuna göre seçilebilir. Bu uygunluk fonksiyonları IAE (mutlak hata integrali), ITAE (zaman ağırlıklı mutlak hata integrali) ve ISE (karesel hata integrali) uygunluk fonksiyonlarıdır. Her bir uygunluk fonksiyonu farklı özellikleri içermektedir. Bu tez çalışmasında IAE uygunluk fonksiyonu dikkate alınarak modeli oluşturulan fırçasız DC motorun kontrolü yapılmıştır. Bu tez çalışmasında Ziegler-Nichols yöntemi ve yukarıda bahsi geçen 9 adet metasezgisel algoritmalar ile ayarlanan PI, PID ve FGPI gibi düşük dereceli kontrolörlerin uygulandığı ve modeli kurulan fırçasız doğru akım motorunun iyi bir performans göstermesi istenmektedir. Tezin bir sonraki bölümü olan bulgular kısmında Ziegler-Nichols yöntemi ve 9 adet metasezgisel algoritmalar ile optimize edilmiş PI, PID ve FGPI denetleyiciler uygulanmış fırçasız DC motor modeline birim basamak girişi

verilerek, sistemin çıkışında elde edilen çıkış sinyalleri yükselme zamanı, yüzde aşım miktarı ve yerleşme zamanları gibi kriterlerle sistem performansı açısından değerlendirilmiştir. Ayrıca kullanılan dokuz adet metasezgisel optimizasyon yöntemlerinin amaç fonksiyonlarının sıfıra yakınsama eğrileri ve çalışma hızları da birbirleriyle karşılaştırılmıştır. Tezin son bölümü olan tartışma ve sonuç kısmında, elde edilen tüm sonuçlar değerlendirilmiş ve en uygun yöntemler belirlenerek çıkarımlar yapılmıştır.

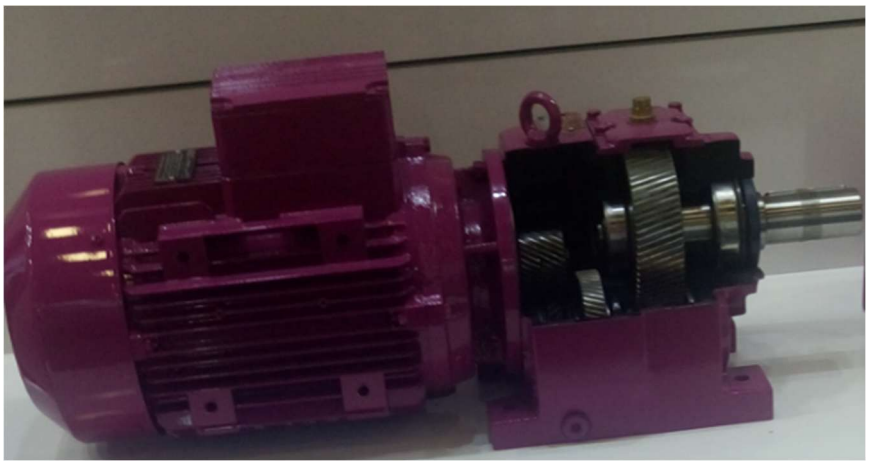
2. KAVRAMSAL ÇERÇEVE

2.1. ELEKTRİK MOTORLARI

Elektrik motorları, elektrik enerjisini mekanik enerjiye çeviren cihazlardır. Elektrik motorları yüksek verimlilik, çevre dostu olması, geniş hız ve güç aralığında çalışabilme avantajlarına sahip olup; bu motorları doğru akım (DC) motorları ve alternatif akım (AC) motorları olmak üzere gruplandırılırlar. DC motorlar da kendi arasında fırçalı DC motor ve fırçasız DC motorlar olmak üzere ikiye ayrılır.

2.1.2. Elektrik Motorlarının Kontrol Uygulamalarında Kullanım Yerleri

Elektrik motorları birçok farklı alanlarda kullanılır. Elektrik motorları endüstriyel, ulaşım, enerji ve güç sistemi, akıllı binalar ve otomasyon, tıbbi cihazlar vb. gibi pek çok kullanım alanı olup; hız, pozisyon ve tork kontrolü açısından önemlidirler.
Endüstriyel alanlarında freze makineleri, konveyör bantları, torna tezgâhları endüstriyel robotlar, vinç, asansörler vb. gibi motor hızının kontrolü uygulamalarında kullanılır. Enerji ve güç sistemi alanlarında güneş takip sistemleri, rüzgâr türbinleri, depolama sistemleri vb. gibi motor hız, pozisyon ve tork kontrol alanlarında kullanılırlar. Örnek bir elektrik motor görüntüsü Şekil.2.1'de gösterilmiştir.



Şekil 2.1: Örnek bir elektrik motoru

2.1.3. Elektrik Motorlarının Çalışma Prensibi

Elektrik motorlarının temel bileşenleri stator, rotor, rulmanlar, fırçalar ve komütatördür. Stator, elektrik motorlarının sabit kısmıdır. Yapısında elektromıknatıs ve sabit mıknatıslar bulunur. Stator, manyetik alan oluşturan ve bu manyetik alanla elektrik üretimi sağlayan motorun en önemli parçalarından biridir. Diğer önemli temel bileşeni ise motorun rotor bileşenidir. Rotor, statorun içindeki dönen kısmıdır. Rotor, stator tarafından oluşturan manyetik alanla etkileşime girerek motor içerisinde döner. Bu dönme sayesinde dönme kuvveti oluşur. Mekanik enerjiden elektrik enerji oluşmasını sağlar. Diğer temel bileşen, motorun rulmanlarıdır. Rulmanlar motorun dönmesi sağlayan bileşenlerdir. Rulmanlar, motorun sürtünmesini azaltarak hızlı dönmesini ve verimini arttıran bileşenlerdir. Diğer temel bileşen ise motorun fırçalar ve komütatördür. Fırçalar ve komütatör, motorlarda rotorun akımını kontrol ederek sürekli dönmesini sağlayan bileşenlerdir. Motorlarda bulunan fırçalar, motorun akımını komütatör yardımıyla rotor sargılarına ulaştırır. Komütatör, bu akımın yönünü düzenleyerek motorun aynı yerde dönmesine olanak sağlar.

Elektrik makinaları çalışma prensibi, Lorentz kuvvetine dayanır. Elektrik motorları stator tarafından oluşturulan manyetik alanı motorun rotorunun üzerindeki bobinlerle etkileşime girerek Lorenz kuvvetini oluşturur. Lorentz kuvveti, tork ve akım oluşturur. Bu tork, rotoru döndürerek; elektrik enerjisini mekanik enerjiye dönüştürür [5].

2.2. DOĞRU AKIM (DC) MOTORLAR

DC motorlar, mekanik enerjiyi elektrik enerjisine çeviren ve adından da anlaşıldığı üzere doğru akımla çalışan elektrik motorlarıdır. DC motorlar, fazla tork üretimi sağlayarak ağır yüklerin taşınmasına olanak sağlar. Elektrik motorlarının hızı, voltaj değişikliği ile kolayca kontrol edilebilir. DC motorların taşıtlar, endüstriyel makinalarda, robotlarda, ev aletleri vb. gibi birçok uygulama alanında kullanımı vardır.

Seri bağlı dc motor, şönt bağlı dc motor, bileşik dc motor, kalıcı mıknatıslı dc motor ve fırçasız dc motorlar dc motor türleridir [3].

2.2.1. Seri Bağlı DC Motor

Seri bağlı dc motorlar, stator ve rotor sargıları birbirlerine seri olarak bağlanan dc motorlardır. Bu motorların temel bileşenleri: stator, rotor, fırçalar ve komütatördür. Bu motorlar, yüksek tork üretir. Motorun hız değişimlerine karşı her zaman yüksek tork üretimi mümkündür. Yük değişimleri, motorun hızıyla ters orantılıdır. Yükteki artış ise motor akımıyla doğru orantılıdır. Bu motorlar yük olmadan çalıştırılırken, hız sınırlayıcı kullanılır.

2.2.2. Şönt Bağlı DC Motor

Stator sargıları ve rotor sargıları birbirlerine paralel bağlı olan motorlardır. Motorun temel bileşenleri stator, rotor, fırçalar, rulmanlar ve motor gövdesidir. Bu motorlar sabit hız ile kararlı çalışırlar. Bu motorlarda stator çevresi, şönt sargılarıyla çevrilidir. Şönt sargıları, DC akım uygulayarak manyetik alan oluşturup; rotorun içinden geçen manyetik kuvvet hatlarını oluşturur. Tork oluşturarak da motorun dönmesi sağlanır.

2.2.3. Bileşik (Compound) DC Motor

Hem seri hem de şönt sargılarının motorda bulunması ile oluşturulan DC motorlardır. Motorun temel bileşenleri stator, rotor, komütatör, fırçalar, rulmanlar ve motor gövdesidir. Bu motor seri ve şönt sargılarının birlikte bulunmasından dolayı yüksek tork ve kararlı çalışma prensibine sahiptir. Bu motorlar, tork ve hız kontrolünde esneklik sağlarlar. Böylece güvenilir performans elde edilir.

2.2.4. Kalıcı Mıknatıslı DC Motor

Sabit kalıcı mıknatıslı dc motorlar, statorda bulunan elektromıknatıs yerine kalıcı mıknatıslar kullanılarak oluşturulan motorlardır. Motorun temel yapı bileşenleri stator, rotor, fırçalar, rulmanlar ve motor gövdesidir. Kalıcı mıknatıslar sabit bir manyetik alan oluşturur. Bu durum motorun sabit bir hız oluşmasını sağlar. Kalıcı mıknatıslar kullanılarak motorun statorunda daha az enerji sağlanarak motorun verimliliği arttırılır.

2.2.5. Fırçasız DC Motorlar

Fırçasız DC motorlar, fırça ve komütatör içermeyen motorlardır. Bu motorların temel bileşenleri stator, rotor, elektronik devreleri, rulmanlar ve motor gövdesidir. Motorun statoru 3 fazlı şekilde düzenlenip; bakır telden yapılmış sargılar içerir. Motorun rotor kısmında kalıcı mıknatıs bulunarak, manyetik alanla etkileşerek döner. Elektronik devreleri hall sensörleri ve elektronik denetleyici kullanılarak motorun sürülmesi sağlanır. Hall sensörleri, rotorun açısını ölçmek için kullanılır. Elektronik denetleyiciler, hall sensörlerden gelen bilgiyi stator sargılarındaki akımın yönünü ayarlayarak rotorun sürekli dönmesini sağlar. Bu motorlar fırçasız kullanıldığından bakımı az ve verimliliği yüksektir. Bu durum, özellikle endüstride yaygın kullanım alanlarına olanak sağlar.

2.2.5.1. Fırçasız DC Motorun Avantajları

Bu avantajlar şunlardır:

- Fırçasız olduğundan ve sürtünme olmadığından verimliliği yüksektir.
- Aşınım olmadığından uzun ömürlüdür.
- Fırça değişimi ve temizliği olmadığından bakımı azdır.
- Daha az ısınma olduğundan yüksek verimlidir.
- Elektronik devreleri motorun hızı, torku ve pozisyonu çok hassas kontrol edilir.
- Daha sessiz çalışır.
- Yüksek tork ve güç üretebilirler.

2.2.5.2. Fırçasız DC Motor Dezavantajları

Bu dezavantajlar şunlardır:

- Fırçasız dc motorlar, sürücü bileşenleri maliyeti fazla olduğunda diğer DC motorlardan daha maliyetlidir.
- Sürücü bileşenlerine bağlı olduğundan elektronik arızalarda motorun çalışmasını etkiler.
- Sensörlerin doğru çalışmaması sistemin performansını etkiler.
- İlk kurulum maliyeti diğer DC motorlara göre yüksektir.

2.2.6. DC Motorların Özelliklerinin Karşılaştırılması

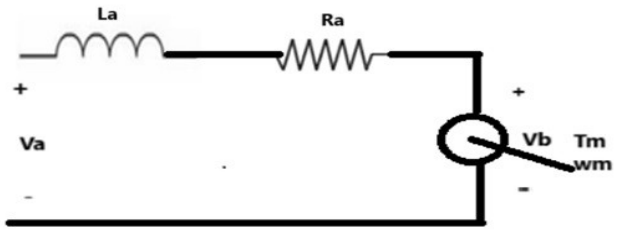
DC motorların çeşitli özelliklerinin bir karşılaştırılması, Tablo 2.1'de verilmiştir [1-5].

Tablo 2.1. DC motorlarının özelliklerinin karşılaştırılması [1-5]

Özellikler	**Seri Bağlı DC Motor**	**Şönt Bağlı DC Motor**	**Bileşik (Com pound) DC Motor**	**Kalıcı Mıknatıslı DC Motor**	**Fırçasız DC Motor (BLDC)**
Yapı	Armatör ve sargılar seri bağlı	Armatör ve sargıları paralel bağlı	Seri ve şönt sargılar bağlı	Sabit mıknatıslar var.	Sabit mıknatıs ve komütatör yok
Verimlilik	Orta	Düşük-orta	Yüksek	Orta	Yüksek
Bakım Gereksinimi	Yüksek	Orta	Yüksek	Düşük	Düşük
Ömür Değeri	Orta süre	Uzun süre	Orta-uzun süreli	Uzun süreli	Uzun süreli
Hız-Tork Karakteristiği	Yük arttıkça hız düşer.	Yük bağımsız hız sabit kalır	Yüksek yükte bile hız kontrolü iyidir	Hız sabittir, tork sabittir	Yüksek hız ve sabit tork sağlanabilir
Hız değer aralığı	Geniş	Sabit	Geniş	Sabit	Geniş
Kullanım Dayanıklığı	Yüksek yükte çalışır.	Hafif yüklerde sabit çalışır	Hem hafif hem ağır yüklerde etkili çalışır.	Hafif yüklerde etkilidir	Hem hafif hem ağır yüklerde etkili
Başlangıç Torku	Çok yüksek	Düşük-orta	Yüksek	Orta	Yüksek
Başlangıç Akımı	Çok yüksek	Düşük	Orta	Düşük	Düşük

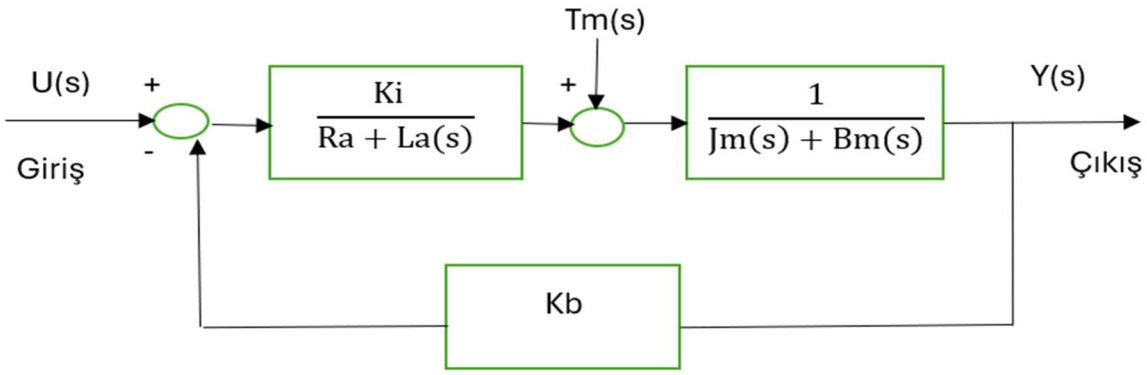
2.3. DC MOTORUN MODELLENMESİ

DC elektrik motorunun temel elektrik devre şeması, Şekil 2.2'de gösterilmiştir [33]. Şekil 2.2'deki devre elemanları, motorun elekriksel davranışını ifade eder. Fırçasız DC motorun çalışması için girişine DC gerilim (Va) verilerek , motor sargılarında bulunan motor direnci (Ra) ve motor endüktans (La) elektrik devre elemanları ile birlikte motorun rotorunun dönmesi sağlanır. Rotor dönmeye başlandığında manyetik alanın etkisiyle ters yönde (Vb) gerilimi oluşur. Ters yönde oluşan bu gerilim, uygulanan gerilime ters yönde meydana gelir. Devrede motorun uygulanan gerilimle oluşan yük değerleri (Tm) ve hız değerlerine (wm) göre akım ve tork değerleri de değişmektedir. Oluşan ters gerilim arttıkça, motorun çektiği akım azaldığı için motorun tork değeri azalır.



Şekil 2.2: Fırçasız DC elektrik motorunun temel **e**lektrik devre şeması [33]

DC motorun blok diyagramı, Şekil 2.3'te gösterilmiştir [34]. Bu diyagramda, DC motorun geri besleme döngüsü kullanılarak DC motorun hız kontrolü anlatılmaktadır. Blok diyagramında dc motorun elektriksel ve mekanik bileşenleri mevcuttur.



Şekil 2.3: DC motor blok diyagramı [34]

Şekil 2.3'te elektrik motorun elektromekanik kuvvet sabiti: KI, motorun sargılardaki elektriksel direnci: Ra, motorun endüktans değeri: La, motorun manyetik ataleti: Jm, motorun sürtünme katsayısı: Bm, motorun geri besleme kazancı: Kb, motorun girişi: U(s) ve motorun çıkışı: Y(s) ile gösterilmiştir. Bu blok diyagramında motorun girişi olarak uygulanan gerilim değeri, elektrik devresi üzerinden bir tork oluşturur. Bu tork, mekanik sistemini ve açısal hızı oluşturur. Dönen motora Kb ile geri besleme yapılır. Böylece bir geri elektromotor kuvveti oluşturularak giriş gerilimine göre hız ayarlaması yapılır.

2.4. FIRÇASIZ DC MOTORUN MATLAB SİMÜLASYON ORTAMINDA MODELLENMESİ

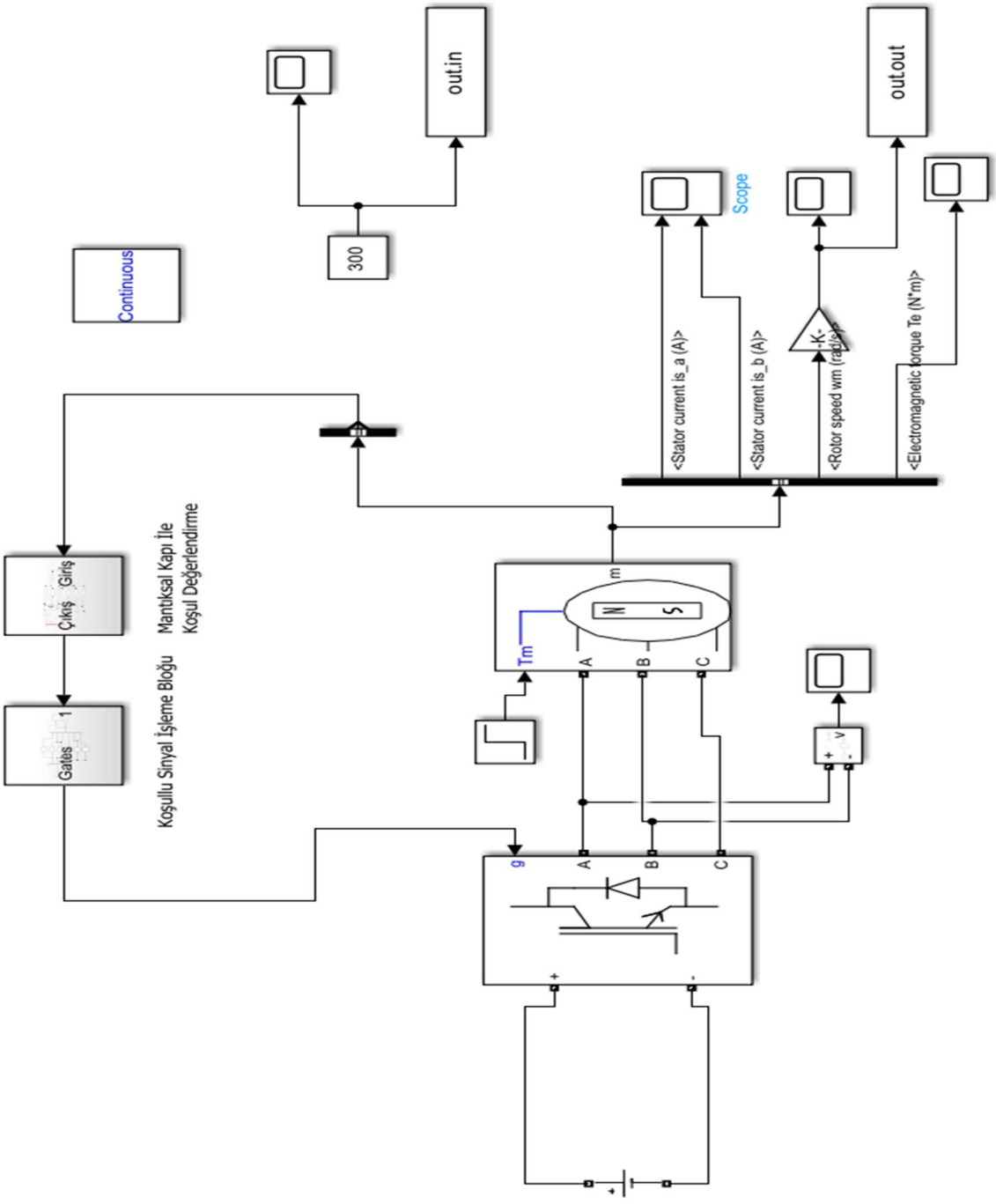
Fırçasız DC motorlar genel olarak üç fazlı besleme ile çalışmaktadır. Bu motorlarda pozisyon, hız ve açı ölçümleri yapmak için geri besleme sensörleri veya enkoderler kullanılmaktadır. MATLAB Simulink ortamında sensörler yerine scope kullanarak hız ölçümü yapılabilmektedir [35].

Motorun MATLAB simulink ortamında sistem modelini oluşturmak için ilk önce dc motorun parametre değerlerinin seçilmesi ya da belirlenmesi gerekmektedir. Bu tezde seçilen motorun parametre değerleri olarak Tablo 2.2'de verilen değerler kullanılmıştır [31,32].

Tablo 2.2: Fırçasız DC motorun seçilen parametre değerleri [31,32]

Parametreler	**Değerler**
Stator Faz Direnci (Rs)	2.8750 ohm
Stator Faz Endüktansı (Ls)	0.0085 H
Mıknatıslar Tarafından Oluşturulan Akı Bağlantısı (λ)	0.175 V. s
Geri EMF Düz Alanı	120 derece
Motorun Atalet (J)	0.0008 kg.m²
Sürtünme Faktörü (Bm)	0.001 N.m.s
Kutup Çifti (P)	4

Fırçasız DC motorunun modeli, Matlab Simulink ortamında Şekil 2.4'te gösterildiği gibi oluşturulmuştur [35]. Motorun hız değerlerinin zamana bağlı değişimi simulink ortamında gözlemlenmiştir. Modelde motorun zaman içinde hız değişimi osiloskop (scope) blokları ile ölçülmüştür. Simülasyon modelinde evrensel köprü, tork ve hız geri besleme blokları ile elektrik güç kaynakları bulunmaktadır.



Şekil 2.4: DC motorun MATLAB Simulink ortamında modeli [35]

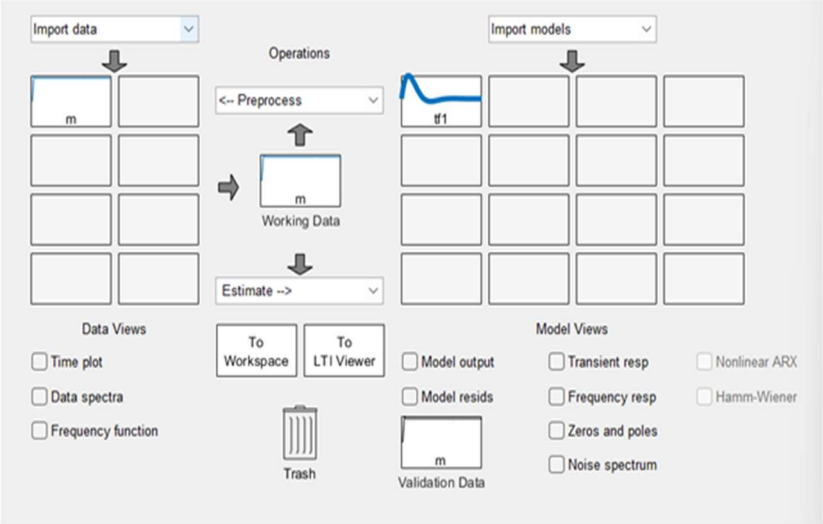
MATLAB Simulink ortamında oluşturulan modelin yapısı elektrik sistemi, mekanik sistem ve kontrol sistemi olmak üzere üç ayrı sistem halinde incelenmiştir. Evrensel köprü DC motora 3 faz gerilim sinyalleri ile motorun beslenmesini sağlayan devre elemanıdır. MATLAB simulink ortamında modellenmiştir. Motorun beslenmesi için 300 V giriş değeri kullanılmıştır. DC motora verilen gerilim değerinin gerçek zamanda çalışabildiği, aynı motor parametreleri kullanılarak yapılan literatürdeki bazı çalışmalardan anlaşılmaktadır. Bu çalışmalarda gerilim değeri 300V'tan hem düşük hem de yüksek giriş gerilimi verildiğinde motorun çalışabildiği gözlemlenmiştir [31, 32].

Motorun kontrolü için oluşturulan kontrol sinyalleri ile motorun hızının kontrol edilmesi sağlanır. Motora verilen giriş sinyalleri motorun Tablo 2.3.'teki motor parametreleri ile çalışmasını sağlar. Motor sargılarına verilen giriş sinyalleri ile manyetik alan oluşturup, rotorun dönmesi sağlanır. Motorun rotorunun dönmesi ile motor hız değerleri oluşur. Giriş değeri ile motorun hız değeri arasında oluşan hata sinyali, simülasyonda belirtilen bloklar tarafından düzenlenir. Hata sinyali giriş değeri ile motorun gerçek hız değerleri arasındaki farktır.
Şekil 2.4'te sinyalleri düzenleyen bloklar simülasyon ortamında mantıksal kapı ile koşul değerlendirme ve koşullu sinyal işleme bloğu olarak adlandırılmıştır. Mantıksal kapı ile koşul değerlendirme bloğu mantıksal devre elemanları ile rotorun anlık hız sinyallerini değerlendirerek, motorun fazlarının sıralamasını kontrol edip, motorun fazlarının doğru sırada anahtarlamasını sağlar. Koşullu sinyal işleme bloğu ise giriş sinyallerini hata sinyallerine göre düzenler.

Simülasyonda motor hız değerlerin scope ile ölçülmesi sonucunda başlangıç hızı 2000 rpm den başlayıp hızlı bir şekilde düşüş gösterip 1833 rpm hız birimde sabit kaldığı gözlemlenmiştir. Bu değer, motorun kararlı duruma ulaştığında çalıştığı hız değeridir ve motorun başlangıçta güçlü bir hızlanma ve tork üretme yeteneğine sahip olduğunu gösterir. Bu yüksek performanslı uygulamalar için istenen bir durumdur. Yüksek verimlilik ve hassasiyet gerektiren uygulamalar için tercih edilebilir ve endüstriyel otomasyon, robotik, elektrikli araçların çekiş sistemleri gibi alanlarda yaygın olarak kullanılır.

2.5. FIRÇASIZ DC MOTORUN TRANSFER FONKSİYONUN OLUŞTURULMASI

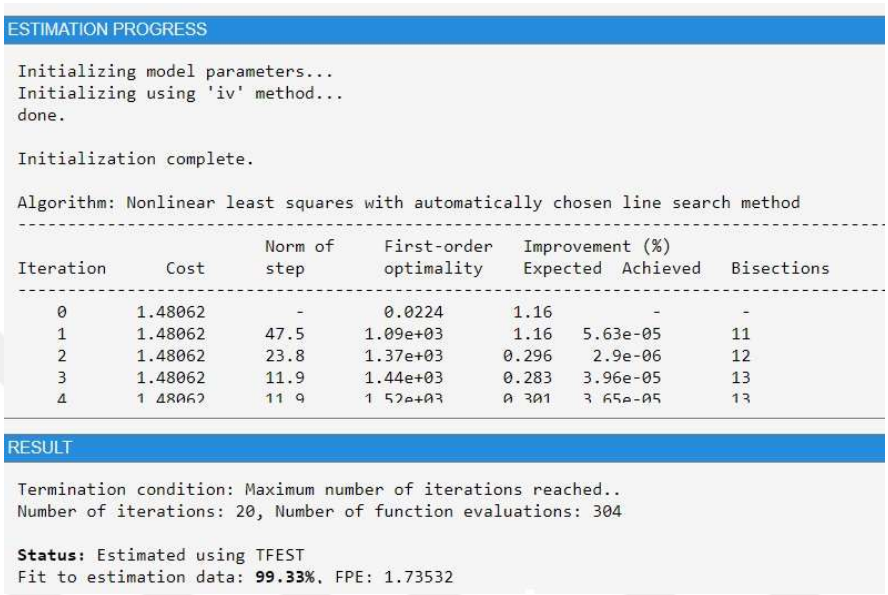
Sistem tanımlama (identification) bir dinamik sistemden elde edilen verilerin o sistemin matematik modelini oluşturmak için kullanılması sürecidir. Veriler kullanılarak sistemin davranışını tahmin eden model oluşturmayı amaçlar [34, 35]. Bu amaçla bu tezde parametre değerleri seçilen dc motorun simülasyon ortamında çalıştırılması sonucu elde edilen zamana göre değişen giriş voltaj değerleri ve çıkışındaki motorun hız verileri kullanılmıştır. Bu veriler dikkate alınarak söz konusu motorun MATLAB System Identification Toolbox kullanılarak transfer fonksiyonu oluşturulmuştur. Transfer fonksiyonu elde edilmeye çalışırken 101 adet giriş ve çıkış verisi kullanılmıştır. Sisteme en uygun transfer fonksiyonunu belirlerken verilerin modele uygunluğunun %90 ve üzeri olması transfer fonksiyonunun doğruluğu açısından önemlidir. Bu amaçla en iyi doğrulama yüzdesini elde edecek şekilde sistemin kutup sayısı ve sıfır sayısı belirlenir. Şekil 2.5'te MATLAB ortamında sistem tanımlama arayüzü gösterilmiştir.



Şekil 2.5: MATLAB ortamında sistem tanımlama arayüz şeması [30,35]

Bu çalışmada, parametre değerleri seçilen dc motordan elde edilen veriler kullanılarak Şekil 2.5'te görülen sistem tanımlama arayüzü kullanılarak bu motorun transfer fonksiyonu elde edilmiştir. Transfer fonksiyonu elde edilirken arayüzde modelimize en uygun kutup ve sıfır sayısı belirlenmiştir. Buradaki motor sistemi için en uygun kutup sayısı 3 ve sıfır sayısı da 1 olarak oluşturulmuştur. Oluşturulan kutup sayısı ve sıfır değerleri ile MATLAB system identification toolbox'da transfer fonksiyonu oluşturulduğunda transfer fonksiyonun doğruluk

değeri %99,33 olarak hesaplanmıştır. FPE (son tahmin hatası) değeri 1.73532 olarak elde edilmiştir. FPE değeri modelin tahmin doğruluğunu bir hata metriği olarak açıklar. Düşük FPE değeri, modelin daha iyi bir tahmin doğruluğuna sahip olduğunu gösterir. Şekil 2.6'da doğruluk değerleri gösterilmiştir.



ESTIMATION PROGRESS

```
Initializing model parameters...
Initializing using 'iv' method...
done.

Initialization complete.

Algorithm: Nonlinear least squares with automatically chosen line search method
```

Iteration	Cost	Norm of step	First-order optimality	Improvement (%) Expected	Achieved	Bisections
0	1.48062	-	0.0224	1.16	-	-
1	1.48062	47.5	1.09e+03	1.16	5.63e-05	11
2	1.48062	23.8	1.37e+03	0.296	2.9e-06	12
3	1.48062	11.9	1.44e+03	0.283	3.96e-05	13
4	1.48062	11.9	1.52e+03	0.301	3.65e-05	13

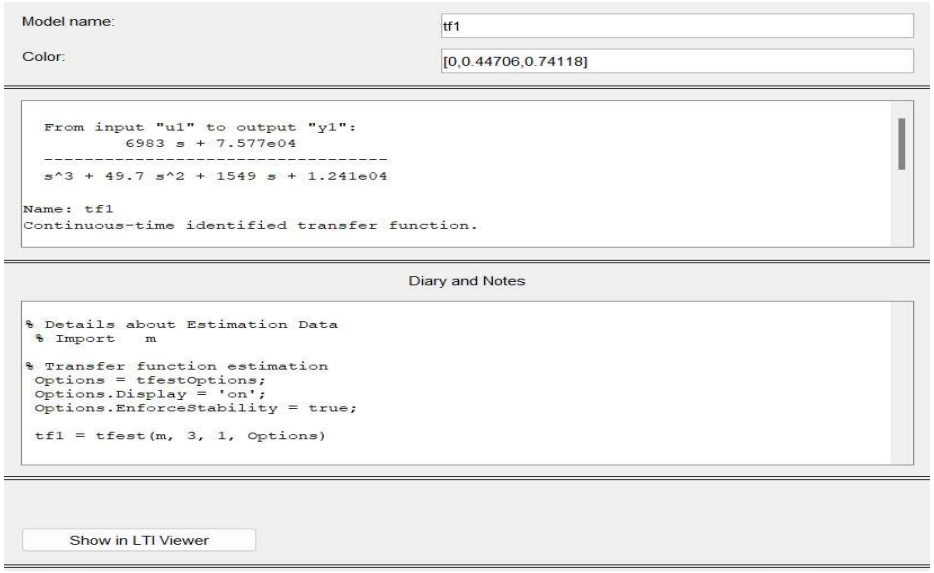
RESULT

```
Termination condition: Maximum number of iterations reached..
Number of iterations: 20, Number of function evaluations: 304

Status: Estimated using TFEST
Fit to estimation data: 99.33%, FPE: 1.73532
```

Şekil 2.6: MATLAB ortamında sistem tanımlama doğruluk değerleri [35]

Bu motor için elde edilen transfer fonksiyon modeline ilişkin verilen giriş ve çıkış verilerinin oldukça iyi uyum sağladığı ve gerçek sistemin davranışına oldukça yakın bir sonuç verdiği görülmüştür. Oluşturulan transfer fonksiyonu, MATLAB ortamında Şekil 2.7'de gösterilmiştir.



Şekil 2.7: MATLAB ortamında motorun sistem tanımlama transfer fonksiyonu [35]

Seçilen parametre değerlerine sahip fırçasız DC motorun MATLAB ortamında sistem tanımlama arayüzü kullanılarak oluşturulan bir sıfıra ve üç kutba sahip transfer fonksiyon modelimiz aşağıdaki bağıntıda verilmiştir.

$$TF = \frac{6983s+7577}{s^3+49.7s^2+1549s+12410} \tag{2.1}$$

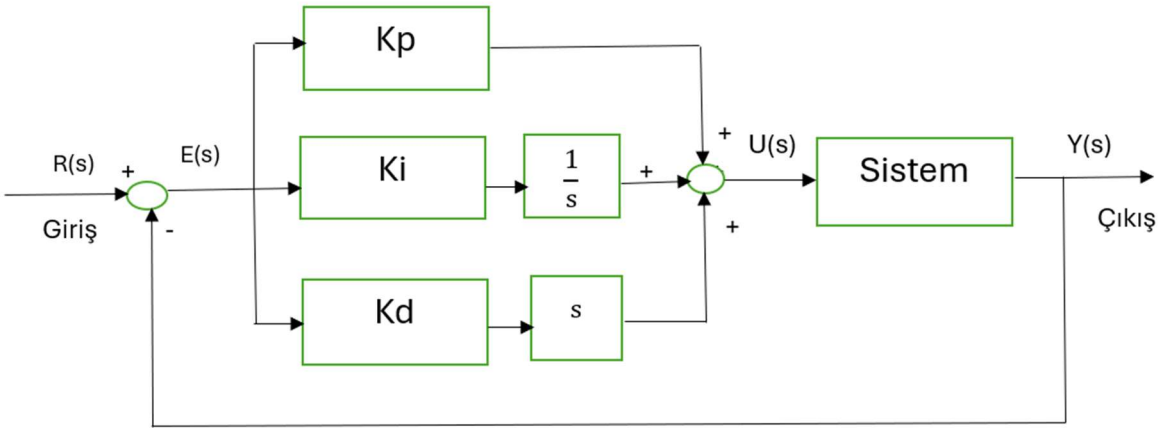
2.6. FIRÇASIZ DC MOTORUN KONTROLÜ

Fırçasız DC motor kontrolünde motorun hızının iyi bir şekilde kontrol edilmesi için farklı yöntemler kullanılır. Bu yöntemler açık çevrim kontrol ve kapalı çevrim kontrol olarak ikiye ayrılır. Fırçasız DC motorlar, mekanik fırçalar yerine elektronik komütasyon ile çalışır. Böylelikle motorun daha verimli çalışması sağlanır. Açık çevrim kontrol, fırçasız DC motorun hızının kontrolünde gerilim sinyallerinin doğrudan ölçülmesini sağlar. Geri besleme olmadığı için motorun kararlılığı düşüktür. Giriş sinyali ve model dikkate alınarak sistem kontrol edilir. Kapalı çevrim kontrol, fırçasız DC motorun hızının belirli bir referans değerine göre kontrolünü sağlar. Referans değerlerinin sensörler yardımıyla karşılaştırılması yapılır veya sensör kullanmadan geri EMF (elektromotor kuvvet) kullanılır. Sensörlerle rotorun hızı sürekli izlenir.

Bu tez çalışmasında düşük dereceli kontrolör yapılarından PID, PI kontrolörler ve elde edilen PI optimizasyon parametrelerinin performansını daha da iyileştirmek amacıyla kullanılan bulanık mantık-PI (FGPI) kontrolör kullanılmıştır.

2.6.1. PID Kontrol

En yaygın kullanılan kapalı çevrim kontrol yöntemlerinden biridir. Şekil 2.8'de PID kontrolün temel bileşenleri gösterilmektedir. PID kontrolörler Kp (oransal kazanç), Ki (integral kazanç) ve Kd (türevsel kazanç) olmak üzere üç temel parametreden oluşur.



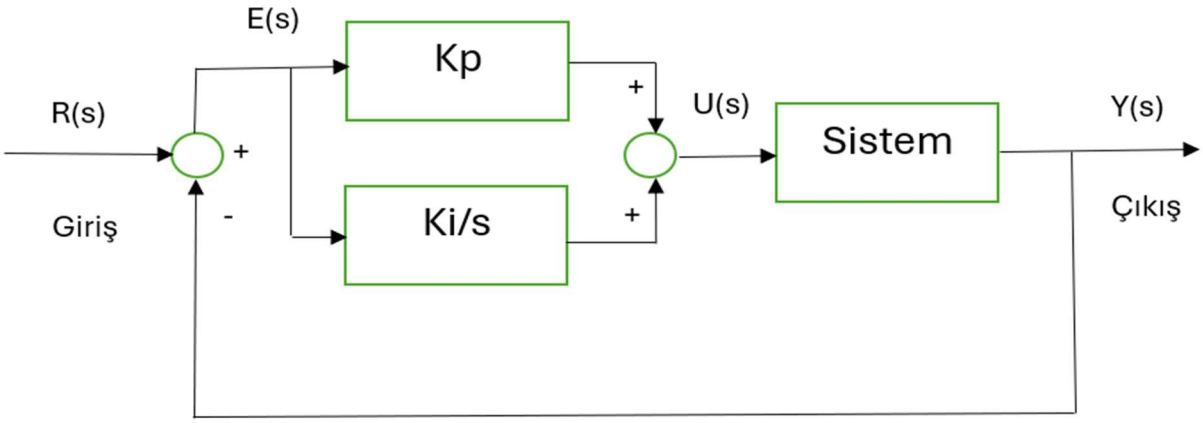
Şekil 2.8: PID kontrol blok diyagramı [14, 16]

PID kontrolörler, Kp, Ki ve Kd kontrol kazanç parametreleri ile DC motorun kalıcı durum yanılgısını (hata) azaltarak motorun doğru bir şekilde çalışmasını sağlar. Kp katsayısı, hata üzerinde anlık düzeltme sağlar. Ki parametresi sistemin kalıcı durum yanılgısını azaltmaya ve zamanla sistemin hızlı bir şekilde referans değerine ulaşmasını sağlamaya çalışır. Kd parametresi salınımları azaltmaya çalışarak sistemin fazla aşım yapmasını engellemeye çalışır. Şekil 2.8'de referans giriş sinyali R(s), çıkış sinyali Y(s), kontrol sinyali U(s) ve hata sinyali E(s) olmak üzere PID kontrolün transfer fonksiyonu aşağıdaki bağıntıda verilmiştir:

$$\frac{U(s)}{E(s)} = Kp + Ki\frac{1}{s} + Kds \tag{2.2}$$

2.6.2. PI Kontrol

PI kontrol, yaygın olarak kullanılan bir diğer kontrol yöntemlerinden biridir. Şekil 2.9'da PI kontrolün temel bileşenleri gösterilmiştir. PI kontrolörler Kp (oransal kazanç) ve Ki (integral kazanç) olmak üzere iki temel parametreden oluşur Burada PID kontrolden farklı olarak Kd türev kazancı bulunmamaktadır.

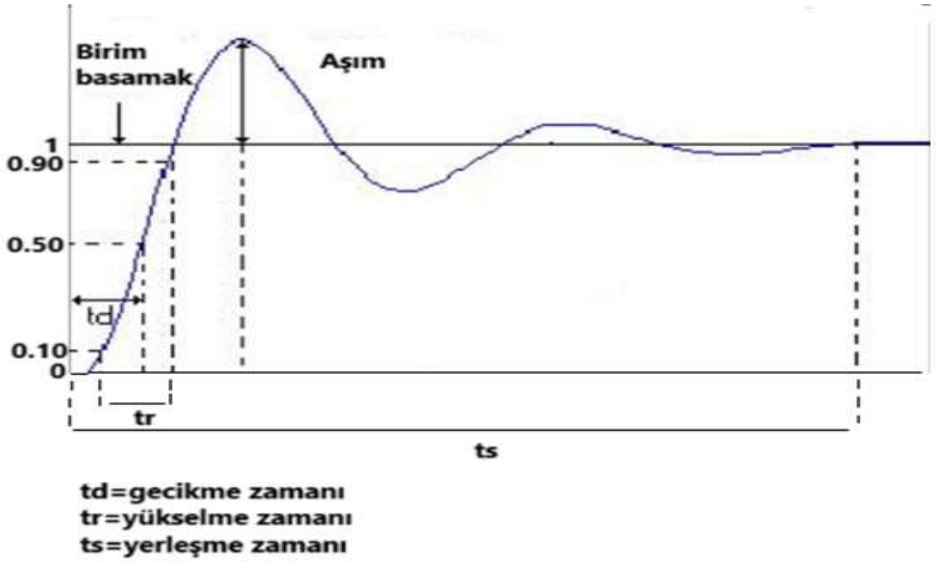


Şekil 2.9: PI kontrol blok diyagramı [14, 16]

PI kontrolörün giriş-çıkış arasındaki transfer fonksiyonu, aşağıdaki bağıntıda verilmiştir:

$$\frac{U(s)}{E(s)} = Kp + Ki\frac{1}{s} \tag{2.3}$$

Şekil 2.10'da bir sisteme birim basamak referans girişi uygulandığında sistem çıkış eğrisi verilmiştir. Bu eğri geçici rejim ve kalıcı cevap kısımlarından oluşur. Bu tez çalışmasında Şekil 2.10'da görülen geçici rejim zaman tanım bölgesi kriterlerinden yükselme süresi, yüzde aşım ve yerleşme süresi değerleri, PI kontrolörlerin oransal ve integral kazanç parametreleri yarımıyla PID kontrolörlerin ise oransal, integral ve türevsel kazanç parametreleri yardımıyla ayarlanabilir. Böylece sistemin performans analizi yapılır. Burada yükselme süresi, literatürde yaygın kullanımında sistem cevabının %10 ile %90 arasında geçen zamanı gösterir. En büyük aşım, cevap eğrisinin tepe değeri ile referans giriş değeri arasındaki farkın referans giriş değerine bölünmesinden elde edilir. Yerleşme süresi ise sistem cevabının %2 veya %5 bandına giriş yaptığı ve bu bandın dışına çıkmadığı zamana kadar gecen süredir.



Sekil 2.10: Birim basamak giriş uygulanmış bir sistemin çıkış eğrisi ve performans kriterleri [16,17]

2.6.3. Bulanık Mantık Kontrol (Fuzzy Logic Control)

Bulanık mantık kontrolörler, özellikle karmaşık ve doğrusal olmayan sistemler için uygundur. Klasik mantıktan farklı olarak 0 ile 1 arasında değer alabilen kontrol yöntemidir. Fırçasız DC motor sistemi için bulanık mantık kontrolör kullanarak hız kontrolü yapılabilir. Bulanık mantık temel bileşenleri bulanıklaştırma, kural tabanı, çıkarım mekanizması ve durulaştırmadır. Bulanık mantık kontrolünün temel bileşenlerini ve işleyişini içeren çalışma yapısı Şekil 2.11.'de gösterilmiştir.

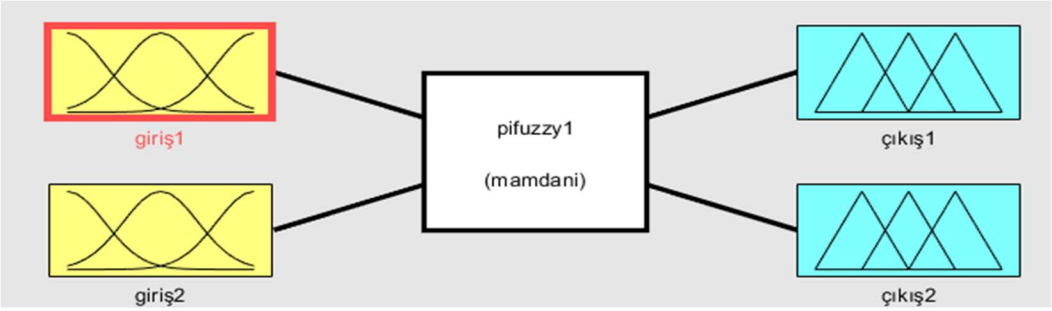


Sekil 2.11: Bulanık mantık kontrolör temel çalışma yapısı [36]

Bulanık mantığın bulanıklaştırma kısmında, sistemin giriş değerleri ölçülerek, bulanık kümeler oluşturulur. Giriş değerlerinin sayısal ifadelerine dilsel anlamda etiket verilir. Örneğin

bir motorun hızı 1000 rpm ise bulanık küme orta hızlı olarak isimlendirilebilir. Bulanık mantığın çıkarım mekanizması kısmında, sistemin üyelik fonksiyonları bulunur. Üçgen, yamuk ve gauss gibi çeşitli üyelik fonksiyonları kullanılmaktadır. Kullanılan en yangın üyelik fonksiyonu üçgen üyelik fonksiyonudur.

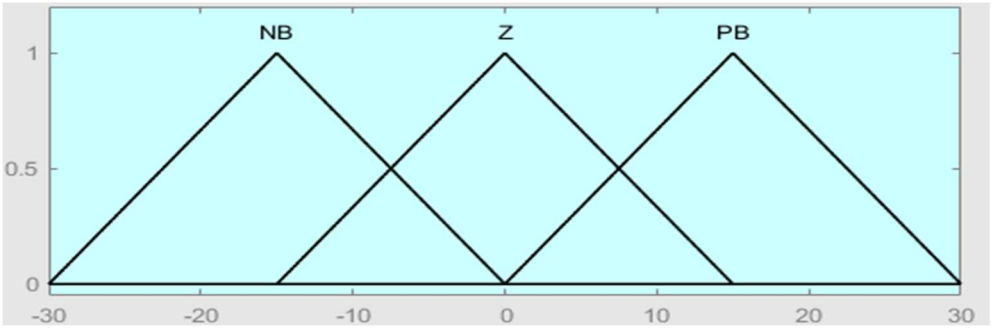
Bulanık mantıkta tanımlanan bulanık kümelerin [0 1] arasında derecelendirilmesi üyelik fonksiyonları tarafından yapılır. Üyelik fonksiyonlarını akademik dilsel anlamda belirtmek için PB (pozitif büyük), PS (pozitif küçük), Z(sıfır), NB (negatif büyük), NS (negatif küçük) vb ifadeler kullanılır. Bu fonksiyonlar kural tablosu kullanılarak değerlendirilir. Kural tablosu sonucunda oluşturulan değerleri sayısal ifadelere dönüştüren bileşen, bulanık mantığın durulaşma kısmıdır. Bulanık mantığı oluşturan bileşenlerin simülasyon ortamında düzenlemesi yapılabilmektedir. MATLAB ortamında bulanık kontrolör çıkarımı Şekil 2.12'de gösterilmiştir [15, 35].



Şekil 2.12: MATLAB ortamında bulanık mantık çıkarımı [15, 35]

Şekil 2.12'de simülasyon ortamında bulanık mantık giriş değişkenleri olarak giriş1 ve giriş2 olarak gösterilmiştir. Giriş değerleri için uygun bulanık kümeler tanımlanır. Bulanık kümelerin aitlik dereceleri üyelik fonksiyonları kullanılarak belirlenir. Bulanık mantık çıkarım mekanizması kullanılarak bulanık çıkış değerleri oluşturulur. Bulanık çıkışlar çıkış1 ve çıkış2 olarak Şekil 2.12'de ifade edilmiştir.

Bulanık mantık kontrolünün değişkenlerinden giriş değişkeninin üyelik fonksiyonlarının ve aralıklarının MATLAB ortamında tanımlanması Şekil 2.13'te görülmektedir. Şekil 2.13'te, örnek olarak seçilen fonksiyon üçgen üyelik fonksiyonudur. Bulanık mantık örnek kural tablosu ise Tablo 2.3'te gösterilmiştir.



Şekil 2.13: MATLAB ortamında üyelik fonksiyonu grafiği [15,35]

Tablo 2.3: MATLAB ortamında bulanık mantık kural örneği [15]

de \ e	NB	Z	PB
NB	NB	PB	PB
Z	Z	NB	Z
PB	PB	PB	PB

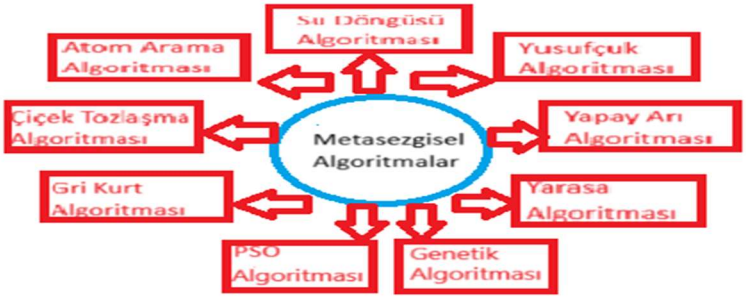
Kontrol edilmek istenen sisteme göre kural tablosu değişebilmektedir. Bu örnek olarak verilen kural tablosunda her bir giriş için 3 üyelik fonksiyonu kullanılmıştır. Üyelik fonksiyonları NB (negatif büyük), Z (sıfır), PB (pozitif büyük) olarak isimlendirilmiştir.

3. YÖNTEM

3.1. METASEZGİSEL OPTİMİZASYON ALGORİTMALARI

Metasezgisel algoritmalar doğadan esinlenerek oluşturulmuş, birçok karmaşık sistemin kontrolünün ve optimizasyon problemlerinin çözümünde kullanılan yöntemlerdir. Metasezgisel algoritmalar, çözüm için geniş bir arama bölgesi oluşturur ve bu arama bölgesinde en iyi çözümü bulmaya yarar. Bu algoritmalar doğadaki canlıların yaşam biçimlerinden, fiziksel ve evrimsel özelliklerinden ilham alır. Bu algoritmalar rastgelelik ile arama içerisinde büyük arama yapılmasını sağlanır.

Pek çok sistemde kullanıldığı gibi bu tezde ele alınan fırçasız bir DC motorun kontrolünde de bazı metasezgisel optimizasyon algoritmaları kullanılarak motor hızının ve torkunun iyileştirilmesi ile daha verimli performans sergilemesi sağlanabilir. Bu bölümde bu tezde kullanılan metasezgisel optimizasyon algoritmalardan yusufçuk optimizasyonu (DA), su döngüsü optimizasyon algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa optimizasyon algoritması (BA) ve çiçek tozlaşma optimizasyon algoritması (FPA) hakkında gerekli bilgiler, algoritmaların sözde (pseudo) kodları ve akış diyagramları anlatılmıştır. Metasezgisel algoritmalar 3 farklı uygunluk fonksiyonuna göre seçilebilir ve sistemlerin kontrolünde bu uygunluk fonksiyonları kullanarak sistemin hedefe en hızlı şekilde ulaşılabilir bu uygunluk fonksiyonları IAE (mutlak hata integrali), ITAE (zaman ağırlıklı mutlak hata integrali) ve ISE (karesel hata integrali) uygunluk fonksiyonlarıdır [30]. Aşağıdaki Şekil 3.1'de bu tezde kullanılan yukarıda bahsi geçen dokuz adet metasezgisel algoritmalar için bir şema gösterilmiştir.



Şekil 3.1: Tez çalışmasında kullanılan metasezgisel optimizasyon algoritmaları şeması

3.1.1. Genetik Algoritma (GA)

Metasezgisel algoritmalar arasında, Genetik algoritma (GA), biyolojik evrim sürecinden esinlenen iyi bilinen bir algoritmadır ve JH Holland tarafından önerilmiştir [19]. Temel yapısı kromozom, mutasyon, seçilim, çaprazlama olarak belirlenmiştir [19, 37-42]. Kromozom ile genetik algoritmalarda gerçek sayı dizileri oluşturulur. Bu dizilere gen adı verilir. Seçim, popülasyondaki bireylerin seçilmesidir. Çaprazlama, seçilen bireylerin yer değiştirerek yeni bireyler üretilmesidir. Mutasyon, kromozomlar arasında genetik değişimlerin oluşturulmasıdır. Aşağıdaki Şekil 3.2'de genlerin dizilimi resmi gösterilmiştir.



Şekil 3.2: Gen dizilimi

Genetik algoritmaya popülasyon değerlerinin rastgele oluşturulmasıyla başlanılır. Uygunluk fonksiyonu oluşturulur. Ardından çaprazlama ve mutasyon yapılarak en iyi değerler oluşturulmaya çalışılır. Eşitlik (3.1) ve (3.2) denklemleriyle çaprazlama yapılır.

$$C_{g1} = \alpha P_{g1} + (1-\alpha) P_{g1} \tag{3.1}$$

$$C_{g2} = \alpha P_{g2} + (1-\alpha) P_{g2} \tag{3.2}$$

Genetik algoritmalarda çaprazlama ve mutasyon yapılarak her iki bireyin özelliklerini taşıyan yeni çözümler üretilmesi sağlanır. Çaprazlama sayesinde bireylerin bilgileri dışında yeni özellikler üretilmesi sağlanır. Çeşitlilik arttırılır. Denklemlerde α katsayısı iki ebeveyn genetik değerlerini hangi oranda birleştireceği konusunda bilgi veren ve 0 ile 1 arasında değer oluşturan sabit çarpandır. P_{g1} ve P_{g2} çaprazlama sonucunda oluşturulan ebeveyn bireylerdir. C_{g1} ve C_{g2} ise yeni oluşturulan ebeveyn bireylerdir. Bu değerler ebeveyn bireyler arasında oluşturulan genetik bilgilerdir. Mutasyon işlemi ile rastgele bir noktada genetik bilginin değişimi, denklem (3.3)'te ifade edilmiştir.

$$X_{ki} = X_{alt} + (X_{ust} - X_{alt}) rand() \tag{3.3}$$

Denklemde X_{ust}, X_{alt} değerleri popülasyonun alt ve üst değerlerini gösterir. X_{ki} yeni mutasyon sonucunda elde edilen denklemlerle oluşturulur. Çaprazlama ve mutasyon yapıldıktan sonra elit bireylerin bir sonraki nesile aktarılarak nesillerin en iyi çözümleri korunmaya çalışılır. Maksimum iterasyon sağlanana kadar algoritma devam eder.

Genetik algoritması (GA) sözde (pseudo) kodu aşağıda gösterilmiştir [19]:

Başlangıç parametrelerini yüklenir.
Genetik başlangıç popülasyonu oluşturulur.
Genetik popülasyonu uygunluk fonksiyonu oluşturulur.
***while** (k<(maksimum iterasyon sayısı))*
***for** (her gen için N)*
Uygunluk değerlerine göre mevcut popülasyondan bireyler seçilir.
Genetik algoritma çaprazlama işlemi uygulanır. (3.1)- (3.2.)
Genetik algoritma mutasyon işlemi uygulanır. (3.3.)

end for

Popülasyon sıralanması ve en iyi bireylerin korunur.

Uygunluk değerlerinin hesaplanır.

En iyi birey çözümü elde edilir.

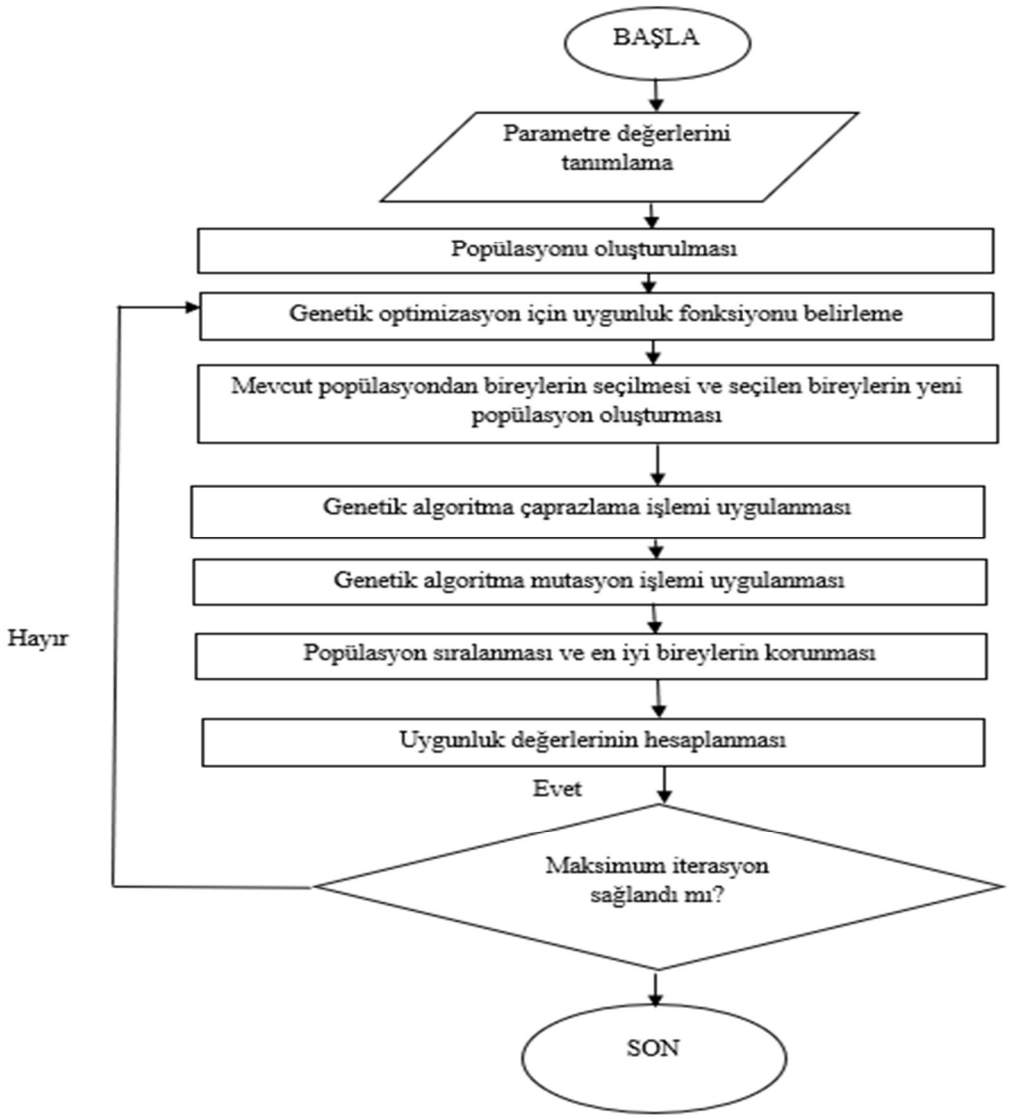
end

k=k+1

end while

Sonuçları göster

GA algoritmanın genel yapısını anlatan akış diyagramı, Şekil 3.8'de gösterilmiştir [19].



Şekil 3.3: Genetik algoritma akış diyagramı [19]

3.1.2. Parçacık Sürü Optimizasyon Algoritması (PSO)

Parçacık sürü optimizasyonu doğadaki canlıların kuş ve balık sürü davranışlarından esinlenerek Kennedy ve Eberhart tarafından oluşturulan metasezgisel bir optimizasyon algoritmasıdır [20]. Parçacık sürü optimizasyonu sürüdeki her bir canlının hem bireysel hem de sürüdeki diğer canlıların davranışlarından deneyim kazanarak öğrenilen bir optimizasyondur. Algoritmada sürüdeki her bir canlının hızı ve davranışlarının konumları dikkate alınır ve her adımda sürüdeki canlıların konumlarının ve hızlarının en iyi güncellemesi yapılarak en iyi çözüm aranır [20, 43-45]. Şekil 3.4'te sürü halinde uçan kuşların birlikte hareketi görselleştirilmiştir.



Şekil 3.4: Kuş sürüsü

Parçacık sürü algoritmasında N adet parçacıklı bir sürü için tüm parçacıkların başlangıç hızı ve konumu mevcuttur. Konum ve hız vektörleri sırasıyla X_i= (i=1,2, 3,.., N) ve V_i= (i=1,2,3…, N) olarak ifade edilir. Bu vektördeki değerler rastgele olmak kaydıyla alt ve üst sınırların içerisinden seçilir. Seçilen popülasyonların uygunluk fonksiyonu oluşturulur. Her bir parçacık için en iyi bireysel ve global konumlar oluşturulur. Bu değerler sırasıyla X_{pbest}ve X_{gbest} olarak ifade edilir. En iyi konum ve global konumlar belirlendikten sonra hız ve pozisyon güncellemesi yapılır. Aşağıdaki (3.5) ve (3.6) denklemlerinde hız ve pozisyonlar gösterilir.

$$V_k^i = wV_k + c1\left(X_{pbest} - X_k\right) + c2rand()(X_{gbest} - X_k) \tag{3.5}$$

$$X_{k+1}^i = X_k^i + V_k^i \tag{3.6}$$

(3.5) ve (3.6) denklemleri ile parçacıkların hız ve konum güncellemeleri yapılmaya çalışılır. Bu denklemlerde c1 ve c2, sosyal ve bireysel öğrenme katsayılarını ifade etmektedir. X_k^i ve V_k^i sırasıyla parçacığın mevcut hız ve konumunu gösterir. Bu hız ve konum vektörleri i' inci parçacığın k'ıncı iterasyonundaki değerlerini ifade etmektedir. X_{k+1}^i ise güncellenmiş konum vektörlerini gösterir. Her parçacığın pozisyonu önceki pozisyon vektörüne eklenerek güncellenir. Ardından parçacıklar güncellenerek en iyi konum bulunur. Her bir parçacığın en iyi konumunu bulmak için gerekli koşullar denklem (3.7.)'de gösterilmiştir.

$$X_{k,pbest} = \begin{cases} X_k & eğer \quad f(X_k) > f(X_{k,pbest}) \\ X_{pbest} & eğer \quad f(X_k) <= f(X_{k,pbest}) \end{cases} \tag{3.7}$$

Denklem (3.7.) her bir parçacığın mevcut pozisyonlarına göre en iyisini seçmek için karşılaştırma yapar. X_{pbest}^k pbest parçacığın k'ıncı iterasyonda bireysel en iyi konumunu, X_v ise parçacığın k'inci iterasyonda mevcut konumunu ifade eder. Parçacığın uygunluk fonksiyonun oluşturduğu konum değerlerinin düşük olması istenilen değere ulaşmayı sağlar. Mevcut pozisyon en iyi bireysel konum değerlerine göre güncellenir. Maksimum iterasyona kadar en iyi bireysel konum değerleri belirlenme işine devam edilir. Elde edilen en düşük uygun değerine ait X_{gbest}^k global konum değeri bulunur.

PSO algoritmasının sözde (pseudo) kodu aşağıda gösterilmiştir [20]:

Parçacık sayısı(N), maksimum iterasyon(v) ve diğer parametreler belirlenir.
Parçacık c1 ve c2 sabit katsayıları oluşturulur.
Parçacık başlangıç pozisyonu Xi ve hızları Vi rastgale belirlenir.
Her bir parçacığın bireysel konumunu X_{pbest} ve X_{gbest} kaydedilir.
***while** (k <(maksimum iterasyon))*
***for** (her parçacık sayısı için N)*
Parçacık uygunluk değerleri hesaplanır.
end
if
Parçacığın pozisyonu alt sınırdan küçükse, pozisyonu alt sınıra eşitlenir. (3.7)
else if
Parçacığın pozisyonu üst sınırdan büyükse, pozisyonu üst sınıra eşitlenir.
end for

Parçacıkların hızları güncellenir. *(3.5)*

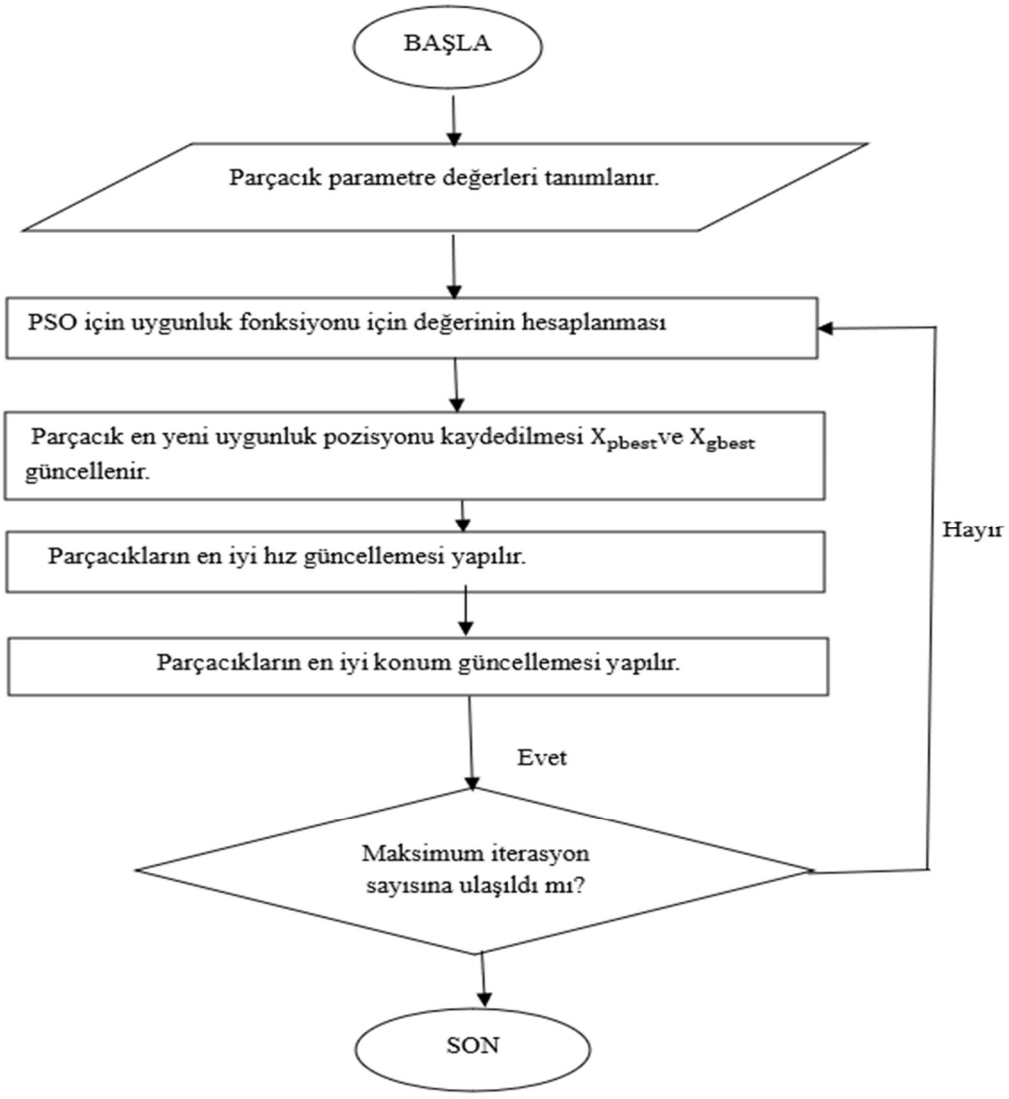
Parçacıkların konumları güncellenir *(3.6)*

k=k+1

end while

Sonuçları göster

PSO algoritmasının genel yapısını anlatan akış diyagramı Şekil 3.5'te gösterilmiştir [20].



Şekil 3.5. PSO akış diyagramı [20]

3.1.3. Yapay Arı Kolonisi Optimizasyon Algoritması (ABC)

ABC algoritması, arıların yiyecek arama davranışlarından esinlenerek Karaboğa, tarafından oluşturulan metasezgisel bir optimizasyon algoritmasıdır [21]. İşçi, gözcü, kaşif arılardan oluşan bir arı kolonisi belirli görevleri içeren bir hiyerarşik yapıya sahiptir. İşçi arılar yiyeceklerin kovana taşınması görevini üstlenmiştir ve bunları kovana taşırken de yiyecek kaynağı hakkındaki bilgileri gözcü arılara iletirler. İşçi arılar, taşıdıkları yiyeceklerin besin kalitesini iyileştirirler. Gözcü arılar, işçi arıların ilgilendiği besin kaynaklarını değerlendirir. Uygunluk değerlerine bağlı olarak kaynak seçimi yapar ve daha iyi çözümler elde etmeye çalışır. Kaşif arılar, işçi arılar ve gözcü arılar tarafından iyileştirilmeyen kaynaklar olma durumunda görev alırlar. Kaşif arılar yiyecek kaynağı keşfetmekle ve tükenen kaynakları yeniden aramakla görevlidirler [21, 46-49]. Şekil 3.6'da bir arı sürüsü resmi gösterilmiştir.



Şekil 3.6: Arı sürüsü

Yapay arı algoritması başlangıç çözüm değerlerini oluşturan bağıntı, denklem (3.8) ile verilmiştir.

$$X_{ij} = X_{alt} + (rand(0,1)(X_{ust} - X_{alt})) \tag{3.8}$$

Burada X_{ij} besin kaynağı konumu ifade etmektedir. Kaynakların rastgele bir şekilde çözüm uzayında bulunması sağlanır. Denklem (3.8)'de ki X_{ust} ve X_{alt} ifadeleri, çözümün alt ve üst sınırlarını göstermektedir.

İşçi arılar tarafından yeni besin kaynakları üretebilmesi denklem (3.9.)'da ifade edilir:

$$X_{k+1}^{i} = X_{k}^{i} + \phi_{ij}(X_{k}^{i} - X_{k}^{k1}) \tag{3.9}$$

Burada X_k^i işçi arıların mevcut besin kaynak konumunu, X_{k+1}^iişçi arıların oluşturduğu yeni besin kaynağı konumunu, X_k^{k1} rastgele seçilen bir kaynağın çözümünü, ϕ_{ij} değeri ise [-1 1] arasında olan rastgele bir sayıyı ifade eder. İşçi arılar mevcut besin kaynağının etrafındaki yeni besin kaynaklarını üretmeye çalışır. Mevcut çözümlerin varlığında yeni çözümler denklem (3.9.) ile bulunur.

İşçi arılar tarafından besin kaynakları taşınan bilgilerin gözcü arılar tarafından kullanılmasını gösteren olasılık değerleri, denklem (3.10)'da verilmiştir.

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_i} \tag{3.10}$$

Burada p_i kaynağın seçilme olasılığını, fit_i uygunluk değerini ve SN ise toplam kaynak sayısını ifade etmektedir. Gözcü arıların kaynak seçiminde kullandıkları uygunluk fonksiyonuna göre seçilme olasılıkları bu denklemde ifade edilmiştir. p_i olasılık değeri, 0 ile 1 arasında bir değer almaktadır. Seçilme olasılık değeri [0 1] aralığında rastgele oluşacak bir değerden büyükse, yeni besin kaynakları denklem (3.9) ile üretilir. Üretilen besin kaynakları ve mevcut besin kaynakları uygunluk değerine bağlı olarak karşılaştırılarak, en uygun besin kaynağı seçimi yapılır. Gözcü arıların yeni besin kaynakları seçiminin ardından, besin kaynaklarının tüketilme durumu araştırılır. Besin kaynaklarının tüketilmesi durumunda yeni kaynaklar üretilir.

Yapay arı kolonisi algoritması (ABC) optimizasyonu sözde kodu aşağıda gösterilmiştir [21]:

ABC algoritması parametreleri tanımlanır.

Rastgele başlangıç popülasyonlarının oluşturulur.

Uygunluk fonksiyonu belirlenir.

***while** (k<Maksimum iterasyon sayısına ulaşana kadar devam ettir)*

***for (**her bir arı sayısı için)*

Yeni komşu çözüm üretir. (3.9.)

Komşu çözüm için uygunluk değeri hesaplanır.

***if** (Mevcut çözüm ve komşu çözüm karşılaştırılarak açgözlü seçim uygula sorgulanır.)*

end

Olasılık temelli seçim yapılır.

***if** (Tükenen kaynakları kontrol edilir.)*

Kaynak tükenmişse, rastgele yeni kaynak üretilir. (3.8.)

end

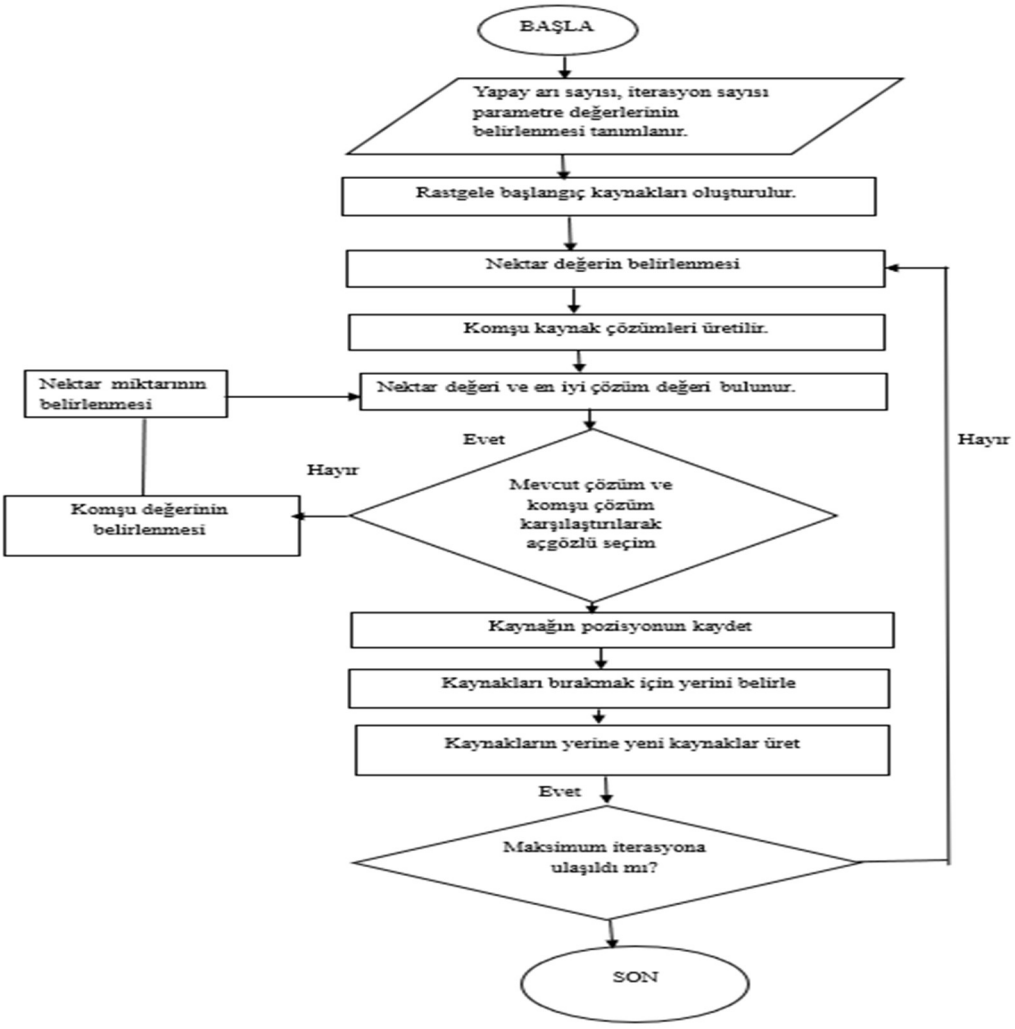
k=k+1

En iyi çözüm değeri belirlenir.

end while

Sonuçları göster

ABC algoritmasının genel yapısını anlatan akış diyagramı Şekil 3.7'te gösterilmiştir [21]:



Şekil 3.7 ABC algoritması akış diyagramı [21]

3.1.4. Çiçek Tozlaşma Optimizasyon Algoritması (FPA)

2012 yılında Xin-She Yang tarafından çiçeklerin hayatta kalması ve çiçek popülasyonlarının üretilmesi üzerine ilham alınarak oluşturulmuş bir algoritmadır [22]. Çiçek tozlaşma optimizasyon algoritması, çiçeklerin biyolojik tozlaşma sürecini model alır [22]. Global ve yerel tozlaşma, algoritmanın temel bileşenleridir. Global tozlaşma biyolojik olarak uzak çiçekler arasında tozlaşmanın gerçekleşmesini ifade eder. Tozlaşma uzun mesafede polen taşıyan böcekler ile gerçekleştirilir. Levy uçuşu global tozlaşma bileşeninde kullanılmaktadır. Levy uçuşu algoritmaların geniş alanlarda çözüm aramalarını sağlar. Yerel tozlaşma yakın mesafede bulunan çiçekler arasındaki tozlaşmayı ifade eder. Yerel tozlaşma aynı bitki ve yakın bitkiler arasındaki polen değişimidir ve polenin kısa mesafelere taşınmasında ölçüt olan katsayıya çiçek üretim katsayısı (p) denir. Bu, global ve yerel tozlaşma süreçleri arasındaki olasılık değeridir. Bu katsayı [0 1] arasındaki belirlenen değerler alarak yerel ve global tozlaşma süreci gerçekleştirilir [22, 50-51]. Şekil 3.8'de çiçeklerin tozlaşması gösterilmiştir.



Şekil 3.8: Çicek tozlaşması [50]

Çiçek tozlaşma algoritmasının başlangıcında rastgele popülasyon oluşturulur. Başlangıç popülasyonun uygunluk değeri belirlenir. Ardından her iterasyonda çicek tozlaşma algoritmasının temel bileşenlerini oluşturan global tozlaşma ve yerel tozlaşma değerleri belirlenmeye çalışılır. Global tozlaşma, denklem (3.11)'de gösterildiği gibi hesaplanır.

$$X_{k+1} = X_k^i + L(g - X_k^i) \tag{3.11}$$

Burada, *L*, levy kuş adım vektörüdür. Levy uçuş kuş adım vektörü canlıların rastgele arama davranışları için kullanılır. g parametresi küresel optimum en iyi çözüm değeridir. Küresel optimum g parametresi 0 ile 1 arasında değer alır. X_{k+1} değeri mevcut çözümü, X_{k+1}^{i} ifadesi global tozlaşma sırasında k'inci iterasyonda oluşan çözüm vektörünü ifade etmektedir. Denklem (3.12)'de global tozlaşma için kullanılan levly uçuşu adım vektörünün hesaplanması gösterilmiştir.

$$L \sim \frac{\Gamma(\lambda)\sin(\frac{\pi\lambda}{2})}{\pi} \frac{1}{s^{1+\lambda}} \quad (s \gg s_0 > 0) \tag{3.12}$$

Levy uçususu denkleminde Γ(λ) gama fonksiyonunu, s adım büyüklüğünü temsil etmektedir. Adım büyüklüğü 0'dan büyük adımlar için gereklidir. Levy uçusu, kısa mesafeli ve uzun mesafeli adımları bir araya getirip çözüm uzayında etkili bir arama yapılmasını sağlar. Denklem (3.13)'te çiçeğin yerel tozlaşma denklemi verilmiştir.

$$X_{k+1} = X_k^i + e_1(X_k^{i1} - X_k^{i2}) \tag{3.13}$$

Bu denklemde e_1 değeri, 0 ile 1 arasında oluşturulan rastgele bir sayıyı temsil etmektedir. X_{k+1} güncellenmiş yeni çözümü gösterirken; X_k^{i1} ve X_k^{i2} ise popülasyonda rastgele seçilen farkı çiceklere ait polen türleridir. Bu değerlerle mevcut değerler güncellenir. Yakın alanlarda arama yapılarak yerel tozlaşma gerçekleştirilir. Denklem (3.13)'te ile çiçeğin yerel tozlaşma denklemi hesaplanır.

FPA algoritması sözde (pseudo) kodu aşağıda gösterilmiştir [22, 50]:

Çiçek optimizasyonu algoritması parametreleri tanımlanır
Başlangıç popülasyonlarının oluşturulur.
Başlangıç popülasyonunda en iyi çözümü bulunur.
[0,1] ile arasında p olasılık değeri oluşturulur.
***while** (k<Maksimum iterasyon sayısı)*
***for** (tüm çiçeklerin popülasyon sayısı)*
***if** r<p*
Levy uçuşu ile global tozlaşma yapılır. (3.12)

else

Yerel tozlaşma değerlerini hesaplanır. (3.13)

end if

En iyi çözüm için güncelleme yapılır.

end for *(En iyi çözümün elde edilir.)*

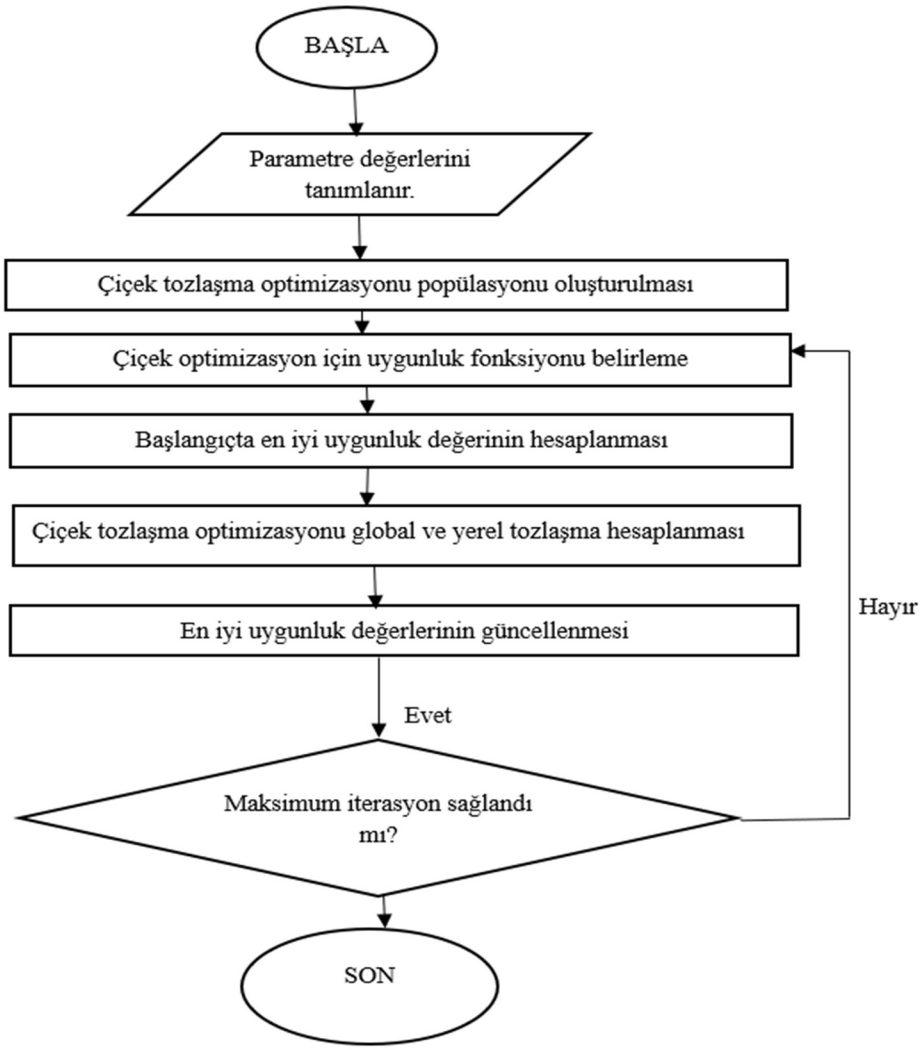
end

k=k+1

end while

Sonuçları göster

FPA algoritmasının genel yapısını anlatan akış diyagramı Şekil 3.9'da gösterilmiştir [22, 49]:



Şekil 3.9: FPA akış diyagramı [22, 50]

3.1.5. Gri Kurt Optimizasyon Algoritması (GWO)

Gri kurt algoritması gri kurt hayvanların avlanması üzerine oluşturulan metasezgisel bir algoritmadır [23]. Algoritmada sosyal sınıflandırma dikkate alınır. Algoritmanın temel bileşenleri alfa (α), beta (β), delta (δ), omega (ω) kurt hiyerarşisi, avlarının yerlerini bulmaya çalışma, avını çevrelemesi ve avlarına saldırmalarıdır. Bu bileşenler kurtların sosyal hiyerarşilerini ifade etmektedir. Alfa kurt, kurtların liderliğini; beta kurt, alfa kurttun yardımcılığını; delta kurt, beta ve alfa kurttun yardımcılığını ve omega kurt ise alt tabaka olarak diğer kurtların keşif ve avlanma işlerine yardımcı olurlar [23, 48, 52, 53]. Gri kurtların avlarını çevreleyerek onları sıkıştırması ve saldırması özellikleri dikkate alınarak bu metasezgisel optimizasyon algoritması oluşturulmuştur. Sekil 3.10'da bir gri kurt ve hiyerarşisi gösterilmiştir.



Şekil 3.10: Gri kurt ve hiyerarşisi

Başlangıçta gri kurt algoritmasında popülasyon değerleri oluşturulur. Alfa, omega ve delta kurt olarak popülasyonun uygunluk değerleri oluşturulur. Kurt popülasyonları uygunluk değerleri en iyi konumdan üçüncü en iyi konuma doğru sırasıyla $X_\alpha, X_\beta, X_\delta$ olarak ifade edilebilir. Geriye kalan konumlar omega grubundandır. Kurt sürüsü tarafından avların çevrelenmesi ve kurtlar arasındaki mesafe eşitlik (3.14)-(3.20) denklemleri ile hesaplanır. Bu mesafeler kurtlar arasındaki her bir konumun belirlenmesinde ve sürünün global olarak çözüme ulaşmasını sağlar. Kurtların sürüdeki lider kurtlara göre olan yakınlığı belirlenir.

$$D_a = \left|C_1 X_\alpha - X_k^i\right| \tag{3.14}$$

. $$X_1 = X_\alpha - A_1 D_a \tag{3.15}$$

$$D_b = \left|C_2 X_\beta - X_k^i\right| \tag{3.16}$$

$$X_2 = X_\beta - A_2 D_b \tag{3.17}$$

$$D_c = \left|C_3 X_\delta - X_k^i\right| \tag{3.18}$$

$$\mathrm{X_3 = X_\delta - A_3 D_c} \tag{3.19}$$

$$X_{yeni} = \frac{X_1 + X_2 + X_3}{3} \tag{3.20}$$

Burada $\mathrm{X_\alpha}$, $\mathrm{X_\beta}$, $\mathrm{X_\delta}$ sırasıyla alfa, beta ve delta kurtlarının pozisyonlarını; $\mathrm{D_a}$, $\mathrm{D_b}$, $\mathrm{D_c}$ mevcut kurt ile lider kurtlar arasındaki mesafeyi ifade etmektedir. Kurtlar arasındaki mesafe belirlendikten sonra kurtlar arasındaki pozisyon (3.15), (3.17) ve (3.19) denklemleri ile hesaplanır. $\mathrm{A_1}$, $\mathrm{A_2}$ ve $\mathrm{A_3}$ sembolleri ile ifade edilen değerler kurtlar arasındaki mesafeyi belirleyen çarpanlardır. Bu denklemlerle kurtların pozisyonu belirledikten sonra, kurtlar arasındaki yeni pozisyonlar, kurtlar arasındaki pozisyonların ortalaması alınarak elde edilir. Bu denklem (3.20)'de gösterilmiştir.

$$A = 2ar - a \tag{3.21}$$

(3.21) denklemi kurtlar arasındaki mesafenin dengelenmesi ile elde edilir. Gri kurt algoritması geniş alanda arama yapılmasını sağlar. R sayısı, 0 ile 1 arasında rastgele bir sayıyı gösterirken; a ise 0 ile 2 arasında adım (iterasyon) sayısını gösterir.

GWO algoritması sözde (pseudo) kodu aşağıda gösterilmiştir [23]:

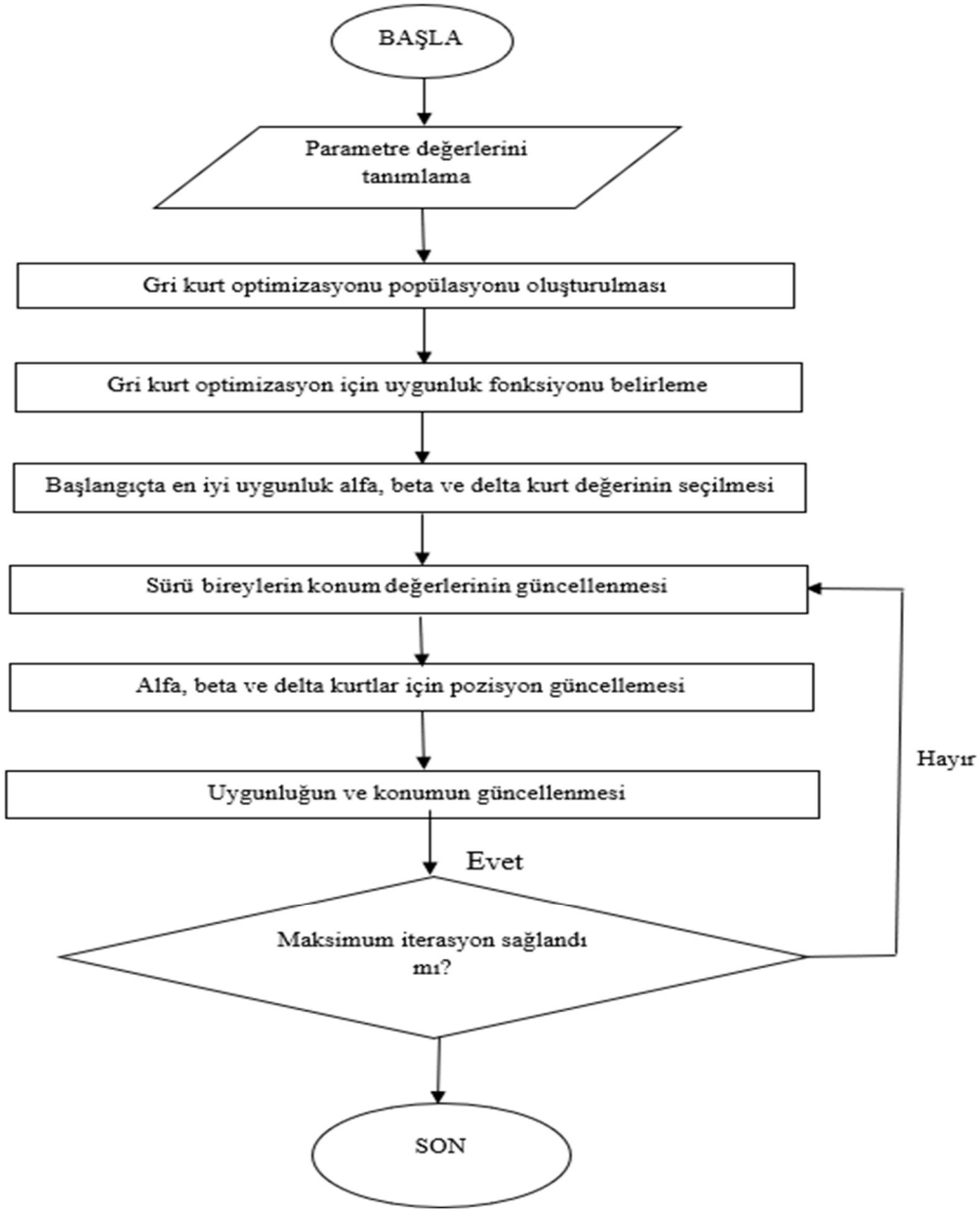
Gri kurt algoritması parametreleri tanımlanır.
Gri kurt başlangıç popülasyonlarının oluşturulur.
Gri kurt optimizasyon için uygunluk fonksiyonu belirlenir.
Başlangıçta en iyi uygunluk alfa, beta ve delta kurt değerinin seçilir.
***while** (k<Maksimum iterasyon sayısına ulaşana kadar devam ettirme)*
***for (**her bir kurt için)*
Alfa, beta ve delta kurtlar için pozisyon güncellenir. (3.20)
end for
Her kurtun uygunluk değerleri belirlenir.
En iyi çözüm değerlerinin güncellemesi yapılır ve kaydedilir.

k=k+1

end while

Sonuçları göster

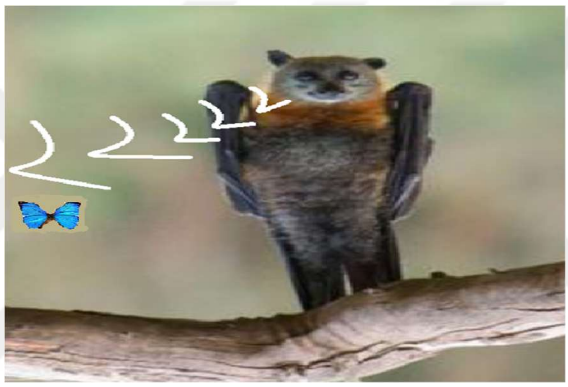
GWO algoritmasının genel yapısını anlatan akış diyagramı Şekil 3.11'de gösterilmiştir [23]:



Şekil 3.11: GWO akış diyagramı [23]

3.1.6. Yarasa Optimizasyonu Algoritması (BA)

Yarasa optimizasyonu algoritması, yarasaların avlanma davranışları model alınarak oluşturulmuş ve yarasaların yankı özelliğine göre oluşturulmuş bir metasezgisel algoritmadır [24]. Algoritmanın temel bileşenleri yankı, hız ve frekans ayarlanması, küresel ve yerel arama yeteneğidir. Yankı, yarasaların ultrasonik ses dalgaları oluşturma özelliği kullanılarak nesnelerin konumu ve uzaklığı ayarlamasıdır. Algoritmada hız ve frekans ayarlaması yapılarak yarasaların avlarını hızlı ve verimli şekilde bulması amaçlanır [24, 54-56]. Küresel ve yerel arama yeteneği ile yarasaların çevrelerini keşfederek en iyi çözümün bulunması sağlanmaya çalışılır. Şekil 3.12.'de yarasanın avını bulması resmedilmiştir.



Şekil 3.12: Yarasanın avını bulması

Başlangıç aşamasında popülasyon değerleri ve algoritma için uygunluk fonksiyonu oluşturulur. Algoritmanın her iterasyonunda yarasanın hız ve frekans güncellemeleri yapılarak yarasaların çözüm uzayında yönlendirilmesi sağlanır. Yarasaların her birinin hızları V_i ve konumları X_i olarak ifade edilir. Her bir yarasanın güncellenen hızı V_{k+1}^i ve konumu X_{k+1}^i olarak algoritmada tanımlanır. Eşitlik (3.22) ve (3.23) ile yarasaların hız ve frekans davranışlarına bağlı en iyi çözüm oluşturulur.

$$Q_i = Q_{min} + r(Q_{min} - Q_{max}) \tag{3.22}$$

$$V_{k+1}^i = V_k^i + (X_{yi} + X_{ybest})Q_i \tag{3.23}$$

Denklem (3.22)'deki Q_i yarasa için belirlenen frekansı, Q_{min} frekans minumum değeri, Q_{max} frekans maksimum değeri ve r 0 ile 1 arasında oluşturulan rastgale sayıyı ifade etmektedir. Denklem (3.23)'deki X_{yi} yarasaların mevcut değerlerini ve X_{ybest} ise o anki en iyi çözüm

değerini ifade eder. Aşağıdaki denklemle yarasaların yerel rastgele yürüyüşü için çözüm değeri oluşturulur.

$$X_{yeni} = X_{eski} + \epsilon L_1 \quad (3.34)$$

Denklem (3.24)'te, L_1 zaman adımındaki ortalama ses şiddetini, ϵ rastgele -1 ile 1 arasındaki sayıyı, X_{eski} eski elde edilen çözüm değerini ve X_{yeni} yeni çözüm değerini göstermektedir. (3.25) ve (3.26) denklemleriyle ses şiddeti ve sinyal emisyon hızı güncellemesi yapılır. Ses şiddeti kullandıkları sinyali güçlendirmek için önemlidir. Sesin emisyon hızı da yarasaların yaydığı sinyalin hızını göstermektedir.

$$A_{k+1}^{i} = \propto_1 A_k^i \quad (3.25.)$$

$$z_{k+1}^{i} = z_k(1 - e - \gamma t) \quad (3.26.)$$

Buradaki denklemde z_i^{k+1} bir sonraki iterasyondaki sinyal emisyon hızını, z_k başlangıç sinyal emisyon hızını, γ güncelleme katsayısı değerini, A_{k+1}^{i} yarasaların bir sonraki iterasyondaki ses şiddetini, A_k^i mevcut iterasyondaki ses şiddetini ve $\propto_1$ayıflatma faktörünü ifade eder.

BA algoritmasının sözde kodu aşağıda gösterilmiştir [24, 54]:

Başlangıç parametreleri yüklenir.
Yarasa başlangıç popülasyonu oluşturulur.
Yarasa popülasyonu uygunluk fonksiyonu oluşturulur.
***while** (k < (maksimum iterasyon sayısı)*
***for** (her bir yarasa sayısı için)*
Yarasa frekans güncellemesi gerçekleştirilir. (3.22)
Yarasa hız güncellemesi gerçekleştirilir. (3.23)
Hata fonksiyonunu hesaplanır.
***if** (En iyi çözümü güncellemesi yapılır.)*
end
Ses şiddeti ve sinyal emülsiyon hızı güncellemesi yapılır. (3.25)-(3.26)

En iyi uygunluk değerinin oluşturulur.

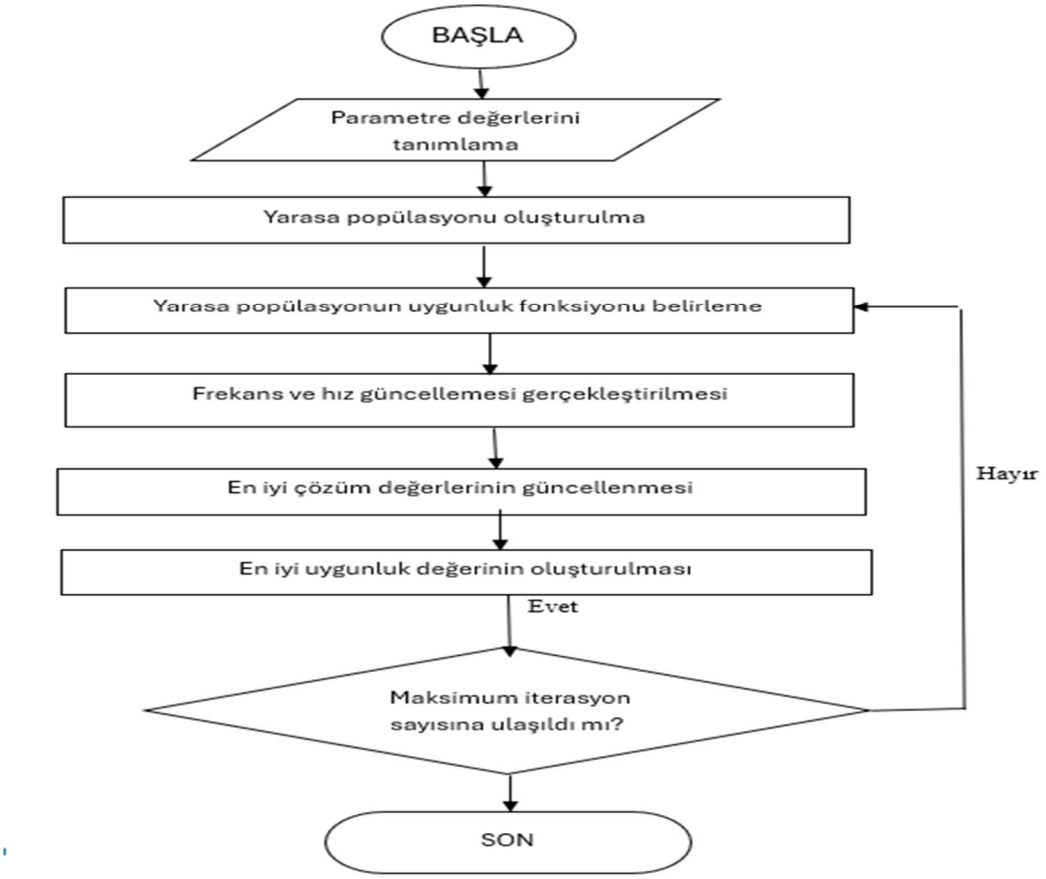
end

k=k+1

end while

Sonuçları göster

BA algoritmasının genel yapısını anlatan akış diyagramı Şekil 3.13'te gösterilmiştir [24, 54]:

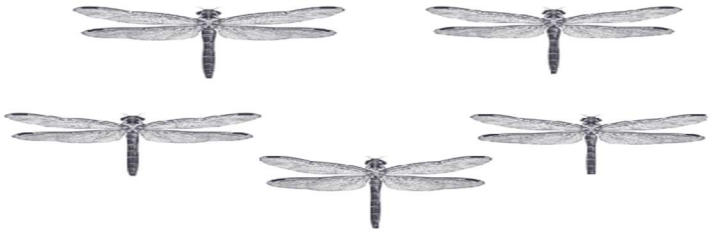


Şekil 3.13: Yarasa algoritması akış diyagramı [24, 54]

3.1.7. Yusufcuk Algoritması (DA)

Yusufçukların yön bulmaları, avlanma ve düşmanlardan kaçma davranışları dikkate alınarak oluşturulmuş bir metasezgisel yöntemdir [25]. Yusufçuk algoritmasının temel bileşenleri ayrışma, hizalanma, uyum, avlanma ve düşmanlardan kaçıştır. Ayrışma, yusufçukların arasındaki çarpışmayı engeller. Hizalama, yusufçukların belirli bir yönde

dizilmesidir. Uyum, yusufçukların belirli bir alanda yakınlaşmasıdır. Avlama, yusufçukların besin için hareket etmesidir. Diğer parametre düşmanlardan kaçış ise yusufçukların iyi olmayan bölgelerden uzaklaşmasıdır [25, 57]. Yusufcukların birbirleriyle hizalanması, Şekil 3.14'te gösterilmiştir.



Şekil 3.14: Yusufcukların birbirleriyle hizalanması

Algoritma başlngıcında popülasyonun rastgele çözüm kümesi oluşturulur. Her yusufçuğun ayrışma hizalama ve uyum vektörleri hesaplanır. Tüm yusufcukların sosyal ve avlanma davranışları denklemlerle ifade edilir. Ayrışma hizalama ve uyum denklemleri aşağıdaki denklemlerle verilmiştir. Denklem (3.27), yusufcukların ayrışma kuvvetini göstermekle birlikte yusufcukların birbirleriyle çarpışmasını önlemek ve uzak durmasını anlatır.

$$S_i = -\sum_j^N (X_i - X_j) \tag{3.27}$$

Burada X_i yusufcuk popülasyonunun i'ci pozisyonunu, X_j ise yusufcuk popülasyonun komşu popülasyonunu, S_i ayrışma kuvvetini temsil etmektedir. Denklem (3.28), yusufcukların hizalama kuvvetini göstermektedir. Bu denklemle yusufcukların komşularının hızlarına göre hizalanması sağlanır.

$$B_i = \frac{1}{N}\sum_{j=1}^N V_j \tag{3.28}$$

Burada, V_j komşu yusufcukların hız popülasyonunu, B_i yusufcukların uyum kuvvetini ifade etmektedir. Denklem (3.29)'da yusufcukların uyum kuvvet denklemi gösterilmektedir. Bu denklem yusufcukların komşularının ortaya doğru çekilebilmesini temsil etmektedir.

$$C_i = \frac{1}{N}\sum_{j=1}^N X_j - X_i \tag{3.29}$$

Yusufcukların sosyal ve avlanma davranışları için bu denklemler oluşturulduktan sonra yiyecek kaynağına çekilme ve düşmandan kaçmak için gerekli denklemler aşağıda verilmiştir:

$$F_i = X_{en} - X_i \tag{3.30}$$

$$Z_i = X_i - X_{kotu} \tag{3.31}$$

Denklem (3.30)'da yusufcukların en iyi yiyecek kaynağına yani çözüme doğru yönelmesi gösterilmiştir. Burada X_{en} en iyi çözüm pozisyonudur. (3.31) denklemi, yusufcukların düşmandan kaçma denklemidir. X_{kotu}, en kötü çözüm pozisyon değeridir. Bu en iyi ve en kötü değerler, ilk iterasyonda uygunluk fonksiyonuna göre bulunur. Her iterasyonda bulunan en iyi çözüm ve en kötü çözümler güncellenir. Elde edilen sosyal ve avlanma devranışlarının düşmandan kaçınma ve yiyecek kaynağına çekilmeye göre güncellemesi yapılır. Denklem (3.32)'de bu durum ifade edilir.

$$\Delta X_{k+1} = (e_2 Z_i + q_1 S_i + z_2 B_i + c C_i + f_1 F_i) + e_2 Z_i + k_1 \Delta X^k \tag{3.32}$$

Burada q_1 ayrılma ağırlık katsayısını, c uyum ağırlık katsayısını, k_1 eylemsizlik ağırlığını, f_1 besin faktörü değerini, e_2 düşman faktörünü ve z_2 sıralanma faktörü değerini ifade etmektedir. Ağırlık katsayıları [0 1] arasında değer almaktadır. ΔX_{k+1} yusufcukların her iterasyon güncellemesi sonucu elde edilen hız adım vektörüdür. Yusufcukların mevcut pozisyonunun güncelemesi denklem (3.33)'te gösterilmiştir.

$$X_{k+1} = \Delta X_{k+1} + X_k \tag{3.33}$$

Burada X_{k+1}, yusufcukların pozisyonlarının her itrerasyonda güncellenmesini ifade eder. Güncelleme yapıldıktan sonra levy uçuşu hesaplanır. Bu yusufcukların optimizasyona girmeyip geniş alanda uçuş yapmasını sağlar. Bu uçuş sayesinde yusufcukların daha geniş bölgelerde çözüm arayışı sağlanır. Bu durum, denklem (3.34)'de gösterilmiştir.

$$X_{k+1} = X_k + Levy(z_3)\, X_k \tag{3.34}$$

$$Levy(x) = 0.01 \frac{r1\sigma}{|r2|^{\frac{1}{\beta}}} \tag{3.35}$$

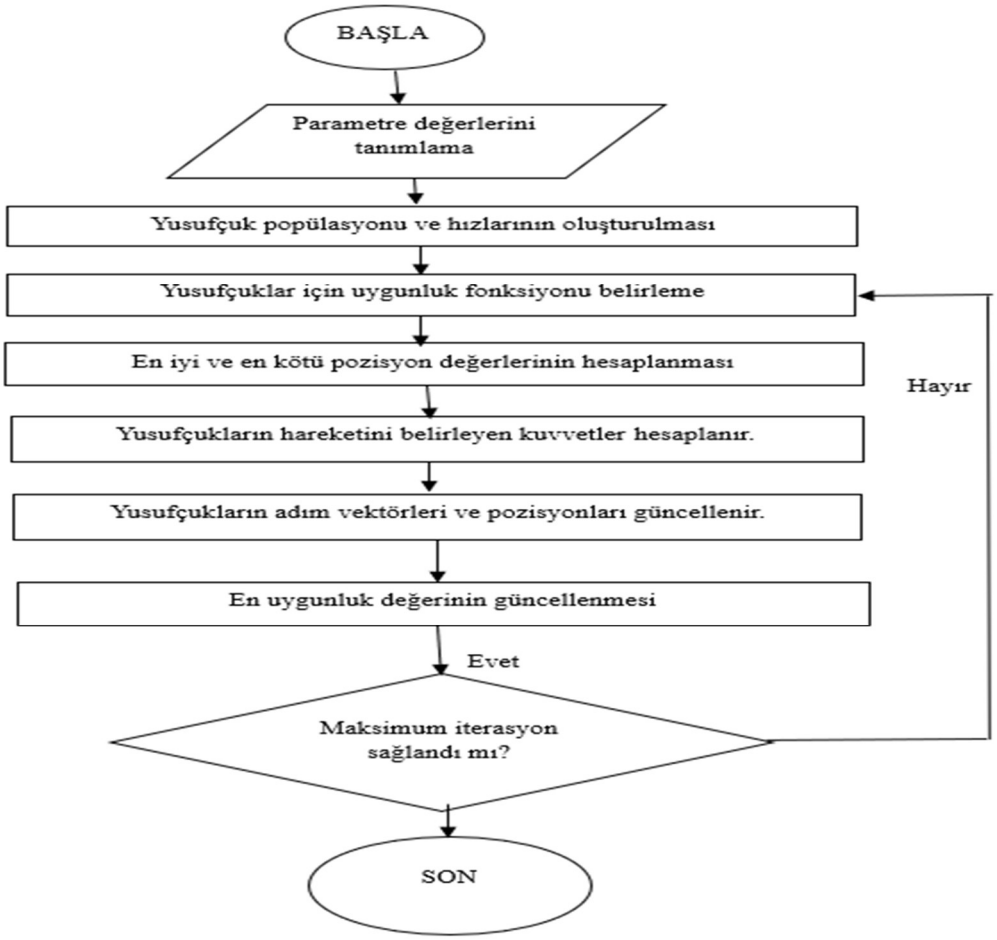
(3.34) denkleminde z_3 konum vektörünün boyutu ve σ levy uçuş katsayısıdır. Denklem (3.35)'te r1 ve r2 katsayıları 0 ile 1 arasındaki rastgele sayıları ifade etmektedir. β, 1.5 değerini alan sabit bir değerdir. Denklem (3.36) ile σ levy uçuş katsayısı aşağıdaki gibi hesaplanır:

$$\sigma = \left(\frac{\Gamma(1+\beta)sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta\left(2^{\frac{\beta-1}{2}}\right)}\right)^{\frac{1}{\beta}} \quad (3.36.)$$

DA algoritmasının sözde kod örneği aşağıdaki gibidir [25, 57]:

Yusufçuk algoritması parametreleri tanımlanması
Başlangıç popülasyonlarından her yusufcuk için hızları ve pozisyonları oluştur.
Uygunluk değerlerini oluştur.
***while** (k<Maksimum iterasyon sayısı)*
***for** (her bir yusufcuk sayısı N için)*
En iyi ve en kötü uygunluk en iyi çözüm değerlerini elde edilir.
Kuvvet değerleri(ayrışma ,hizalama ve uyum) hesaplanır. (3.27)-(3.28)-(3.29)
Komşuluk yarıçapını güncellenir.
***if** (Yusufçuk optimizasyonunda komşuluk var mı?)*
Adım vektörlerini ve pozisyonu güncellenir. (3.32)
else if
Konum vektörü güncellenir. (3.33)
else if
Yusufçuk optimizasyonu sınırlarının kontrolü yapılır.
end
k=k+1
end while
Sonuçları göster

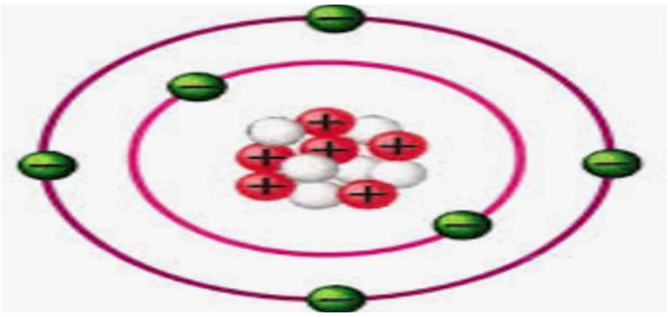
DA algoritmasının akış diyagramı Şekil 3.21'de gösterilmiştir [25, 57]:



Şekil 3.15: Yusufçuk algoritması akış diyagramı [25, 57]

3.1.8. Atom Arama Optimizasyon Algoritması (ASO)

Atom arama algoritması (ASO), atomların doğal hareketlerinden ve birbirleri ile olan etkileşimlerinden esinlenerek oluşturulmuş bir metasezgisel algoritmadır [26]. Atom arama algoritması atomların çekim kuvvetleri ve coulomb yasası üzerine kurulmuştur. Yapısı, çekim ve itme kuvvetleri, kuvvet hesaplama, enerji depolama içermektedir [26, 58, 59]. Her bir atomun toplam kuvveti, diğer tüm atomların etkileri göz önünde bulundurularak hesaplanır. Şekil 3.16'da atomların birbirleriyle olan etkileşimi gösterilmiştir.



Şekil 3.16: Atomların birbirleriyle olan etkileşimi

Atom arama algoritmasında ilk olarak başlangıç popülasyonu oluşturulur. Daha sonra elde edilen popülasyonun uygunluk fonksiyonu oluşturulur. Uygunluk fonksiyonu oluşturulduktan sonra atom arama algoritmasının atomların birbirlerini çekim veya itme kuvvetine bağlı olarak etkileri gözlenir. Her atom en iyi atoma doğru hareket eder ve bunlar en iyi bir atoma kovalent bağ ile bağlıdır. Atom sayısının birbirlerine olan kısıtı aşağıdaki denklemle ifade edilmiştir:

$$\theta_k^i = (|X_k - X_{best}|^2) - b_{best}{}^2) \tag{3.37}$$

Burada θ_i^k , atom sayısının uygunluk değerine bağlı oluşturulan değerdir. X_k, k'ıncı iterasyondaki atomların konumunu, X_{best}, atomun en iyi değerini ve b_{best} ise sabit bağ sayısını göstermektedir.

$$Pi_k^i = \lambda_t(X_k^{best} - X_k^{best})) \tag{3.38}$$

(3.38) kısıtlama kuvveti denkleminde Pi_l^k, atomların birbirleriyle itme ve çekme kuvvetlerini temsil etmektedir. λ_t zamanla değişen çarpan değeridir. X_{best}^k en iyi konumu, X_t^v değeri atomun d boyutundaki değeridir. $\lambda_t = \beta e^{-20t}$ zaman çarpımı bu değere göre hesaplanır. β ise çarpan ağırlığıdır. Bu değer çalışmada 0,2 olarak alınmıştır. (3.39) ve (3.40) denklemleri ile atomun kütlesi hesaplanır.

$$M_k^i = e^{-\frac{F_{iti}(t) - F_{ibest}(t)}{F_{itkotu}(t) - F_{ibest(t)}}} \tag{3.39}$$

$$M_k^i = \frac{M_k^i}{\sum_{j=1}^{N} M_l^k} \tag{3.40}$$

Burada M_i^k atomun kütlesini, $F_{iti}(t)$ atomun uygunluk fonksiyonu değerini, $F_{ibest}(t)$ en iyi uygunluk değerini ve $F_{ıtkotu(t)}$ ise en kötü değeri temsil eder. Komşuluk sayısını veren denklem aşağıdadır:

$$K(t) = N - (N-2) \cdot \sqrt{\frac{t}{k}} \tag{3.41}$$

Burada K(t) değeri komşuluk sayı değerini, N toplam atom sayını, k toplam iterasyon sayısını gösterir. Aşağıdaki (3.42) ve (3.43) denklemleriyle atomların hız ve pozisyonları güncellenir.

$$V_{k+1}^i = rand_k^i V_k^i + Pi_k^i \tag{3.42}$$

$$X_{k+1}^i = X_{k+1}^i + V_k^i \tag{3.43}$$

Her bir iterasyonda atomların hız ve pozisyon değerleri yenilenir. (3.42) ve (3.43) denklemlerinde V_{t+1}^v : zamana bağlı hızı, X_{t+1}^v: zamana bağlı konumu, Pi_k^i ise zamana bağlı ivmeyi ifade etmektedir.

ASO sözde kodu(pseudo) aşağıda gösterilmiştir [26, 58]:

Atom algoritması parametreleri tanımlanır.

Başlangıç popülasyonlarının oluşturulur.

Her bir atom uygunluk fonksiyonu belirlenir.

while (k<Maksimum iterasyon sayısı)

***for** (Her bir atom değeri için)*

Atomların toplam kuvvetinin hesaplanır. (3.39)

Her bir atom için hız ve pozisyon güncellemesi yapılır.

***if** (atom optimizasyonunda komşuluk var mı?)*

Atomların en iyi konuma doğru hareket etmeleri belirlenir.

En iyi uygunluk konum pozisyonu güncellenir. (3.43)

En iyi uygunluk hız güncellenir. (3.42)

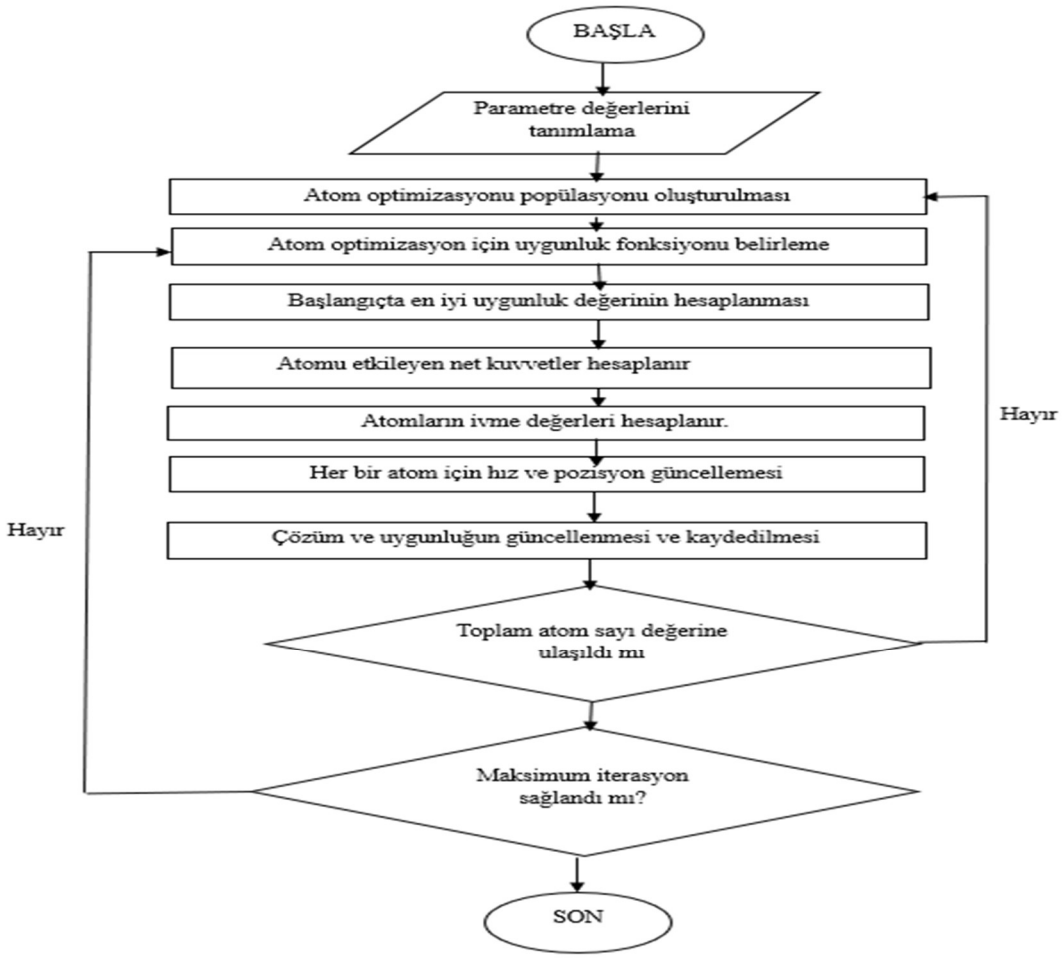
end

k=k+1

end while

Sonuçları göster

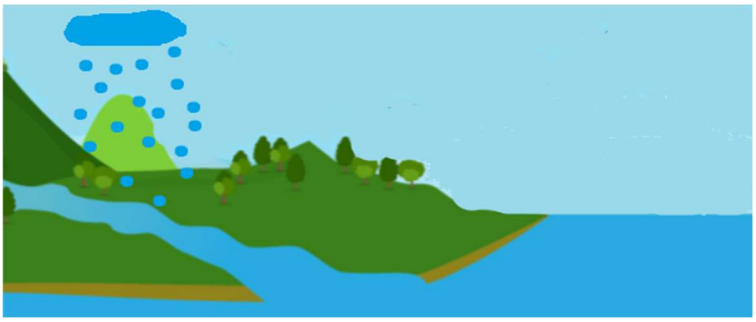
ASO algoritmasının genel yapısını anlatan akış diyagramı, Şekil 3.17'de gösterilmiştir [26,58]:



Şekil 3.17: Atom arama algoritması akış diyagramı [26, 58]

3.1.9. Su Döngüsü Algoritması (WCA)

Su döngüsü algoritması, suyun doğadaki akışından esinlenilerek oluşturulmuş bir metasezgisel algoritmadır [27]. Su döngüsü algoritması yağmur sularının nehirlere ve akarsulara düşmesini ve ardından su damlacıklarının denizlere ulaşıp atmosfere doğru buharlaşmasını matematiksel olarak modelleyen bir algoritmadır. Algoritmada popülasyon içinde çözüm değerleri nehirler, denizler, su damlacıkları olarak belirlenmiştir. Bunların içerisinde en iyi çözüm denizler olarak belirlenmiştir [27, 60-62]. Her bir iterasyonda su damlacıklarının nehirlere ve oradan da denize hareketleri tekrarlanır. Şekil 3.18'de yağmur damlacıkların nehire düşmesi gösterilmiştir.



Şekil 3.18: Yağmur damlacıkların nehire düşmesi

Yağmur damlası ile başlangıç popülasyonu oluşturulur. Su döngüsü algoritmasında her bir yağmur damlası sayısı vektörü $N_{var}=(X_1, X_2 \ldots\ldots X_{var})$ olarak ifade edilir. Başlangıç popülasyonu oluşturmak için

$$N_{pop} x\ N_{var} \tag{3.44}$$

boyutunda bir yağmur damlası matrisi oluşturulur. Burada N_{pop} toplam yağmur damlasını ve N_{var} ise her bir yağmur damla sayısını temsil eder. Rastgele oluşturulan yağmur su damlalarını ifade eden X matrisi aşağıda verilmiştir:

$$X = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_{N_{var}}^1 \\ \vdots & \vdots & \vdots & X_{N_{var}}^2 \\ X_1^{N_{Pop}} & X_2^{N_{pop}} & \vdots & X_{N_{var}}^{N_{pop}} \end{bmatrix} \tag{3.45}$$

Başlangıç popülasyonu uygunluk fonksiyonlarında değerlendirildiğinde, popülasyonun en iyi uygunluk değerine sahip deniz ve nehir değişkenleri belirlenir. Deniz ve nehirlerin toplam sayısı denklem (3.46) ve yağmur damlacıklarının sayısı denklem (3.47) ile belirlenir. Akışkan yoğunluğuna bağlı olarak yağmur damlalarının denizlere ve nehirlere akış miktarının hesabı denklem (3.48) ile ifade edilir.

$$N_{sr} = N_1 + 1 \tag{3.46}$$

$$N_{yd} = N_{pop} - N_{sr} \tag{3.47}$$

$$ND_i = round\left\{ \left| \frac{fit_i}{\sum_{k=1}^{SR} fit_k} \right| N_{yd} \right\} \tag{3.48}$$

Burada N_{sr} deniz ve nehirlerin toplam sayısını, N_1 nehir sayısını, N_{yd} yağmur damlacıkları sayısını, fit_i belli bir i sürecindeki uygunluk fonksiyonunu, SR tüm süreç değerini

ve ND_i akış miktarını ifade eder. Su damlacıklarının nehire doğru hareketinin ardından denizlere ulaşması denklem (3.49) ve denklem (3.50) ile ifade edilmiştir.

$$X_{yenisu} = X_{su} + rc_3(X_{nehir} - X_{su}) \tag{3.49}$$

$$X_{yeni} = X_{nehi} + rc_3(X_{deniz} - X_{nehi}) \tag{3.50}$$

Burada (3.49) ve (3.50) denklemlerinde r çarpanı 0 ile 1 arasında oluşan rastgele bir sayıdır. c_3 0 ile 2 arasında değer alan sabit bir çarpandır. Sabit çarpan değeri damlacıkların ne kadar geniş aralıkta olacağı ile belirlenir. Değer büyüdükçe su damlacıklarının nehirlere daha büyük su kütleleri ile ulaşmasını sağlar. Bu denklemde X_{yenisu} su damlacıklarının yeni pozisyonunu, X_{su} su damlacıklarının mevcut konumunu, X_{deniz} su damlacıklarının denizdeki konumunu ve X_{nehir} nehirlerin konumunu ifade eder. Su damlacıkları nehire ulaştıktan sonra nehirden denize ulaşması sırasında oluşacak yeni konum denklem (3.50) ile gösterilmiştir. Nehirler ve denizler arasındaki su seviyesi yönünü ve suyun akış yönü belirmek için denklem (3.51) ve denklem (3.52) verilmiştir:

$$|X_{deniz} - X_{nehir}| < dmax \tag{3.51}$$

$$dmax = dmax - \frac{dmax}{maksimum\ iterasyon} \tag{3.52}$$

Denklem (3.51)'de kullanılan dmax değeri nehirler ve denizler arasındaki su seviyesini belirleyerek, buharlaşma sürecini kontrol eden değerdir. Bu değer çok küçük olduğunda buharlaşma sürecinin son ermesi sağlanır. Buharlaşma sonrası yağmur suyunun yağması ile oluşacak yeni çözüm değerleri denklem (3.53) ile elde edilmiştir.

$$X_{yeniakış} = X_{deniz} + \sqrt{u_1} rand(1, dim) \tag{3.53}$$

Burada $X_{yeniakış}$ yeni arama pozisyonlarını temsil eder. Arama için dim çözüm uzayında boyutu, u_1 denize yakın arama oranını ve $rand(1, dim)$ rastgele bir değeri gösterir. Denize yakın arama oranı genellikle [0 1] arasında bir değer alır.

WCA algoritması sözde kod(pseduo) örneği aşağıdaki gösterilmiştir [27, 60]:

Başlangıç su damlacığı sayısı (N) ve diğer parametreler tanımlanır.

Su damlacıkları başlangıç popülasyonu oluşturulur.

Hata fonksiyonunu hesaplanır.

***while** (k <maksimum iterasyon sayısı)*

***for** her bir su damlacığı için (N)*

Su damlacıklarının nehre doğru hareketini güncellenir. (3.49)

Nehrin denize doğru hareketini güncellenir. (3.50)

Uygunluk fonksiyonu güncellenir.

***if** su damlacığının uygunluk değeri en iyi çözümden küçükse*

En iyi çözümü güncelle (denizi güncelle) (3.53)

***else if** (Buharlaşma koşulunun sağlanıp sağlanmadığını kontrol edilir.)(3.51)*

end

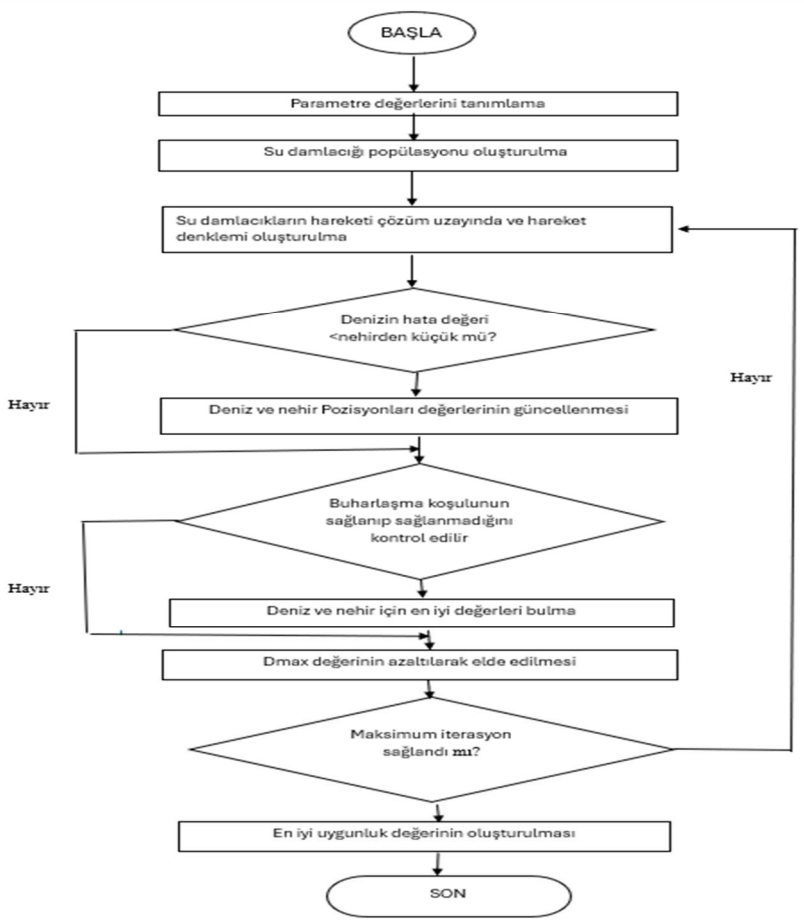
dmax değerini her iterasyonda azaltılır.

k=k+1

end while

Sonuçları göster

WCA algoritması genel yapısını anlatan akış diyagramı Şekil 3.19'da gösterilmiştir [27, 60]:



Şekil 3.19: Su döngüsü algoritması akış diyagramı [27, 60]

3.2. KLASİK KONTROL YÖNTEMİ

Klasik kontrol yöntemleri, kontrol edilmek istenen sistemlerin istenilen performansının elde edilebilmesi için kullanılan yöntemlerdir. Sıklıkla kullanılan klasik kontrol yöntemlerinden biri de Ziegler-Nichols yöntemidir.

3.2.1. Ziegler Nichols Yöntemi

Bu yöntem, PID ve PI denetleyicilerin parametrelerini ayarlamak için sıklıkla kullanılan bir yöntemdir [13]. Ziegler- Nichols yöntemi için iki kritik parametre vardır. Bu parametreler Ku ve Tu parametreleridir. Ku parametresi kritik kazanç parametresi olarak adlandırılır. Bu parametre Kp kazanç değerine bağlı olarak değişir. Kp kazancı artıkça sistem kararsız hale gelir. Sürekli salınım yapar ve kararsızlık yapacak değerine kritik kazanç denir. Salınım sırasındaki bu salınım periyoduna ise kritik periyot Tu değeri denilir. Bu iki Ku ve Tu değeri kullanılarak

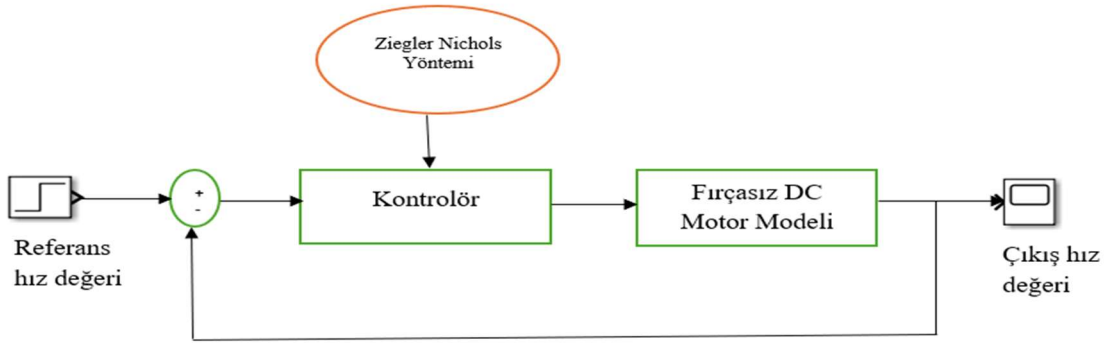
Tablo 3.1'deki Ziegler Nichols kazanç parametreleri hesaplama formülleri yardımıyla PID kontrolör parametreleri hesaplanır [13].

Tablo 3.1: Ziegler Nichols kazanç parametreleri hesaplama formülleri [13, 63]

Kontroller	Kp	Ki	Kd
PI	0.45Ku	$\frac{0.54Ku}{Tu}$	-
PID	0.6Ku	$\frac{1.2Ku}{Tu}$	$\frac{0.6KuTu}{8}$

3.2.2 Ziegler Nichols Yönteminin Fırçasız DC Motora Uygulanması

Ziegler Nichols yöntemi, fırçasız DC motorların hızlarının kontrolünü sağlamak ve motorun istenilen performansını elde etmek için klasik bir yöntem olarak kontrolör parametrelerinin ayarlanmasında kullanılabilir. Şekil 3.20'de Ziegler Nichols yönteminin fırçasız DC motor sistemine uygulanması gösterilmiştir.



Şekil 3.20: Ziegler Nichols yönteminin fırçasız DC motor sistemine uygulanması [64]

Ziegler Nichols yöntemi fırçasız DC motor modelimize uygulayarak PI ve PID kontrolör parametreleri belirlenmiştir. Kontrol parametrelerini belirlemek için motorun kritik kazanç parametresi ve kritik periyot parametresi hesaplanmıştır. Bu parametre değerleri sırasıyla Ku=0.6600 ve Tu=0.0524 olarak hesaplanmıştır. Bu değerler Tablo 3.1'de gösterilen Ziegler Nichols hesaplama formülleri kullanılarak bulunmuş ve fırçasız DC motora uygulanmıştır. Motora uygulanan kontrolör parametreleri Tablo 3.2'de gösterilmiştir.

Tablo 3.2 Ziegler-Nichols yöntemi ile bulunan PI ve PID kontrolör parametreleri

Kontroller	Kp	Ki	Kd
PI	0.2970	6.7983	-
PID	0.3960	15.1074	0.0026

3.3. METASEZGİSEL ALGORİTMALARIN FIRÇASIZ DC MOTORA UYGULANMASI

Bu çalışmada fırçasız bir DC motorun kontrolünde kullanılan PI, PID ve elde edilen PI optimizasyon parametrelerinin performansını daha da iyileştirmek amacıyla kullanılan bulanık Mantık (Fuzzy Logic)-PI (FGPI) denetleyicileri optimize etmek amacıyla önceki bölümlerde anlatılan dokuz adet metasezgisel optimizasyon algoritmaları kullanılmıştır. Bu algoritmalarının motor modeline uygulanmaları esnasında bulunan değerlerin kıyaslanabilmesi için uygunluk fonksiyonlarına gereksinim vardır. Literatürde sıklıkla kullanılan mutlak hatanın toplamı (Integral Absolute Error, IAE), zaman ağırlıklı mutlak hata toplamı (Integral Time-weighted Absolute Error, ITAE) ve hata karelerinin toplamı (Integral Square Error, ISE) adlı uygun fonksiyonları aşağıda anlatılmıştır:

IAE Fonksiyonu:

Günümüzde kontrol sistemlerinde yaygın kullanılan performans ölçütüdür. Sistem modellemelerinde IAE mutlak istenilen değer ile gerçek değer arasındaki farkın integralinin alınmasıyla hesaplanır ve hesap formülü aşağıdadır:

$$IAE = \int_0^T |e(t)|\, dt \tag{3.54}$$

Burada e, hatayı gösterir. IAE fonksiyonu en küçük hatayı bile hesaba kattığından hata oranının en düşük seviyeye indirmekte yardımcı olur. IAE sistemin motor kontrollerinde kararlığını arttırır. Bu nedenle motorun hız değişimleri kontrolünde tercih edilir.

ITAE Fonksiyonu:

ITAE fonksiyonu istenilen değer ile gerçek değer arasındaki farkın alınıp zaman çarpımının integralin alınmasıyla hesaplanır. Bu fonksiyon uzun süren hata oranlarını düzeltir ve formülü aşağıdadır.

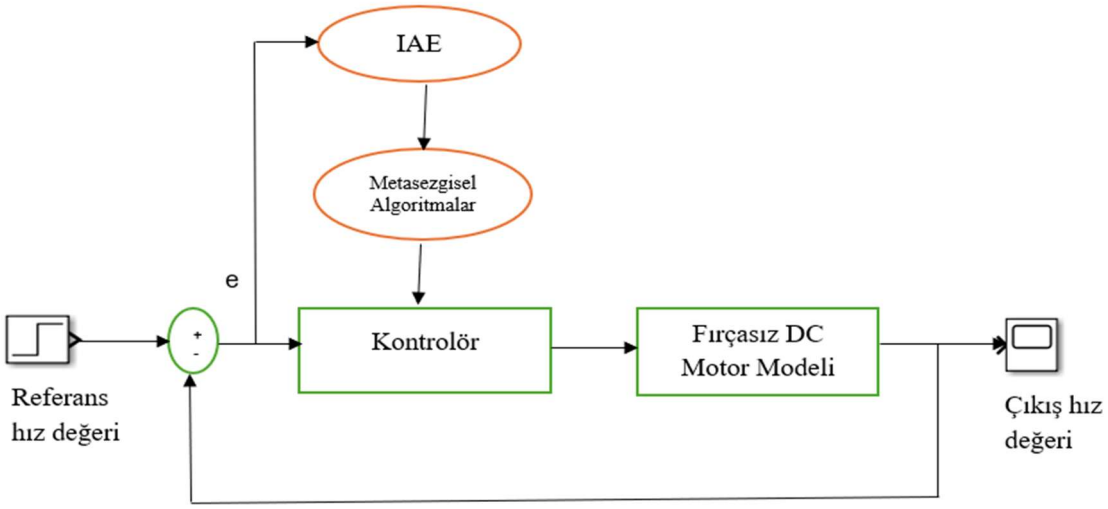
$$ITAE = \int_0^T t|e(t)|\, dt \tag{3.55}$$

ISE Fonksiyonu:

ISE fonksiyonu istenilen değer ile gerçek değer arasındaki farkın alınıp karesinin integralin alınmasıyla hesaplanır ve formülü aşağıdadır.

$$ISE = \int_0^T e(t)^2 dt \tag{3.56}$$

Metasezgisel algoritmalarının fırçasız DC motora uygulanmasında *e* hata işaretini en küçük düzeyde tutmak ve sistem cevap eğrisinin oturma zamanının küçük olmasını garantileyebilmek amacıyla bu çalışmada IAE fonksiyonundan yararlanılmıştır. İlk olarak PID ve PI denetleyicilerin başlangıç parametreleri, tezde anlatılan GA, PSO, GWO, BA, ASO, FPA, WCA, DA ve ABC optimizasyon algoritmalara uygun biçimde elde edilir. Sonra bu kontrolör parametre değerleri motor modelinde yazılarak IAE hesabı ve uygunluk değerleri elde edilir. Bu işlem popülasyon birey sayısı kadar yapılır. Tüm iterasyon sayısı boyunca tüm algoritmalar için tekrarlayarak IAE değeri en düşük olan K_P, K_I , K_D kontrolör parametre değerleri belirlenir. Şekil 3.21'de fırçasız DC motor modeline uygulanan PI, PID denetleyicilerin metasezgisel optimizasyon algoritmalarının motor sistemine uygulanışı ve algoritmanın arzu edilen sonuçlara ulaşması için IAE fonksiyonunun e hata kolu ile olan ilişkisi resmedilmiştir.



Şekil 3.21: Metasezgisel algoritmalarla optimize edilmiş kontrolörlerin uygulandığı bir fırçasız DC motor modelinin sistem blok diyagramı [48,65]

GA, PSO, GWO, BA, ASO, FPA, WCA DA ve ABC optimizasyon algoritmaları için başlangıçta belirlenen parçacık sayısı, iterasyon sayısı ve parametrelerin alt üst limit değerleri Tablo 3.3'te verilmiştir.

Tablo 3.3: Metasezgisel Algoritmalarda Kullanılan Başlangıç Değerleri

Parametre	**Değer**
Parçacık Sayısı	20
İterasyon Sayısı	100
Parametrelerin alt limiti	[0.001 0.001 0.001]
Parametrelerin üst limiti	[30 30 30]

3.3.1. Metasezgisel Algoritmalarla Optimize Edilmiş PID Kontrolörün Fırçasız DC Motor Modeline Uygulanması

PID kontrolör fırçasız DC motorun hız kontrolünde etkilidir. PID kontrolörün oransal, integral ve türev kazanç parametreleri kullanılarak sistemin performansı kontrol edilir. Bu bölümde Şekil 3.22'de görülen sistem blok diyagramında Tablo 3.3'te verilen başlangıç değerleri kullanılarak GA, PSO, GWO, BA, ASO, FPA, WCA DA ve ABC algoritmaları ile optimize edilmiş PID kontrolörler kullanılarak simülasyonlar sonucunda elde edilen PID denetleyici parametreleri Tablo 3.4'te verilmiştir.

Tablo 3.4: Metasezgisel algoritmalar optimize edilmiş PID kontrolör parametre değerleri

Metasezgisel Optimizasyon Algoritmalar	**Kp**	**Ki**	**Kd**
DA-PID	1.828	10	0.0175
WCA-PID	5.3026	9.0195	0.0935
PSO-PID	0.4162	10	0.0079
ASO-PID	1.0424	10	0.0174
GWO-PID	0.4208	10	0.0080
BA-PID	1.1397	10	0.0293
GA-PID	2.9196	9.7903	0.1510
ABC-PID	1.1669	10	0.0288
FPA-PID	2.4735	7.8427	0.0238

Bu kontrolör parametreleri IAE uygunluk fonksiyonu kullanılarak oluşturulmuştur. Algoritmanın daha hassas optimizasyon yapabilmesi için 0.001 örnekleme zamanı alınmıştır.

3.3.2. Metasezgisel Algoritmalarla Optimize Edilmiş PI Kontrolörün Fırçasız DC Motor Modeline Uygulanması

Bu bölümde Şekil 3.22'de görülen sistem blok diyagramında Tablo 3.3'te verilen başlangıç değerleri kullanılarak GA, PSO, GWO, BA, ASO, FPA, WCA, DA ve ABC algoritmaları ile optimize edilmiş PID kontrolörler kullanılarak simülasyonlar sonucunda elde edilen PID denetleyici parametreleri Tablo 3.5'te verilmiştir.

Tablo 3.5: Metasezgisel algoritmalar optimize edilmiş PI kontrolör parametre değerleri

Metasezgisel Optimizasyonu Algoritmalar	Kp	Ki
DA-PI	0.3804	3.9705
WCA-PI	0.3156	4.5231
GWO-PI	0.2493	3.7000
GA-PI	0.3736	4
ASO-PI	0.2902	4
PSO-PI	0.2500	3.9800
BA-PI	0.3727	4
ABC-PI	0.2493	4.3000
FPA-PI	0.3696	4.3632

Oluşturulan parametreler IAE uygunluk fonksiyonu kullanılarak oluşturulmuştur.

3.3.3. Metasezgisel Algoritmalarla Optimize Edilmiş PI Tabanlı FGPI Kontrolörün Fırçasız DC Motor Modeline Uygulanması

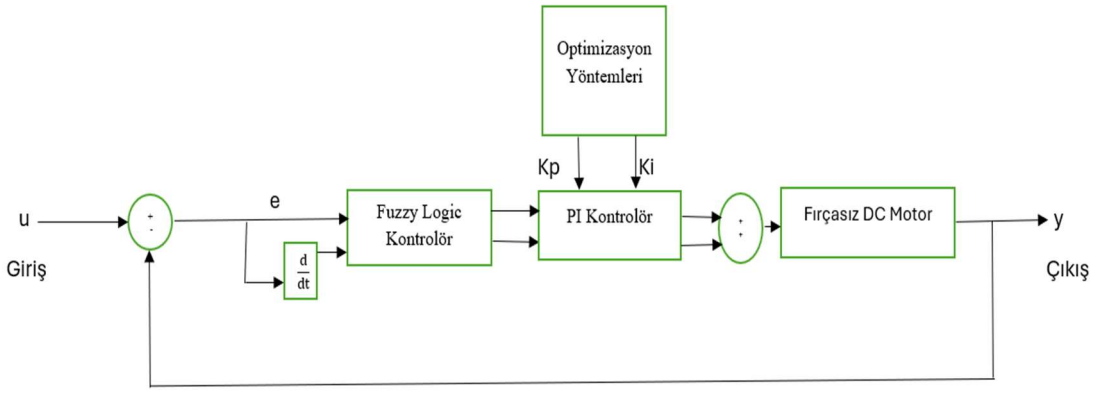
Bulanık mantık kavramı ilk olarak Zadeh L.A. tarafından ortaya atılan bir kavramdır [66]. Bulanık mantığın bir sistem üzerine ilk uygulamaları Mamdani E. tarafından, bir buhar motoru ve kazan sistemini kontrol etmek için kullanılmıştır [67].

Bulanık mantık, doğrusal olmayan ve belirsizlik içeren sistemlerin kontrolünde kullanılan etkili bir kontrol yöntemidir. Bu kontrol yöntemi, sistemin dinamik yapısındaki doğrusal belirsizlikleri yönetme kapasitesine sahiptir. Bu çalışmada fırçasız DC motorların kontrolünde PI kontrolör ile elde edilen optimizasyon parametreleri, başlangıç değeri alınarak bulanık mantık PI kontrolde kullanılmıştır.

Fırçasız DC motorun kontrolünde bulanık mantık ve PI kontrolör hibrit bir yapıda birleştirilerek, sistemin dinamik davranışları gözlemlenmiştir [68-70]. Bu tezin bir amacı da, metasezgisel algoritmalarla optimize edilmiş PI kontrolör parametrelerinden faydalanılarak

oluşturulan FGPI (bulanık mantık-PI) kontrolör ile sistem modeli elde edilmiş fırçasız dc motorun performansının daha da iyileştirilmesidir.

FGPI kontrolör ve metasezgisel optimizasyon yöntemlerinin bir arada kullanıldığı fırçasız dc motor sisteminin blok diyagramı, Şekil 3.22'de verilmiştir.



Şekil 3.22: Optimize edilmiş FG-PI kontrolörlü fırçasız dc motor sisteminin blok diyagramı [15, 36]

Şekil 3.22'de görüldüğü üzere, FGPI kontrolöründe iki giriş bulunmaktadır. Bu girişler hatanın değişimi ve çıkış arasındaki farklardır. Girişlerden biri sistemin çıkışındaki hız değeri (y) ile referans sinyali (u) arasındaki motor hız hatasıyken; ikincisi ise hızdaki değişim hatasıdır. Bulanık mantık çıkışında 9 adet metasezgisel algoritma ile Kp ve Ki parametreleri optimize edilmiş PI kontrolör bulunmaktadır. Böylece Şekil 3.22'deki gibi oluşturulan FG-PI kontrolörler kullanılarak fırçasız DC motor sistemimizin çıkışları bu tezde, yüzde aşımlar, yerleşme süreleri ve yükselme süreleri gibi performans kriterleri açısından karşılaştırılmıştır. Zamana bağlı PI kontrolör parametrelerinin ayarlanması (3.57) ve (3.58) denklemleri ile gösterilebilir [15].

$$Kp(t) = Kp + \Delta Kp(t) \tag{3.57}$$

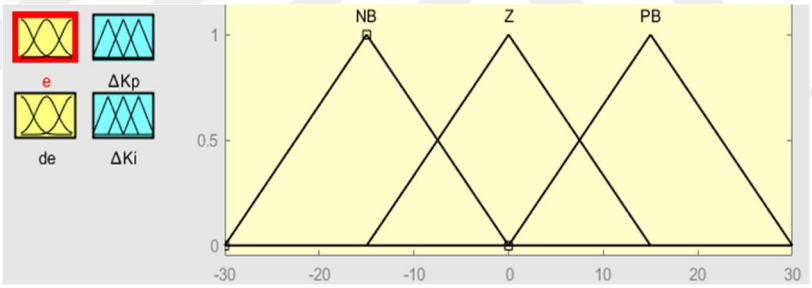
$$Ki(t) = Ki + \Delta Ki(t) \tag{3.58}$$

Denklemlerde gösterilen Kp ve Ki kontrol parametreleri için metasezgisel algoritmalarla başlangıç değerleri optimize edilerek sisteme entegre edilmiştir. Zamanla değişen ΔKp(t) ve ΔKi(t) metasezgisel algoritmalar tarafından iyileştirilen bileşenlerdir. Bu

terimler, sistemin dinamik davranışlarına uyum sağlamak ve doğrusal olmayan etkileri yönetmek amacıyla kontrol parametrelerini sürekli olarak güncelleyerek performansı artırır. FGPI kontrolöründe, girişler hata ve hata değişimi ile PI kontrolör parametre değerleri arasındaki ilişkiyi tanımlayan bir dizi kural bulunur. Üyelik fonksiyonlarından Şekil 3.23.a.'daki gibi üçgen üyelik fonksiyonu bulanık kontrolde kullanılarak değerlendirilme yapılabilir.

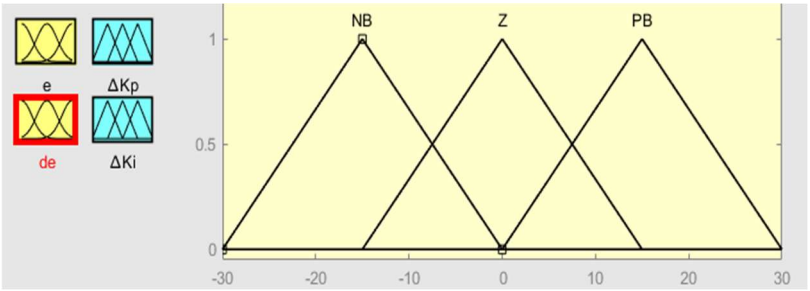
FGPI denetleyicinin Matlab ortamında simülasyonu yapılırken bulanık mantık (fuzzy logic) kontrolünün hata ve hata değişim oranları öncelikle üyelik fonksiyonlarına göre bulanıklaştırılır. Her bir giriş ve çıkış üyelik fonksiyonları için ayrı ayrı değer aralıkları oluşturulur. NB (negatif büyük), Z (sıfır) ve PB (pozitif büyük) terimleri, negatif, sıfır ve pozitif durumları tanımlamak için üyelik fonksiyonlarında kullanılır. Bulanık mantık üyelik fonksiyonları değerleri MATLAB fuzzy logic toolbox kullanılarak Şekil 3.23'te ayrı ayrı oluşturulmuştur [35].

Motor hız hatası üyelik fonksiyonu gösterimi Şekil 3.23.a.'da gösterilmiştir [35]. Bu çalışmada motor hız hatası giriş üyelik fonksiyonlarının aralığı, (-30 30) olarak belirlenmiştir.



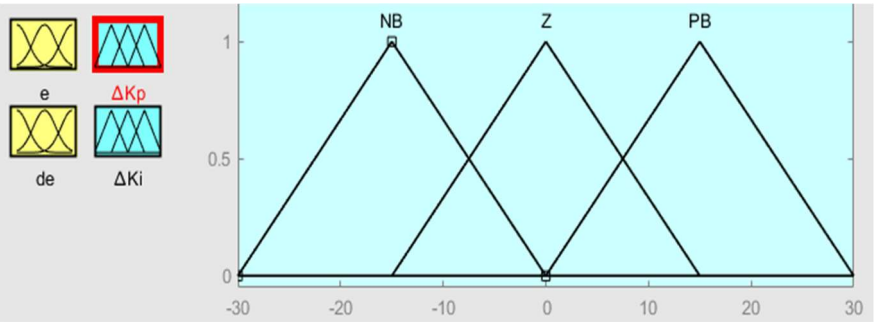
Şekil 3.23.a: Motor hız hatası giriş üyelik fonksiyonlarının aralığı [35]

Motor hızındaki değişim hatası üyelik fonksiyonu gösterimi de Şekil 3.23.b'de gösterilmiştir [35]. Motor hızındaki değişim üyelik fonksiyonlarının aralığı da (-30 30) olarak belirlenmiştir.



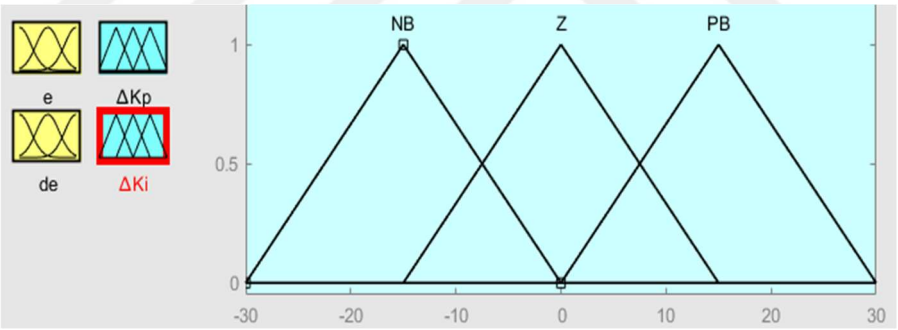
Şekil 3.23.b: Motor hızının değişim hatası üyelik fonksiyonlarının aralığı [35]

Motor çıkışındaki ΔKp üyelik fonksiyonu gösterimi Şekil 3.23.c'de gösterilmiştir [35]. Burada da üyelik fonksiyonlarının aralığı (-30 30) olarak belirlenmiştir.



Şekil 3.23.c: Motor çıkışındaki ΔKp üyelik fonksiyonlarının aralığı [35]

Motor çıkışındaki ΔKi üyelik fonksiyonu gösterimi Şekil 3.23.d'de gösterilmiştir [35]. ΔKi üyelik fonksiyonlarının aralığı da (-30 30) olarak belirlenmiştir.



Şekil 3.23.d: Motor çıkışındaki ΔKi üyelik fonksiyonlarının aralığı [35]

Bulanık mantık kontrol, üyelik fonksiyonlarının belirlenmesinin ardından kural tablosu oluşturulmuştur. Giriş değişkenlerine ait üç üyelik fonksiyonu değerlerine göre, Kp ve Ki parametrelerinin çıkışları için belirlenen kural tabloları Tablo 3.6 ve Tablo 3.7'de verilmiştir.

Tablo 3.6: Bulanık mantık kontrol yönteminin çıkışındaki $\Delta Kp(t)$ kontrol parametresine ilişkili kurallar

de \ **e**	NB	Z	PB
NB	N	Z	Z
Z	Z	Z	Z
PB	PB	PB	PB

Tablo 3.7: Bulanık mantık kontrol yönteminin çıkışındaki $\Delta Ki(t)$ kontrol parametresine ilişkili kurallar

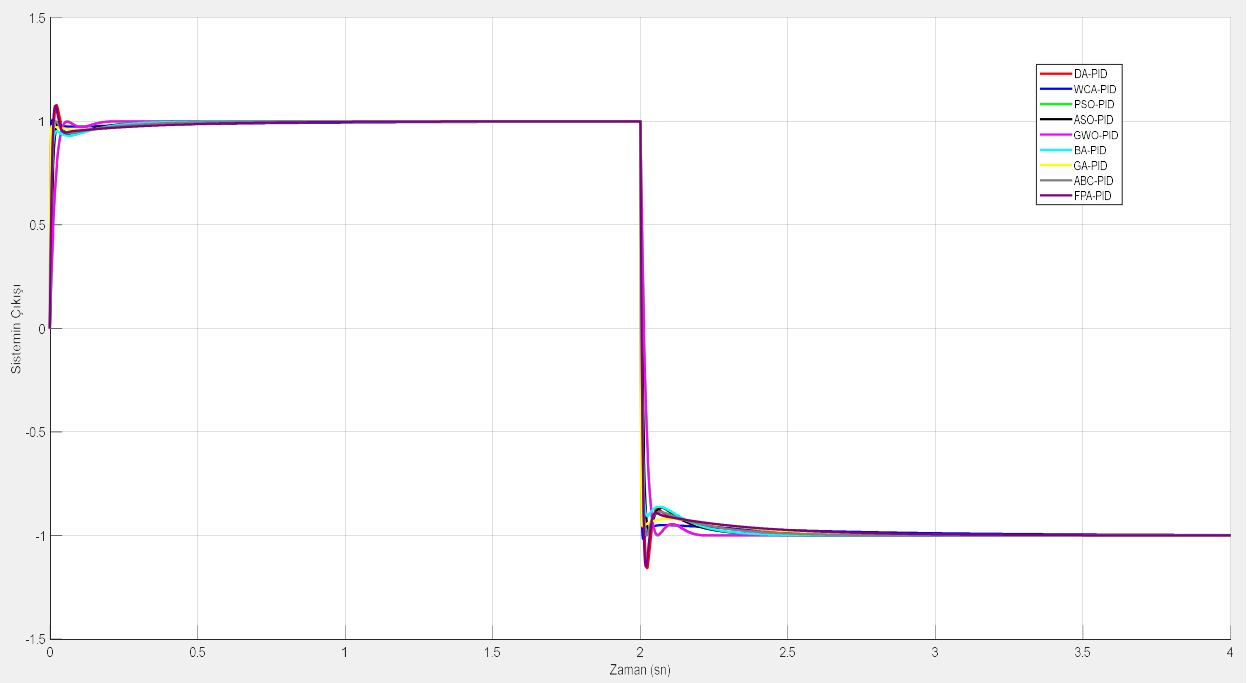
de \ **e**	NB	Z	PB
NB	PB	Z	PB
Z	Z	Z	PB
PB	Z	Z	NB

4. BULGULAR

Birim basamak referans giriş altında fırçasız DC motor modelimizin PID, PI ve FGPI kontrolünde yusufçuk algoritması (DA), su döngüsü algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa algoritması (BA) ve çiçek tozlaşma algoritması (FPA) olmak üzere 9 adet metasezgisel algoritmanın performans kriterlerini birbirleriyle karşılaştırabilmek için yükselme zamanı, yerleşme zamanı ve aşım kriterlerinden yararlanılmıştır. Sistemin performans kriterlerine göre birim basamak referans girişine bağlı olarak sistemin çıkışındaki hem pozitif hem de negatif bölgede sistemin davranışları incelenmiştir.

3.4. PID KONTROLÖR İÇİN SONUÇLAR

DA, WCA, PSO, GA, GWO, ABC, ASO, BA ve FPA kullanılarak optimize edilen PID kontrolörlerin fırçasız DC motor sistem modelimize birim basamak referans giriş altında uygulanması sonucu 0-2 saniye ve 2-4 saniye arasında elde edilen sistem cevap eğrileri hem pozitif hem de negatif bölgede birlikte Şekil 4.1'de gösterilmiştir.

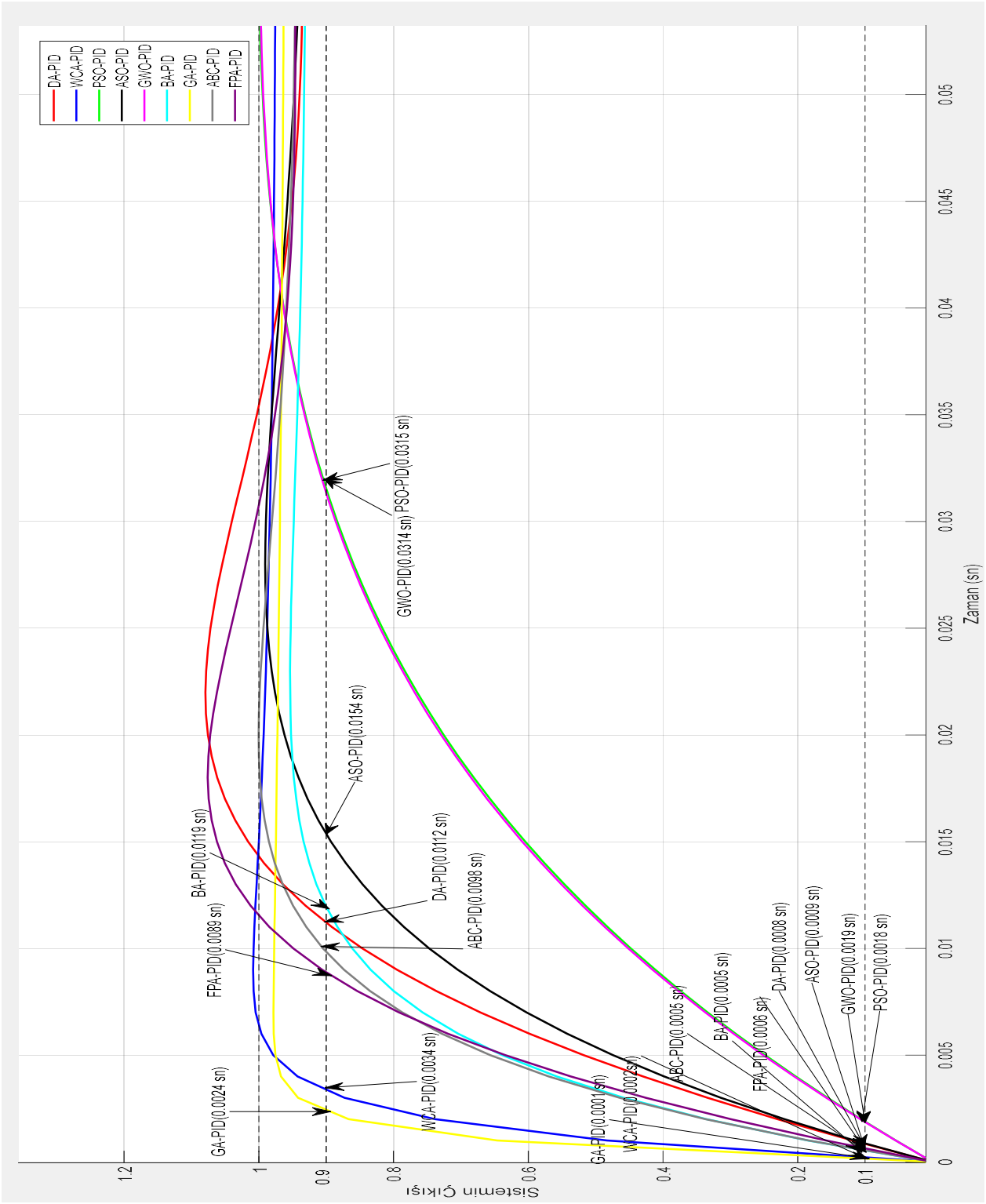


Şekil 4.1: Optimize edilmiş PID kontrolörler uygulanmış bir fırçasız DC motor sisteminin (0-4) saniye aralığında cevap eğrileri

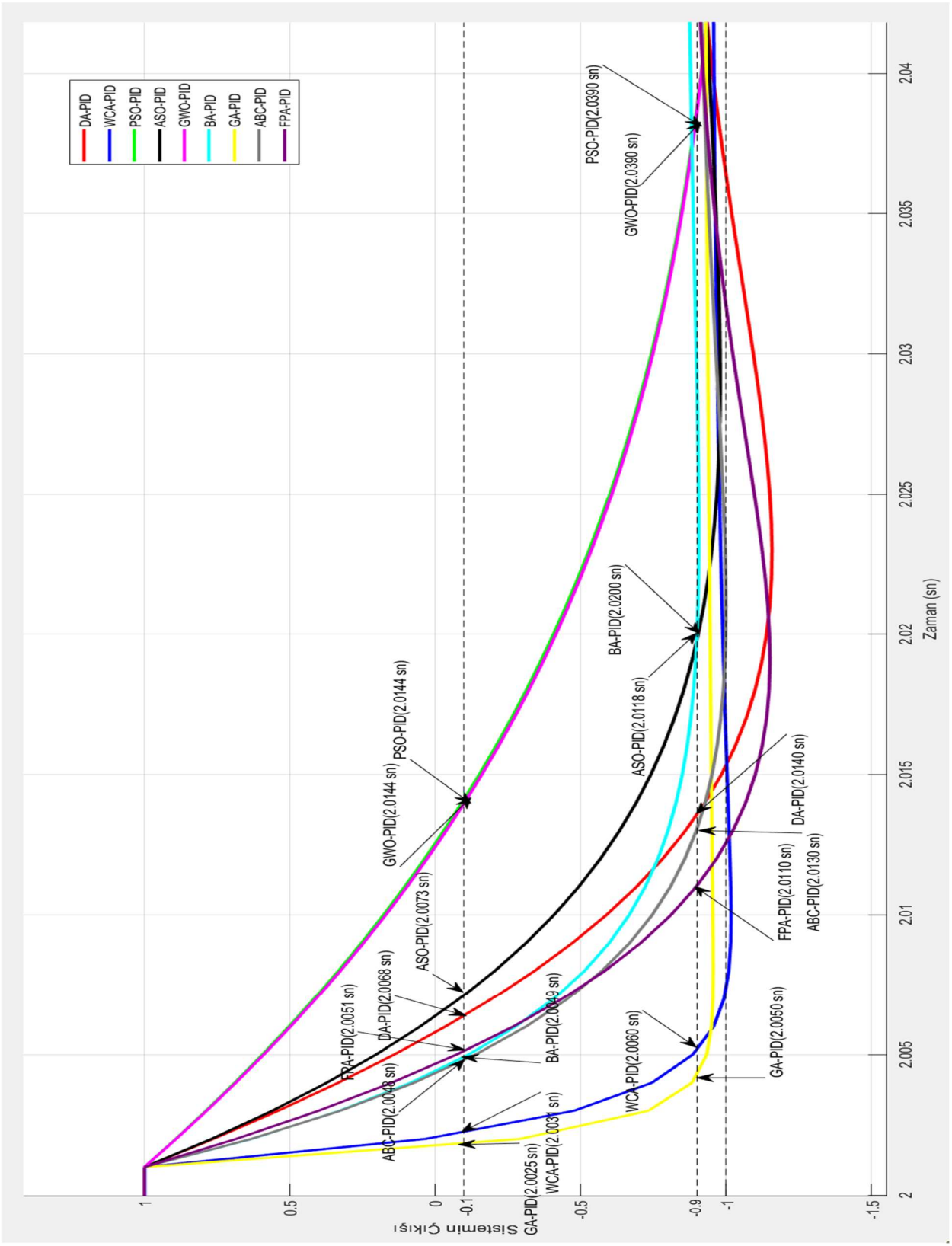
Metasezgisel optimizasyon tabanlı PID kontrolörlere sahip fırçasız DC motor modelimizin sistem çıkışlarının birbirleri ile performans kriterlerinden yükselme zamanları açısından karşılaştırılması için Şekil 4.2 (pozitif bölge) ve Şekil 4.3'teki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Hem pozitif bölgede (0-2 sn arası) hem de negatif bölgede (2-4 sn arası) fırçasız DC motorun çıkış işaretlerinden yükselme zamanları için %10'ından %90'ına ulaşma süreleri dikkate alınarak ayrı ayrı karşılaştırılmaları yapılmıştır. Şekil 4.2'den 0 ile 2 saniye arasındaki pozitif bölgede yükselme zamanı değerlerinin küçükten büyüğe doğru GA-PID (0.0023 sn), WCA-PID (0.0032 sn), FPA-PID (0.0083 sn), ABC-PID (0.0093 sn), DA-PID (0.0104 sn), BA-PID (0.0114 sn), ASO-PID (0.0145 sn), GWO-PID (0.0295 sn) ve PSO-PID (0.0297 sn) olarak sıralı olduğu görülür. Şekil 4.3'ten (2-4) saniye arasındaki negatif bölgede yükselme zamanı değerleri ise küçükten büyüğe doğru GA-PID (0.0025 sn), WCA-PID (0.0029 sn), ASO-PID (0.0045 sn), FPA-PID (0.0059 sn), DA-PID (0.0072 sn), BA-PID (0.0151 sn), GWO-PID (0.0246 sn) ve PSO-PID (0.0246 sn) olarak sıralanmıştır.

Metasezgisel optimizasyon tabanlı PID kontrolörlere sahip sistem çıkışlarının birbirleri ile performans kriterlerinden yerleşme zamanları açısından karşılaştırılması için Şekil 4.4 (pozitif bölge) ve Şekil 4.5'teki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Hem pozitif bölgede (0-2 sn arası) hem de negatif bölgede (2-4 sn arası) fırçasız DC motorun çıkış işaretlerinden yerleşme zamanları için %2'lik yerleşme bandı dikkate alınarak ayrı ayrı karşılaştırılmaları yapılmıştır. Şekil 4.4'ten 0 ile 2 saniye arasındaki pozitif bölgede yerleşme zamanı değerlerinin küçükten büyüğe doğru PSO-PID (0.1336 sn), GWO-PID (0.1338 sn), ASO-PID (0.2041 sn), BA-PID (0.2178 sn), WCA-PID (0.2415 sn), ABC-PID (0.2556 sn), DA-PID (0.2576 sn), GA-PID (0.3401 sn) ve FPA-PID (0.3592 sn) olarak sıralı olduğu görülür. Şekil 4.5'ten 2 ile 4 saniye arasındaki negatif bölgede yerleşme zamanı değerleri ise küçükten büyüğe doğru PSO-PID (2.1588 sn), GWO-PID (2.1589 sn), ASO-PID (2.2748 sn), BA-PID (2.2868 sn), ABC-PID (2.3725 sn), DA-PID (2.3893 sn), GA-PID (2.5192 sn) ve WCA-PID (2.6412 sn) olarak sıralanmıştır.

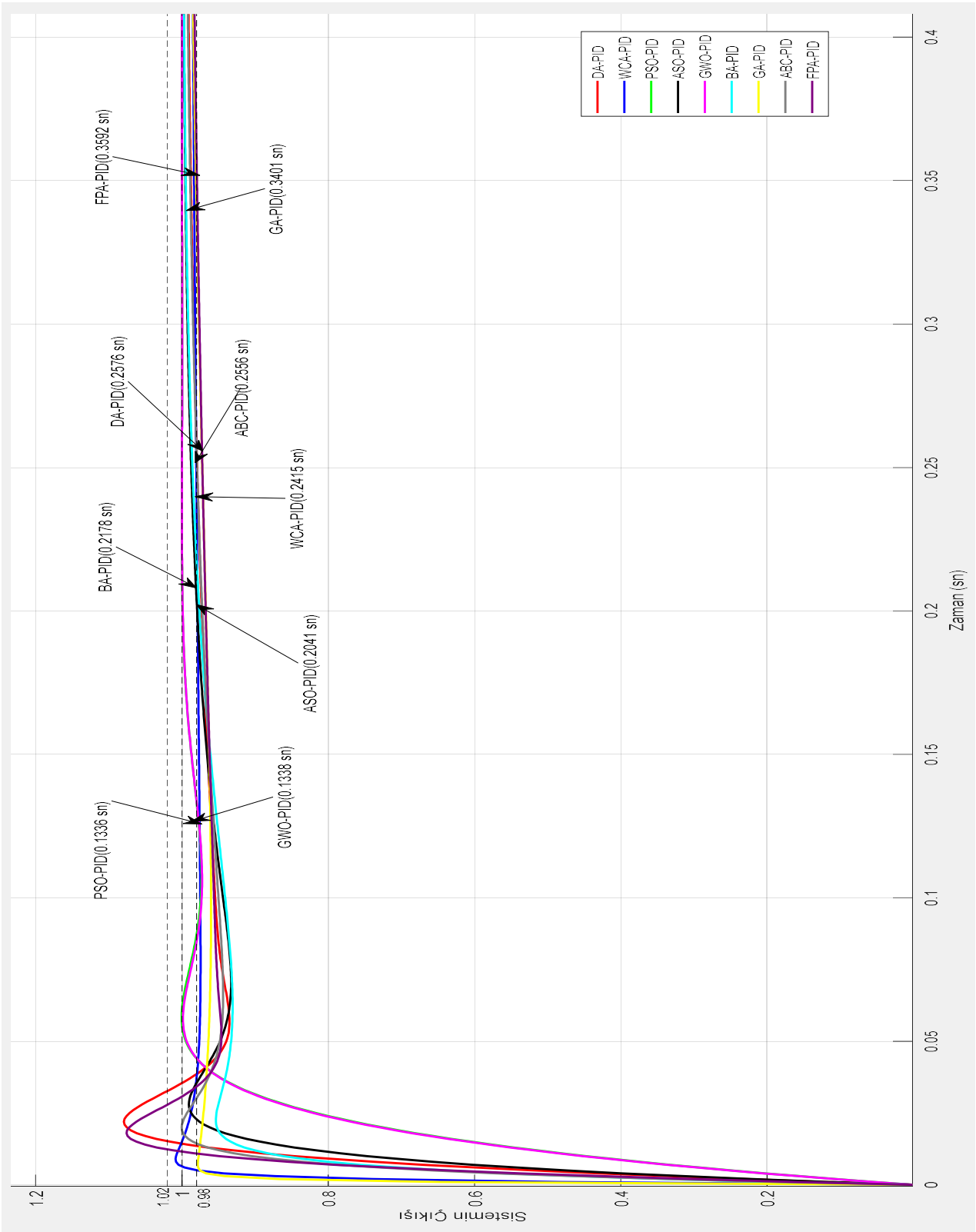
Metasezgisel optimizasyon tabanlı PID kontrolörlere sahip sistem çıkışlarının birbirleri ile performans kriterlerinden yüzde aşım açısından karşılaştırılması için Şekil 4.6 (pozitif bölge) ve Şekil 4.7'deki (negatif bölge) sistem cevap eğrileri kullanılmıştır.



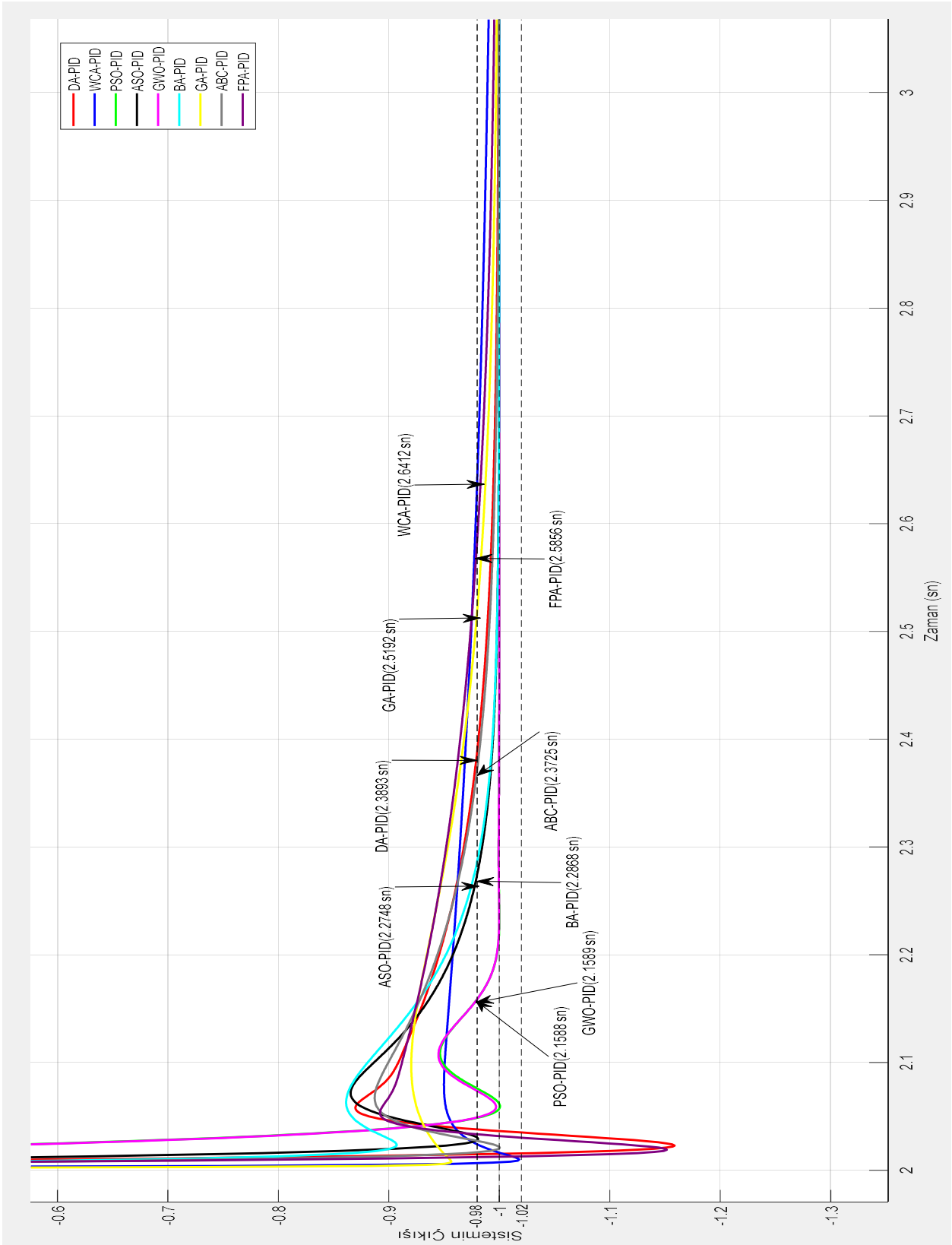
Şekil 4.2: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin pozitif bölgedeki yükselme zamanları (%10-%90 kriterine göre)



Şekil 4.3: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin negatif bölgedeki yükselme zamanları (%10-%90 kriterine göre)

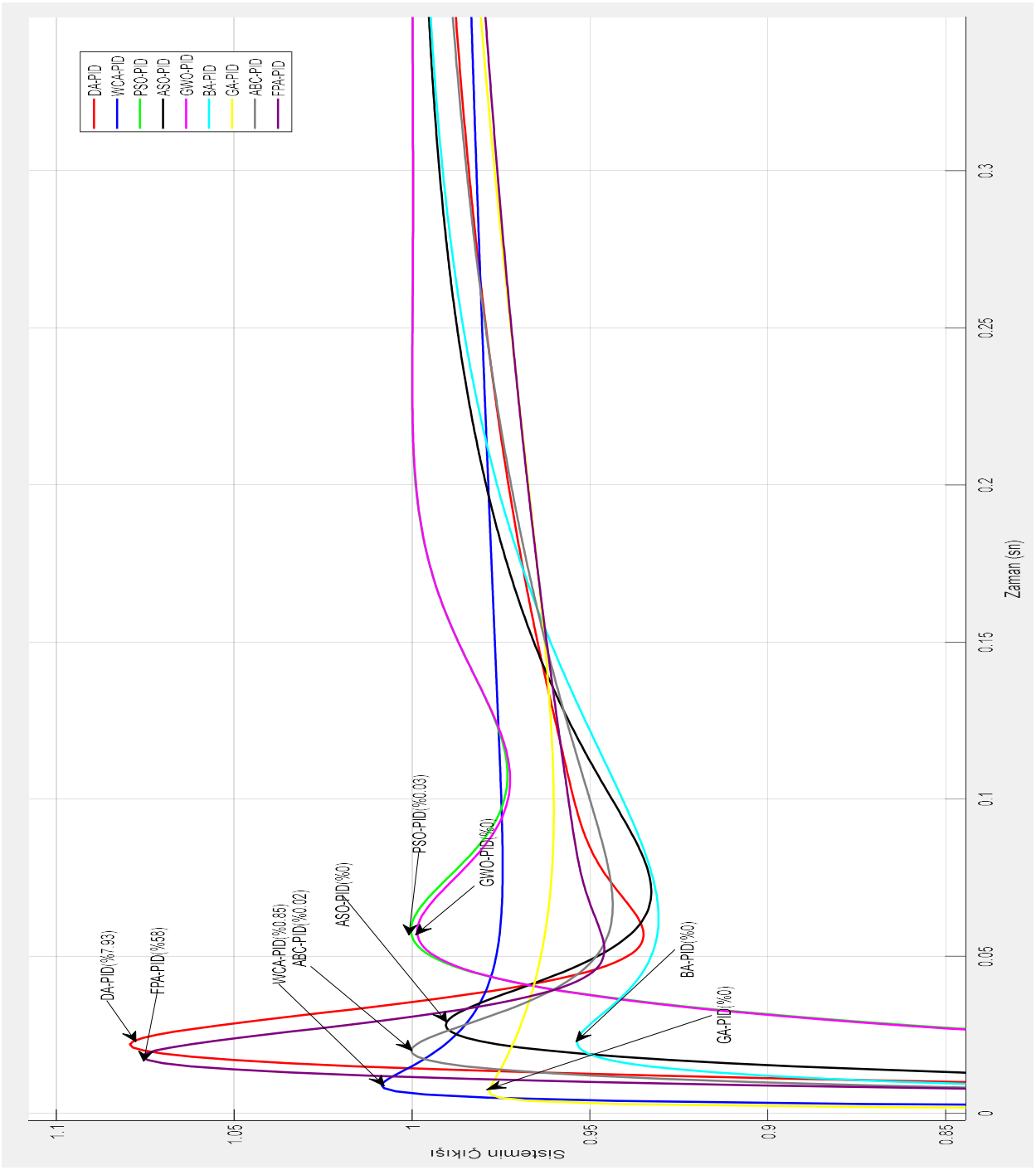


Şekil 4.4: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin pozitif bölgedeki yerleşme zamanları (%2 bandına göre)



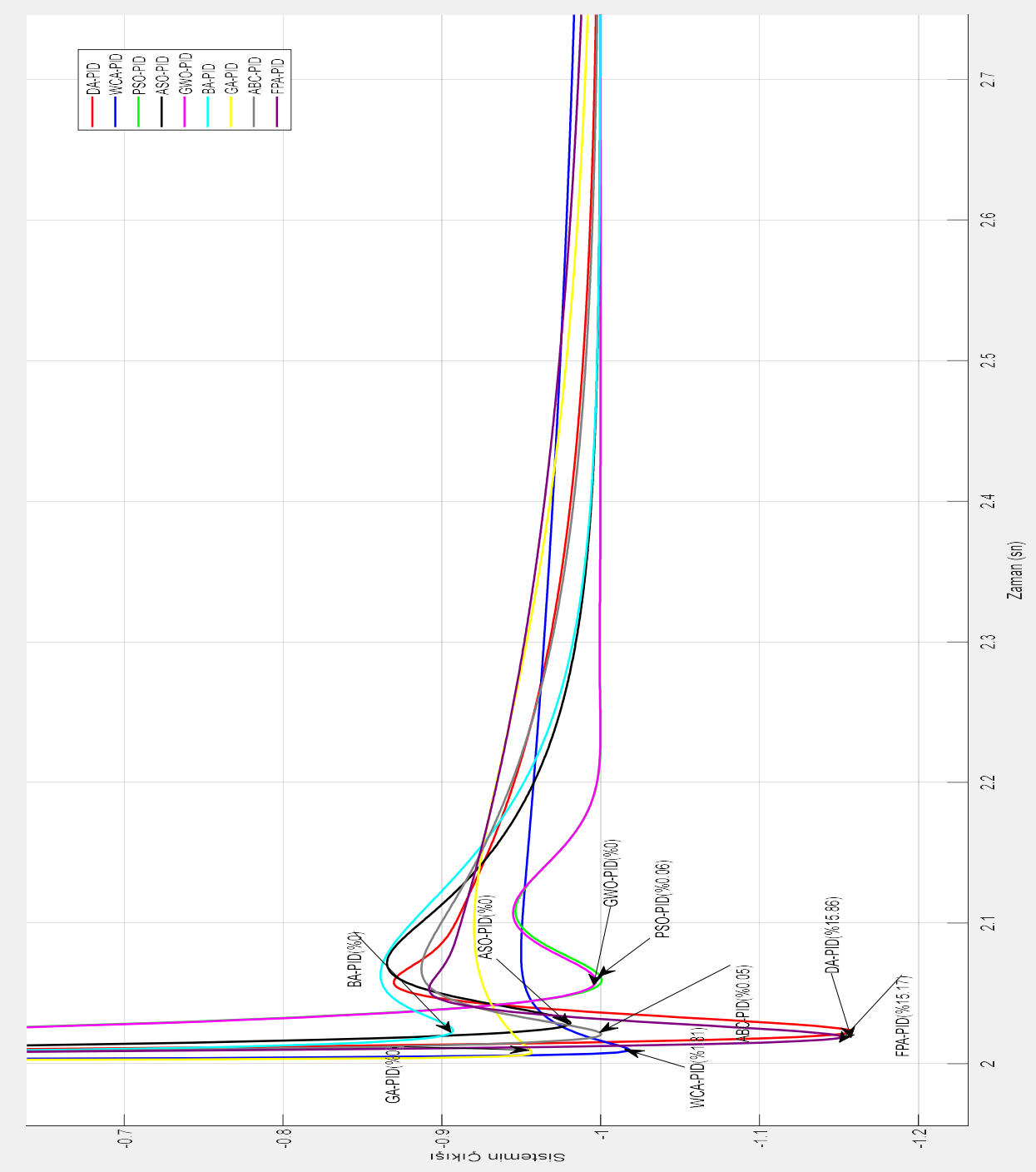
Şekil 4.5: Metasezgisel algoritmalarla optimize edilmiş PID kontrolöre sahip sisteme ilişkin negatif bölgedeki yerleşme zamanları (%2 bandına göre)

Şekil 4.6'dan (0-2) saniye arasındaki pozitif bölgede yüzde aşım değerlerinin küçükten büyüğe doğru GWO-PID (0%), BA-PID (0%), ASO-PID (0%), GA-PID (0%), ABC-PID (0.02%), PSO-PID (0.03%), WCA-PID (0.85%), DA-PID (7.93%) ve FPA-PID (7.58%) ve olarak sıralı olduğu görülür.



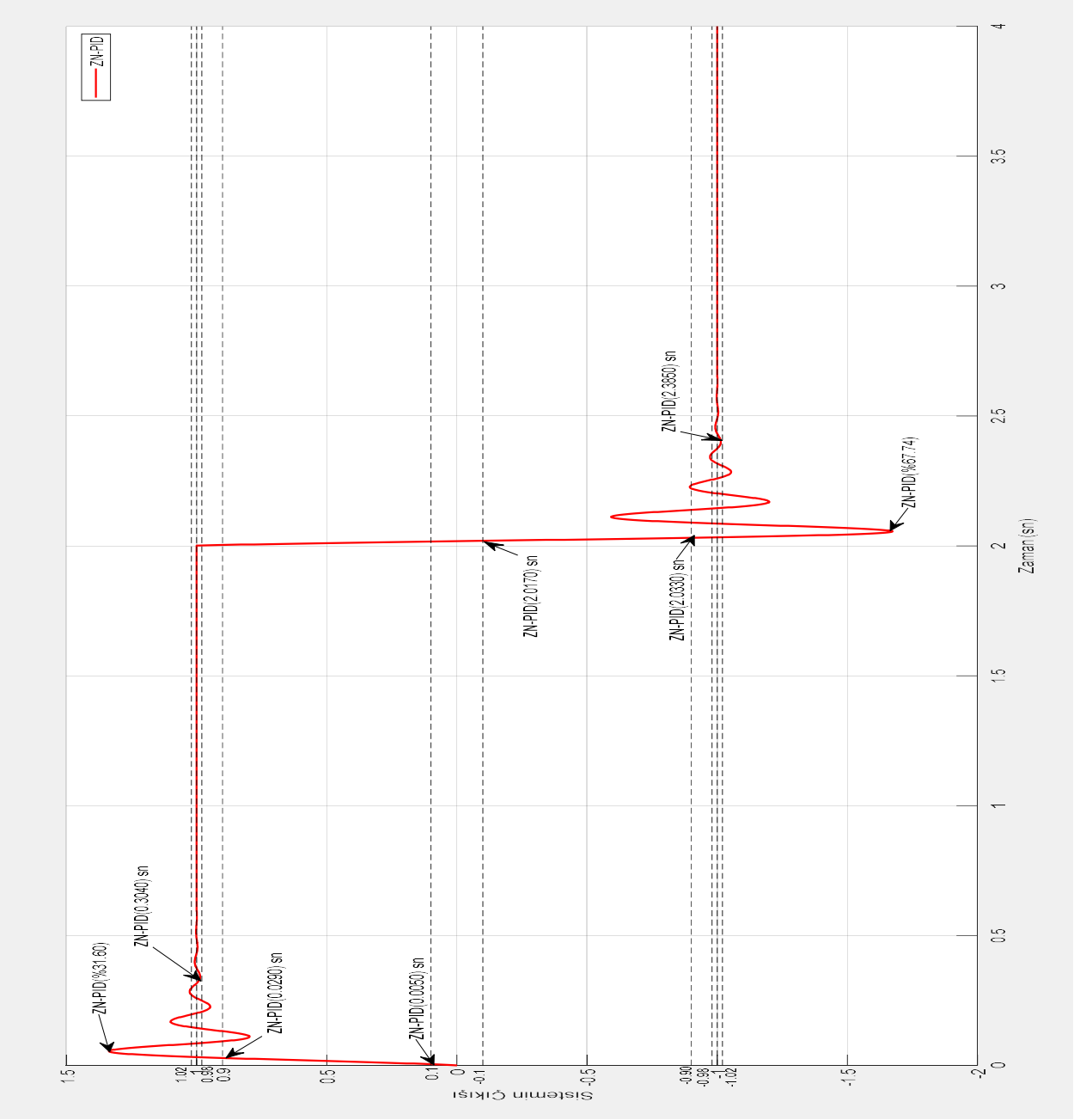
Şekil 4.6: Metasezgisel algoritmaların optimize edilmiş PID kontrolöre sahip sisteme ilişkin pozitif bölgedeki yüzde aşım değerleri

Şekil 4.7'den (2-4) saniye arasındaki negatif bölgede yüzde aşım değerlerinin küçükten büyüğe doğru GWO-PID (0%), BA-PID (0%), ASO-PID (0%), GA-PID (0%), PSO-PID (0.06%), ABC-PID (0.05%), WCA-PID (1.81%), FPA-PID (15.17%) ve DA-PID (15.86%) olarak sıralı olduğu görülür.



Şekil 4.7: Metasezgisel algoritmaların optimize edilmiş PID kontrolöre sahip sisteme ilişkin negatif bölgedeki yüzde aşım değerleri

Ziegler-Nichols yöntemi ile optimize edilmiş PID kontrolöre ilişkin pozitif ve negatif bölgelerdeki yerleşme zamanı (%2), yükselme zamanları (%10-%90) ve yüzde aşım değerleri Şekil 4.8'de gösterilmiştir.



Şekil 4.8: Ziegler-Nichols kontrol yöntemi ile optimize edilmiş PID kontrolöre ilişkin pozitif ve negatif bölgelerdeki yerleşme zamanları (%2), yükselme zamanları (%10-%90) ve yüzde aşım değerleri

Fırçasız DC motor modelimizin kontrolünde PID kontrolör parametrelerini ayarlamak için kullanılan klasik Ziegler-Nichols yöntemi kullanılması sonucu bulunan yükselme zamanı, yerleşme zamanı ve yüzde aşım miktarları Şekil 4.8'de, Tablo 4.1 ve Tablo 4.2'de gösterilmiştir. Buna göre pozitif bölge için yerleşme süresi 0.3040 saniye ve yükselme zamanı (%10'dan %90'a ulaşma süresi) 0.024 saniye olarak bulunmuştur. Negatif bölge için ise yerleşme süresi 2.3850 saniye ve yükselme zamanı ise 0.016 saniye olarak elde edilmiştir. Ayrıca pozitif bölge için %31.6 aşım görülürken; negatif bölge için ise %67.74 aşım yaptığı görülmüştür.

Şekil 4.2 ile Şekil 4.8 arasındaki tüm sistem cevap eğrilerine ilişkin şekillerde hem pozitif hem negatif bölgelerde görülen yükselme zamanı, yerleşme zamanı ve yüzde aşım değerleri, Tablo 4.1 ve Tablo 4.2'de topluca derlenerek tüm yöntemler için performans karşılaştırılması yapılmıştır.

Tablo 4.1: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PID kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin pozitif bölge (0-2 sn arası) performans kriter değerleri

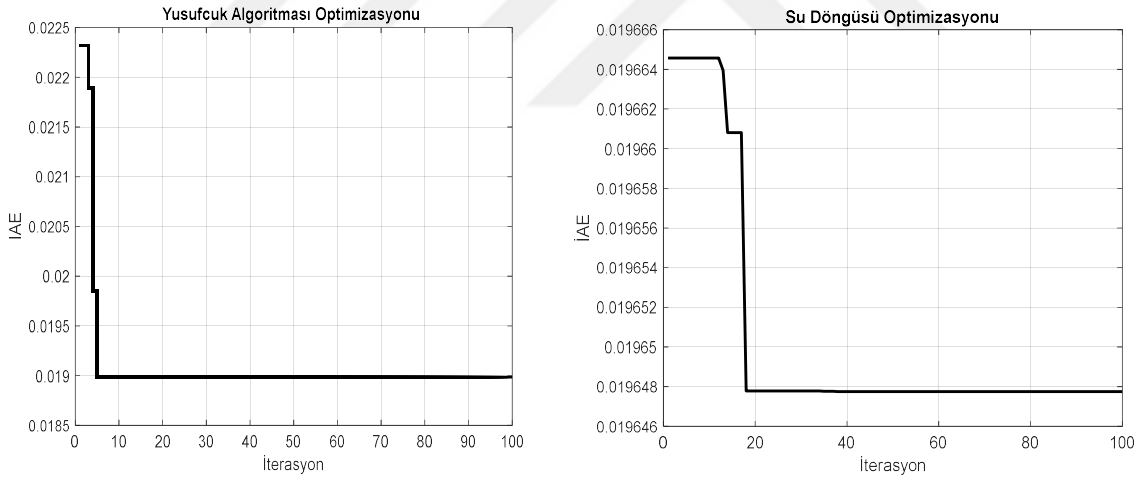
Optimizasyon Kontrol Yöntemleri	Aşım (%)	tr (yükselme süresi) %10-%90	ty (yerleşme süresi) %2
DA-PID	7.93	0.0104 sn (0.0112-0.0008)	0.2576 sn
WCA-PID	0.85	0.0032 sn (0.0002-0.0034)	0.2415 sn
GWO-PID	0	0.0295 sn (0.0019-0.0314)	0.1338 sn
GA-PID	0	0.0023 sn (0.0001-0.0024)	0.3401 sn
ASO-PID	0	0.0145 sn (0.0009-0.0154)	0.2041 sn
PSO-PID	0.03	0.0297 sn (0.0018-0.0315)	0.1336 sn
ABC-PID	0.02	0.0093 sn (0.0005-0.0098)	0.2556 sn
BA-PID	0	0.0114 sn (0.0005-0.0119)	0.2178 sn
FPA-PID	7.58	0.0083 sn (0.0006-0.0089)	0.3592 sn
ZN-PID	31.60	0.0240 sn (0.0050-0.0290)	0.3040 sn

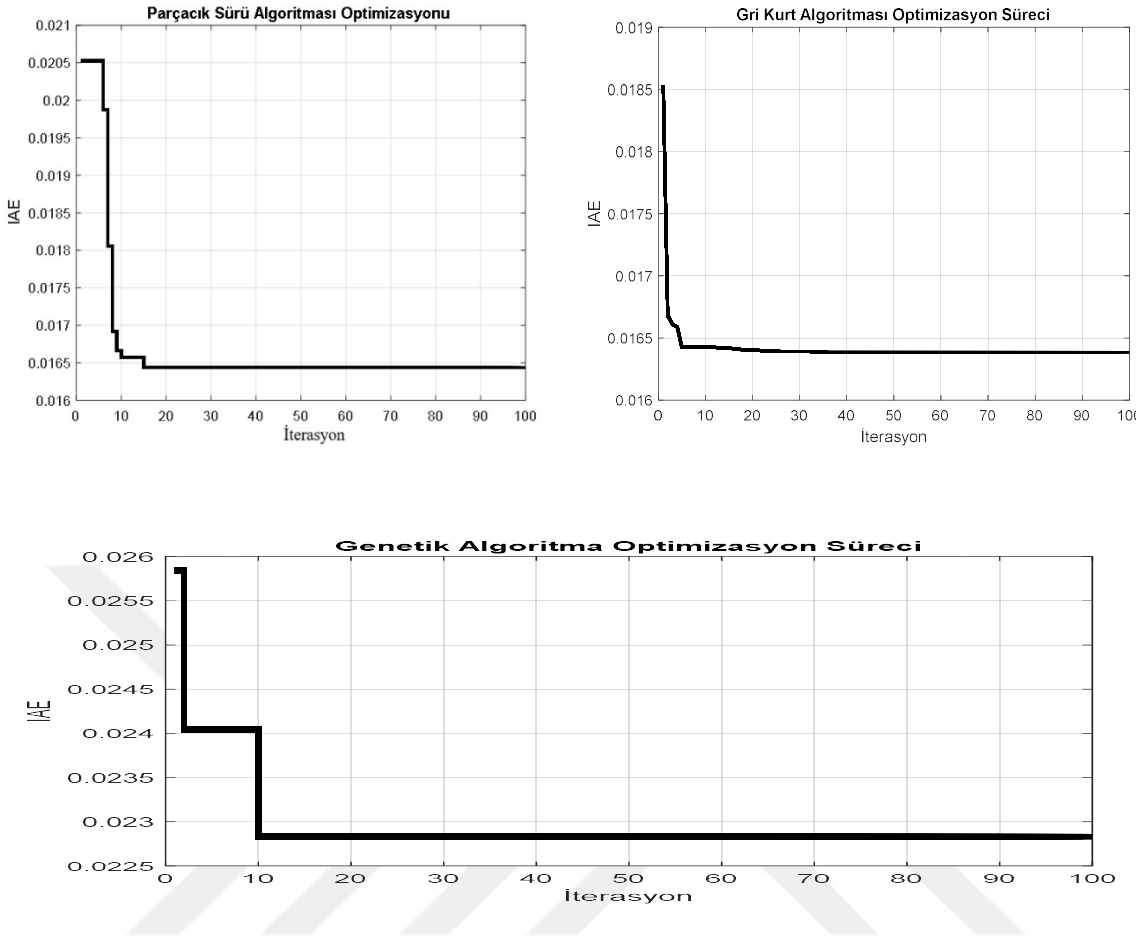
Tablo 4.2: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PID kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin negatif bölge (2-4 sn arası) performans kriter değerleri

Optimizasyon Kontrol Yöntemleri	Aşım (%)	tr (yükselme süresi) %10-%90	ty (yerleşme süresi) %2
DA-PID	15.86	0.0072 sn (2.0068-2.0140)	2.3893 sn
WCA-PID	1.81	0.0029 sn (2.0031-2.0060)	2.6412 sn
GWO-PID	0	0.0246 sn (2.0144-2.0390)	2.1589 sn
GA-PID	0	0.0025 sn (2.0025-2.0050)	2.5192 sn
ASO-PID	0	0.0045 sn (2.0073-2.0118)	2.2748 sn
PSO-PID	0.06	0.0246 sn (2.0144-2.0390)	2.1588 sn
ABC-PID	0.05	0.0082 sn (2.0048-2.0130)	2.3725 sn
BA-PID	0	0.0151 sn (2.0049-2.0200)	2.2868 sn
FPA-PID	15.17	0.0059 sn (2.0051-2.0118)	2.5856 sn
ZN-PID	67.74	0.0160 sn (2.0170-2.0330)	2.3850 sn

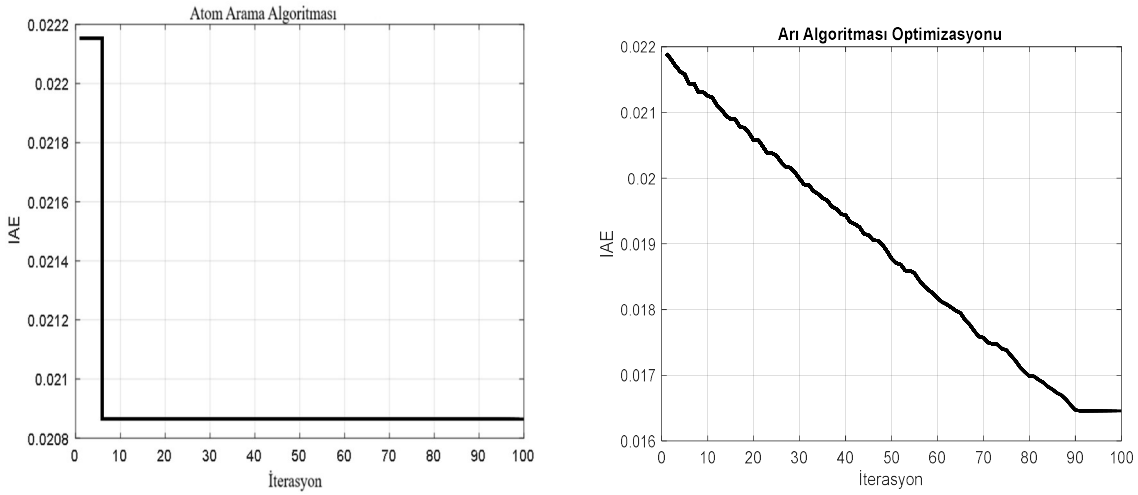
Tablo 4.1 ve Tablo 4.2'den PID kontrolöre sahip fırçasız dc motor sistemimiz için metasezgisel algoritmalar ile Ziegler-Nichols yöntemi performans kriterleri açısından kıyaslandığında; yerleşme süreleri, yükselme zamanları ve yüzdesel aşımlar için genelde metasezgisel algoritmaların hem negatif hem de pozitif bölgede sistemin daha iyi performans sergilediği görülmüştür. Dokuz adet metasezgisel algoritmaların kendi aralarında performans karşılaştırılması yapıldığında ise hem pozitif hem de negatif bölge için %2 bandına göre yerleşme sürelerinde GWO-PID ve PSO-PID'nin en iyi sonucu verdiği görülür. %10'ından %90'ına ulaşma süreleri açısından yükselme süreleri karşılaştırıldığında ise hem pozitif hem de negatif bölge için en iyi sonucu GA-PID ve WCA-PID'nin verdiği görülür. Yüzde aşım değerleri açısından karşılaştırıldıklarında ise GWO-PID, GA-PID, ASO-PID ve BA-PID için motor sisteminin hiç aşım yapmadığı görülür. Tüm bu sonuçlar üç performans kriteri açısından topluca değerlendirildiğinde fırçasız dc motor sistemimiz için GWO-PID'nin yerleşme süresi ve yüzde aşım performans kriterleri ile öne çıktığı görülür.

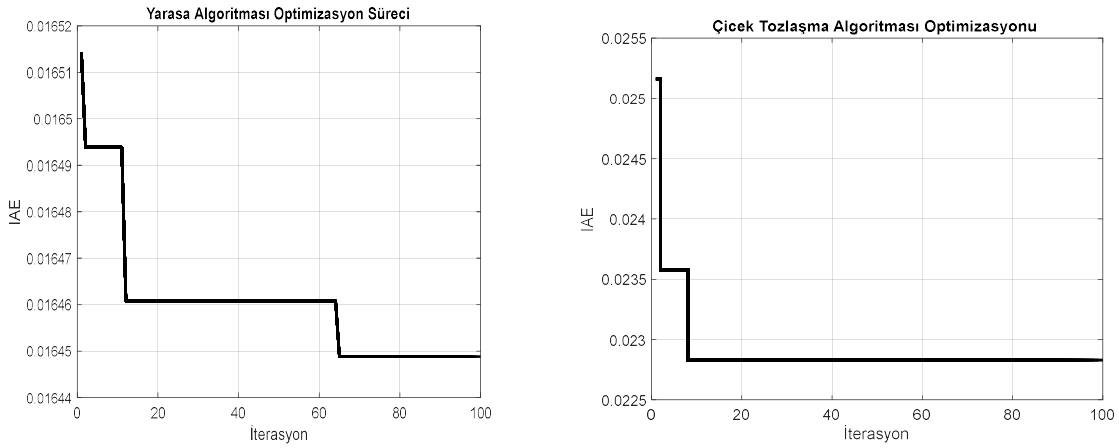
Bu tezde kullanılan tüm metasezgisel algoritmaların IAE amaç fonksiyonu sıfıra yaklaşım eğrileri, Şekil 4.9 ve Şekil 4.10'da gösterilmiştir.





Şekil 4.9: DA,WCA,GWO,GA,PSO metasezgisel algoritmalarının PID kontrolör yakınsama eğrileri





Şekil 4.10: ASO, ABC, BA, FPA metasezgisel algoritmalarının PID kontrolör yakınsama eğrileri

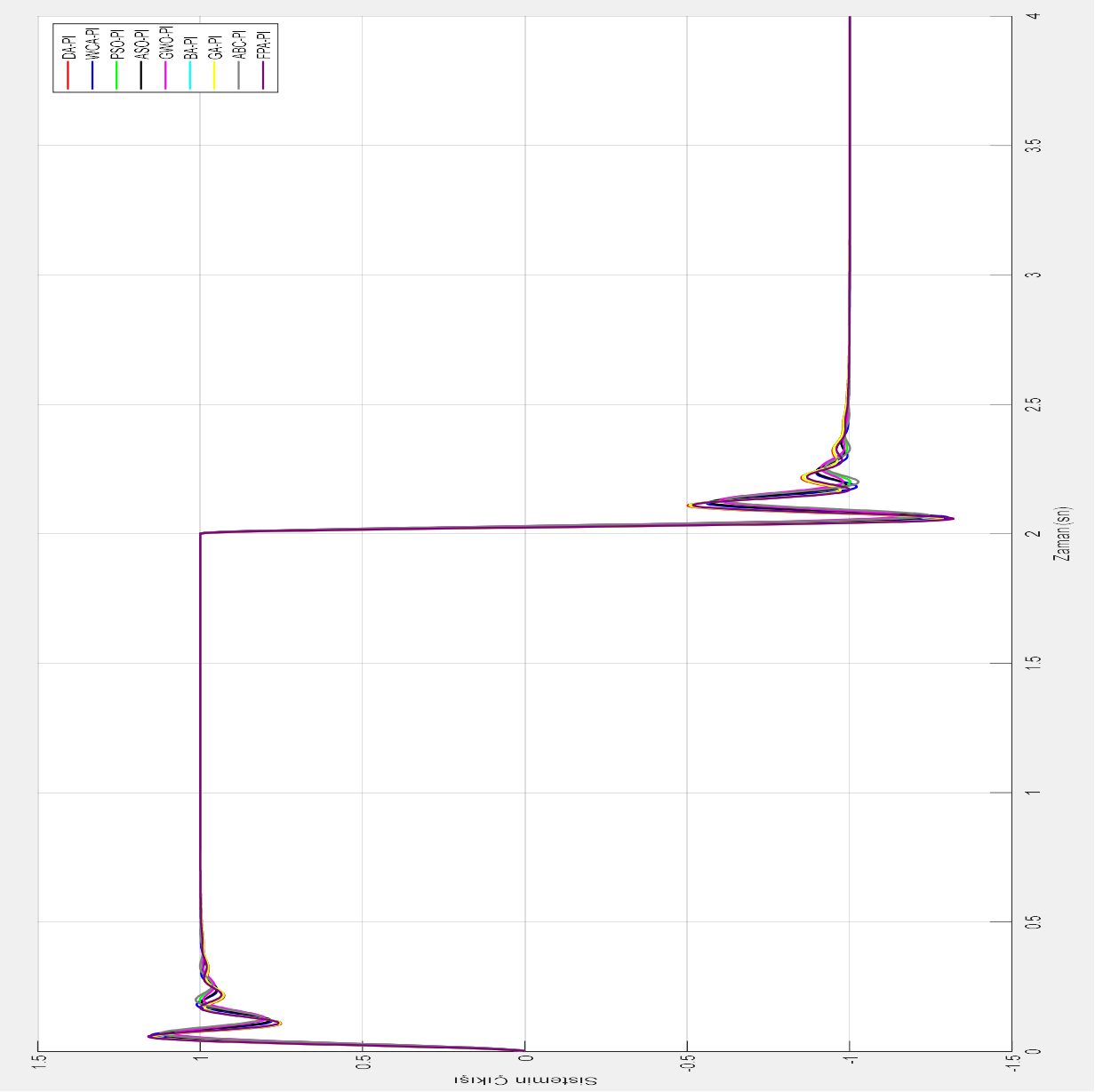
Metasezgisel algoritmalar ile amaç fonksiyonlarından sıfıra yakın değer elde edilmeye çalışılarak hatalar minimize edilmeye uğraşılır. Şekil 4.9 ve Şekil 4.10'daki metasezgisel algoritmaların yakınsama eğrileri incelendiğinde 0.0164 ile optimizasyon sürecinde en düşük değeri alan GWO algoritmasıdır. Metasezgisel algoritmaların bilgisayar ortamında optimizasyon çalışma süreleri Tablo 4.3'te verilmiştir. Bu süreler hesaplanırken AMD Ryzen 7 7840HS w/ Radeon 780M Graphics 3.80 GHz işlemcili bir bilgisayar kullanılmıştır. Tablo 4.3'ten FPA-PID algoritmasının çalışma süresi olarak en kısa optimizasyon süresine sahip olduğu görülür.

Tablo 4.3: PID kontrolöre sahip sistem için metasezgisel algoritmaların optimizasyon çalışma süreleri

Metasezgisel Algoritmalar	**Çalışma Süreleri (sn)**
DA-PID	12.7636
WCA-PID	9.1017
GWO-PID	10.2305
GA-PID	9.1123
ASO-PID	9.1296
PSO-PID	7.1209
ABC-PID	11.2772
BA-PID	7.2450
FPA-PID	6.4330

3.4. PI KONTROLÖR İÇİN SONUÇLAR

Yusufçuk algoritması (DA), su döngüsü algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa algoritması (BA) ve çiçek tozlaşma algoritması (FPA) kullanılarak optimize edilen PI kontrolörlerin fırçasız DC motor sistem modelimize birim basamak referans giriş altında uygulanması sonucu 0-2 saniye ve 2-4 saniye arasında elde edilen sistem cevap eğrileri hem pozitif hem de negatif bölge için birlikte Şekil 4.11'de gösterilmiştir.

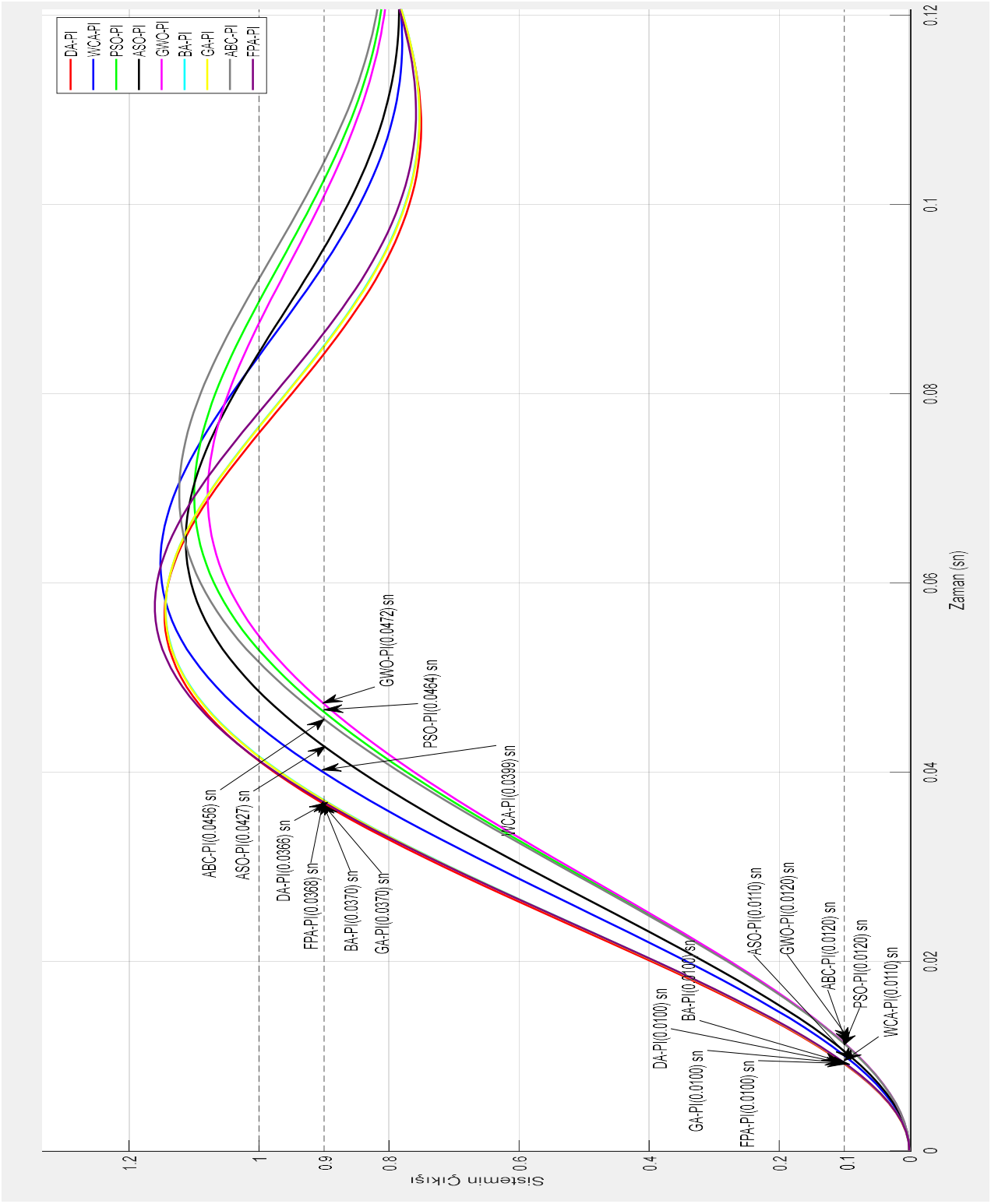


Şekil 4.11: Optimize edilmiş PI kontrolörler uygulanmış bir fırçasız DC motor sisteminin (0-4) saniye aralığında cevap eğrileri

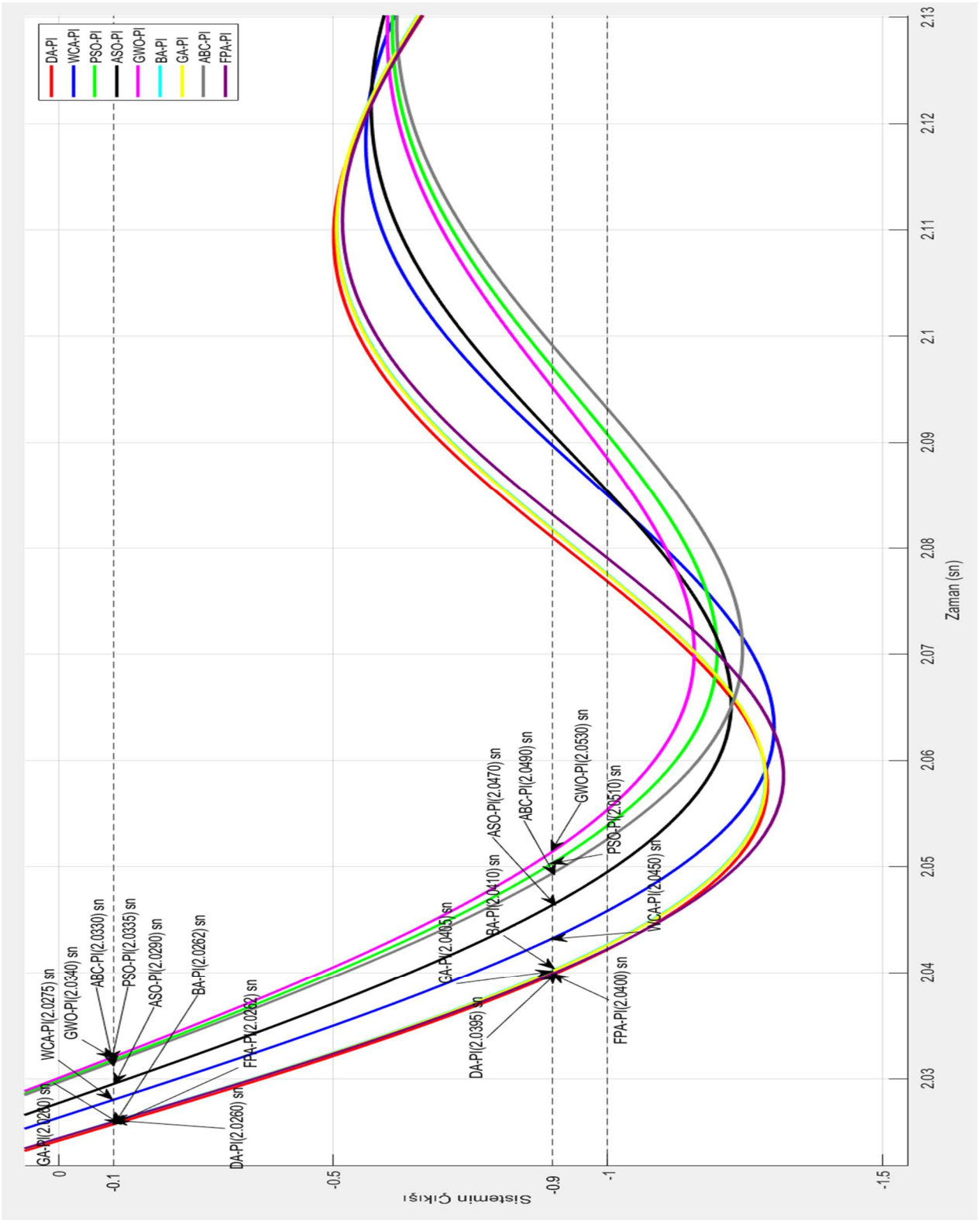
Metasezgisel optimizasyon tabanlı PI kontrolörlere sahip fırçasız DC motor modelimizin sistem çıkışlarının birbirleri ile performans kriterlerinden yükselme zamanları açısından karşılaştırılması için Şekil 4.12 (pozitif bölge) ve Şekil 4.13'teki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Hem pozitif bölgede (0-2 sn arası) hem de negatif bölgede (2-4 sn arası) fırçasız DC motorun çıkış işaretlerinden yükselme zamanları için %10'ından %90'ına ulaşma süreleri dikkate alınarak ayrı ayrı karşılaştırılmaları yapılmıştır. Şekil 4.12'den 0 ile 2 saniye arasındaki pozitif bölgede yükselme zamanı değerlerinin küçükten büyüğe doğru DA-PI (0.0266 sn), FPA-PI (0.0268 sn), BA-PI (0.0270 sn), GA-PI (0.0270 sn), WCA-PI (0.0289 sn), ASO-PI (0.0317 sn), ABC-PI (0.0336 sn), PSO-PI (0.0344 sn) ve GWO-PI (0.0352 sn) olarak sıralı olduğu görülür. Şekil 4.13'ten (2-4) saniye arasındaki negatif bölgede yükselme zamanı değerleri ise küçükten büyüğe doğru DA-PI (0.0135 sn), FPA-PI (0.0138 sn), GA-PI (0.0145 sn), BA-PI (0.0148 sn), ABC-PI (0.0160 sn), WCA-PI (0.0175 sn), PSO-PI (0.0175 sn), ASO-PI (0.0180 sn) ve GWO-PI (0.0190 sn) olarak sıralanmıştır.

Metasezgisel optimizasyon tabanlı PI kontrolörlere sahip sistem çıkışlarının birbirleri ile performans kriterlerinden yerleşme zamanları açısından karşılaştırılması için Şekil 4.14 (pozitif bölge) ve Şekil 4.15'teki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Hem pozitif bölgede (0-2 sn arası) hem de negatif bölgede (2-4 sn arası) fırçasız DC motorun çıkış işaretlerinden yerleşme zamanları için %2'lik yerleşme bandı dikkate alınarak ayrı ayrı karşılaştırılmaları yapılmıştır. Şekil 4.14'te 0 ile 2 saniye arasındaki pozitif bölgede yerleşme zamanı değerlerinin küçükten büyüğe doğru WCA-PI (0.2703 sn), ASO-PI (0.2830 sn), ABC-PI (0.2875 sn), PSO-PI (0.2910 sn), GWO-PI (0.2965 sn), FPA-PI (0.3313 sn), GA-PI (0.3483 sn), BA-PI (0.3483 sn) ve DA-PI (0.3495 sn) olarak sıralı olduğu görülür. Şekil 4.15'ten 2 sn ile 4 saniye arasındaki negatif bölgede yerleşme zamanı değerleri ise küçükten büyüğe doğru ABC-PI (2.3010 sn), PSO-PI (2.3069 sn), FPA-PI (2.3706 sn), WCA-PI (2.3720 sn), ASO-PI (2.3873 sn), GWO-PI (2.3938 sn), BA-PI (2.4008 sn), GA-PI (2.4049 sn) ve DA-PI (2.4378 sn) olarak sıralanmıştır.

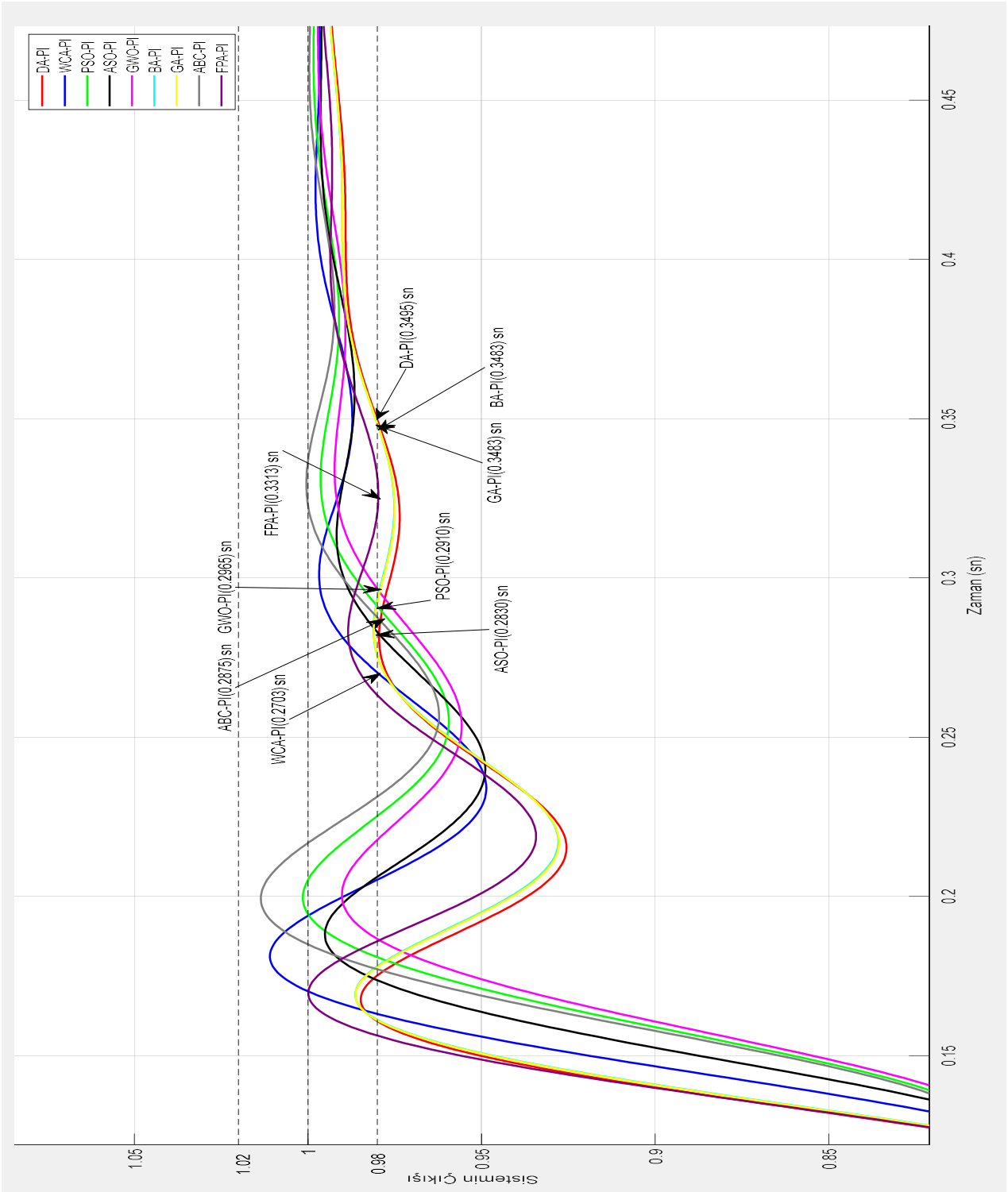
Metasezgisel optimizasyon tabanlı PID kontrolörlere sahip sistem çıkışlarının birbirleri ile performans kriterlerinden yüzde aşım açısından karşılaştırılması için Şekil 4.16 (pozitif bölge) ve Şekil 4.17'deki (negatif bölge) sistem cevap eğrileri kullanılmıştır



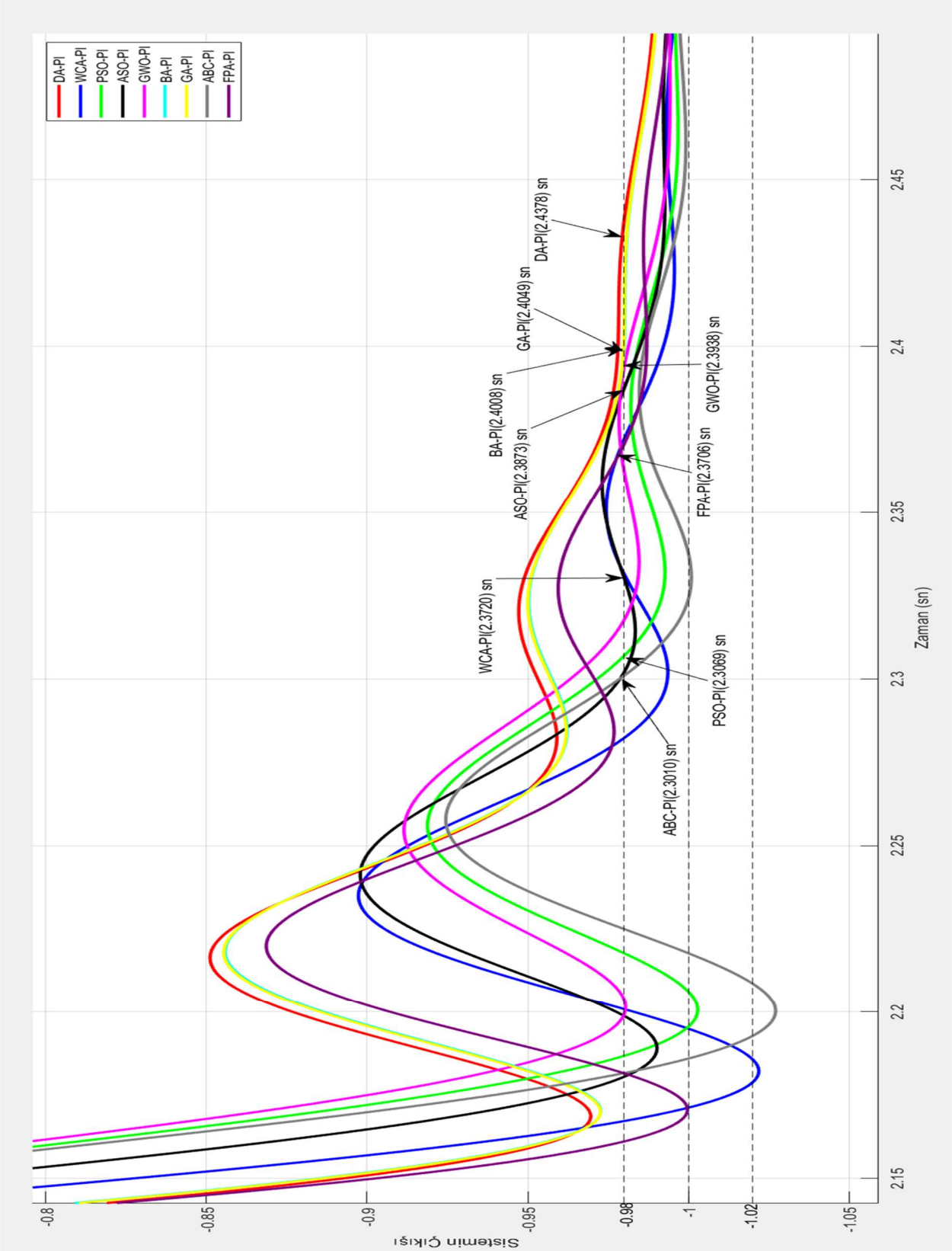
Şekil 4.12: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yükselme zamanları (%10-%90 kriterine göre)



Şekil 4.13: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin negatif bölgedeki yükselme zamanları (%10-%90 kriterine göre)

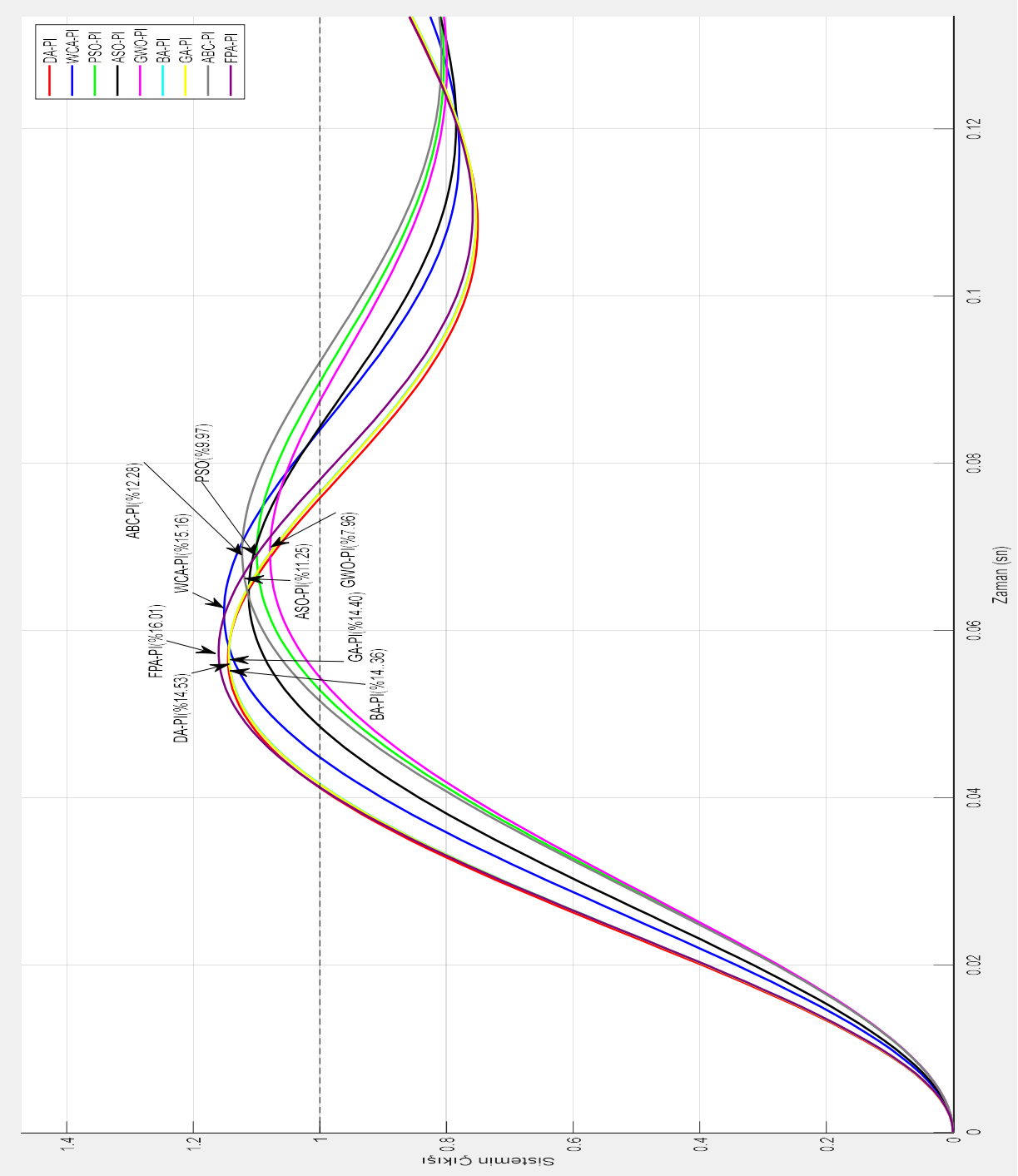


Şekil 4.14: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yerleşme zamanları (%2 bandına göre)



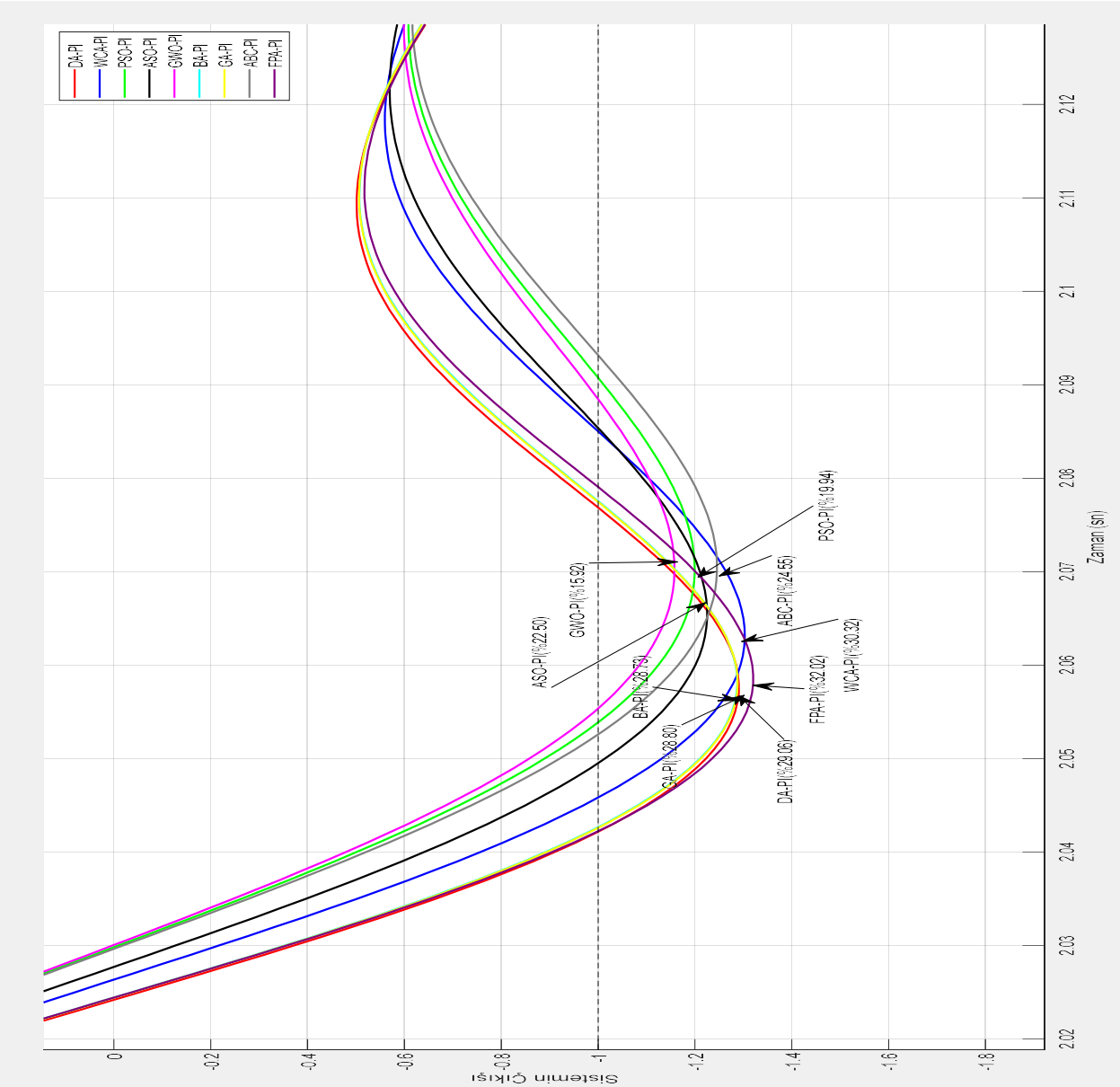
Şekil 4.15: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin negatif bölgedeki yerleşme zamanları (%2 bandına göre)

Şekil 4.16'dan (0-2) saniye arasındaki pozitif bölgede yüzde aşım değerlerinin küçükten büyüğe doğru GWO-PI (7.96%), PSO-PI (9.97%), ASO-PI (11.25%), ABC-PI (12.28%), BA-PI (14.36%), GA-PI (14.40%), DA-PI (14.53%), WCA-PI (15.16%) ve FPA-PI (16.01%) ve olarak sıralı olduğu görülür.



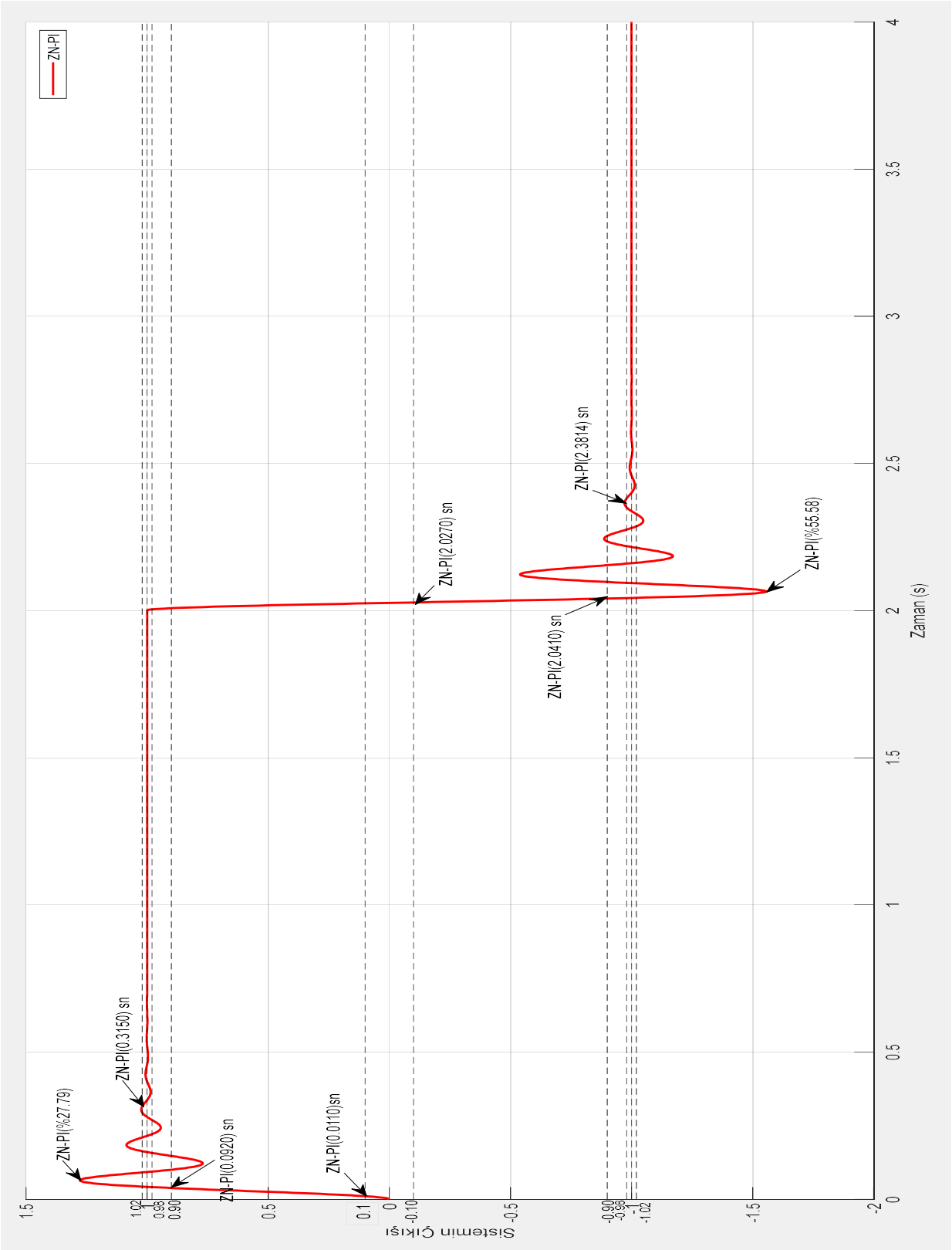
Şekil 4.16: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yüzde aşım değerleri

Şekil 4.7'den (2-4) saniye arasındaki negatif bölgede yüzde aşım değerlerinin küçükten büyüğe doğru GWO-PI (15.92%), PSO-PI (19.94%), ASO-PI (22.50%), ABC-PI (24.55%), BA-PI (28.73%), GA-PI (28.80%), DA-PI (29.06%), WCA-PI (30.32%) ve FPA-PI (32.02%) olarak sıralı olduğu görülür.



Şekil 4.17: Metasezgisel algoritmaların optimize edilmiş PI kontrolöre sahip sisteme ilişkin negatif bölgedeki yüzde aşım değerleri

Ziegler-Nichols yöntemi ile optimize edilmiş PID kontrolöre ilişkin pozitif ve negatif bölgelerdeki yerleşme zamanı (%2), yükselme zamanları (%10-%90) ve yüzde aşım değerleri Şekil 4.18'de gösterilmiştir.



Şekil 4.18: Ziegler-Nichols kontrol yöntemi ile optimize edilmiş PI kontrolöre ilişkin pozitif ve negatif bölgelerdeki yerleşme zamanları (%2), yükselme zamanları (%10-%90) ve yüzde aşım değerleri

Fırçasız DC motor modelimizin kontrolünde PI kontrolör parametrelerini ayarlamak için kullanılan klasik Ziegler-Nichols yöntemi kullanılması sonucu bulunan yükselme zamanı, yerleşme zamanı ve yüzde aşım miktarları Şekil 4.18'de, Tablo 4.4 ve Tablo 4.5'te gösterilmiştir. Buna göre pozitif bölge için yerleşme süresi 0.3150 saniye, yükselme zamanı (%10'dan %90'a ulaşma süresi) 0.0810 saniye ve aşım (27.79%) olarak bulunmuştur. Negatif bölge için ise yerleşme süresi 2.3814 saniye, yükselme zamanı 0.014 saniye ve aşım (55.58%) olarak elde edilmiştir. Ayrıca pozitif bölge için %27.79 aşım görülürken; negatif bölge için ise %55.58 aşım yaptığı görülmüştür.

Şekil 4.12 ile Şekil 4.18 arasındaki tüm sistem cevap eğrilerine ilişkin şekillerde hem pozitif hem negatif bölgelerde görülen yükselme zamanı, yerleşme zamanı ve yüzde aşım değerleri, Tablo 4.4 ve Tablo 4.5'te topluca derlenerek tüm yöntemler için performans karşılaştırılması yapılmıştır.

Tablo 4.4: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PI kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin pozitif bölge (0-2 sn arası) performans kriter değerleri

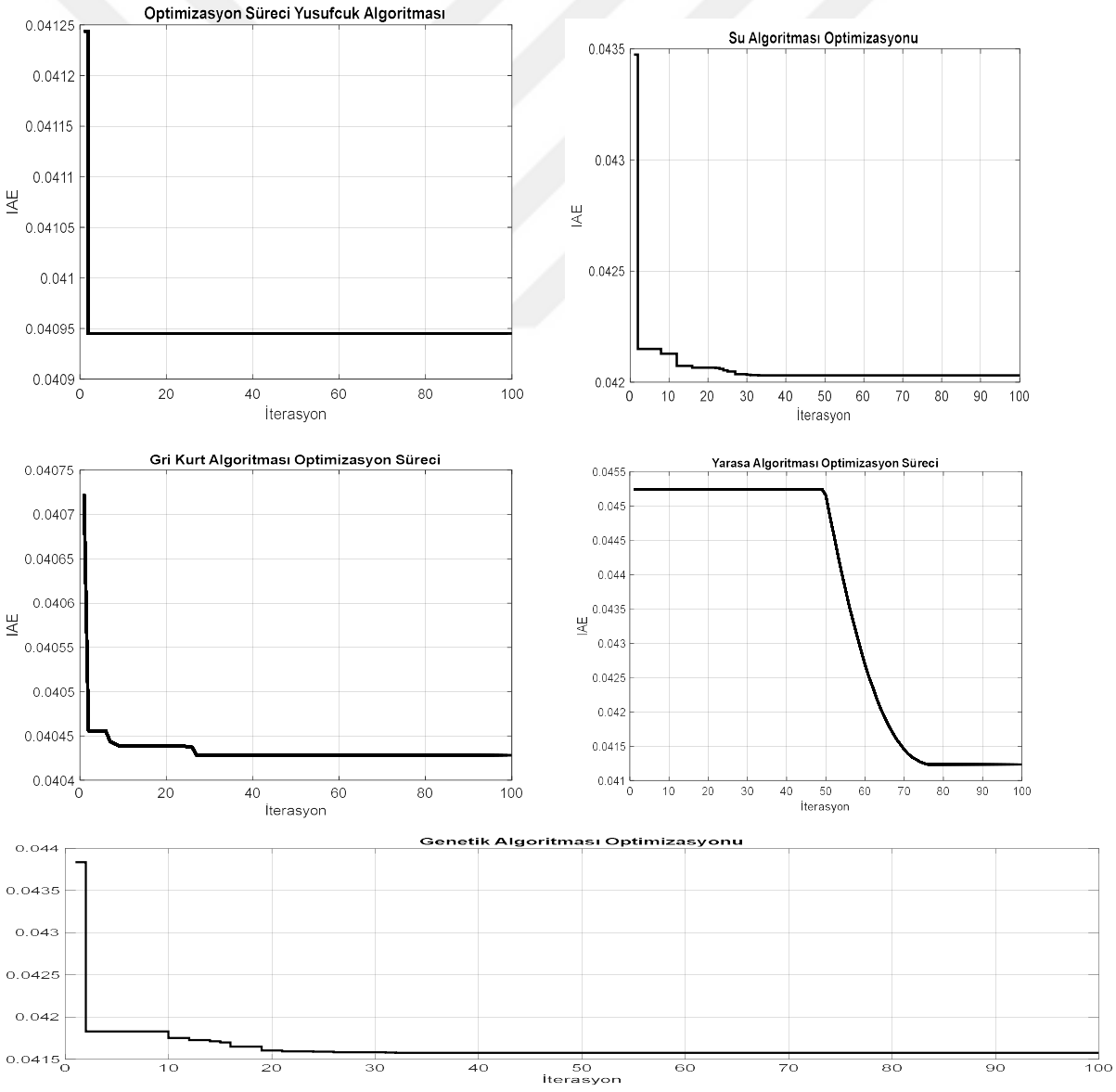
Optimizasyon Kontrol Yöntemleri	Aşım (%)	tr (yükselme süresi) %10-%90	ty (yerleşme süresi) %2
DA-PI	14.53	0.0266 sn (0.0100-0.0366)	0.3495 sn
WCA-PI	15.16	0.0289 sn (0.0110-0.0399)	0.2703 sn
GWO-PI	7.96	0.0352 sn (0.0120-0.0472)	0.2965 sn
GA-PI	14.40	0.0270sn (0.0100-0.0370)	0.3483 sn
ASO-PI	11.25	0.0317 sn (0.0110-0.0427)	0.2830 sn
PSO-PI	9.97	0.0344 sn (0.0120-0.0464)	0.2910 sn
ABC-PI	12.28	0.0336 sn (0.0120-0.0456)	0.2875 sn
BA-PI	14.36	0.0270 sn (0.0100-0.0370)	0.3483 sn
FPA-PI	16.01	0.0268 sn (0.0100-0.0368)	0.3313 sn
ZN-PI	27.79	0.0810 sn (0.0110-0.0920)	0.3150 sn

Tablo 4.5: Metasezgisel algoritmalar ve Ziegler Nichols yöntemi ile optimize edilmiş PI kontrolörlerin uygulandığı sistemin çıkış eğrilerine ilişkin negatif bölge (2-4 sn arası) performans kriter değerleri

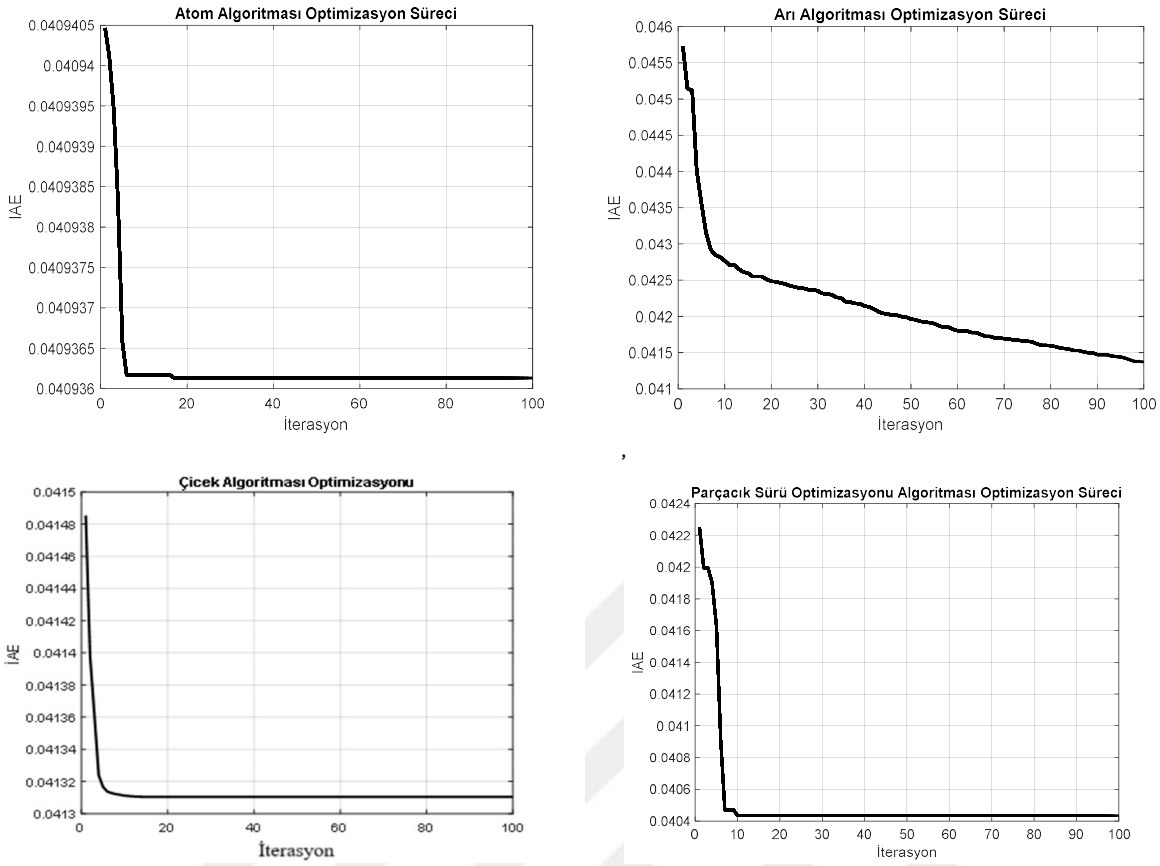
Optimizasyon Kontrol Yöntemleri	Aşım (%)	tr (yükselme süresi) %10-%90	ty (yerleşme süresi) %2
DA-PI	29.06	0.0135 sn (2.0260-2.0395)	2.4378 sn
WCA-PI	30.32	0.0175 sn (2.0275-2.0450)	2.3720 sn
GWO-PI	15.92	0.0190 sn (2.0340-2.0530)	2.3938 sn
GA-PI	28.80	0.0145 sn (2.0260-2.0405)	2.4049 sn
ASO-PI	22.50	0.0180 sn (2.0290-2.0470)	2.3873 sn
PSO-PI	19.94	0.0175 sn (2.0335-2.0510)	2.3069 sn
ABC-PI	24.55	0.0160 sn (2.0330-2.0490)	2.3010 sn
BA-PI	28.73	0.0148 sn (2.0262-2.0410)	2.4008 sn
FPA-PI	32.02	0.0138 sn (2.0262-2.0400)	2.3706 sn
ZN-PI	55.58	0.0140 sn (2.0270-2.0410)	2.3814 sn

Tablo 4.3 ve Tablo 4.4'ten PI kontrolöre sahip fırçasız dc motor sistemi için için metasezgisel algoritmalar ile Ziegler-Nichols yöntemi performans karşılaştırılması yapıldığında %2 bandına göre yerleşme süreleri açısından pozitif bölge için WCA-PI ve ASO-PI'nın en iyi sonucu verdiği görülürken; negatif bölge için PSO-PI, ABC-PI ve WCA-PI'nın en iyi sonucu verdiği görülür. %10'ından %90'ına ulaşma süreleri açısından yükselme süreleri karşılaştırıldığında ise hem pozitif hem de negatif bölge için en iyi sonucu DA-PI, FPA-PI ve BA-PI'nın verdiği görülür. Yüzde aşım değerleri açısından karşılaştırıldıklarında ise ZN-PI için motor sisteminin en fazla aşım yaptığı görülür. Yüzde aşım için karşılaştırıldıklarında ise hem pozitif hem de negatif bölge için en az aşımı GWO-PI'nın yaptığı görülür.

Bu tezde kullanılan tüm metasezgisel algoritmaların IAE amaç fonksiyonu sıfıra yaklaşım eğrileri, Şekil 4.19 ve Şekil 4.20'de gösterilmiştir.



Şekil 4.19: DA,WCA,GWO,GA,BA metasezgisel algoritmalarının PI kontrolör yakınsama eğrileri



Şekil 4.20: ASO, ABC, PSO, FPA metasezgisel algoritmalarının PI kontrolör yakınsama eğrileri

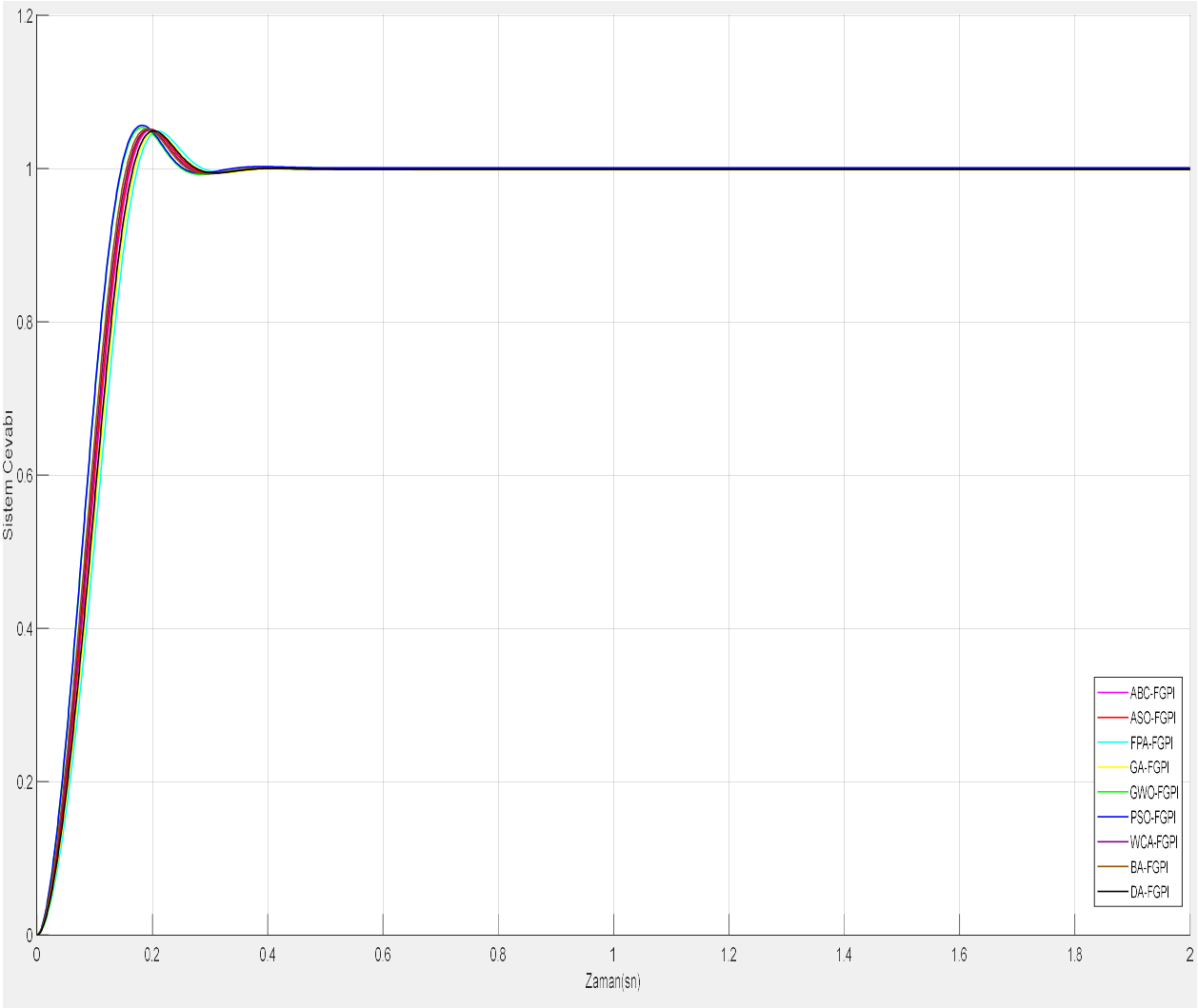
Şekil 4.19 ve Şekil 4.20'deki metasezgisel algoritmaların yakınsama eğrileri incelendiğinde optimizasyon sürecinde en düşük değeri olan 0.0404 ile PSO ve ona çok yakın 0.04043 ile GWO algoritmalarıdır. Tablo 4.7'den FPA-PI algoritmasının çalışma süresi olarak en kısa optimizasyon süresine sahip olduğu görülür.

Tablo 4.7: PI kontrolöre sahip sistem için metasezgisel algoritmaların optimizasyon çalışma süreleri

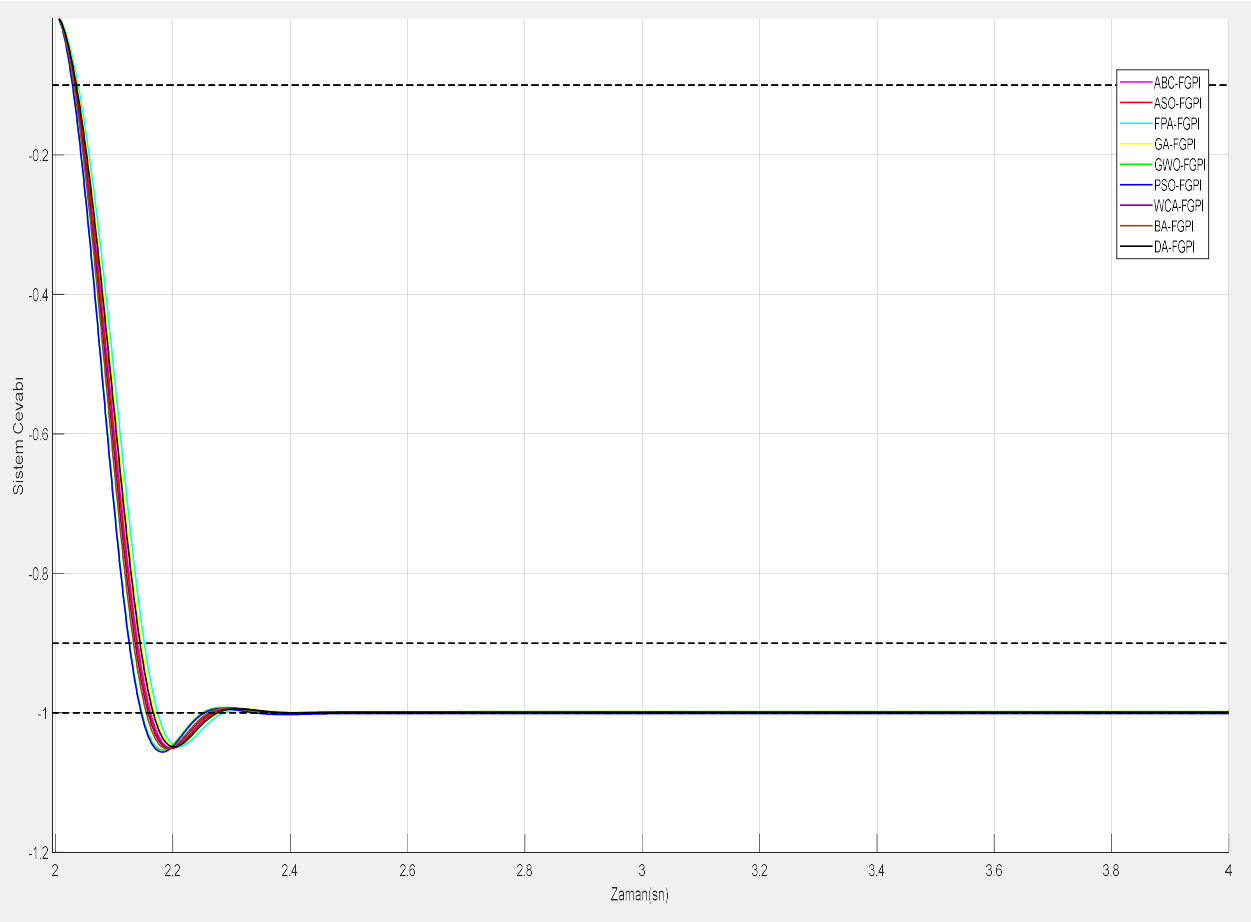
Metasezgisel Algoritmalar	**Çalışma Süreleri (sn)**
DA-PI	9.8068
WCA-PI	9.2448
GWO-PI	10.2287
GA-PI	9.5299
ASO-PI	10.5965
PSO-PI	11.1391
ABC-PI	11.9919
BA-PI	8.5955
FPA-PI	4.5475

4.3 BULANIK MANTIK PI (FGPI) KONTROLÖR İÇİN SONUÇLAR

Birim basamak referans giriş altında fırçasız DC motor modelimizin FGPI kontrolünde yusufçuk algoritması (DA), su döngüsü algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa algoritması (BA) ve çiçek tozlaşma algoritması (FPA) olmak üzere 9 adet metasezgisel algoritmanın performans kriterlerini birbirleriyle karşılaştırabilmek için 0-2 saniye ve 2-4 saniye arasında elde edilen sistem cevap eğrileri hem pozitif hem de negatif bölgede ayrı ayrı olmak üzere Şekil 4.21 ve Şekil 4.22'de gösterilmiştir.



Şekil 4.21: Optimize edilmiş FGPI kontrolörler uygulanmış bir fırçasız DC motor sisteminin 0-2 saniye aralığında cevap eğrileri



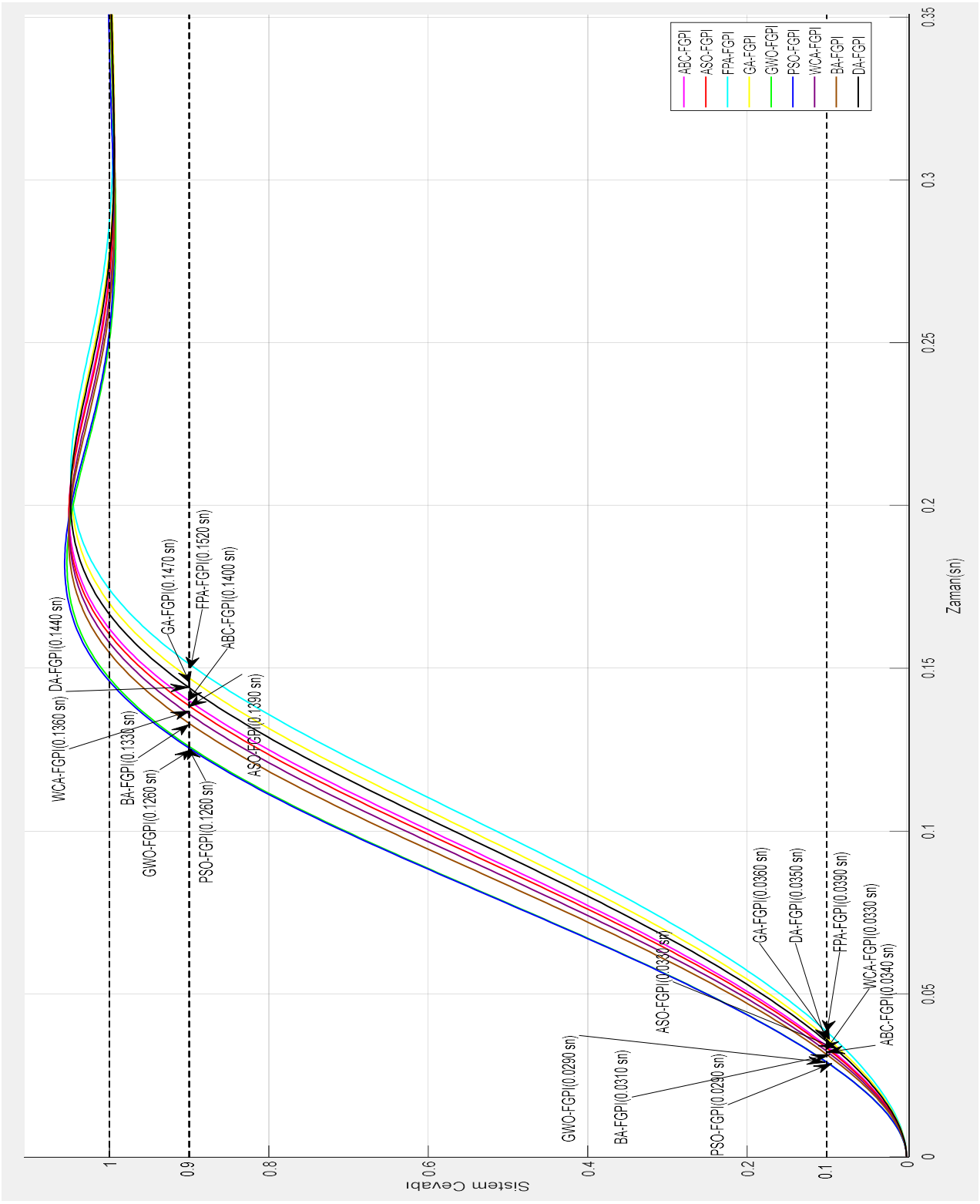
Şekil 4.22: Optimize edilmiş FGPI kontrolörler uygulanmış bir fırçasız DC motor sisteminin 2-4 sn aralığında cevap eğrileri

Metasezgisel optimizasyon tabanlı FGPI kontrolörlere sahip fırçasız DC motor modelimizin sistem çıkışlarının birbirleri ile performans kriterlerinden yükselme zamanları açısından karşılaştırılması için Şekil 4.23 (pozitif bölge) ve Şekil 4.24'teki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Şekil 4.23'ten 0 ile 2 saniye arasındaki pozitif bölgede yükselme zamanı değerlerinin küçükten büyüğe doğru GWO-FGPI (0.0970 sn), PSO-FGPI (0.0970 sn), BA-FGPI (0.1020 sn), WCA-FGPI (0.1030 sn), ABC-FGPI (0.1060 sn), ASO-FGPI (0.1060 sn), DA-FGPI (0.1090 sn), GA-FGPI (0.1110 sn) ve FPA-FGPI (0.1130 sn) olarak sıralı olduğu görülür. Şekil 4.24'ten (2-4) saniye arasındaki negatif bölgede yükselme zamanı değerleri ise küçükten büyüğe doğru GWO-FGPI (0.0970 sn), PSO-FGPI (0.0990 sn), BA-FGPI (0.1010 sn), WCA-FGPI (0.1040 sn), ASO-FGPI (0.1060 sn), ABC-FGPI (0.1080 sn), GA-FGPI (0.1100 sn) DA-FGPI (0.1100 sn) ve FPA-FGPI (0.1130 sn) olarak sıralanmıştır.

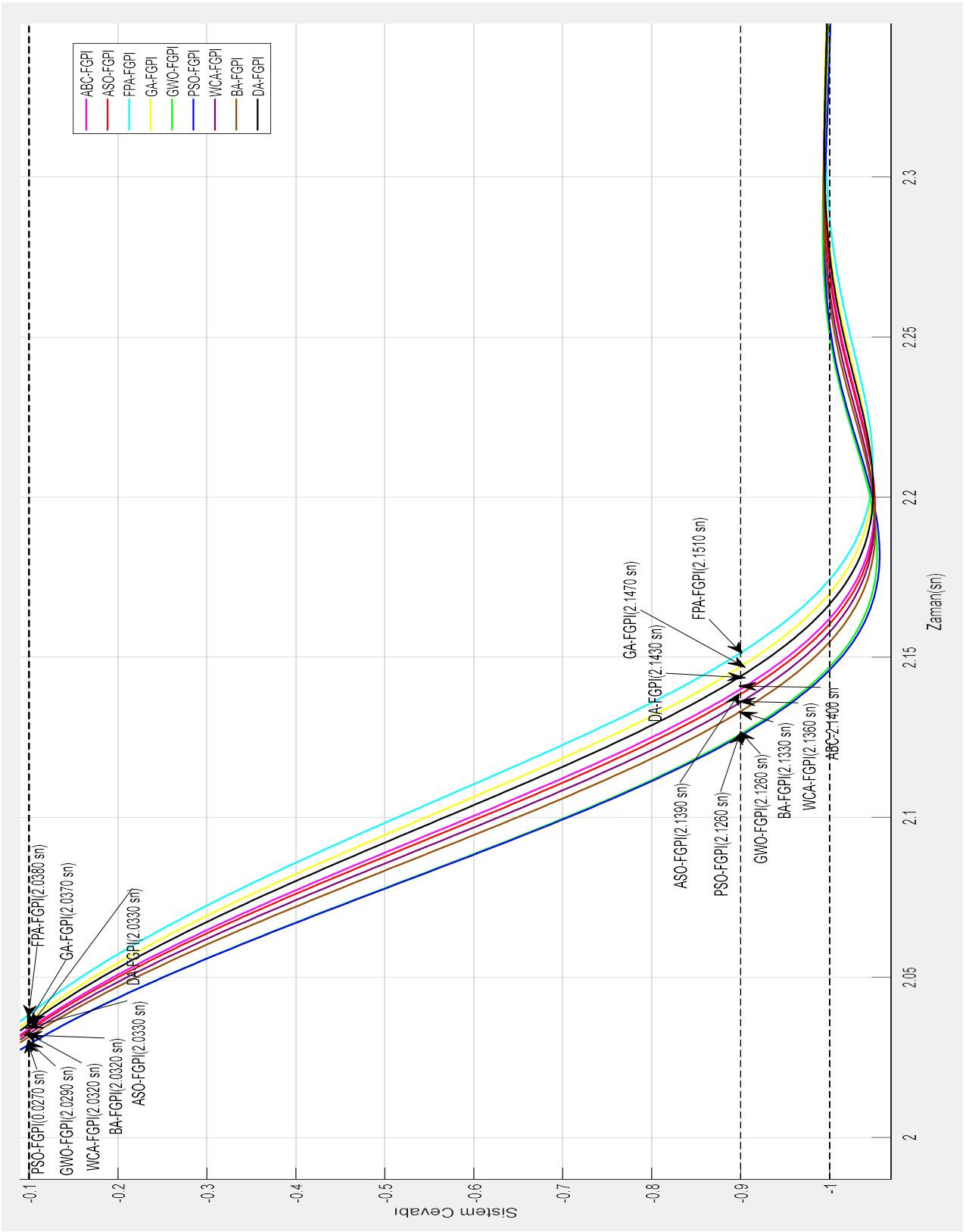
Metasezgisel optimizasyon tabanlı FGPI kontrolörlere sahip sistem çıkışlarının birbirleri ile performans kriterlerinden yerleşme zamanları açısından karşılaştırılması için Şekil 4.25 (pozitif bölge) ve Şekil 4.26'daki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Hem pozitif bölgede (0-2 sn arası) hem de negatif bölgede (2-4 sn arası) fırçasız DC motorun çıkış işaretlerinden yerleşme zamanları için %2'lik yerleşme bandı dikkate alınarak ayrı ayrı karşılaştırılmaları yapılmıştır. Şekil 4.25'ten 0 ile 2 saniye arasındaki pozitif bölgede yerleşme zamanı değerlerinin küçükten büyüğe doğru GWO-FGPI (0.2260 sn), PSO-FGPI (0.2270 sn), BA-FGPI (0.2340 sn), WCA-FGPI (0.2370 sn), ASO-FGPI (0.24 sn), ABC-FGPI (0.2420 sn), DA-FGPI (0.2460 sn), GA-FGPI (0.2480 sn) ve FPA-FGPI (0.2540 sn) olarak sıralı olduğu görülür. Şekil 4.26'dan 2 ile 4 saniye arasındaki negatif bölgede yerleşme zamanı değerleri ise küçükten büyüğe doğru GWO-FGPI (2.2250 sn), PSO-FGPI (2.2270 sn), BA-FGPI (2.2340 sn), WCA-FGPI (2.2370 sn), ASO-FGPI (2.2400 sn), ABC-FGPI (2.2430 sn), DA-FGPI (2.2460 sn) GA-FGPI (2.2480 sn) ve FPA-FGPI (2.2550 sn) olarak sıralanmıştır.

Metasezgisel optimizasyon tabanlı FGPI kontrolörlere sahip sistem çıkışlarının birbirleri ile performans kriterlerinden yüzde aşım açısından karşılaştırılması için Şekil 4.27 (pozitif bölge) ve Şekil 4.28'deki (negatif bölge) sistem cevap eğrileri kullanılmıştır. Şekil 4.27'den (0-2) saniye arasındaki pozitif bölgede yüzde aşım değerlerinin küçükten büyüğe doğru GA-FGPI (4.76%), FPA-FGPI (4.85%), DA-FGPI (4.9%), WCA-FGPI (5.05%), ABC-FGPI (5.06%), ASO-FGPI (5.11%), BA-FGPI (5.14%), GWO-FGPI (5.34%) ve PSO-FGPI (5.62%) olarak sıralı olduğu görülür. Şekil 4.28'den (2-4) saniye arasındaki negatif bölgede yüzde aşım değerlerinin küçükten büyüğe doğru GA-FGPI (4.8%), FPA-FGPI (4.9%), DA-FGPI (4.98%), WCA-FGPI (5.1%), ABC-FGPI (5.1%), ASO-FGPI (5.2%), BA-FGPI (5.2%), GWO-FGPI (5.3%) ve PSO-FGPI (5.7%) olarak sıralı olduğu görülür.

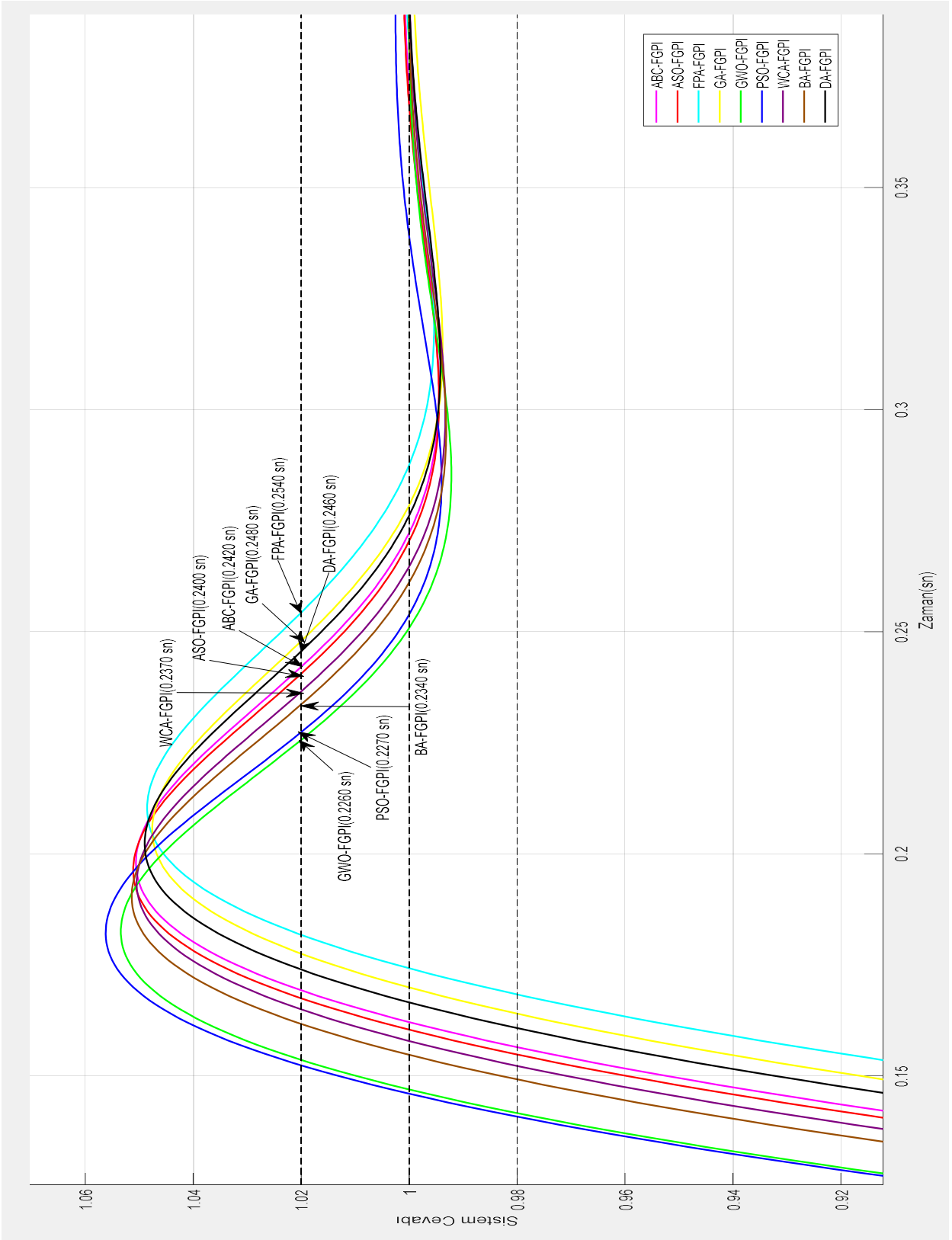
Ziegler-Nichols kontrol yöntemi tabanlı FGPI kontrolörün uygulandığı sisteme ilişkin yerleşme zamanı (%2), yükselme zamanı (%10-%90) ve aşım değerleri, pozitif bölge için Şekil 4.29'da gösterilmişken; negatif bölgeye ilişkin olanlar ise Şekil 4.30'da gösterilmiştir.



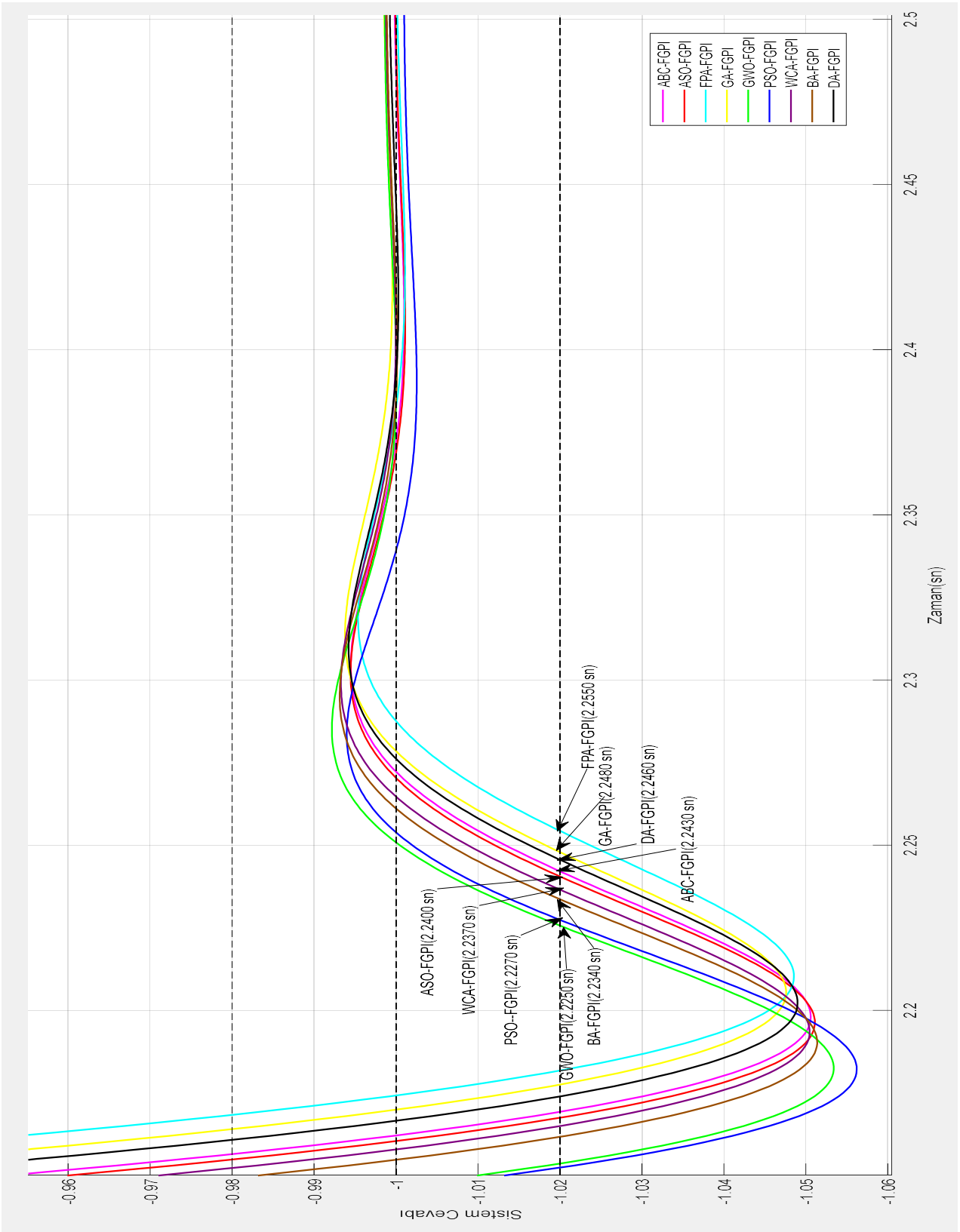
Şekil 4.23: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yükselme zamanları (%10-%90 kriterine göre)



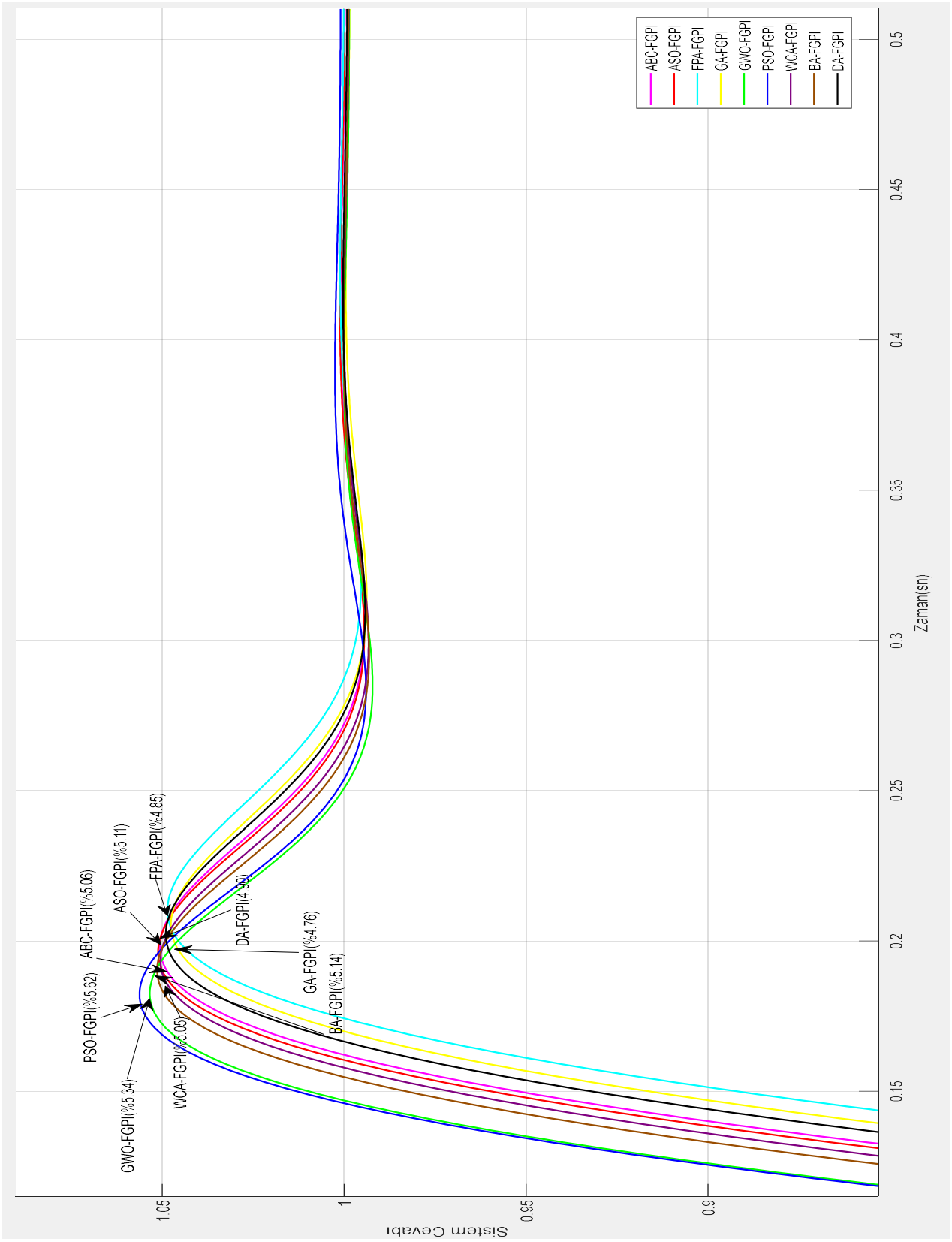
Şekil 4.24: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin negatif bölgedeki yükselme zamanları (%10-%90 kriterine göre)



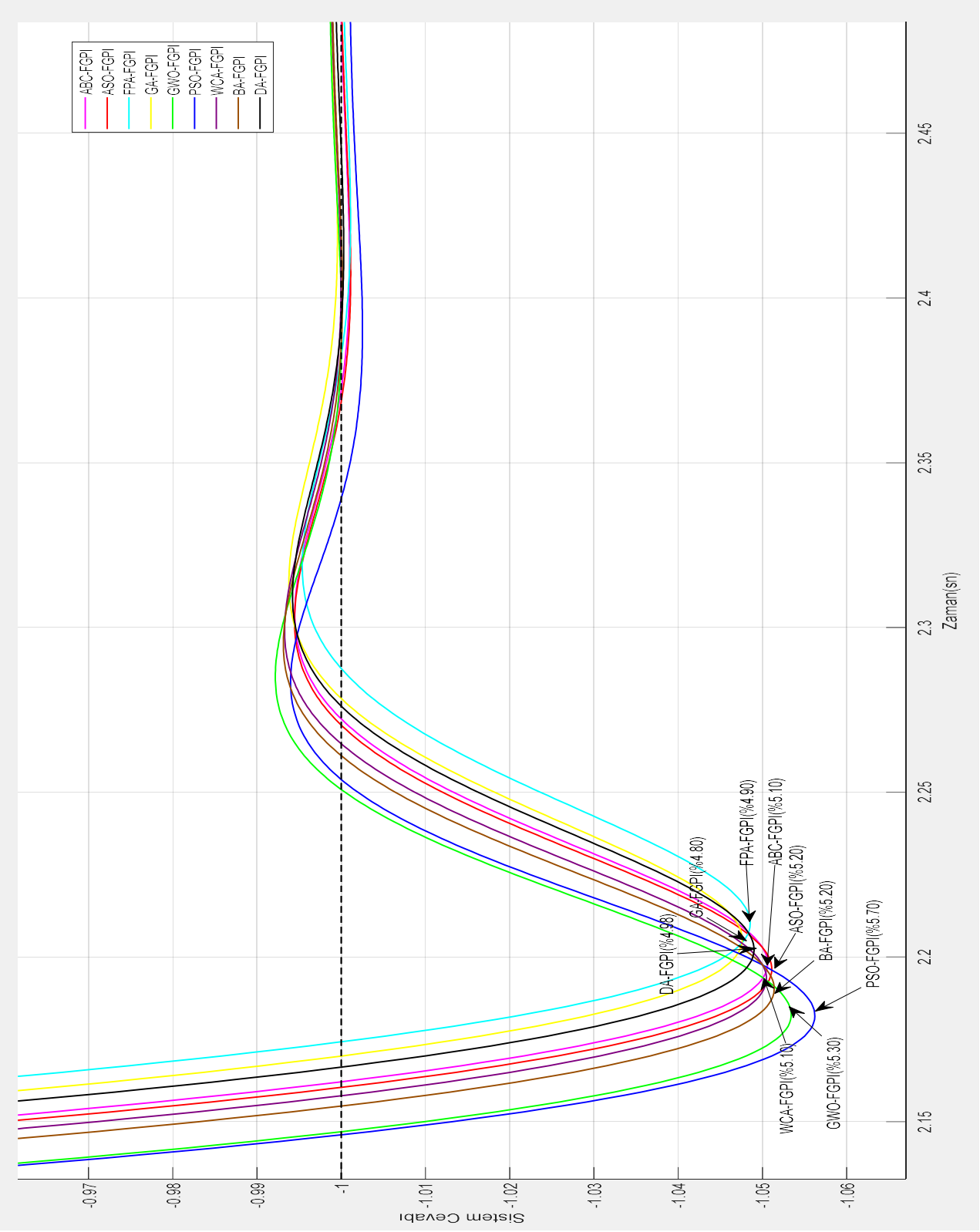
Şekil 4.25: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin pozitif bölgedeki yerleşme zamanları (%2 bandına göre)



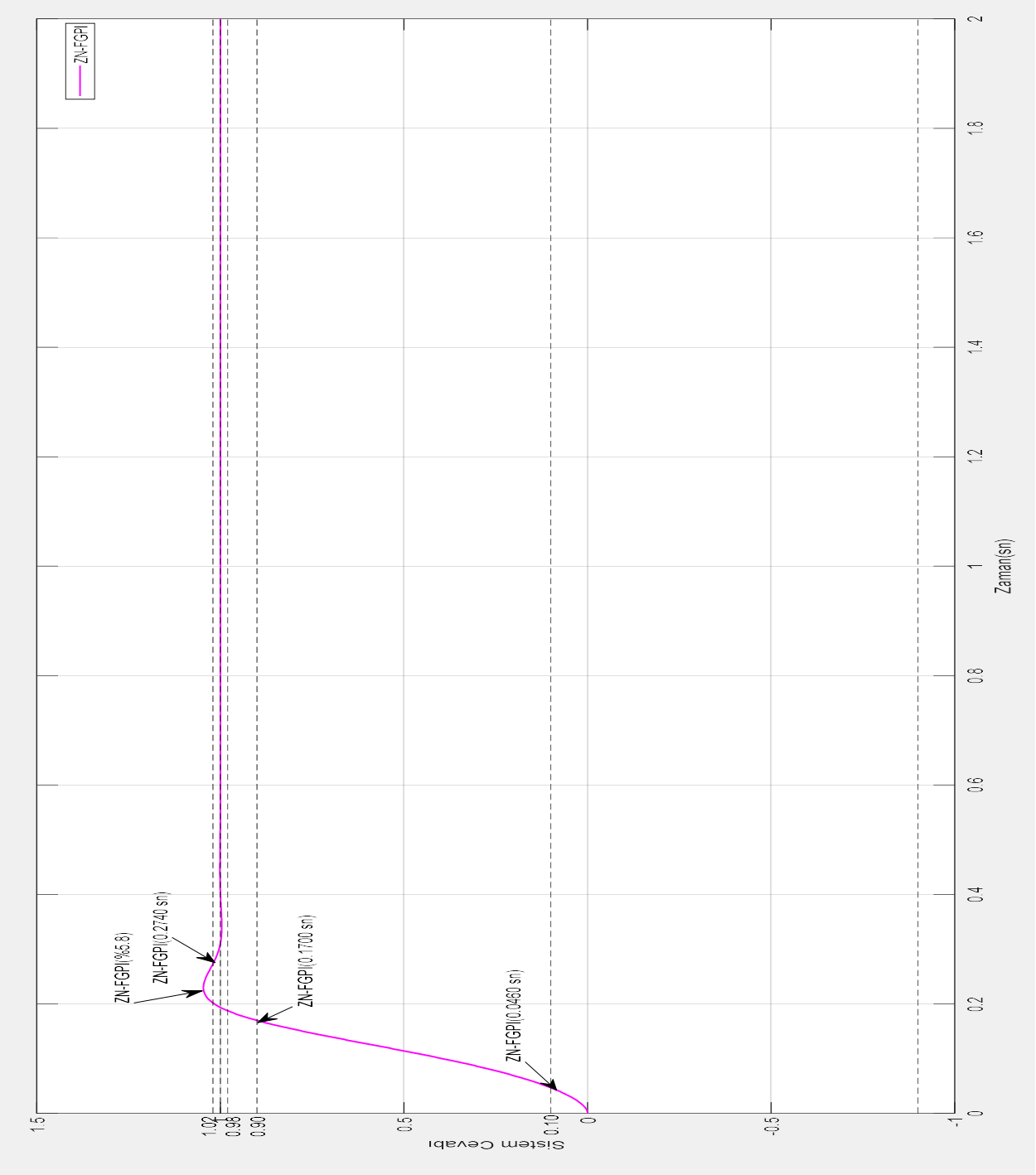
Şekil 4.26: Metasezgisel algoritmalarla optimize edilmiş FGPI kontrolöre sahip sisteme ilişkin negatif bölgedeki yerleşme zamanları (%2 bandına göre)



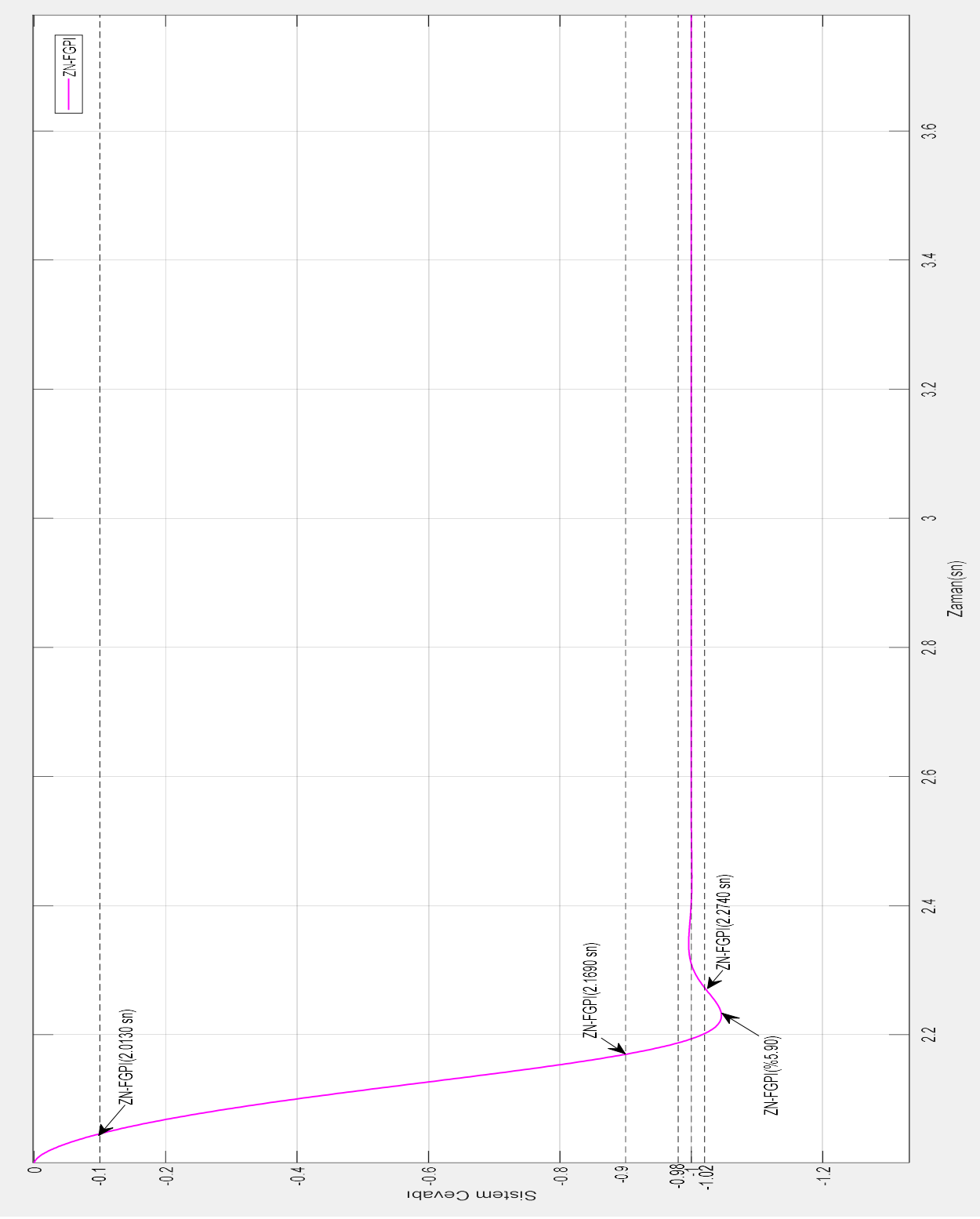
Şekil 4.27: Metasezgisel algoritmaların optimize edilmiş FGPI kontrolöre ilişki pozitif bölgedeki aşım değerleri



Şekil 4.28: Metasezgisel algoritmaların optimize edilmiş FGPI kontrolöre ilişkin negatif bölgedeki aşım değerleri



Şekil 4.29: Ziegler-Nichols yöntemi ile optimize edilmiş FGPI kontrolörün uygulandığı sisteme ilişkin pozitif bölgedeki yerleşme zamanı (%2), yükselme zamanı (%10-%90) ve yüzde aşım değerleri



Şekil 4.30: Ziegler-Nichols yöntemi ile optimize edilmiş FGPI kontrolörün uygulandığı sisteme ilişkin negatif bölgedeki yerleşme zamanı (%2), yükselme zamanı (%10-%90) ve yüzde aşım değerleri

Fırçasız DC motor modelimizin kontrolünde FGPI kontrolör parametrelerini ayarlamak için klasik Ziegler-Nichols yöntemi kullanılması sonucu bulunan yükselme zamanı, yerleşme zamanı ve yüzde aşım miktarları Şekil 4.29, Şekil 4.30, Tablo 4.8 ve Tablo 4.9'da gösterilmiştir. Buna göre pozitif bölge için yerleşme süresi 0.2740 saniye, yükselme zamanı (%10'dan %90'a ulaşma süresi) 0.1240 saniye ve yüzde aşım 5.8% olarak bulunmuştur. Negatif bölge için ise yerleşme süresi 0.2740 saniye, yükselme zamanı ise 0.1560 saniye ve aşım 5.9% olarak bulunmuştur.

Şekil 4.23 ile Şekil 4.30 arasındaki tüm sistem cevap eğrilerine ilişkin şekillerde hem pozitif hem negatif bölgelerde görülen yükselme zamanı, yerleşme zamanı ve yüzde aşım değerleri, Tablo 4.8 ve Tablo 4.9'da topluca derlenerek tüm yöntemler için performans karşılaştırılması yapılmıştır.

Tablo 4.8: Metasezgisel algoritmaları ve Ziegler Nichols kontrol yöntemi ile optimize edilmiş FGPI kontrolöre ilişkin pozitif bölge değerleri (0-2 sn)

Optimizasyon Yöntemleri	Aşım (%)	tr (yükselme süresi) %10-%90	ty (yerleşme süresi) %2
DA-FGPI	4.90	0.1090 sn (0.0350-0.1440)	0.2460 sn
WCA-FGPI	5.05	0.1030 sn (0.0330-0.1360)	0.2370 sn
GWO-FGPI	5.34	0.0970 sn (0.0290-0.1260)	0.2260 sn
GA-FGPI	4.76	0.1110 sn (0.0360-0.1470)	0.2480 sn
ASO-FGPI	5.11	0.1060 sn (0.0330-0.1390)	0.2400 sn
PSO-FGPI	5.62	0.0970 sn (0.0290-0.1260)	0.2270 sn
ABC-FGPI	5.06	0.1060 sn (0.0340-0.1400)	0.2420 sn
BA-FGPI	5.14	0.1020 sn (0.0310-0.1330)	0.2340 sn
FPA-FGPI	4.85	0.1130 sn (0.0390-0.1520)	0.2540 sn
ZN-FGPI	5.80	0.1240 sn (0.0460-0.1700)	0.2740 sn

Tablo 4.9: Metasezgisel algoritmaları ve Ziegler Nichols kontrol yöntemi ile optimize edilmiş FGPI kontrolöre ilişkin negatif bölge değerleri (2-4 sn)

Optimizasyon Yöntemleri	Aşım (%)	tr (yükselme süresi) %10-%90	ty (yerleşme süresi) %2
DA-FGPI	4.98	0.1100 sn (2.0330-2.1430)	2.2460 sn
WCA-FGPI	5.10	0.1040 sn (2.0320-2.1360)	2.2370 sn
GWO-FGPI	5.30	0.0970 sn (2.0290-2.1260)	2.2250 sn
GA-FGPI	4.80	0.1100 sn (2.0370-2.1470)	2.2480 sn
ASO-FGPI	5.20	0.1060 sn (2.0330-2.1390)	2.2400 sn
PSO-FGPI	5.70	0.0990 sn (2.0270-2.1260)	2.2270 sn
ABC-FGPI	5.10	0.0970 sn (2.0320-2.1400)	2.2430 sn
BA-FGPI	5.20	0.1010 sn (2.0320-2.1330)	2.2340 sn
FPA-FGPI	4.90	0.1130 sn (2.0380-2.1510)	2.2550 sn
ZN-FGPI	5.90	0.1560 sn (2.0130-2.1690)	2.2740 sn

Tablo 4.8 ve Tablo 4.9'dan FGPI kontrolöre sahip fırçasız dc motor sistemimiz için metasezgisel algoritmalar ile Ziegler-Nichols yöntemi performans kriterleri açısından kıyaslandığında yerleşme süreleri, yükselme zamanları ve yüzdesel aşımlar için genelde metasezgisel algoritmaların hem negatif hem de pozitif bölgede sistemin daha iyi performans sergilediği görülmüştür. Dokuz adet metasezgisel algoritmaların kendi aralarında performans karşılaştırılması yapıldığında ise hem pozitif hem de negatif bölge için %2 bandına göre yerleşme sürelerinde GWO-FGPI ile PSO-FGPI'nın en iyi sonucu verdiği görülür. Pozitif bölgede %10'ından %90'ına ulaşma süreleri açısından yükselme süreleri karşılaştırıldıklarında GWO-FGPI ile PSO-FGPI'nın en iyi sonucu verdiği görülürken; yüzde aşım değerleri karşılaştırmasında ise GA-FGPI'nın en iyi sonucu verdiği görülür. Negatif bölge için kıyaslandıklarında ise %10'ından %90'ına ulaşma süreleri açısından yükselme sürelerinde GWO-FGPI, ABC-FGPI ile PSO-FGPI'nın en iyi sonucu verdiği görülürken; yüzde aşım değerleri karşılaştırılmasında ise GA-FGPI'nın en iyi sonuçları verdiği görülür.

5. TARTIŞMA

Elektrik motorları günümüzde endüstriyel uygulamaları, yenilebilir enerji alanları, ulaşım, otomasyon, kontrol uygulamaları gibi birçok alanda yaygın olarak kullanılmaktadır [1-12]. Elektrik motorlarının kontrol edilmesi için çeşitli yöntemler bulunmaktadır. Bu yöntemlerle motorun hız, konum ve torku kontrol edilebilmektedir [4,5]. Bu motorların hız, konum ve tork kontrolü için PI, PID ve bulanık mantık tabanlı kontrolörler sıklıkla kullanılmasına rağmen; tam istenilen performansı sağlayamamaları gibi durumlarda klasik yöntemlerin [13, 14] yanı sıra başta metasezgisel optimizasyon algoritmaları gibi pek çok yapay zeka yöntemleri daha hassas kontrolörler tasarlamak için literatürde yaygın olarak kullanılmaktadır. [15-18]. Günümüzde parametre değerleri ayarlanarak optimize edilmiş PI, PID ve FGPI gibi farklı yapıdaki pek çok kontrolörlerin uygulandığı değişik sistem modellerinin performanslarını karşılaştırmak için yüzde aşım, yerleşme zamanı ve yükselme zamanı kriterlerini inceleyen literatürde birçok bilimsel çalışmalar mevcuttur [16,17].

Yılmaz [15] çalışmasında, PID ve bulanık mantık kontrolör parametrelerini, GA ve PSO kullanılarak optimize etmiştir. GA'nın diğer yöntemlere kıyasla performans açısından daha iyi sonuçlar verdiğini görülmüştür.

Gökçe ve diğ. [18] çalışmasında, DC motor hız kontrolü için ZN ve PSO algoritmalarını karşılaştırmıştır. Farklı referans ve yük tipleri kullanılarak yapılan çalışmada, PSO yönteminin ZN'ye göre daha iyi performans gösterdiğini belirtmişlerdir. Bu tez çalışmasında da PSO ile optimize edilmiş PID kontrolörün DC motor modelimize uygulandığında, hesaplanan yerleşme sürelerinin Ziegler-Nichols yöntemininkinden daha küçük olduğu simülasyon sonuçlarından görülmüştür.

Bu çalışmada ele alınan algoritmalardan biri olan genetik algoritma (GA), Holland [19] tarafından doğal seçim ve genetik prensiplere dayalı olarak geliştirilmiştir. Parçacık sürü optimizasyonu (PSO) algoritması Kennedy ve Eberhart [20] tarafından sürü davranışlarından esinlenerek oluşturulmuştur. Yapay arı kolonisi (ABC) algoritması Karaboğa [21] tarafından arıların yiyecek arama davranışlarına dayalı olarak geliştirilmiştir. Çiçek tozlaşma algoritması (FPA), Yang [22] tarafından çiçeklerin biyolojik tozlaşma süreçlerinden esinlenerek tasarlanmıştır. Gri kurt optimizasyonu (GWO), Mirjalili ve diğ. [23] tarafından gri kurtların avlanma stratejilerinden ilham alınarak geliştirilmiştir. Yarasa algoritması (BA), Yang [24] tarafından yarasaların yankı mekanizmasına dayalı olarak oluşturulan metasezgisel optimizasyon algoritmasıdır. Yusufçuk algoritması (DA), Mirjalili [25] tarafından

yusufçukların avlanma ve göç davranışlarından esinlenerek geliştirilmiştir. Atom arama optimizasyonu (ASO), Zhao ve diğ. [26] tarafından coulomb yasasına dayalı olarak oluşturulmuştur. Su Döngüsü Algoritması (WCA), Eskander ve arkadaşları [27] tarafından suyun döngüsel hareketlerinden ilham alınarak tasarlanmıştır.

Korkmaz ve diğ**.** [39] çalışmasında, dinamik katsayılara sahip PID kontrolör kullanılmıştır. Aynı zamanda genetik algoritma kullanılarak optimize edilen PID kontrolörlerin performanslarını karşılaştırmışlardır. PID kontrolörün katsayılarını belirlemek için sisteme giriş ve çıkışına bağlı bir hata fonksiyonu tanımlanmış ve bu yöntemle kontrol performansı iyileştirilmiştir.

Sezer [48] çalışmasında, gaz türbinli bir elektrik santrali modeline PSO, ABC, GWO, güve-alev optimizasyonu (MFO), balina optimizasyon algoritması (WOA), salp sürü algoritması (SSA), sinüs-kosinüs algoritması (SCA) ve ASO gibi metasezgisel algoritmalar ile parametreleri optimize edilmiş kesir dereceli PID kontrolörler uygulamış ve sistem cevaplarını zaman tanım bölgesi geçici rejim parametreleri cinsinden karşılaştırmıştır. Simülasyon sonuçlarında, GWO algoritmasının genelde daha iyi performans sergilediğini ifade etmiştir.

Vanchinathan ve diğ. [56] çalışmasında, sensörsüz fırçasız doğru akım motorunun rotor hızının kontrolü için kesirli mertebeden orantılı integral türev (FOPID) denetleyicisi kullanılmıştır. Kontrol denetleyici tasarımında yarasa algoritması (BA) tabanlı bir optimizasyon yöntemi kullanmıştır. DC motorun hız çıkış sinyalleri, yükselme zamanı, yüzde aşım miktarı ve yerleşme zamanı gibi kriterler dikkate alınarak değerlendirilmiştir. Simülasyon sonuçları, geçici ve kalıcı rejim zaman bölgesi özelliklerinin değerlendirilmesi amacıyla, yapay arı kolonisi (ABC) optimizasyon yöntemi ve değiştirilmiş genetik algoritma (MGA) ile karşılaştırılmıştır. Önerilen BA tabanlı FOPID denetleyicinin, geçici durum özelliklerinin iyileştirilmesi ve kalıcı durum hatasının azaltılmasında etkili olduğu gözlemlenmiştir. Bu tez çalışmasında ise [56]'da kullanılan BA algoritmasıyla birlikle dokuz adet farklı metasezgisel optimizasyon algoritması kullanılmış olup bu çalışmadan farklı olarak düşük dereceli PI, PID ve FGPI kontrolörler kullanarak DC fırçasız motorun hız kontrolü incelenmiştir. DC motorun çıkış sinyalleri, yükselme zamanı, yüzde aşım miktarı ve yerleşme zamanı gibi kriterler dikkate alınarak, karşılaştırılmış ve değerlendirilmiştir.

Premkumar ve diğ. [70] çalışmasında, fırçasız doğru akım sürücüsünün hız kontrolü için bulanık orantı türev denetleyici (FPD) ve bulanık orantı türev integral denetleyici (FPID) tasarımı yapılmıştır. Söz konusu denetleyicilerin optimizasyonu parçacık sürüsü optimizasyonu (PSO), guguk kuşu arama algoritması (CSA) ve yarasa algoritması (BA) gibi doğadan ilham

alan optimizasyon yöntemleri kullanılarak gerçekleştirilmiştir. Kontrol denetleyicileri, aşım ve yükselme süresi gibi kriterlerin yanı sıra kök ortalama karesel hata (RMSE), IAE, ITAE ve ISE gibi uygunluk fonksiyonları ile değerlendirilmiştir. Bu analizler, farklı ayar hızı ve yük bozulması gibi çalışma koşulları altında gerçekleştirilmiştir.

Metasezgisel algoritmalar üç farklı uygunluk fonksiyonuna göre seçilebilir. Literatürde bu uygunluk fonksiyonları IAE, ITAE ve ISE uygunluk fonksiyonlarıdır. Bu tez çalışmasında IAE uygunluk fonksiyonu dikkate alınarak modeli oluşturulan bir fırçasız DC motorun kontrolü ve performans analizi yapılmıştır. Bu tez çalışmasında klasik Ziegler-Nichols yöntemi ve DA, WCA, PSO, GA, GWO, ABC, ASO, BA ve FPA isimli dokuz adet metasezgisel algoritmalar ile ayarlanan PI, PID ve FGPI gibi düşük dereceli kontrolörler, modeli kurulan bir fırçasız doğru akım motor modeline uygulanarak, sistemin çıkışında elde edilen çıkış sinyalleri yükselme zamanı, yüzde aşım miktarı ve yerleşme zamanları gibi kriterlerle sistem performansı açısından değerlendirilmiştir. Ayrıca kullanılan dokuz adet metasezgisel optimizasyon yöntemlerinin amaç fonksiyonlarının sıfıra yakınsama eğrileri ve çalışma hızları da birbirleriyle karşılaştırılmıştır. Böylece bu tezde bir fırçasız dc motorun performansını iyileştirmek için edilen tüm sonuçlar değerlendirilmiş ve en uygun yöntemlerin belirlenmesi hedeflenmiştir.

6. SONUÇ ve ÖNERİLER

Fırçasız DC motorlar, yalnızca verimlilikleri ve tork özellikleri nedeniyle değil, aynı zamanda doğru akım (DC) ile beslenmenin avantajlarına sahip olmaları ve fırça kullanmanın dezavantajlarını ortadan kaldırmaları nedeniyle de endüstriyel uygulamalarda giderek daha fazla kullanılmaktadır. Fırçasız DC motorlar geniş bir hız aralığına sahiptir, bu nedenle hız kontrolü önemli bir konudur. Başlangıç akımı, başlangıç torku, yerleşme süresi, yükselme süresi vb. gibi bir hız performansı incelenirken odaklanılması gereken birçok parametre vardır. Motor hızını kontrol etmek için PID, PI ve bulanık mantık PI (FGPI) gibi kontrolörler önem kazanır ve böylece bu tip kontrolörlerin parametrelerini optimize etmek de önemli bir husus olarak karşımıza çıkar. Bu amaçla bu tezde, öncelikle parametre değerleri literatürden seçilen bir fırçasız doğru akım motorunun MATLAB Simulink ortamında sistem tanımlama (system identification toolbox) kullanılarak sistem modeli oluşturulmuştur. Daha sonra birim basamak referans giriş altında fırçasız DC motor modelimizin PID, PI ve FGPI kontrolünde yusufçuk algoritması (DA), su döngüsü algoritması (WCA), parçacık sürü optimizasyonu (PSO), genetik algoritma (GA), gri kurt optimizasyonu (GWO), yapay arı kolonisi (ABC), atom arama optimizasyonu (ASO), yarasa algoritması (BA) ve çiçek tozlaşma algoritması (FPA) olmak üzere 9 adet metasezgisel algoritmanın ve klasik Ziegler-Nichols yönteminin performans kriterlerini birbirleriyle karşılaştırabilmek için 0-2 saniye ve 2-4 saniye arasında elde edilen sistem cevap eğrileri hem pozitif hem de negatif bölgede yerleşme süreleri, yükselme süreleri ve yüzde aşım kriterleri açısından değerlendirilmiştir.

Birim basamak giriş altında PID kontrolör uygulanmış fırçasız DC motor sistemimiz için metasezgisel algoritmalar ile Ziegler-Nichols yöntemi performans kriterleri açısından kıyaslandığında sistem cevaplarının pozitif bölge için %2 bandına göre yerleşme sürelerinde 0.1338 sn ile GWO-PID ve 0.1336 sn ile PSO-PID' nin en iyi sonucu verdiği görülmüştür. Negatif bölge için ise %2 bandına göre yerleşme sürelerinde 2.1589 sn ile GWO-PID ve 2.1588 sn ile PSO-PID'nin en iyi sonucu verdiği görülmüştür. %10'ından %90'ına ulaşma süreleri açısından yükselme süreleri karşılaştırıldığında pozitif bölge için en iyi sonucu 0.0023 sn ile GA-PID ve 0.0032 sn ile WCA-PID' nin verdiği görülmüştür. Negatif bölge için ise %10'ından %90'ına ulaşma süreleri açısından yükselme sürelerinde 0.0025 sn ile GA-PID ve 0.0029 sn ile WCA-PID' nin verdiği görülmüştür. Yüzde aşım değerleri açısından karşılaştırıldıklarında ise GWO-PID, GA-PID, ASO-PID ve BA-PID için motor sistemimizin hiç aşım yapmadığı görülmüştür. Tüm bu sonuçlar topluca değerlendirildiğinde optimize edilmiş PID kontrolör

uygulanmış fırçasız DC motor sistemimiz için GWO-PID' nin yerleşme süresi ve yüzde aşım performans kriterleri ile öne çıktığı görülür. Metasezgisel algoritmaların yakınsama eğrileri incelendiğinde 0.0164 ile optimizasyon sürecinde en düşük değeri alan GWO-PID algoritmasıdır FPA-PID'nin ise algoritmanın çalışma süresi olarak en kısa optimizasyon süresine sahip olduğu görülmüştür.

PI kontrolör uygulanmış fırçasız DC motor sistemimiz için metasezgisel algoritmalar ile Ziegler-Nichols yöntemi performans kriterleri açısından kıyaslandığında sistem cevaplarının pozitif bölge için %2 bandına göre yerleşme sürelerinde 0.2703 sn ile WCA-PI ve 0.2830 sn ile ASO-PI'nın en iyi sonucu verdiği görülmüştür. Negatif bölge için ise %2 bandına göre yerleşme sürelerinde 2.3010 sn ile ABC-PI ve 2.3069 sn ile PSO-PI'nın en iyi sonucu verdiği görülmüştür. %10'ından %90'ına ulaşma süreleri açısından yükselme süreleri karşılaştırıldığında pozitif bölge için en iyi sonucu 0.0266 sn ile DA-PI ve 0.0268 sn ile FPA-PI'nin verdiği görülmüştür. Negatif bölge için ise %10'ından %90'ına ulaşma süreleri açısından yükselme sürelerinde 0.0135 sn ile DA-PI ve 0.0138 sn ile FPA-PI' nin verdiği görülmüştür. Yüzde aşım değerleri açısından karşılaştırıldıklarında ise hem pozitif hem negatif bölge için en az aşımı GWO-PI'nın yaptığı görülmüştür. Sonuçlar değerlendirildiğinde optimize edilmiş PI kontrolör uygulanmış fırçasız DC motor sistemimiz için GWO-PI' nın yüzde aşım performans kriteri ile ve DA-PI'nın da yükselme süresi açısından öne çıktığı görülür. Metasezgisel algoritmaların yakınsama eğrileri incelendiğinde optimizasyon sürecinde en düşük değeri olan 0.0404 ile PSO ve ona çok yakın 0.04043 ile GWO algoritmalarıdır. FPA-PI'nın ise algoritmanın çalışma süresi olarak en kısa optimizasyon süresine sahip olduğu görülmüştür.

Bulanık mantık ve optimize edilmiş PI kontrolör hibrit bir yapıda birleştirilerek elde edilen bulanık mantık (FGPI) denetleyicilerin uygulandığı fırçasız DC motor sistemimiz için metasezgisel algoritmalar ile Ziegler-Nichols yöntemi performans kriterleri açısından kıyaslandığında sistem cevaplarının pozitif bölge için %2 bandına göre yerleşme sürelerinde 0.2260 sn ile GWO-FGPI ve 0.2270 sn ile PSO-FGPI' nın en iyi sonucu verdiği görülmüştür. Negatif bölge için ise %2 bandına göre yerleşme sürelerinde ise 2.2250 sn ile GWO-FGPI ve 2.2270 sn ile PSO-FGPI' nın en iyi sonucu verdiği görülmüştür. %10'ından %90'ına ulaşma süreleri açısından yükselme süreleri karşılaştırıldığında pozitif bölge için en iyi sonucu 0.0970 saniyelik aynı değerle GWO-FGPI ve PSO-FGPI' nın verdiği görülmüştür. Negatif bölge için ise %10'ından %90'ına ulaşma süreleri açısından yükselme sürelerinde ise aynı değer olan 0.0970 sn ile GWO-FGPI ve PSO-FGPI ve bu değere çok yakın olan 0.0990 sn ile PSO-FGPI' nın verdiği görülmüştür. Yüzde aşım değerleri açısından karşılaştırıldıklarında ise en az aşımı

pozitif bölge için 4.76% ile GA-FGPI'nın ve negatif bölge için de 4.8% ile GA-FGPI'nın yaptığı görülmüştür.

Bu çalışmada bahsi geçen PI ve FGPI denetleyicilerin fırçasız DC motor modelimize uygulanması sonucunda sisteminin cevap eğrilerinin pozitif ve negatif bölgelerinde performans kriterleri karşılaştırıldığında; FGPI denetleyicilerin PI kontrolörlere göre genelde yüzde aşımları azaltırken, yükselme sürelerini ise arttırdığı gözlemlenmiştir.

GWO ve PSO ile optimize edilmiş PID ve FGPI kontrolörler fırçasız DC motor modelimize uygulandığında, hesaplanan yerleşme sürelerinin diğer metasezgisel algoritmalarınkinden ve Ziegler-Nichols yöntemininkinden daha küçük olduğu simülasyon sonuçlarından görülmüştür. Fırçasız DC motorlarda hız kontrolü önemli bir konu olduğundan, en küçük yerleşme sürelerini veren GWO ve PSO algoritmaları bu açıdan önem arz etmektedir. Metasezgisel algoritmalarla optimize edilmiş PID, PI ve FGPI için aşım yüzdeleri dikkate alındığında fırçasız DC motor modelimizde PID kontrolör kullanmanın diğer iki kontrolöre göre daha az yüzdesel aşımlar ve genelde daha küçük yükselme süreleri elde etmeyi sağladığı görülmüştür.

Metasezgisel algoritmaların hem PID hem de PI kontrolörler için yakınsama eğrileri karşılaştırıldığında genelde GWO algoritmasının daha düşük hata değeri olduğu ve çalışma süreleri kıyaslandığında ise FPA algoritmasının optimizasyon sonuçlarına daha hızlı ulaştığı görülmüştür.

Geleceğe yönelik çalışmalar olarak farklı motor modellerine daha farklı yapılardaki kontrolörler uygulayarak makine öğrenmesi yöntemleri ve bazı hibrit yapay zekâ yöntemleri kullanılarak çeşitli performans analizleri yapılması düşünülmektedir.

KAYNAKLAR

[1] Wildi, T., 2014*, Electrical machines, drives and power systems*, 6th ed., Pearson education Inc, London, ISBN:978-1-292-02458-5

[2] Hughes, A. and Drury, B., 2019, *Electric motors and drives: fundamentals, types and applications,* 5th ed., Elsevier, ISBN:978-0-08-102615-1

[3] Chapman, S.J., 2011*, Electric machinery fundamentals*, 5th ed., McGraw-hill education, New York, ISBN 978-0-07-352954-7

[4] Sen, P.C., 2013, *Principles of electric machines and power electronics*, 3rd ed., John Wiley and Sons, Inc., Hoboken, New Jersey, ISBN: 978-1-118-07887-7

[5] Batan, E., 2019, *Matlab simulink ortamında kullanılabilen arduino temelli kontrol deney seti tasarımı*, Yüksek Lisans Tezi, Tarsus Üniversitesi Lisansüstü Eğitim Enstitüsü

[6] Küçük, S., 2022, Elektriğin tarihi, *Sosyal bilimler dergisi*, 56 (1), 38-55

[7] Falconer, I., 2022, Electromagnetism in the encyclopaedias*, Newsletter of the Institute of physics history of physics group*, 39, 20-36

[8] Ciba, S., 2019, *Fırçasız DC motorlarının performans analizi ve denetimi*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü

[9] Büyük, S., 2018, *Elektrik motorlarında enerji verimliliği mevzuatının Türkiye pazarına etkisinin analizi*, Yüksek Lisans, İstanbul Teknik Üniversitesi

[10] Çetin, E., 2021, Endüstriyel dönüşüm süreci ve stratejik insan kaynakları dönüşümünün ilişkisi, *Journal of social, humanities and administrative sciences*, 7 (42), 1291-1302

[11] Mindell, D.A., 2002, *Between human and machine: feedback, control, and computing before cybernetics*, 1th ed, Johns Hopkins University Press, Baltimore, ISBN:978-0801868955

[12] Bennett, S., 1984, Nicolas Minorsky and the automatics steering of ships, *IEEE control systems magazine*, 4 (4), 10-15

[13] Ziegler, J.G. Nichols, N.B., 1942, Optimum settings for automatic controllers, *Transactions of the ASME*, 64 (8), 759–765.

[14] Develi, H., 2004, *Ziegler Nichols yöntemi ve migo yaklaşımı*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü

[15] Yılmaz, G., 2023, Comparison of different methods for optimization of pıd controller gain coefficients, *Kırklareli university journal of engineering and science*, 9 (2), 254-264

[16] Ogata, K., 2010, *Modern control engineering*, Pearson, 567-641

[17] Aksözek, M., İn, S., Arıkuşu, Y.S., Bayhan, N., Tiryaki, H., 2023, Yapay zekâ yöntemleri ile bir kömür yakıtlı santralde kontrol uygulaması, *Otomatik kontrol ulusal toplantısı bildirileri*, 14-16 Eylül 2023, İstanbul, Türkiye, 724-730

[18] Gökçe, O.C., Durusu, V., Ünal, R., 2022, Farklı yük çeşitleri için parçacık sürü optimizasyonu ve Ziegler Nichols metodunun DC motor hız kontrolü probleminde karşılaştırılması, *Avrupa bilim ve teknoloji dergisi,* 33, 88-92

[19] Holland, J.H., 1975, *Adaptation in natural and artificial systems*, University of Michigan press, Ann Arbor, MI

[20] Kennedy, J. and Eberhart, R., 1995, Particle swarm optimization, *Proceedings of ıeee ınternational conference on neural networks*, 4, 1942-1948

[21] Karaboğa, D., 2005, An Idea Based on Honey Bee Swarm for Numerical Optimization, *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 200, 1-10

[22] Yang, X.S., 2012, Flower pollination algorithm for global optimization, unconventional computation and natural computation, *Lecture notes in computer science*, 7445, 240-249

[23] Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014, Grey wolf optimizer, *Advances in engineering software*, 69, 46-61.

[24] Yang, X.S., 2010, *A new metaheuristic bat-ınspired algorithm*, Nature ınspired cooperative strategies for optimization (NICSO 2010)*,* Springer, Berlin, Heidelberg, 65-74

[25] Mirjalili, S., 2015, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems*, Neural computing and applications*, 27, 1053-1073

[26] Zhao, W., Wang, L., Zhang, Z., 2019, A Novel atom search optimization for dispersion coefficient estimation in groundwater, *Future generation computer systems*, 91, 601-610

[27] Eskander, H., Sadollah, A., Bahreininejad, A., Shukor, A.H.A., 2012, Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers and structures*, 151–166

[28] Koçaş, G., 2023*, Elektrikli araçlar için bldc motor sürücü tasarımı*, Yüksek Lisans Tezi, Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü

[29] Bayraktar, H.C., Balık, H.H., 2015, The Control of brushless dc motor, *International journal of electronics, mechanical and mechatronics engineering*, 5 (2), 919-932

[30] Anshory, I., Robandi, I., Wirawan, W., 2018, Parameters ıdentification bldc motor: ınstrumentations and transfer functions, *The 3rd annual applied science and engineering conference,* 12 September 2018, 1-4

[31] SudhanshuMitra, SaidaNayak, R., Prakash, R., 2015, Modeling and simulation of bldc motor using matlab/simulink environment, *International research journal of engineering and technology,* 2(8), 199-203

[32] Manda, P., Veeramalla, S.K., 2021, Brushless dc motor modeling and simulation in the matlab/sımulınk software environment, *Advances in modelling and analysis b*, 64(1-4), 27-33

[33] Koç, H., Berzener, B.S., Yaren, T., Kizir, S., Kizir, S., 2022, Gerçek zamanlı dc motor konum-hız ve konum-hız-tork kaskad PID denetleyici performanslarının karşılaştırılması, *Mühendislik bilimleri dergisi*, 15 (2), 123-134

[34] Kaplan, K., Kuncan, M., Polat, H., Tepe, B., Ertunç, H.M., 2020, PID ve bulanık mantık tabanlı dc motorun gerçek zamanlı konum kontrolü, *Iğdır Üniversitesi fen bilimleri enstitüsü dergisi*, 10 (2), 900-916

[35] The MathWorks, Inc., 2024, MATLAB R2024a(Lisans numarası: 41084692), https://www.mathworks.com

[36] Zeb, K., Rahman, M.A., Yang, Y., Xu, D., Lei, Q., 2019, Design of fuzzy-PI and fuzzy-sliding mode controllers for single-phase two-stages grid-connected transformerless photovoltaic ınverter, *Electronics*, 8 (5), 1-19

[37] Munciño, D.M., Damian Ramírez, E.A., Cruz-Fernández, M., Montoya-Santiyanes, L.A., Rodríguez-Reséndiz, J., 2024, Metaheuristic and heuristic algorithms-based ıdentification parameters of a direct current motor, *Algorithms*, 17 (5), 209, 1-22

[38] Rodriguez-Maya, N.E., Flores, J.J., Graff, M., 2024, Models to classify the difficulty of genetic algorithms to solve continuous optimization problems, *Natural computing*, 23, 431–451

[39] M. Korkmaz, Ö. Aydoğdu and H. Doğan, 2012, Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller, *International symposium on ınnovations in ıntelligent systems and applications*, 02 Jul 2012, Trabzon, Turkey, 1-5

[40] Bayhan, N., Arıkuşu, Y.S., Tiryaki, H., 2023, PI control based on parameter space approach supported metaheuristic optimization algorithms and anfıs in a natural gas combined cycle power plant, *Optimal control applications and methods*, 44 (2), 846-865

[41] Yusuf, A., Bayhan, N., Ibrahim, A.A., Tiryaki, H., Ballıkaya, S., 2021, Geometric optimization of thermoelectric generator using genetic algorithm considering contact resistance and thomson effect, *International journal of energy research*, 45 (6), 9382–9395

[42] Katoch, S., Chauhan, S.S., Kumar, V., 2021, A Review on genetic algorithm: past, present, and future, *Multimedia tools and applications*, 80 (5), 8091–8126

[43] Arıkusu, Y., Bayhan, N., Tiryaki, H., 2019, Comparison of conventional and modern controllers for a gas turbine power plant, *Elektrica*, 19 (2), 146-157

[44] Gad, A.G., 2021, Particle swarm optimization algorithm and ıts applications: a systematic review, *Archives of computational methods in engineering*, 28 (3), 1421–1448

[45] Uçmaz, B.N., Yakut, Y.B., 2024, PEM yakıt pillerinde pso tabanlı pı ve fopı kontrollü dc/dc dönüştürücülerine ilişkin performansların karşılaştırılması, *Dicle Üniversitesi fen bilimleri mühendislik dergisi*, 15 (1), 23-29

[46] Yönem, E., Akay, R., 2020, Yapay arı koloni algoritması ile eğitilmiş tekrarlayıcı sinir ağlarının robot navigasyonu için kullanılması, *Avrupa bilim ve teknoloji dergisi*, 318-324

[47] Kaya, B., Eke, İ., 2020, Yapay arı kolonisi algoritması ile yapılan geliştirmeler ve sonuçları, *Verimlilik dergisi*, 1, 100-115

[48] Sezer, K.S., 2021, *Metasezgisel algoritmalarla optimize edilmiş kesir dereceli PID kontrolör ile gaz türbin elektrik santralinin kontrolü*, Yüksek Lisans Tezi, İstanbul Üniversitesi-Cerrahpaşa, Lisansüstü Eğitim Enstitüsü

[49] Sezer, K.S., Bayhan, N., Tiryaki, H., 2020, Fractional order pıd controller design with metaheuristic algorithms for power plant, *4th Engineers of future ınternational student symposium*, 7-8 May, Zonguldak, Türkiye, 71-75

[50] Yavuz, S., 2019, *Çiçek tozlaşması algoritması ile devre süresi modellerinin geliştirilmesi*, Yüksek Lisans Tezi, Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü

[51] Fidan, Ş., Zaloğlu, M., Erkan, E., 2023, Meta-Heuristik optimizasyon algoritmalarının sistem tanımlama problemine uygulanması ve performans karşılaştırması, *2nd International conference on engineering, natural and social sciences*, 4-6 April 2023, Konya, Turkey, All sciences proceedings, 516-526

[52] Doğan, C., 2019, *Balina optimizasyon algoritması ve gri kurt optimizasyonu algoritmaları kullanılarak yeni hibrit optimizasyon algoritmalarının geliştirilmesi*, Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü

[53] Yusuf, A., Koç, T., Arıkusu, Y.S., Tiryaki, H., Bayhan, N., Ballıkaya, S., 2022, Modeling of a thermoelectric cooler system, design and optimization of the system's controller, *Sādhanā - Academy Proceedings in Engineering Sciences,* 47(3), 1-5

[54] Ekinci, S., 2015, Power system stabilizer design for multi-machine power system using bat search algorithm, *Sigma journal of engineering and natural sciences*, 33, 627-637

[55] Lakshmi, P., Venkateswara Rao, B., Devarapalli, R., Rai, P., 2020, Optimal power flow for a power system using bat algorithm to reduce transmission line losses with svc, *Proceedings of the ınternational conference on emerging frontiers in electrical and electronics technologies,* 10-11 July 2020, ISBN:978-1-7281-7549-2

[56] Vanchinathan, K., Valluvan, K. R., A, 2018, Metaheuristic optimization approach for tuning of fractional-order PID controller for speed control of sensorless bldc motor, *Journal of circuits systems and computers*, 27 (8), 1-19

[57] Doğruer, T., 2023, DC motorun hız kontrolü için kesir dereceli PID kontrolör tasarımı ve dayanıklılık analizi, *Adıyaman Üniversitesi mühendislik bilimleri dergisi*, 19, 15-28

[58] Ekinci, S., Demirören, A., Zeynepligil, H.L., Hekimoğlu, B., 2020, An opposition-based atom search optimization algorithm for automatic voltage regulator system, *Journal of the faculty of engineering and architecture of Gazi University*, 35 (3), 1141-1157

[59] Eker, E., Kayri, M., Ekinci, S., 2019, Kısıtlı optimizasyon problemlerinin çözümü için atom arama optimizasyon algoritması, *DÜMF mühendislik dergisi*, 10 (3), 841-851

[60] Kalyan, C.H., Goud, B.S., Bajaj, M., Kumar, M.K., Ahmed, E.M., Kamel, S., 2022, Water-cycle-algorithm-tuned ıntelligent fuzzy controller for stability of multi-area multi-fuel power system with time delays*, Mathematics*, 10 (3), 1-15

[61] Bakır, H., Merabet, A., Kiehbadroudinezhad, M., 2023, Optimized control of a hybrid water pumping system ıntegrated with solar photovoltaic and battery storage:towards sustainable and green water-power supply, *Energies*, 16 (13), 1-16

[62] Sadollah, A., Eskandar, H., Lee, H.M., Yoo, D.G., Kim, J.H., 2016, Water cycle algorithm: a detailed standard code*, SoftwareX*, 5, 37–43

[63] Sabzevari, D., Kargar, S., Zanjani, S.M., 2016, Mathematical modeling and designing pıd controller for a quadrotor and optimize ıts step response by genetic algorithm, *Majlesi journal of electrical engineering,* 10 (4), 17-24

[64] Meshram, P.M., Kanojiya, R.G., 2012, Tuning of PID controller using Ziegler-Nichols method for speed control of dc motor, *Proceedings of the IEEE International conference on advanced engineering*, 30-31 March 2012, ISBN: 978-81-909042-2-3, 117-122

[65] Kaçtı, V., 2021, *Henry gaz çözünürlük optimizasyonu ve farklı sistemlerdeki kontrolör tasarımına uygulanması*, Yüksek Lisans Tezi, Batman Üniversitesi, Fen Bilimleri Enstitüsü

[66] Zadeh, L.A., 1965, Fuzzy sets, *Information and control*, 8, 338-353

[67] Mamdani, E.H., 1974, Application of fuzzy algorithms for control of simple dynamic plant, *Proceedings of the ınstitution of electrical engineers*, 121 (12), 1585-1588

[68] Song, B., Xiao, Y., Xu, L.,2020, Design of fuzzy PI controller for brushless dc motor based on pso–gsa algorithm, *Systems science-control engineering: an open access journal*, 8(1), 67–77

[69] Mohamed, H.A., Khattab, H.A., Mobarka, A., Morsy, G.A., 2024, Hybrid fuzzy logic controller for boost converter operating in pv energy system, *Engineering research journal*, 40 (2), 73-78

[70] Premkumar K., Manikandan B., V., 2016, Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor, *Engineering science and technology, an ınternational journal,* 19, 818–840

İNTİHAL RAPORU İLK SAYFASI



Mert AKSÖZEK

ORİJİNALLİK RAPORU

%7	%5	%5	%2
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	acikbilim.yok.gov.tr İnternet Kaynağı	%2
2	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	<%1
3	hdl.handle.net İnternet Kaynağı	<%1
4	Armenta, Michelle. "The Lived Experience of Educators Who Teach Students Who Have Attention Deficit Hyperactivity Disorder (ADHD): A Phenomenological Study", Sakarya Universitesi (Turkey), 2024 Yayın	<%1
5	acikerisim.sakarya.edu.tr İnternet Kaynağı	<%1
6	dergipark.org.tr İnternet Kaynağı	<%1
7	Günel, Nihal Şenel. "Bağış Kampanyalarında Teleton Yayıncılığının Rolü : Türkiye'de Afet Dönemlerinde Bağış Davranışlarını Belirleyen	<%1