T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

METASEZGİSEL ALGORİTMALARLA OPTİMİZE EDİLMİŞ KESİR DERECELİ PID KONTROLÖRLER İLE GAZ TÜRBİN ELEKTRİK SANTRALİNİN KONTROLÜ

Kunter Sercan SEZER

DANIŞMAN
Dr. Öğr. Üyesi Nevra BAYHAN

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik-Elektronik Mühendisliği Programı

İSTANBUL-2021

Bu çalışma, 22.01.2021 tarihinde aşağıdaki jüri tarafından Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektrik-Elektronik Mühendisliği Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Dr. Öğr. Üyesi Nevra BAYHAN(Danışman)
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Dr. Öğr. Üyesi Hasan TİRYAKİ
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Doç. Dr. Ali Fuat ERGENÇ
İstanbul Teknik Üniversitesi
Elektrik-Elektronik Fakültesi

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa'nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Tez çalışmam süresince fikirleri ve deneyimleri ile bana yol gösteren ve beni destekleyen değerli tez danışmanım Dr. Öğr. Üyesi Nevra Bayhan'a, yapıcı önerileri ile desteklerini esirgemeyen Dr. Öğr. Üyesi Hasan Tiryaki'ye, fikir paylaşımları ve diğer yardımları için İstanbul Üniversitesi-Cerrahpaşa Doktora Öğrencisi Yılmaz Arıkuşu'na en içten teşekkürlerimi sunarım.

Ayrıca bana her zaman destek olan aileme ve arkadaşlarıma, tez sürem boyunca desteğini hiç esirgemeyen iş arkadaşım Mühendis Alparslan Çağlak'a yürekten teşekkür ederim.

Ocak 2021 Kunter Sercan SEZER

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **iv**
İÇİNDEKİLER ..... **v**
ŞEKİL LİSTESİ ..... **vi**
TABLO LİSTESİ ..... **ix**
SİMGE VE KISALTMA LİSTESİ ..... **x**
ÖZET ..... **xii**
SUMMARY ..... **xiv**
1 GİRİŞ **1**
2 GENEL KISIMLAR **6**
2.1 PID KONTRALÖRLER 6
2.2 KESİR DERECELİ PID KONTROLÖRLER 9
3 MALZEME VE YÖNTEM **13**
3.1 GAZ TÜRBİNLİ GÜÇ SANTRAL MODELİ 13
3.2 KESİR DERECELİ PID KONTROLÖRÜN SANTRAL MODELİNE UYGULANMASI 21
3.3 METASEZGİSEL OPTİMİZASYON ALGORİTMALARI 21
3.3.1 Parçacık Sürüsü Optimizasyon Algoritması (PSO) 23
3.3.2 Yapay Arı Kolonisi Optimizasyon Algoritması (ABC) 26
3.3.3 Gri Kurt Optimizasyon Algoritması (GWO) 30
3.3.4 Güve-Ateş Optimizasyon Algoritması (MFO) 33
3.3.5 Balina Optimizasyon Algoritması (WOA) 36
3.3.6 Salp Sürüsü Optimizasyon Algoritması (SSA) 41
3.3.7 Sinüs-Kosinüs Optimizasyon Algoritması (SCA) 45
3.3.8 Atom Arama Optimizasyon Algoritması (ASO) 49
3.4 SEZGİSEL OPTİMİZASYON ALGORİTMALARININ FOPID GÜÇ SANTRAL KONTROLÖRÜNE UYGULANMASI 54
4 BULGULAR **60**
5 TARTIŞMA VE SONUÇ **74**
6 KAYNAKLAR **78**
7 ÖZGEÇMİŞ ..... Error! Bookmark not defined.

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Klasik PID kontrolör sistemi. 6

Şekil 2.2: Örnek bir sistemin birim basamak cevabı [40]. 7

Şekil 2.3: S-düzleminde PID kontrolör blok diyagramı. 8

Şekil 2.4: s-düzleminde FOPID kontrolör. 12

Şekil 2.5: (a) PID türev-integral kuvvet haritası (b) FOPID türev-integral kuvvet haritası [44]. 12

Şekil 3.1: Basit gaz türbini çevrimi [38, 39, 46, 47]. 13

Şekil 3.2: Basitleştirilmiş santral modeli 14

Şekil 3.3: Basitleştirilmiş gaz türbin modeli [38, 39, 46, 47,49]. 15

Şekil 3.4: Türbin-Jeneratör ilişkisi [38, 39, 46, 47,49]. 16

Şekil 3.5: Türbin-Jeneratör güç aktarım modeli [38, 39, 46, 47,49]. 18

Şekil 3.6: Jeneratör uyarılma blok diyagramı [39, 46, 47,49]. 18

Şekil 3.7: Açısal hız değişimi çıkışlı jeneratör modeli [39, 46, 47,49]. 18

Şekil 3.8: Basitleştirilmiş jeneratör modeli [39, 46, 47,49] 19

Şekil 3.9: Genel Santral Modeli [39, 46, 47,49]. 20

Şekil 3.10: Santral Modeli [39, 46, 47,49]. 20

Şekil 3.11: FOPID kontrolör uygulanmış yeni santral modeli. 21

Şekil 3.12: Sezgisel algoritma kategorileri 22

Şekil 3.13: Kuş Sürüsü 23

Şekil 3.14: PSO Akış Diyagramı [28]. 25

Şekil 3.15: Arılar 26

Şekil 3.16: ABC Algoritması Akış Diyagramı [29]. 29

Şekil 3.17: Gri kurt sürü hiyerarşisi ve sürü. 30

Şekil 3.18: GWO akış diyagramı [30]. 32

Şekil 3.19: Güve Kelebeği. 33

Şekil 3.20: Güve Kelebeği Yörüngesi [67]. 33

Şekil 3.21: Ateş sayısının *l*'ye bağlı olarak değişimi [31]. 34

Şekil 3.22: *t*'ye bağlı M_i değişimi [31]. 35

Şekil 3.23: MFO Akış Diyagramı. 36

Şekil 3.24: Kambur balina ve kabarcık ağı. 37

Şekil 3.25: Arama uzayında balina hareketine *A*'nın etkisinin temsili gösterimi [32] 38

Şekil 3.26: 2 boyutlu arama uzayında balina hareketine *l*'nın etkisi [32]. 39

Şekil 3.27: WOA akış diyagramı [32]. 40

Şekil 3.28: Salp Sürüsü 41

Şekil 3.29: SSA Algoritma sıraısnda salp sürüsünün hareketi (2 boyutlu) [33]. 42

Şekil 3.30: SSA akış diyagramı 44

Şekil 3.31: r1 değerinin $Xt + 1$ konumuna etkisi ve 2 genlikli sinüs-kosinüs fonksiyonları [34, 73, 74]. 46

Şekil 3.32: r1 değerinin $Xt + 1i$ konumuna etkisi [34, 73, 74]. 46

Şekil 3.33: SCA simülasyonu sırasında fonksiyon dağılım aralığının zamana(yaklaşım sayısına) bağlı dağılımı ($a = 2$) [34, 73]. 47

Şekil 3.34: SCA akış diyagramı [34] 48

Şekil 3.35: Atomlar ve aralarındaki kuvvetler [35] 50

Şekil 3.36: ASO akış diyagramı. 53

Şekil 3.37: FOPID kontrolör ile denetlenen gaz türbinli elektrik güç santral modeli 55

Şekil 4.1: 0-100sn aralığında santral sistemi çıkış sinyali. 60

Şekil 4.2: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yükselme zamanları (%0-%100) 62

Şekil 4.3: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yükselme zamanları (%10-%90) 63

Şekil 4.4: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yükselme zamanları (%0-%100) 64

Şekil 4.5: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yükselme zamanları (%10-%90) 65

Şekil 4.6: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yerleşme zamanları (%5)..... 67

Şekil 4.7: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yerleşme zamanları (%5)..... 68

Şekil 4.8: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yerleşme zamanları (%2)..... 69

Şekil 4.9: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yerleşme zamanları (%2)..... 70

Şekil 4.10: Optimizasyon algoritmaları yaklaşım eğrileri (PSO, ABC, WOA, SSA) 72

Şekil 4.11: Optimizasyon algoritmaları yaklaşım eğrileri (GEO, MFO, SCA, ASO) 73

TABLO LİSTESİ

Sayfa No

Tablo 3.1: Basitleştirilmiş gaz türbin modeli parametre değerleri [38, 39, 46, 47,49]..... 15

Tablo 3.2: Basitleştirilmiş jeneratör modeli parametre değerleri [39, 46, 47,49]. 19

Tablo 3.3: FOPID optimizasyonu için PSO simülasyon parametre değerleri..... 56

Tablo 3.4: FOPID optimizasyonu için ABC simülasyon parametre değerleri..... 56

Tablo 3.5: FOPID optimizasyonu için GWO simülasyon parametre değerleri..... 57

Tablo 3.6: FOPID optimizasyonu için MFO simülasyon parametre değerleri. 57

Tablo 3.7: FOPID optimizasyonu için WOA simülasyon parametre değerleri..... 58

Tablo 3.8: FOPID optimizasyonu için SSA simülasyon parametre değerleri..... 58

Tablo 3.9: FOPID optimizasyonu için SCA simülasyon parametre değerleri. 59

Tablo 3.10: FOPID optimizasyonu için ASO simülasyon parametre değerleri. 59

Tablo 4.1: Pozitif bölge geçici hal kriter ve değerleri (0-50sn)..... 71

Tablo 4.2: Negatif bölge geçici hal kriter ve değerleri (50-100sn) 71

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
K_P	: Oransal Kazanç Katsayısı
K_I	: İntegral Kazanç Katsayısı
K_D	: Türev Kazanç Katsayısı
e(t)	: t anındaki hata
λ	: Kesir dereceli PID kontrolör transfer fonksiyonunda integralin kuvvet sayısı
μ	: Kesir dereceli PID kontrolör transfer fonksiyonunda türevin kuvvet sayısı
K	: Hız regülatörü kazancı
Tb	: Hız düzenleyicisi birim zaman sabit değeri
Ti	: Hız düzenleyici ikincil zaman sabit değeri
Ky	: Yakıcı sistemin kazanç sabit değeri
Ty	: Yakıcı sistemin zaman sabit değeri
Tyo	: Yanma odası zaman sabit değeri
Tt	: Türbin dinamik özelliklerine bağlı zaman sabit değeri
Tk	: Kuvvetlendirme zaman sabit değeri
Tu	: Uyarma zaman sabit değeri
Tj	: Jeneratör zaman sabit değeri
Z	: Sönüm sabit değeri
H	: Hız düzenleyici sabit değeri veya kayması
X	: 5 boyutlu arama uzayındaki konum ifadesi
X_t^i	: i numaralı arama yapan ajanın t iterasyonundaki konumu
X_{t+1}^i	: i numaralı arama yapan ajanın t+1 iterasyonundaki konumu
v_t^i	: i numaralı arama yapan ajanın t iterasyonundaki hızı
v_{t+1}^i	: i numaralı arama yapan ajanın t+1 iterasyonundaki hızı
$\emptyset_i$	: [0,1] aralığında değişen rastgele çarpanlardır
D	: Iki ajanın birbirine olan uzaklığıdır
ub	: Arama uzayının üst sınırı
lb	: Arama uzayının alt sınırı
λ(t)	: Langrajyan çarpanı
u(t)	: t anındaki kontrolör çıkış işareti

y(t)	: t anındaki sistem çıkış işareti
$\Gamma(z)$	: Euler Gama fonksiyonu
$\propto$	: kesir dereceli integra-türev kuvveti
$PI^{\lambda}D^{\mu}$	: kesir dereceli PID kontrolör genelleştirilmiş gösterimi
ω	: açısal hız
a	: açısal ivme

Kısaltmalar	**Açıklama**
PID	: Oransal-İntegral-Türev Kontrolör
FOPID	: Kesir Dereceli Oransal-İntegral-Türev Kontrolör
IAE	: Bozucu Etkilerin Değişimine Bağlı Hatanın Mutlak Değerinin İntegralinin En Küçüklenmesine Dayalı Bir Kontrolör Katsayı Ayarlama Yöntemi
ITAE	: Bozucu Etkilerin Değişimine Bağlı Hatanın Mutlak Değerinin Zaman Ağırlıklı İntegralinin En Küçüklenmesine Dayalı Bir Kontrolör Katsayı Ayarlama Yöntemi
ITSE	: Bozucu Etkilerin Değişimine Bağlı Hatanın Karesinin Zaman Ağırlıklı İntegralinin En Küçüklenmesine Dayalı Bir Kontrolör Katsayı Ayarlama Yöntemi
ISE	: Bozucu Etkilerin Değişimine Bağlı Hatanın Karesinin En Küçüklenmesine Dayalı Bir Kontrolör Katsayı Ayarlama Yöntemi
FOMCON	: Kesir dereceli modelleme ve kontrol (Fractional Order Modeling and Control)
PSO	: Parçacık sürüsü optimizasyonu (Particle Swarm Optimization)
ABC	: Yapay arı kolonisi optimizasyonu (Artificial Bee Colony)
GWO	: Gri kurt optimizasyonu (Grey Wolf Optimization)
MFO	: Güve-ateş optimizasyonu (Moth-Flame Optimization)
WOA	: Balina optimizasyonu algoritması (Whole Optimization Algorithm)
SCA	: Sinüs-kosinüs algoritması (Sine-Cosine Algorithm)
SSA	: Salp sürüsü algoritması (Salp Swarm Algorithm)
ASO	: Atom arama optimizasyonu (Atom Search Optimization)

ÖZET

YÜKSEK LİSANS TEZİ

METASEZGİSEL ALGORİTMALARLA OPTİMİZE EDİLMİŞ KESİR DERECELİ PID KONTROLÖRLER İLE GAZ TÜRBİN ELEKTRİK SANTRALİNİN KONTROLÜ

Kunter Sercan SEZER

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman : Dr. Öğr. Üyesi Nevra BAYHAN

Günümüzde elektrik üretim santrallerinde klasik kontrol yöntemleri sıklıkla kullanılmaktadır. Klasik kontrol yöntemleri, santrallerin uzun çalışma süreleri ve hattaki yük kapasitesinin değişken olması sebebiyle oluşan dinamiklere yeterli cevabı sağlamayabilmektedir. Bu gibi durumlara çözüm olması için daha esnek ve etkin yapılarıyla modern kontrol yöntemlerinin kullanımı giderek önem kazanmaktadır.

Kullanımı oldukça yaygın olan klasik orantı-integral-türev (PID) kontrolörler günümüzde birçok kontrol sisteminde bulunduğu gibi elektrik santrallerinde de kullanılmaktadır; ancak bazı koşullarda gereksinimleri karşılamayabilmekte, santralin kurulumundan günümüze kadar değişen ortam şartları ve yaşlanmanın da etkisiyle mevcut kontrolörlerden daha üstün sonuçlar vermesi beklenebilmektedir. Bu gibi durumlarda kullanılabilirliğinin değerlendirilmesi için klasik PID kontrolörün genelleştirilmiş modern hali olan kesir dereceli PID (FOPID) kontrolörler bu tezde ele alınmıştır.

Bu amaçla bu tezde; bilgisayar ortamında simülasyonlar yapılabilmesi için sistem modeli olarak basitleştirilmiş bir gaz türbinli elektrik santral modeli ele alınmış ve FOPID kontrolör ile sistem denetimi sağlanmıştır. FOPID kontrolörlerin optimizasyonu, yapıları gereği yüksek matematik bilgisi gerektirmektedir ve bu işlemler klasik matematik yöntemleri ile yapıldığında oldukça fazla zaman almaktadır. Bu çalışmada; optimizasyon yöntemi olarak parçacık sürüsü optimizasyonu (PSO), yapay arı kolonisi optimizasyonu (ABC), gri kurt optimizasyonu (GWO), güve-ateş optimizasyonu (MFO), balina optimizasyonu (WOA), salp sürüsü

optimizasyonu(SSA), sinüs-kosinüs optimizasyon (SCA), atom arama optimizasyonu (ASO) metasezgisel algoritmalarının kullanımıyla FOPID kontrolörlerin optimizasyonu ele alınmıştır.

Bu tezde basitleştirilmiş gaz türbinli santral modelinin kontrolü için FOPID kontrolör içeren sistemin simülasyonu yapılıp, yukarıda bahsi geçen metasezgisel algoritmalar ile optimizasyonu sağlanarak tasarlanmış olan FOPID kontrolörler yardımıyla santral çıkışındaki istenmeyen güç dalgalanmalarına oldukça kısa sürede cevap verilmesi sağlanmıştır. Bahsedilen optimizasyon algoritmaları ile parametre değerleri belirlenmiş FOPID kontrolörün uygulandığı basitleştirilmiş gaz türbinli elektrik güç santral modeli çıkış işaretleri; yükselme süresi, yüzde en büyük aşım değeri ve yerleşme süresi kriterlerine göre karşılaştırılmıştır.

Elde edilen simülasyon sonuçları, GWO algoritması ile tasarlanmış FOPID kontrolörün, ele alınan santral modeli için uygun ve kullanışlı olduğunu göstermektedir. GWO algoritması ile optimize edilmiş FOPID kontrolör uygulanmış basitleştirilmiş gaz türbinli elektrik güç santral modeli çıkış işaretinin yükselme ve yerleşme sürelerinin, diğer algoritmalar ile bulunan FOPID kontrolör parametre değerleri kullanıldığında elde edilen sistem çıkış sinyallerindeki yükselme ve yerleşme sürelerine kıyasla daha küçük olduğu görülmüştür. FOPID kontrolör ile denetlenen sistemin çıkış işaretleri karşılaştırıldığında en az aşım yapan iki işaretin sırasıyla küçükten büyüğe ABC ve GWO algoritmaları ile optimize edilmiş olduğu anlaşılmıştır. Sonuç olarak, sistem çıkış işaretlerindeki yükselme süresi, yerleşme süresi ve yüzde en büyük aşım değeri kriterleri bir arada değerlendirilerek, enerji üretim sektöründe FOPID kontrolör kullanılan modern kontrol sistemlerinin optimizasyonu için PSO, ABC, GWO, MFO, WOA, SSA, SCA, ASO algoritmaları ile çözümler sunulmuştur.

Ocak 2021, 101 sayfa.

Anahtar kelimeler: kesir dereceli PID, metasezgisel algoritma, optimizasyon karşılaştırma, gaz türbinli santral, güç kontrol, parçacık sürüsü optimizasyonu (PSO), yapay arı kolonisi optimizasyonu (ABC), gri kurt optimizasyonu (GWO), güve-ateş optimizasyonu (MFO), balina optimizasyonu (WOA), salp sürüsü optimizasyonu(SSA), sinüs-kosinüs optimizasyon (SCA), atom arama optimizasyonu (ASO)

SUMMARY

M.Sc. THESIS

CONTROL OF GAS TURBINE POWER PLANT WITH FRACTIONAL ORDER PID CONTROLLERS OPTIMIZED BY METAHEURISTIC ALGORITHMS

Kunter Sercan SEZER

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Electrical and Electronic Engineering

Supervisor : Assist. Prof. Dr. Nevra BAYHAN

Nowadays; classical control methods are used in general of power generation plants. Classical control methods; due to the long operating times of the power plants and the variable load capacity on the grid, it may not provide the necessary response to the dynamics that occur. It is becoming increasingly important to use modern control methods with more flexible and effective structures to solve such situations.

Classical proportional-integral-derivative (PID) controllers, which are widely used, are also used in power plants as well as in many control systems today; however, it may not meet the requirements in some conditions, and it can be expected to give superior results than existing controllers due to the changing environmental conditions and aging. Fractional-order PID (FOPID) controllers, one of the modern controllers, are discussed in this thesis in order to be evaluated in such cases.

For this purpose, in this thesis; in order to make simulations in computer environment, a simplified gas turbine power plant model has been considered as the system model and system control has been provided with the FOPID controller. Optimization of FOPID controllers requires high mathematical knowledge due to their nature, and these operations take quite a lot of time when done with classical mathematical methods. In this study; by using particle swarm optimization (PSO), artificial bee colony optimization (ABC), gray wolf optimization (GWO), moth-fire optimization (MFO), whale optimization (WOA), salp swarm optimization (SSA),

sine-cosine optimization as optimization method (SCA), atomic search optimization (ASO) metaheuristic algorithms; optimization of FOPID controllers is discussed.

In this thesis, for the control of the simplified gas turbine power plant model, the system containing the FOPID controller is simulated and it is provided to respond to the unwanted power fluctuations at the power plant output in a very short time with the help of FOPID controllers designed by mentioned metaheuristic algorithms. The simplified gas turbine electric power plant model output signals where the FOPID controller whose parameter values are determined with the help of mentioned optimization algorithms is applied; was compared according to the rise time, maximum overshoot and settling time criteria.

The obtained simulation results show that the FOPID controller which is optimized by GWO algorithm are suitable and useful for the power plant model. It has been observed that the rise and settling times provided by output signal of the simplified gas turbine electric power plant model which contains FOPID controller designed by GWO algorithm are smaller compared to the rise and settling times in the output signals when using the FOPID controller parameter values obtained by other algorithms. And also it seems that the overshoots of the output signal which belongs to system contains FOPID controller designed by GWO and ABC algorithms are smaller then the others. As a result, by evaluating rise time, settling time and overshoot ratio criterias of system output signals together, PSO, ABC, GWO, WOA, MFO, SSA, SCA, ASO algorithms have been presented as solutions for the optimization of modern control systems which includes FOPID controller in the power generation sector.

January 2021, 101 pages.

Keywords: fractional order PID, metaheuristic algorithm, comparison of optimization, gas turbine power plant, power control, particle swarm optimization(PSO), artificial bee colony(ABC), grey wolf optimization(GWO), moth-flame optimization(MFO), whole optimization algorithm(WOA), salp swarm algorithm(SSA), atomic search algorithm (ASO)

1 GİRİŞ

Son yıllardaki teknolojik gelişmelerle birlikte kontrol sistemlerinin kullanım alanları her geçen gün artmaktadır. Günlük yaşantımızda kontrol sistemleri otomobiller, akıllı ev aletleri, endüstriyel üretim bantları, uzay taşıtları, askeri alanlar ve medikal cihazlar vb. çeşitli amaçlarla hayatımızın her alanında yer alır hale gelmiştir. Genellikle sıcaklık, nem, basınç, akı, hız parametrelerinin denetim altında tutulması için kontrol sistemlerine ihtiyaç duyulmaktadır. Kontrol sistemleri, emek ve nitelikli iş gücünün daha verimli kullanılmasına imkan sağlarken aynı zamanda da insan yaşamını kolaylaştırır ve yüksek hassasiyetle denetim yapılabilmesini mümkün kılar.

Tarihte bilinen ilk basit kontrol sistemi milattan önce Antik Yunan Toplumları tarafından bulunmuştur ve su seviyesinin sabit tutulması için kullanılmıştır [1]. Bu buluş basit olmasının yanı sıra oldukça işlevseldir. İlk mekanik su saatinin icadının da önünü açmıştır ve günümüzde hala basit tasarımı ile kullanılarak, şamandıra adı ile bilinir [2, 3]. 16. yüzyıla gelindiğinde modern çağın getirdiği gereksinimler doğrultusunda yakıt kontrolü ve sıcaklık sabitleme işlemleri için ilk geri beslemeli kontrol sistemi Holland tarafından geliştirilmiştir [3, 4]. 18. yüzyılda Watt tarafından bir buhar türbinli makinenin hız kontrolünü yapmak için ilk otomatik geri beslemeli kontrol sistemi geliştirilmiştir [3,4]. Watt'ın "flyball governer" adını verdiği bu icat hassas şekilde ayarlandığında bir buhar türbinli motorun dönüş hızının yüksek hassasiyetle kontrol edilebilmesine imkan sağlamış ve sonrasında "flyball governer" farklı birçok sisteme de uygulanmıştır. 1830'larda Faraday'ın indüksiyon kanununu tanımlaması, elektrik motorunun temellerini atmış, 1880'lerin sonunda ise Tesla'nın alternatif akımı bulması ile birlikte alternatif akım motorunu tasarlaması, sanayileşmeyi hızlandırmıştır [4]. 19.yy'a gelindiğinde sanayi devrimi sonrasında; üretimin artması ve çalışan bazında iş alanlarının daralarak çalışanların üretim bandının bir parçası olması, üretilen ürünün bütününden koparak özelleşmiş bir bölümünden sorumlu olması yapılan işin birçok kez tekrarlanarak rutine dönüşmesine sebep olmuştur. Özelleşmiş ve tekrarlar hale gelmiş işlemlerin, gelişimini sürdüren kontrol sistemleri ile de yapılabileceğinin anlaşılması ile otomatik kontrol sistemlerine ilgi artmıştır. 1900'lü yıllarda insanlar fabrikalarda günümüzde makinelerin yaptığı birçok işi kendileri, insan gücü ile yapmaya çalışıyordu. Kontrol sistemlerinin gelişmesi ile insan gücünün yerini büyük ölçüde modern kontrol sistemleri almıştır ve kontrol sistemlerinin

gelişimi hızla devam etmiştir. 1911 yılında Sperry [5] kontrol sistemlerinde bulunan kontrolör çeşitlerinden klasik orantı-integral-türev (PID) kontrolöre benzer ilk denetleyiciyi geliştirmiştir. Ardından 11 yıl sonra 1922'de mühendis Minorsky ilk kez bir yayında klasik PID kontrolöre benzer düşük dereceli kontrolörün teknik analizini yapmıştır [6]. Minorsky PID kontrolörü gemi direksiyon (dümen) sistemlerine uyarlaması ile tanınmıştır [5, 7]. Tasarım ve uygulanabilirliğinin kolay olması, gereksinimleri çoğu zaman yeteri kadar karşılayabiliyor olması, maliyetinin düşük olması gibi sebeplerle kontrol sistemlerinin birçoğunda PID kontrolörler ya da benzeri düşük dereceli denetleyicilerden faydalanılmıştır [7]. Ancak PID kontrolörlerin de istenileni karşılamaya yeterli gelmediği durumlarda daha hassas kontrolörlere ihtiyaç duyulabilmektedir.

17. yüzyılın sonlarında l'Hopital'ın Leibniz'e yolladığı mektubunda ilk kez bahsettiği kesir dereceli diferansiyel denklemler fikri, gelecek yıllarda farklı bir PID kontrolör yaklaşımı geliştirilmesine fayda sağlamıştır. Türev ve integral dereceleri tam sayı olması gerekmeyen, herhangi bir reel sayı değerini alabilen diferansiyel denklemlerle ifade edilen sistemler, kesir dereceli sistemler olarak tanımlanırlar. 18. yüzyılda Liouville ve Holmgren çalışmalarını bu konu üzerine sürdürmüş ve 19. yüzyıla gelindiğinde ise Riemann bu alandaki çalışmaları devam ettirmiştir [8-10].

Kesir dereceli diferansiyel denklem fikrinin olgunlaşması ile birlikte klasik PID kontrolörlerin gereksinimleri karşılamadığı durumlarda kullanılabilecek, klasik PID sistemlerden daha hassas kontrol yöntemleri geliştirebilmek adına PID kontrolörlerdeki türev ve integral kuvveti olarak kesir dereceli sayıların kullanılabileceğini düşünen Podlubny, 1999 yılındaki yayınında ilk kez "Frational Order Proportional Integral Derivative" (FOPID) kontrolör yani kesir dereceli PID kontrolör fikrini öne sürmüştür [11].

FOPID kontrolörlerin kontrol sistemlerinde kullanılırken optimize edilmesi gerektiğinde parametre sayısının klasik PID kontrolörden fazla olması ve yüksek matematik hesaplamaları gerektirmesi yüzünden bu sürec bilim insanları için zorlayıcı olmuştur. Konvansiyonel PID kontrolör tasarım yöntemlerinden Ziegler Nichols, Cohen-Coon, Astrom-Hagglund, Chien-Hrones-Reswick metodları FOPID kontrolörlere de uyarlanmaya çalışılmıştır [12-16]. FOPID kontrolörlerin optimizasyonu konusunda klasik matematik yöntemleri kullanılarak, faz payı, kazanç payı, kazanç geçiş frekansı temelli yöntemlere dayanan birçok çalışma yapılmıştır [17-

20]. Keyser ve diğ. 2018 yılında kesir dereceli PID kontrolör parametrelerinin optimizasyonu için basit sinüs testini kullanarak bir optimizasyon yöntemi önermiştir [21].

Son yıllarda bilgisayar yardımı ile matematiksel problemlerin çözümü eskiye nazaran kolaylaşmıştır, bu sebeple günümüz modern sistemlerinin kontrolü için FOPID kontrolör kullanılan çalışmalar da giderek artmaktadır. Gül ve Tan'ın yapmış olduğu çalışmada güneş panelleri ve rüzgar gülü gibi yenilenebilir enerji kaynaklarının çıkış güç kontrolünde FOPID kontrolörler sıradan PID kontrolörlere göre daha dayanıklı ve kararlı sonuçlar vermiş, dış etkenler sebebiyle sistem güç çıkış sinyalindeki dalgalanmalar azalmış ve toplam harmonik bozulmanın daha az olduğu belirlenmiştir [22]. Malek'in yapmış olduğu doktora çalışmasında ise bir güneş enerji santralinin hat ile bağlantısını sağlayan düzenleyici kontrol sisteminde FOPID kontrolör kullanılmıştır [23]. Göktürk'ün 2009 yılında yaptığı çalışmada PID ve FOPID kontrol sistemleri karşılaştırılmıştır [24]. PID sistemlerde parametre sayısı üç iken, FOPID sistemlerde parametre sayısı beş olduğundan optimizasyon parametrelerinin artması sonuçlara da olumlu yansımıştır [11, 13, 22, 24]. FOPID kontrolörlerin eskiye nazaran kullanım alanlarının genişlediğini ortaya koyan ve endüstride bu kontrolörün tercih edilme sebeplerini inceleyen farklı bir çalışmada ise FOPID kontrolörlerin enerji tasarrufu sağladığı ve hassas kontrole harcanan eforu azalttığına değinilmiştir [25]. Bir diğer çalışmada da klasik PID ve FOPID kontrolörlerin, hata karelerinin toplamı (ISE, Integral Squere Error) ve zaman ağırlıklı hata karelerinin toplamı (ITSE, Integral Time Squere Error) yöntemleri kullanılarak kıyaslaması yapılmış ve FOPID kontrolörlerin daha üstün olduğu belirtilmiştir [26].

FOPID kontrolörler, yapıları gereği üç parametreli klasik PID kontrolörlerden farklı olarak beş parametreye sahiptirler. Klasik PID kontrolörlere göre daha karmaşık yapıya sahip kesir dereceli PID kontrolörleri optimize edebilmek için bu tezde günümüz modern metasezgisel optimizasyon algoritmaları alanındaki literatür araştırması sonucunda başarılı olabileceği düşünülen, son dönemde kullanım sıklığı artan bazı algoritmalar incelenmiş ve hangilerinin kullanılacağı kararlaştırılmıştır.

Usta'nın [27] no'lu çalışmasında metasezgisel optimizasyon algoritmaları hakkında genel bir bakış edinilebilmesi için yaygın kullanılan, popüler algoritmaların esin kaynakları temel alınarak karşılaştırmaları yapılmış ve sınıflandırılmıştır.

1975 yılında Kennedy ve Eberhart'ın çalışmaları sonucunda parçacık sürüsü optimizasyon (Particle Swarm Optimisation, PSO) algoritması bulunmuş ve o günden günümüze kadar geçen

süre boyunca çok sayıda sisteme uygulanmış olup optimizasyon problemlerine alternatif bir çözüm olarak sunulmuştur [28]. 2005 yılında yapılan bir çalışma sonucunda bulunmuş olan yapay arı kolonisi (Artificial Bee Colony, ABC) optimizasyon algoritması metasezgisel optimizasyon algoritmaları alandaki önemli çalışmalardan biridir [29]. ABC algoritmasında arı kolonilerinin besin ararken gösterdiği davranışlarından esinlenilmiştir. Grup halinde hareket eden, farklı içgüdüler ile avlanan hayvanlardan bir diğeri ise kurtlardır. 2014 yılında Mirjalili ve diğ. tarafından yayınlanan gri kurt optimizasyon algoritması da kurtların avlanma tarzlarından esinlenmiştir [30]. Bir diğer yaklaşım ise güve kelebeklerinin ışığı takip etmesinden ve ışığın etrafında dönme şekillerini inceleyerek ortaya çıkarılmış olan güve-ateş optimizasyonudur (Moth-Flame Optimizasion, MFO) [31]. Bu yöntemlere benzer şekilde farklı hayvan davranışları ve içgüdülerinden esinlenerek Mirjalili farklı algoritmalar da ortaya koymuştur. Bunlar arasında balina optimizasyonu algoritması (Whale Optimization Algorithm, WOA) ve salp sürüsü algoritması (Salp Swarm Algorithm) önde gelen yöntemlerdendir [32, 33].

Sezgisel yöntemler arasında optimizasyon problemleri açısından iyi sonuçlarıyla öne çıkan ve bu tezde kullanılan matematik tabanlı bir diğer yöntem olan sinüs-kosinus algoritması, 2016 yılında geliştirilmiştir [34].

2019 yılında jeoloji alanında çok bilinmeyenli denklemlerin çözümü için yüklü parçacıkların potansiyel enerjileri sebebiyle birbirini itme ve çekme kuvvetlerinden esinlenilerek oluşturulan fizik tabanlı olan atom arama algoritması (Atom Search Optimization, ASO) Zhao ve diğ. tarafından geliştirilmiştir [35]. Zhao ve diğ.'nin çalışmasında ASO algoritması parçacık sürüsü optimizasyonu ve genetik algoritma ile karşılaştırılarak alternatif bir yöntem olarak sunulmuştur [28, 36]. Genetik algoritma, Darvin'in genetik seçilim teorisinden esinlenilerek oluşturulmuş bir optimizasyon algoritmasıdır [37].

Bu tezde; elektrik üretimi için tasarlanmış olan gaz türbinli elektrik santralinin güç kontrol sisteminde, FOPID kontrolörün etkin şekilde performans sağlayacağı ön görülüp, bu santralin [38,39]'daki basitleştirilmiş modeli için son yıllardaki yaygın kullanımları ile önde gelen farklı optimizasyon algoritmaları kullanılarak performans analizi karşılaştırılması yapılmıştır.

Bu tezin amacı; [38, 39]'daki bir enerji üretim gaz türbinli santral modelinin güç kontrolünde kullanılan FOPID kontrolörün PSO, ABC, GWO, MFO, WOA, SCA, SSA, ASO yöntemleri

kullanılarak optimize edilmesi ve elde edilen sonuçlar doğrultusunda optimizasyon algoritmalarının karşılaştırılmasının yapılmasıdır.

Yapılmış olan bu tez çalışması giriş, genel kısımlar, malzeme ve yöntem, bulgular, tartışma ve sonuç olmak üzere beş bölümden oluşmaktadır.

İkinci bölüm olan genel kısımlar başlığı altında, klasik PID kontrolör ve FOPID kontrolör yapıları hakkında teorik bilgiler anlatılmıştır ve farklılıklar ortaya koyulmuştur.

Üçüncü bölüm olan malzeme ve yöntem kısmında öncelikle, gaz türbinli güç santralinin çalışma prensibi açıklanmıştır ve bu santralin basitleştirilmiş modeli hakkında teorik bilgiler verilmiştir. Ardından, ele alınan santral modeline uygulanarak karşılaştırması yapılacak olan optimizasyon algoritmaları hakkında teorik bilgiler, akış diyagramları, sözde kod örnekleri verilmiştir. Bu algoritmalar sistem modeline uygulanırken kullanılan parametreler açıklanarak, algoritmaların en uygun parametre değerlerini saptaması için ihtiyaç duyulan amaç fonksiyonu çeşitleri (IAE: mutlak hatanın toplamı, ISE: hata karelerinin toplamı, ITAE: zaman ağırlıklı mutlak hata toplamı, ITSE: zaman ağırlıklı hata karelerinin toplamı) hakkında teorik içerik aktarılmıştır. Son olarak algoritmaların ele alınan basitleştirilmiş gaz türbinli elektrik güç santral modeline uygulanması sonucunda elde edilen optimize edilmiş kontrolör parametreleri belirlenmiştir.

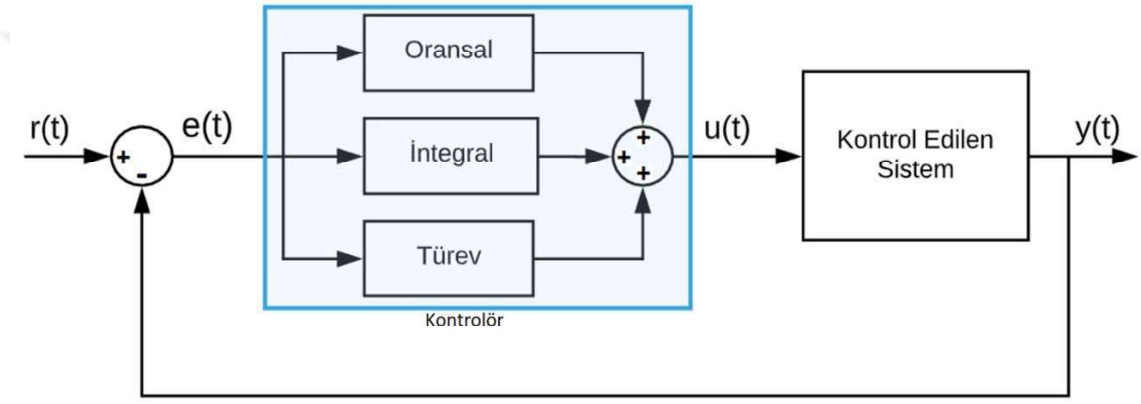
Dördüncü bölüm olan bulgular kısmında optimize edilmiş FOPID kontrolör parametreleri ile denetlenen sistem modeli girişine uygulanan giriş sinyali ve sistemin buna karşılık cevabı irdelenerek kontrolör geçici durum performans karşılaştırmaları yükselme süresi, aşım miktarı ve yerleşme süresi kriterlerine bakılarak yapılmıştır. Kontrolör performanslarından ve optimizasyon algoritmaları amaç fonksiyonlarının sıfıra yakınsama eğrileri değerlendirilerek kullanılan metasezgisel optimizasyon algoritmalarının ele alınan model için performansları kıyaslanmıştır.

Beşinci ve son bölüm olan tartışma ve sonuç başlığı altında ise elde edilen bulgular değerlendirilmiş, karşılaştırmalar yapılmış ve bu doğrultuda çıkarımlar belirtilmiştir.

2 GENEL KISIMLAR

2.1 PID KONTRALÖRLER

Şekil 2.1'de PID kontrolör içeren bir sistem verilmiştir. PID kontrolörler geri beslemeli sistemlerde bulunur ve kontrol edilen sistemin çıkış sinyalini (Şekil 2.1'de y(t)) istenilen seviyede tutmak için kontrol edilen sistemin giriş sinyalinin (Şekil 2.1'de u(t)) düzenlemesini yaparlar.



Şekil 2.1: Klasik PID kontrolör sistemi.

PID kontrolör üç ana bileşenden oluşur ve bunlar sırasıyla: P, I, D

P: proportional gain (oransal kazanç)

I: integration gain (integral kazancı)

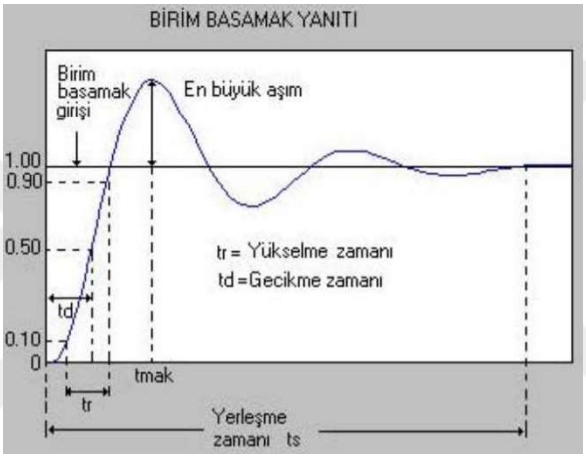
D: derivative gain (türev kazancı)

Klasik PID kontrolörün zamana (t: zaman) bağlı ifadesi için Şekil 2.1'den haraketle aşağıdaki denklem (2.1) yazılabilir.

$$u(t) = K_P\, e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \qquad (2.1)$$

PID kontrolörlerde K_P, K_I ve K_D şeklinde üç parametrenin ayarlanması gerekir. Oransal kazanç parametresi girdi ile çıktı arasındaki katsayısal büyüklük farkını doğrudan etkiler. Bu kazanç değeri K_P ise, K_P değerinin artması istenilen çıkış değerine daha çabuk ulaşılmasını

sağlar. K_P'nin büyük bir değer olması aynı zamanda aşımlara da sebep olabilir. İstenilen çıkış değerine ilk ulaşımdan sonra kalıcı durum yanılgılarının azaltılması için integral etkisi kullanılır. Türev etkisi ise geçici yanıtın iyileştirilmesi için kullanılır. İntegral kazanç katsayısı K_I, türev kazanç katsayısı K_D olarak ifade edilir. Şekil 2.2'de örnek bir sistemin birim basamak cevabı verilmiştir. Buradaki geçici durum cevabı incelendiğinde görülebilecek en büyük aşım, yükselme zamanı, yerleşme zamanı değerleri K_P, K_I ve K_D parametrelerinin ayarlanması ile istenilen hale getirilebilir.



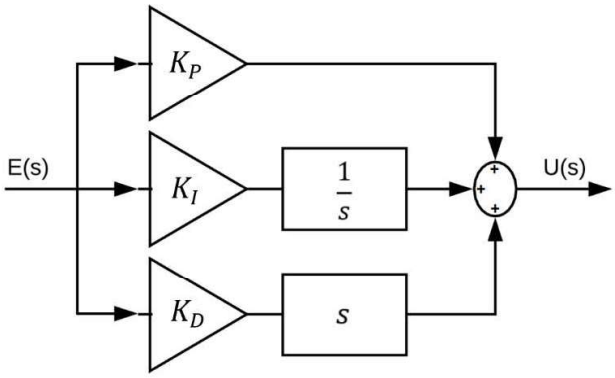
Şekil 2.2: Örnek bir sistemin birim basamak cevabı [40].

Yüzde en büyük aşım, sistem cevabının en büyük değeri ile sürekli hal değeri arasındaki farkın sürekli hal değerine oranının yüzdelik ifadesidir [40]. Yükselme zamanı, sistem cevabının %0'ından %100'üne ya da %10'nundan %90'ına ulaşıncaya kadar geçen süredir. Literatürde genellikle aşırı sönümlü sistemler için %10-%90, sönümlü sistemler için %0-%100 aralığında geçen süre yükselme zamanı olarak kabul edilmektedir [40]. Yerleşme zamanı, sistem cevabının belirlenmiş olan tolerans bandına giriş yaptığı ve bu bandın dışına tekrar çıkmadığı ana kadar geçen süredir. Literatürde yaygın olarak %2 ve %5'lik tolerans bandı kullanıldığı bilinmektedir [40].

PID kontrolörün Laplace düzlemindeki ifadesi ise (2.2) numaralı denklemde verilmiştir. Bu denklemde U(s), PID kontrolörün çıkışının s-düzlemindeki ifadesini; E(s) ise PID kontrolörün girişinin s-düzlemindeki ifadesini temsil etmektedir. Buna göre PID kontrolörün transfer fonksiyonu aşağıdaki gibidir:

$$\frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D\, s \qquad (2.2)$$

Böylece s-düzleminde PID kontrolörün blok diyagramı Sekil 2.3'de görülmektedir.



Şekil 2.3: s-düzleminde PID kontrolör blok diyagramı.

Tam dereceli PID kontrolörün bu üç katsayısı için işlevsellikler şunlardır:

K_P, meydana gelen hata sinyaline bağlı olarak PID kontrolör çıkışına aktarılacak olan değer doğrusal orantılı etkilenerek değişir. Sistem cevabının hızlıca referans değere yaklaşmasını sağlarken K_P değerinin büyüklüğü aşım değerinin artmasına sebep olabilir.

K_I, sistemin belirlenen referans değere yerleşmesinde en büyük etkiye sahip olan parametredir. Kalıcı hal hatasını azaltır.

K_D, sistem cevap süresini kısalmasına yardımcı olurken aşımı azaltıcı etkisi vardır. Yüksek değerler uygulandığında salınım ve aşımlara sebep olabilir [41-43].

Kontrol edilecek sistemin gereksinimleri doğrultusunda K_P, K_I ve K_D parametrelerinin optimizasyonu ile sistem çıkış sinyalinde minimum aşım değeri, en kısa yerleşme zamanı ve en kısa yükselme zamanı elde edilmeye çalışılır. Bu optimizasyon işlemi için klasik matematiksel optimizasyon yöntemleri kullanılabilirken; modern kontrol yöntemlerinin de kullanıldığı literatürde sıklıkla görülmektedir. PID kontrolör optimizasyonu için birçok bilgisayar uygulaması mevcuttur. Bu uygulamalar yardımıyla optimizasyonlar başarılı şekilde yapılabilmesine rağmen daha iyi sonuçlar elde etmek isteyen araştırmacılar farklı algoritmalar geliştirmek için çalışmalarına devam etmektedirler.

2.2 KESİR DERECELİ PID KONTROLÖRLER

Kesir dereceli matematik integral ve türevin tam sayı olmayan kuvvetleri için oluşturulmuş bir genellemedir. Kesir dereceli integral-türev operatörü ${}_aD_t^{\propto}$ denklem (2.3)'te verilmiştir.

$$ {}_aD_t^{\propto} = \begin{cases} \dfrac{d^{\propto}}{dt^{\propto}} & \mathrm{Re}(\propto) > 0 \\ 1 & \mathrm{Re}(\propto) = 0 \\ \displaystyle\int_a^t (dt)^{\propto} & \mathrm{Re}(\propto) < 0 \end{cases} \tag{2.3} $$

Burada $\propto$ türev ve integral kuvvetini ifade etmektedir. $\propto \in \mathbb{R}$ olmalıdır ve karmaşık sayı da olabilir. t limit değerlerini, a ise rastgele bir sayıyı temsil eder.

Kesir dereceli matematik için Grünwald–Letnikov ve Riemann-Liouville literatürde en yaygın kullanılan tanımlardandır [10].

Grünwald–Letnikov kesir dereceli türevi $\propto$ dereceli bir integral-türev denklemi olarak (2.4)'deki gibi ifade etmiştir.

$$ {}_aD_t^{\propto} f(t) = \lim_{h \to \infty} \frac{\Delta_h^{\propto} f(t)}{h^{\propto}} \tag{2.4} $$

Burada; h adım sayısıdır ve

$$ \Delta_h^{\propto} f(t) = \sum_{j=0}^{\infty} (-1)^j \binom{\propto}{j} f(t - jh) \tag{2.5} $$

denklemi ile ifade edilir. Denklem (2.5)'de toplam işlemi r sonlu sayısına kadar hesaplandığında;

$$ \sum_{j=0}^{r} (-1)^j \binom{\propto}{j} = \frac{1}{\Gamma(n-\propto)} \frac{\Gamma(r+1-\propto)}{\Gamma(r+1)} \tag{2.6} $$

sonucuna ulaşılacak ve $h = \frac{t-a}{r}$ kabul edilip denklem (2.4)'te yerine yazıldığında denklem (2.7) ve (2.8) elde edilir.

$$ {}_aD_t^{\propto} f(t) = \lim_{h\to\infty} \frac{1}{h^{\propto}} \sum_{j=0}^{\left[\frac{(t-a)}{h}\right]} (-1)^j \binom{\propto}{j} f(t-jh) \tag{2.7} $$

$$ {}_aD_t^{\propto} f(t) = \lim_{r\to\infty} \left\{ \frac{\left(\frac{(t-a)}{h}\right)}{\Gamma(-\propto)} \sum_{j=0}^{[r-1]} \frac{\Gamma(j-\propto)}{\Gamma(j+a)} f\left(t - j\left(\frac{t-a}{h}\right)\right) \right\} \tag{2.8} $$

Riemann-Liouville tanımı denklem ise (2.9)'de verilmiştir [10].

$$ {}_aD_t^{\propto} f(t) = \frac{1}{-\propto} \int_a^t (t-\tau)^{-\propto-1} f(\tau) d\tau, \quad a < 0 \tag{2.9} $$

n-1<∝<*n* olmak koşuluyla Denklem (2.9)'ün *n.* dereceden türevi alındığında Denklem (2.10) elde edilir.

$$ {}_aD_t^{\propto} f(t) = \frac{1}{\Gamma(n-\propto)} \frac{d^n}{dx^n} \int_a^t (t-\tau)^{n-\propto-1} f(\tau) d\tau, \quad n > 0 \tag{2.10} $$

Denklem (2.10), $f(t)$ fonksiyonunun (n-∝). dereceden kesir dereceli türevine denk gelmektedir. Bu denklemde n tamsayıdır. Denklem (2.10) hesaplanırken ihtiyaç duyulan Euler Gama Fonksiyonu,

$$ \Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt, \quad t > 0 \tag{2.11} $$

eşitliği ile bulunur. Burada z ≥ 0 ve tam sayı olmalıdır. Denklem (2.11)'nın (z+1) için integrali alındığında denklem (2.12) elde edilir.

$$\Gamma(z+1) = \int_0^{\infty} e^{-t} t^{(z+1)-1} dt = \int_0^{\infty} e^{-t} t^{z} dt$$
$$= [-e^{-t} t^{z}]_{t=0}^{t=\infty} + z \int_0^{\infty} e^{-t} t^{z-1} dt = z\Gamma(z) \quad (2.12)$$

Kesir dereceli matematikte Laplace dönüşümü $\propto \in \mathbb{R}$ ve x(0) = 0 şartları sağlandığında Denklem (2.13)'dan faydalanılarak yapılır [10].

$$\mathcal{L}\{D^{\propto}\mathrm{x(t)}\} = s^{\propto} X(s) \quad (2.13)$$

İlk olarak Podlubny kesir dereceli matematiği kullanarak PID kontrolör için daha genel bir yapı oluşturarak türev ve integral kuvvetlerinin tam sayılardan oluşması durumunu genişleterek kuvvetlerin kesirli sayılar da olabileceğini çalışmasında anlatmıştır [11].

Bir önceki bölümde, tam dereceli PID kontrolörlerin oransal, integral, türev etkisi olmak üzere üç parametresinin sistem çıktısına etkisi incelenmiştir. Klasik PID kontrolörlerde K_P , K_I, K_D şeklinde üç parametre varken; kesir dereceli PID (FOPID) kontrolörlerde ise beş parametre (K_P , K_I, K_D, λ, μ) mevcuttur. Klasik PID kontrolörlerden farklı olarak integral ve türev etkilerinin kesir dereceli kuvvetleri de mevcuttur. FOPID kontrolörün zaman düzlemindeki fonksiyonu denklem (2.14)'te verilmiştir.

$$u(t) = K_P\, e(t) + K_I D^{-\lambda} e(t) + K_D\, D^{\mu} e(t) \quad (2.14)$$

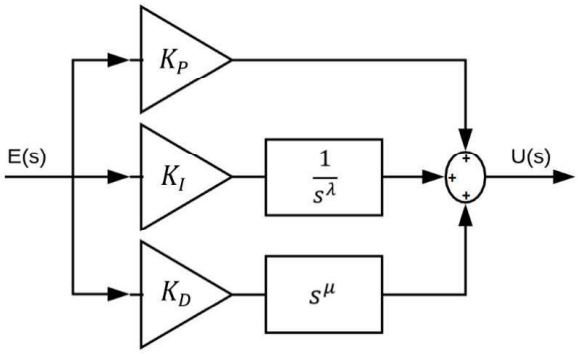
Eşitlik (2.14)'de "λ" ve "μ" parametreleri sırasıyla türev ve integral terimlerinin kesirli sayılar da olabilen, $\lambda \in \mathbb{R}$, $\mu \in \mathbb{R}$ koşullarını sağlayan kuvvetleridir.

FOPID kontrolörün s-düzlemindeki transfer fonksiyonu eşitlik (2.15)'de verilmiştir.

$$\frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s^{\lambda}} + K_D\, s^{\mu} \quad (2.15)$$

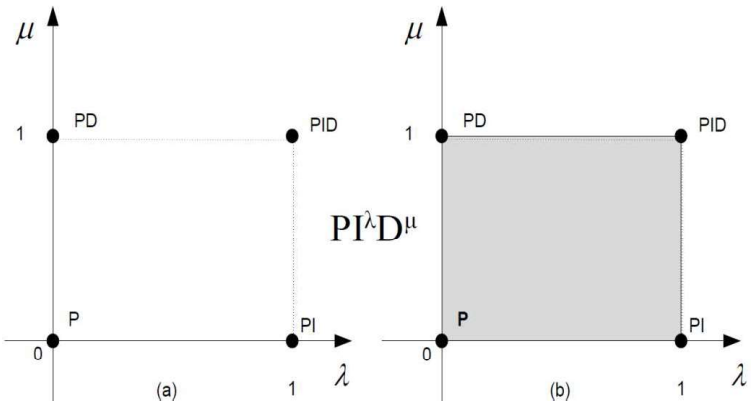
Klasik PID kontrolörlerde "λ" ve "μ" parametrelerinin her ikisi de 1'e eşitken FOPID kontrolörlerde bu parametreler kesirli sayılar da dahil olmak üzere farklı sayılar da olabilirler.

Şekil 2.4'te FOPID kontrolörün blok diyagramı verilmiştir.



Şekil 2.4: s-düzleminde FOPID kontrolör.

FOPID kontrolörlerin klasik PID kontrolörlerden daha hassas ve dayanıklı olmasının temel sebebi transfer fonksiyonunda bulunan 5 parametrenin daha hassas optimizasyona imkan tanımasıdır. "λ" ve "μ" parametrelerinin kesir dereceli sayılar olabilmesi, daha geniş bir parametre uzayı oluşturmakta ve bu iki parametrenin kesir sayılı değerler alabilmesi hassasiyet ve dayanıklılık ayarlarının daha titiz bir şekilde yapılabilmesine imkan sağlamaktadır. Şekil 2.5'te verilmiş olan klasik PID ve FOPID türev-integral etki kuvvet haritaları (λ=[0,1] ve μ=[0,1]) incelendiğinde, Şekil 2.5(a)'da "λ" ve "μ" değerlerinin 0 veya 1 tamsayılarının olabileceği, Şekil 2.5(b) incelendiğinde ise "λ" ve "μ" değerlerinin ek olarak 0 ile 1 arasındaki kesir dereceli sayıları da alabileceği görülebilmektedir. Şekil 2.5(b)'de koyu renkli taralı alan "λ" ve "μ"nün alabileceği değerleri simgelemekte, Şekil 2.5(a)'deki koyu renkli dört nokta ise klasik (tam sayılı) PID tip kontrolörlerde "λ" ve "μ"nün alabileceği değerleri simgelemektedir. Ancak parametre sayısının artması arama uzayını genişlettiği için, FOPID kontrolörlerde parametrelerin belirlenmesini ve optimizasyonunu zorlaştırmaktadır.



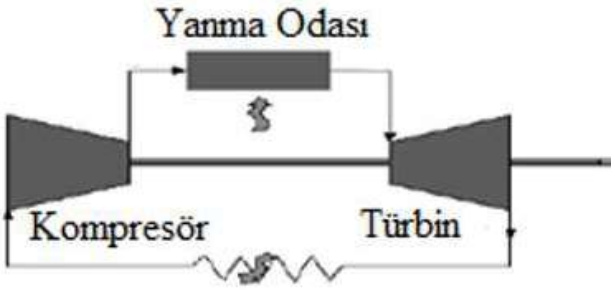
Şekil 2.5: (a) PID türev-integral kuvvet haritası (b) FOPID türev-integral kuvvet haritası [44].

3 MALZEME VE YÖNTEM

3.1 GAZ TÜRBİNLİ GÜÇ SANTRAL MODELİ

Gaz türbinleri enerji üretiminde önemli yer tutmaktadır. Dünya genelinde gaz kaynaklı elektrik üretimi % 22 seviyesindedir [45]. Enerji üretimindeki en yüksek payı kömürün ardından gaz kaynaklı üretim almaktadır. PID kontrolörlerin kullanıldığı yerlerden biri de gaz türbinli güç santralleridir ve bu tezde de basitleştirilmiş bir gaz türbinli güç santrali modeli ele alınacaktır [38, 39, 46, 47].

Genel gaz türbinli santral çalışma prensibi şu şekilde ifade edilebilir; Şekil 3.1'da akış şeması verilmiş olan basit gaz türbini çevriminde kompresöre giriş yapan gaz sıkıştırılarak yanma odasına aktarılır. Yanma odasında yakıt ile karıştırılan gaz ateşlenir ve sıcaklığın etkisi ile gazın genleşmesi sağlanır. Genleşerek hacmi artan gaz, gaz türbinine doğru yönlendirilerek türbinin hareketi sağlanır. Türbin milinde elde edilen işin bir kısmı kompresör tarafından bir kısmı da aktarıcı organlar tarafından absorbe edilir. Geriye kalan ise üretilen net işe eşittir ve türbin mili ile açısal hız olarak jeneratöre aktarılır ve elektrik enerjisi üretilir.



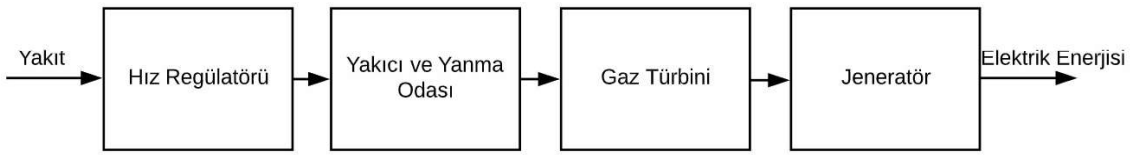
Şekil 3.1: Basit gaz türbini çevrimi [38, 39, 46, 47].

Gaz türbinlerinde en basit modelleme hava standardı kabulü ile mümkündür. Bu tezde Çiftkaya ve Tiryaki'nin çalışmalarında incelediği basitleştirilmiş gaz türbinli güç santral modeli kullanılmış ve hava standardı kabul edilmiştir [38, 46]. Akışkan gaz olarak hava kullanıldığı ve santral çevrimi süresince ideal gaz özelliklerini taşıdığı kabul edilmiştir. Yanma işlemi göz ardı edilerek, çevrime katılan akışkanın dışarıdan bir ısı kaynağı ile sabit basınçta ısıtıldığı varsayılmıştır. Türbinden çıkan gaz ve kompresörün absorbe işlemleri ihmal edilmiş ve türbinden çıkan egzoz gazının sabit basınçta sistem dışındaki bir kaynağa ısı vermesi sonrasında kompresöre tekrar giriş yaptığı kabul edilmiştir.

Belirtildiği şekilde hava standartları kabulü ile meydana getirilen döngülere ideal hava çevrimi denir. Bu kabullerin yapıldığı çevrimlerde aynı zamanda farklı kabuller de yapılabilir. Örneğin; hal değişimlerinin tersinir olduğu kabul edilmiştir. Ayrıca hesaplamaların kolay yapılabilmesi ve yakın sonuçlar elde edilebilmesi için hal değişimleri sırasında özgül ısıların oda sıcaklığı değerlerindeki gibi sabit kaldığı kabul edilmiştir [38, 39, 46, 47]. Bu şekilde belirtilen şartlarda oluşturulan model, soğuk hava standartları kullanılarak oluşturulan model şeklinde adlandırılır.

Bu tezde ele alınmış olan santral modelinin iki ana ünitesinden biri gaz türbini ve bağlantı sistemleri iken ikincisi jeneratördür [38, 39, 46, 47]. Gaz türbini ve bağlantı sistemleri ise üç alt bölümden oluşur; hız regülatörü, yakıcı ve yanma odası, gaz türbini.

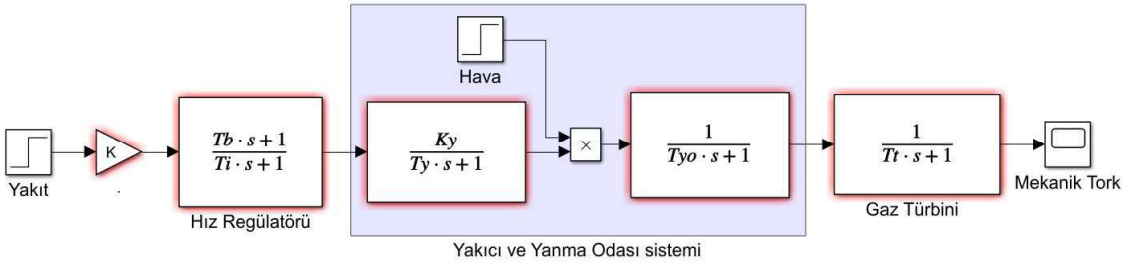
Şekil 3.2'de ele alınan basitleştirilmiş santral modeli diyagramı verilmiştir.



Şekil 3.2: Basitleştirilmiş santral modeli

Hız regülatörü, türbin dönüş hızının kontrolünü sağlar. Enerji kullanımındaki değişimler sebebiyle sistem yük tarafında değişiklikler olacaktır. Bu değişim sistem üzerinde türbin dönüş hızını etkiler. Dönüş hızı etkilendiğinde ise jeneratör ile yük arasında frekans farklılıkları oluşmaya başlar. Bu istenmeyen durum olan frekans değişimini engellemek için türbin dönüş frekansındaki değişime göre üretimin arttırılması ya da azaltılması gerekir. Aksi taktirde frekans sabit tutulamaz. Hız regülatörü yakıt sistemini kontrol eder ve ateşlemeyi jeneratör frekansını sabit tutacak şekilde düzenler [39, 47, 48, 49]. Böylece türbin hız kontrolü sağlanmış olur. Gaz türbini çıktısı olarak mekanik torka dönüşmüş olan enerji, jeneratör vasıtası ile elektrik enerjisine dönüşür.

Yukarıda bahsedilen varsayımlar, ihmaller ve konu edinilen bilgiler doğrultusunda, genelleştirilmiş basit gaz türbin modeli Şekil 3.3 'deki gibi gösterilebilir [38, 39, 46, 47].



Şekil 3.3: Basitleştirilmiş gaz türbin modeli [38, 39, 46, 47,49].

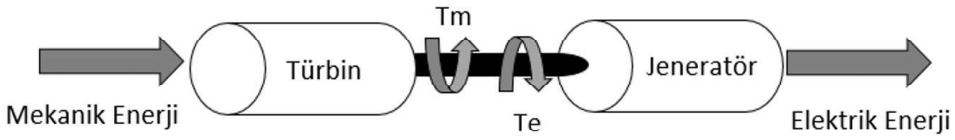
Şekil 3.3'deki modelde kullanılmış olan parametreler ve değerleri Tablo 3.1'de verilmiştir.

Tablo 3.1: Basitleştirilmiş gaz türbin modeli parametre değerleri [38, 39, 46, 47,49].

Parametre	Değer	Açıklama
K	1	Hız regülatörü kazancı
Tb	0.65	Hız düzenleyicisi birim zaman (gecikme) sabit değeri (saniye)
Ti	0.1	Hız düzenleyicisi ikincil zaman (gecikme) sabit değeri (saniye)
Ky	1	Yakıcı sistemin kazanç sabit değeri
Ty	0.05	Yakıcı sistemin zaman (gecikme) sabit değeri (saniye)
Tyo	0.4	Yanma odası zaman (gecikme) sabit değeri (saniye)
Tt	0.1	Türbin dinamik özelliklerine bağlı zaman (gecikme) sabit değeri (saniye)

Şekil 3.3'deki diyagramda görülebilen sistem blokları, bir çalışma sırasında İstanbul ilimizin Avcılar semtinde bulunan kombine çevrim doğalgaz santralinden alınan deneysel bilgiye göre türbin dönüşünü etkileyen en önemli bölümler (hız regülatörü, yakıcı ve yanma odası, gaz türbini) olarak belirtilmiştir [38, 48] ve bu sistem bloklarındaki Tablo 3.1'de verilmiş olan parametre değerlerinin türbin dönüş hızına etkisinin büyük olduğu anlaşılmıştır. Bu parametreler haricindekiler, türbin dönüş hızına etkileri minimal oldukları için deneysel çalışmalarda kullanılmak amaçlı modelleme sırasında ihmal edilerek basitleştirilmiş gaz türbin modeli elde edilmiştir [38, 39, 46, 47,49]. Türbin bölümünde üretilmiş olan mekanik tork mil aracılığıyla jeneratöre aktarılır ve elektrik enerjisine dönüştürülür.

Şekil 3.4'de türbin ve jeneratör arasındaki mekanik aktarım ilişkisi modellenmiştir. Burada T_m mekanik döndürme momentini, T_e ise elektriksel döndürme momentini ifade etmektedir.



Şekil 3.4: Türbin-Jeneratör ilişkisi [38, 39, 46, 47,49].

Türbin sistemi ile yakılan yakıttan elde edilen enerji hızlandırıcı bir etki gösterir ve T_m artar. Jeneratörün beslemekte olduğu hattaki tüketimin artması ise yavaşlatıcı etki gösterir ve T_e'nin artmasına sebep olur. Bu durumda $T_e > T_m$ olduğunda santral ortak mili yavaşlama eğilimi gösterir ve frekansın azalması söz konusu olur. Frekansın sabit tutulması için yakıt tüketiminin arttırılarak T_m'nin artması ve T_e ile eşitlenerek dengenin sağlanması gerekir. $T_m < T_e$ olduğu durumlarda ise yakıt tüketiminin azaltılarak T_m'nin azalması ve T_e ile eşitlenerek kuvvet dengesinin sağlanması frekansın korunması için gereklidir. Bu işlemler santralin işleyişi süresince defalarca tekrarlanır. Bu denetleme işleminin santralin kontrol altında tutulması ve üretilen elektriğin sabit frekansta kalması için gereklidir. Bu noktada türbin ile jeneratör arasındaki ilişkiyi ifade eden denklemler aşağıda verilmiştir.

$$T_{net} = T_m - T_e = I\,\alpha \tag{3.1}$$

$$M = I.\omega \tag{3.2}$$

$$\omega = \omega_0 + \Delta\omega = \omega_0 + \alpha.t \tag{3.3}$$

$$\Delta\omega = \alpha.t \tag{3.4}$$

Burada;

T_{net} : net ivmelendirme momenti (tork)

I : santral eylemsizlik momenti

α: açısal ivme

M: açısal momentum

ω: açısal hız

P_m: mekanik güç

P_e: elektriksel güç

P_{net} : net gücü

Δω : t süredeki açısal hız değişimi

olarak tanımlanmıştır.

Mekanik ve elektriksel güç dengesinin türbin dönüş hızı ile ilişkisi, aşağıdaki denklemlerden çıkarılabilir.

$$P_{net} = P_m - P_e \tag{3.5}$$

$$P_{net} = \omega \,.\, T_{net} = \omega \,.\, (I \,.\, \alpha) = M \,.\, \alpha \tag{3.6}$$

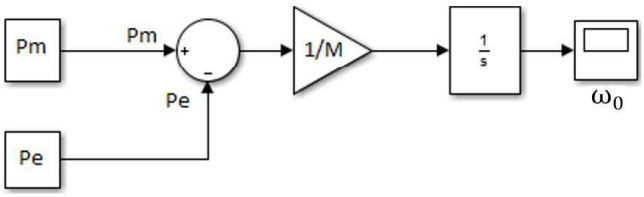
$$P_{net} = M \,.\, \alpha = M.\frac{d\omega_0}{dt} \tag{3.7}$$

$$P_{net} = P_m - P_e = M.s\,\omega_0 \tag{3.8}$$

Denklem (3.7)'nin Laplace dönüşümü sonrasında Denklem (3.8) elde edilir ve s-düzleminde açısal hız değişiminin matematiksel ifadesi Denklem (3.9)'daki gibi bulurnur.

$$\omega_0 = (P_m - P_e)\frac{1}{Ms} \tag{3.9}$$

Bu denkleme göre sistemin model diyagramı Şekil 3.5'te verilmiştir.

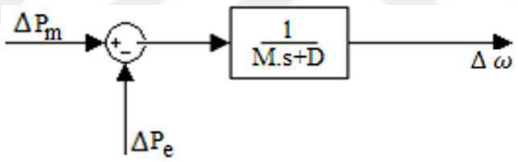


Şekil 3.5: Türbin-Jeneratör güç aktarım modeli [38, 39, 46, 47,49].

Yükteki değişimlerin açısal hıza etkisini matematiksel olarak ifade ettiğimizde denklem (3.10) elde edilir. ΔP_L , yükteki güç değişimini ifade etmektedir.

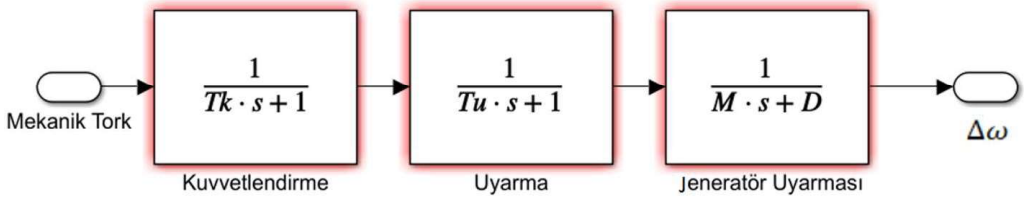
$$\Delta P_L = \mathrm{D}.\Delta\omega \tag{3.10}$$

Sönüm sabiti (D), yükteki değişimin yüzdelik ifadesinin açısal hızda oluşturduğu değişimin yüzdelik ifadesine oranını temsil etmektedir. Örnek olarak; yük güç seviyesinin %1'lik artışı açısal hıza %1 olarak yansıyabilir. Bu durumda D=1 olacaktır. Şekil 3.6'da D'nin sisteme etkisini gösteren blok diyagramı gösterilmiştir.



Şekil 3.6: Jeneratör uyarılma blok diyagramı [39, 46, 47,49].

Sönüm sabitinin de blok diyagramına eklenmesi ile jeneratör uyarılması basitçe ifade edilmiş olur. Buna ek olarak jeneratör uyarılmasının önüne gerilim düzenleyicisinin kuvvetlendirme ile uyarma blokları da eklendiğinde Şekil 3.7'de verilmiş olan blok diyagramı elde edilir.



Şekil 3.7: Açısal hız değişimi çıkışlı jeneratör modeli [39, 46, 47,49].

Bu çalışmada amaçlanan güç kontrol çalışması doğrultusunda model çıkışının güç olması gerekecektir. Bu doğrultuda denklem (3.11)'de verilen eşitlik ile H hız regülasyon sabiti sistem modeline eklenir.

$$H = \frac{\Delta\omega}{P_{jn}} = \frac{\Delta f}{P_{jn}} = \frac{f_2 - f_1}{P_{jn}} \quad (3.11)$$

Eşitlik (3.11)'de;

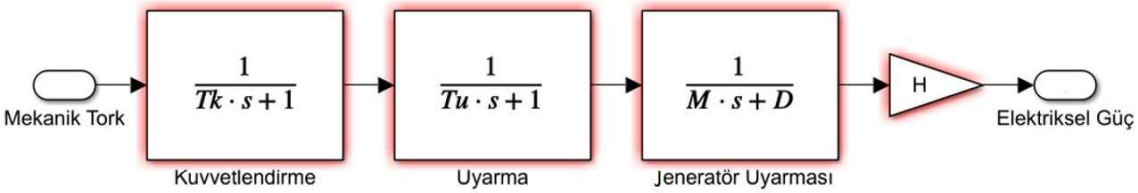
P_{jn}: Jeneratör nominal aktif çıkış gücü (MW)

f_2: Yüksüz durumda frekans (Hz)

f_1: Nominal güçte frekans (Hz)

H: Hız regülasyon sabiti (Hz/MW)

göstermektedir. Burada H, Şekil 3.7'deki modelde sistem çıkışına yerleştirilirse, çıkışı aktif güç olan Şekil 3.8'deki model elde edilir.



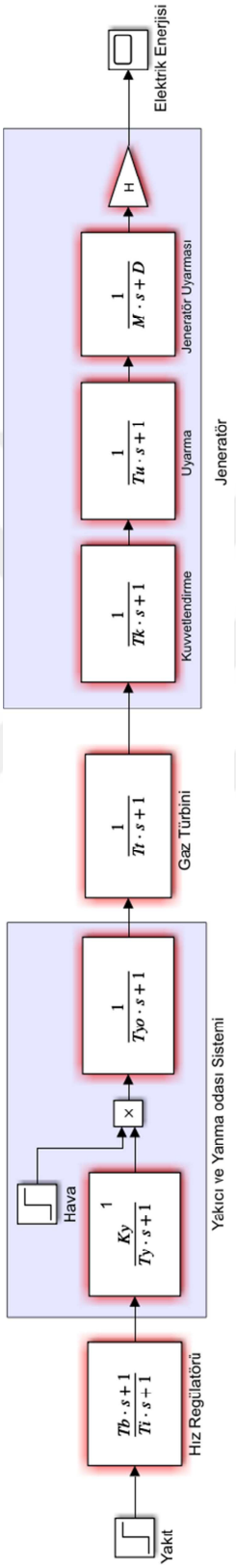
Şekil 3.8: Basitleştirilmiş jeneratör modeli [39, 46, 47,49] .

Şekil 3.8'daki jeneratör modelinde kullanılmış olan parametreler ve kabul değerleri Tablo 3.2'de verilmiştir [50, 51].

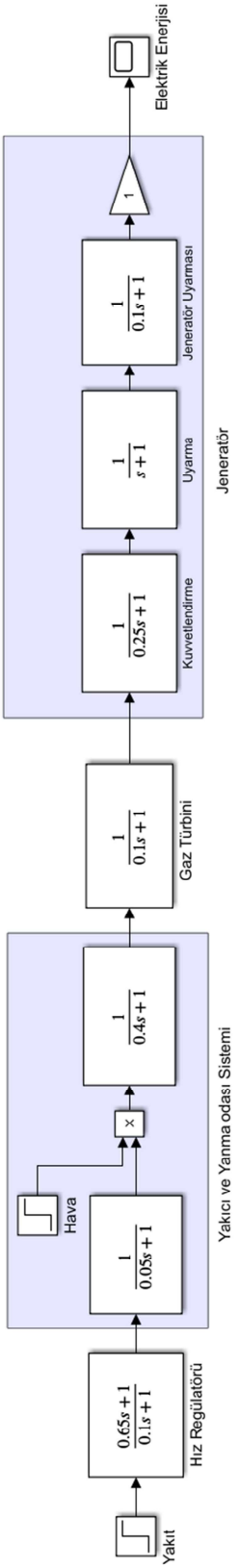
Tablo 3.2: Basitleştirilmiş jeneratör modeli parametre değerleri [39, 46, 47,49].

Parametre	Değer	Açıklama
Tk	0.25	Kuvvetlendirme zaman (gecikme) sabit değeri (saniye)
Tu	1	Uyarma zaman (gecikme) sabit değeri (saniye)
M	0.1	Jeneratör zaman (gecikme) sabit değeri (saniye)
D	1	Sönüm sabit değeri
H	1	Hız düzenleyici sabit değeri veya kayması (Hz/MW)

Şekil 3.3 ve Şekil 3.8'de sırasıyla verilmiş olan parametrik türbin ve jeneratör modellerinin birleştirilmesi ile elde edilen basitleştirilmiş genel santral modeli Şekil 3.9'da verilmiştir. Tablo 3.2 ve Tablo 3.3'te verilmiş olan parametre değerleri Şekil 3.9'da verilmiş olan genel santral modelinde yerleştirildiğinde Şekil 3.10 elde edilir.



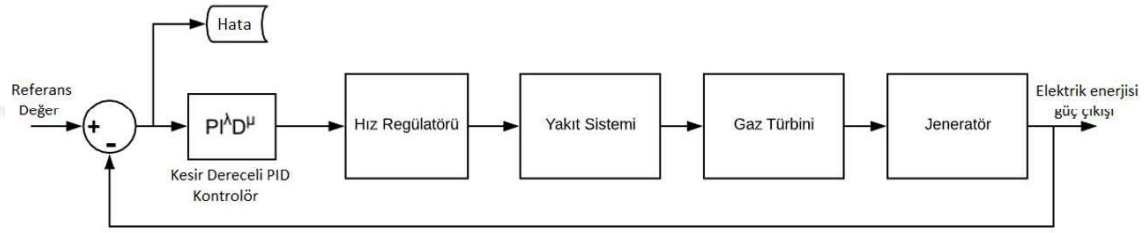
Şekil 3.9: Genel Santral Modeli [39, 46, 47,49].



Şekil 3.10: Santral Modeli [39, 46, 47,49].

3.2 KESİR DERECELİ PID KONTROLÖRÜN SANTRAL MODELİNE UYGULANMASI

Bu tezde Matlab standart uygulama eklentilerine [52] ilave olarak Tepljakov ve diğerlerinin [53,54] hazırlamış olduğu FOMCON toolbox kullanılmıştır. Şekil 3.2'de verilen basitleştirilmiş santral modeline FOPID kontrolör uygulanmasıyla elde edilen yeni sistem modeli Şekil 3.11'de verilmiştir.



Şekil 3.11: FOPID kontrolör uygulanmış yeni santral modeli.

3.3 METASEZGİSEL OPTİMİZASYON ALGORİTMALARI

Teknolojinin gelişmesi ve araştırmaların eskiye kıyasla daha hızlı ilerlemesi sebebiyle karmaşık sistemleri optimize etme gereksinimi her geçen gün artmaktadır. Sistemlerin karmaşıklık seviyesi arttıkça optimizasyon işlemi de o kadar zorlaşmaktadır. Bu tez kapsamında kullanılan FOPID kontrolörler 5 parametreye sahiptir ve bu parametrelerin geleneksel klasik yöntemler ile hesaplanabilmesi kolay değildir. Bunun yanında bazı durumlarda kesin sonuçlara ulaşmak mümkün olmayabilir ve bu durumlarda yaklaşım yöntemlerinin kullanılması daha iyi sonuçlar verebilmektedir. Parametre optimizasyonu için kullanılan yaklaşım yöntemlerinin önde gelenleri metasezgisel algoritmalardır. Bu algoritmalar, bir amacı gerçekleştirmek ya da istenen sonuca ulaşmak için alternatif yöntemleri belli bir kural dizilimine uyarak deneyen ve hedefe en çok yaklaşan çözümü bulmaya yarayan bilgisayar destekli yöntemlerdir. Bunlar probleme özel, kesin sonuca varılan çözümler geliştirmez; ancak genel yöntemi uygulayarak çözüme yakın sonuçlar verirler. Böyle optimizasyon algoritmaları, basit ve anlaşılır olması, hızlı sonuç vermesi gibi bazı avantajlarından dolayı günümüzde yaygın kullanılan yöntemlerdendir.

Bu algoritmalar genelde doğadaki yapılardan esinlenilerek geliştirilmiştir. Örneğin bunlardan biri, sürü algoritmalarıdır. Sürü halinde yaşayan canlılardan esinlenilerek geliştirilmişlerdir. Şekil 3.12'de bazı temel algoritma kategorileri gösterilmiştir. Esinlenilen kaynağa göre genel sınıflandırma aşağıdaki şekildeki gibi yapılabilir [27].



Şekil 3.12: Sezgisel algoritma kategorileri.

Son yıllarda sezgisel algoritmalar ile ilgili yapılan çalışmalar incelendiğinde bunlardan bazılarının performansları ile öne çıktığı görülebilmektedir. Bu tezde bilim dünyasında kabul görmüş ve dikkat çeken metasezgisel algoritmalardan PSO, ABC, GWO, MFA, WOA, SSA, SCO, ASO Şekil 3.11'deki santral modeli üzerinde uygulanarak değerlendirmeler yapılmıştır [28-34][55, 56]. Bu optimizasyon algoritmaları popülasyon tabanlı yöntemler izler. Popülasyonun ve popülasyon üyelerinin her birinin sosyal davranışları benzetimler ile matematiksel olarak ifade edilerek optimizasyon sorunlarına çözüm olarak sunulmuştur. Örneğin; hayvan tabanlı algoritmalarda hayvan sürüleri popülasyonu temsil ederken popülasyon üyesi olan her bir hayvan ise algoritmalarda partikül (parçacık) olarak kabul edilen bireyleri temsil eder. Popülasyon üyeleri D boyutlu olduğu kabul edilen arama uzayı içerisinde hareket ettirilerek istenilen sonuca en yakın sonuçları veren partikül (parçacık) konumu tespit edilmeye çalışılır. Burada D, optimize edilecek parametre sayısıdır.

FOPID kontrolör transfer fonksiyonunda K_P, K_I , K_D, λ, μ olmak üzere beş parametre mevcuttur. Bu sebeple optimizasyon algoritmalarının uygulanışı sırasında 5 boyutlu bir arama uzayı oluşturulmuş ve bu arama uzayının sınırları algoritmalar uygulanmadan önce belirlenmiştir. Bu tezde algoritmaların kullanım amacı, sınırları başlangıçta belirlenmiş olan beş boyutlu bir arama uzayı içerisinde istenilene en yakın sonuçları veren parametre değerlerini belirlemektir.

Popülasyon bireylerinin (partikül, parçacık) hareketleri popülasyonun türüne göre çeşitlilik gösterebilmektedir ve buna bağlı olarak farklı popülasyon tabanlı algoritmalar ortaya koyulabilmiştir. Optimizasyon algoritmalarının istenilen sonuca yakınlaşma, sonuca ulaşma süresi gibi değerlendirme kriterleri, parçacıkların hareketlerine bağlıdır. Algoritmalarda bireylerin davranışsal hareketlerinin matematiksel benzetimi yapılmış olan her bir parçacığın konumu (X_t) ve hızı (V_t) vardır. Bu parametrelerin algoritma çalışma süresince matematiksel hareket benzetimleri doğrultusunda belirlenen yeni değerleri algoritma başarısı üzerinde direkt olarak etkilidir.

3.3.1 Parçacık Sürüsü Optimizasyon Algoritması (PSO)

Parçacık sürü optimizasyonu (Particle Swarm Optimization, PSO) algoritması kuş sürülerinden esinlenilerek geliştirilmiştir ve popülasyon tabanlı bir yöntemdir [28]. Bu algoritmada kuşların sürü içerisindeki sosyal davranış ve hareketlerinin benzetiminden faydalanılarak kuş sürüsünün popülasyonu, bir kuşun da bir parçacığı temsil ettiği kabul edilmiştir [57]. Şekil 3.13'da bir kuş sürüsünün hareketi sırasındaki fotoğrafı verilmiştir.



Şekil 3.13: Kuş Sürüsü

PSO algoritması başlangıcında n adet parçacıktan oluşan bir popülasyonda her bir parçacığın başlangıç konumu (X_0^i) ve başlangıç hızı (V_0^i) vardır. Başlangıç konumları sınırları belirlenmiş olan arama uzayı içerisinden rastgele olarak belirlenir. Bu konumlar optimize edilen fonksiyona uygulanarak istenilen sonuca yakınlıkları hesaplanır. Ardından benzetim yolu ile belirlenmiş yöntem yardımıyla parçacık hızları hesaplanır ve parçacıkların arama uzayı içerisinde hareket etmesi sağlanır. Parçacıkların bu hareketleri sırasında her bir parçacığın rastladığı, istenilen sonuca en yakın değerleri veren konumu barındıran her bireye özel bir değişken $X_t^{i,pbest}$ mevcuttur. Global değişken olarak kabul edilen X_t^{gbest}, popülasyon içerisindeki bütün bireyler arasında rastlanmış olan en iyi sonucu veren konum değerini barındırır. Bu bilgiler doğrultusunda $i \in [1,\ldots,n]$ ve $n>1$ olduğu durumda i numaralı parçacığın t anındaki (t iterasyonundaki) hızı V_t^i, konumu X_t^i aşağıdaki şekilde hesaplanır [58-60]. Buna göre aşağıdaki ifadeler yazılabilir.

$$V_{t+1}^i = V_t^i + c1 * \text{rand}(\quad) * \left(X_t^{i,pbest} - X_t^i\right) + c2 * \text{rand}(\quad) * \left(X_t^{gbest} - X_t^i\right) \tag{3.12}$$

$$X_{t+1}^i = X_t^i + V_{t+1}^i \tag{3.13}$$

Denklem (3.12)'de rand() fonksiyonu [0,1] aralığında rastgele bir sayı üretmektedir. c1 ve c2 hız değişkenlerinin kuvvet çarpanlarıdır. Konumların güncellenme işlemi maksimum deneme

sayısı (iterasyon sayısı) kadar tekrarlanır ve sonuç olarak en iyi neticeleri veren (istenilen sonuca en yakın değerleri veren) parçacık konumu X^{gbest} bulunmuş olur.

$X_t^{i,pbest}$'in belirlenmesinde kullanılan koşul, denklem (3.14)'de verilmiştir. Burada *f*() ile ifade edilen amaç fonksiyonu girdi değerleri doğrultusunda fonksiyonun ürettiği değerler konumların uygunluklarını (fitness) ifade eder. Uygunluk (fitness) değeri düşük olan konum istenilen sonuca daha yakın çıktı değerleri verdiği anlamına gelir.

$$X_t^{i,pbest} = \begin{cases} X_t^{i,pbest}, & eğer\ f(X_t^i) > f(X_t^{i,pbest}) \quad f: \mathbb{R}^n \to \mathbb{R} \\ X_t^i \quad , & eğer\ f(X_t^i) \leq f(X_t^{i,pbest}) \quad f: \mathbb{R}^n \to \mathbb{R} \end{cases} \tag{3.14}$$

Küresel (global) en iyi değeri veren konumun belirlenme şekli matematiksel olarak denklem (3.15)'de ifade edilmiştir. Bireysel en iyi konumlar arasındaki en küçük uygunluk değerine sahip konum global en iyi konum

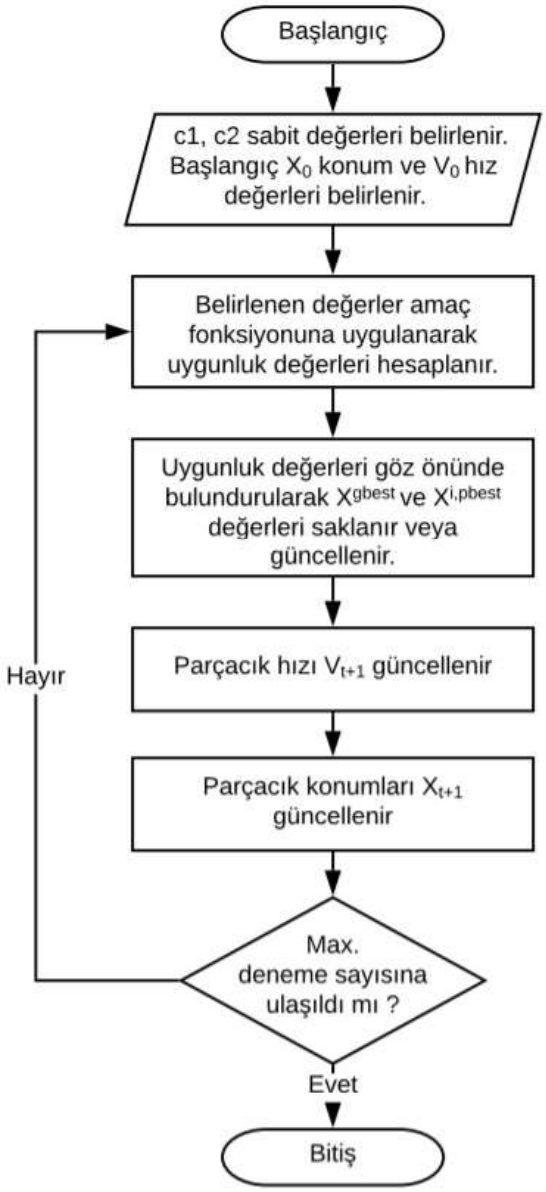
$$X_t^{gbest} = min\{X_t^{i,pbest}\}, \quad i \in [1, \dots, n] \text{ ve } n > 1 \tag{3.15}$$

olarak belirlenir.

PSO algoritmasının sözde kod örneği (Pseudo kod örneği) aşağıda verilmiştir [28].

Parçacık sayısı (n) belirlenir ve konumları rastgele atanır X_0^i *i=1,2,...,n)*
Sabit başlangıç değerleri c1, c2 belirlenir
t=1
while *(t < max. deneme sayısı)*
    for *her parçacık için, n*
        Herbir parçacık için model simulasyonu yapılır
        Uygunluk değerleri (fitness) belirlenir
        if $f(X_t^{i,pbest}) > f(X_t^i)$
            $X_t^{i,pbest} = X_t^i$
            if $f(X_t^{gbest}) > f(X_t^{i,pbest})$
                $X_t^{gbest} = X_t^{i,pbest}$
            end if
        end if
    end for
    V parçacık hız değerleri hesaplanır (3.12)
    X parçacık konumları güncellenir (3.13)
    t=t+1
end while
return X_t^{gbest}

Şekil 3.14'de PSO algoritması akış diyagramı verilmiştir [28].



Şekil 3.14 : PSO Akış Diyagramı [28].

3.3.2 Yapay Arı Kolonisi Optimizasyon Algoritması (ABC)

Yapay Arı Kolonisi algoritması (Artificial Bee Colony, ABC) arıların koloni olarak içgüdüsel yemek arama davranışlarından esinlenilerek Karaboğa tarafından geliştirilmiş bir optimizasyon algoritmasıdır [29]. Sürü zekasına dayanan bu algoritma, optimizasyon problemlerini çözmek için kullanılan bilgisayar ortamında geliştirilmiş bir yöntemdir. Şekil 3.15'da arıların kovan içerisindeki hareketlerine dair bir fotoğraf verilmiştir.



Şekil 3.15: Arılar

Arı kolonileri popülasyon içerisinde hiyerarşik bir yapıya dayanan görev dağılımlarına sahiptirler. Arı kolonilerinde 3 tip arı vardır; kraliçe arı, erkek arılar ve işçi arılar. Kolonideki birçok iş işçi arılar tarafından üstlenilir. Bu yapıdan esinlenen ABC algoritması üç ana bölümden oluşur [61-64];

- Arıların yani parçacıkların kovan çevresinde dolaşarak besin kaynaklarını tespit etmeleri gerekir. Arılar kolonisi için bir besin kaynağının kalitesi, besinin kovana uzaklığı ve besinin çıkartılması için gerekli uğraşa bağlıdır.
- Görevli işçi arılar, kovan çevresinde bulunmuş olan besin kaynaklarından besinin kovana taşınmasından sorumludurlar ve bu işlem sırasında kovana geldiklerinde besin kaynağı hakkındaki bilgileri kovandaki gözcü arılara aktarırlar.
- Kolonideki herhangi bir iş üstlenmemiş arılar gözcü ve izci işçi arılar olmak üzere ikiye ayrılır. Gözcü işçi arılar besin getiren işçi arıları izleyerek onlardan aldıkları bilgiler doğrultusunda besin taşıma işini üstlenmek için besin kaynağına yönelirler, izci işçi arılar ise civarda dolaşarak yeni besin kaynakları keşfetmeye çalışırlar.

ABC algoritmasının ilk aşamasında kovan çevresindeki besinleri temsilen başlangıç besin konumları atanır. Bu işlem denklem (3.16) kullanılarak yapılır.

$$X_0^i = X_{min} + \text{rand}()(X_{max} - X_{min}), \quad i \in [1, \dots, n] \text{ ve } n > 1 \tag{3.16}$$

Burada X_0^i, belirlenmiş olan arama uzayı içerisindeki n adet parçacık (besin) konumunu ifade eder. X_{min} arama uzayının alt sınırlarını ve X_{max} arama uzayının üst sınırlarını ifade eder. rand() fonksiyonu ise [0,1] aralığında eşit olasılıklara sahip rastgele sayıları ifade eder. Belirlenmiş olan besin konumlarının uygunluk (fitness) değerleri hesaplanır

ABC algoritmasının ikinci aşamasında belirlenmiş olan besinlerin yakınlarına aşağıdaki denklem (3.17) kullanılarak görevli işçi arılar yönlendirilir.

$$X_{t+1}^i = X_t^i + \emptyset_i * (X_t^i - X_t^k) \tag{3.17}$$

X_{t+1}^i , besinin komşu konumunu ve X_t^i değişkeninin yeni değerini sembolize etmektedir. X_t^i mevcut besin konumu; X_t^k bilinen besin kaynaklarından rastgele birinin konumudur. $\emptyset_i$ ise [-1,1] aralığında rastgele değer alan ağırlık katsayısıdır. Besinlere yollanan görevli işçi arılardan alınan bilgi doğrultusunda aşağıdaki denklem (3.18) ile besinlerin olasılık değerleri hesaplanır. Bu değer daha sonra gözcü arılar tarafından kullanılacaktır.

$$p_i = \frac{fit_i}{\sum_{i=1}^{n} fit_i}, \qquad i \in [1, \dots, n] \text{ ve } n > 1 \tag{3.18}$$

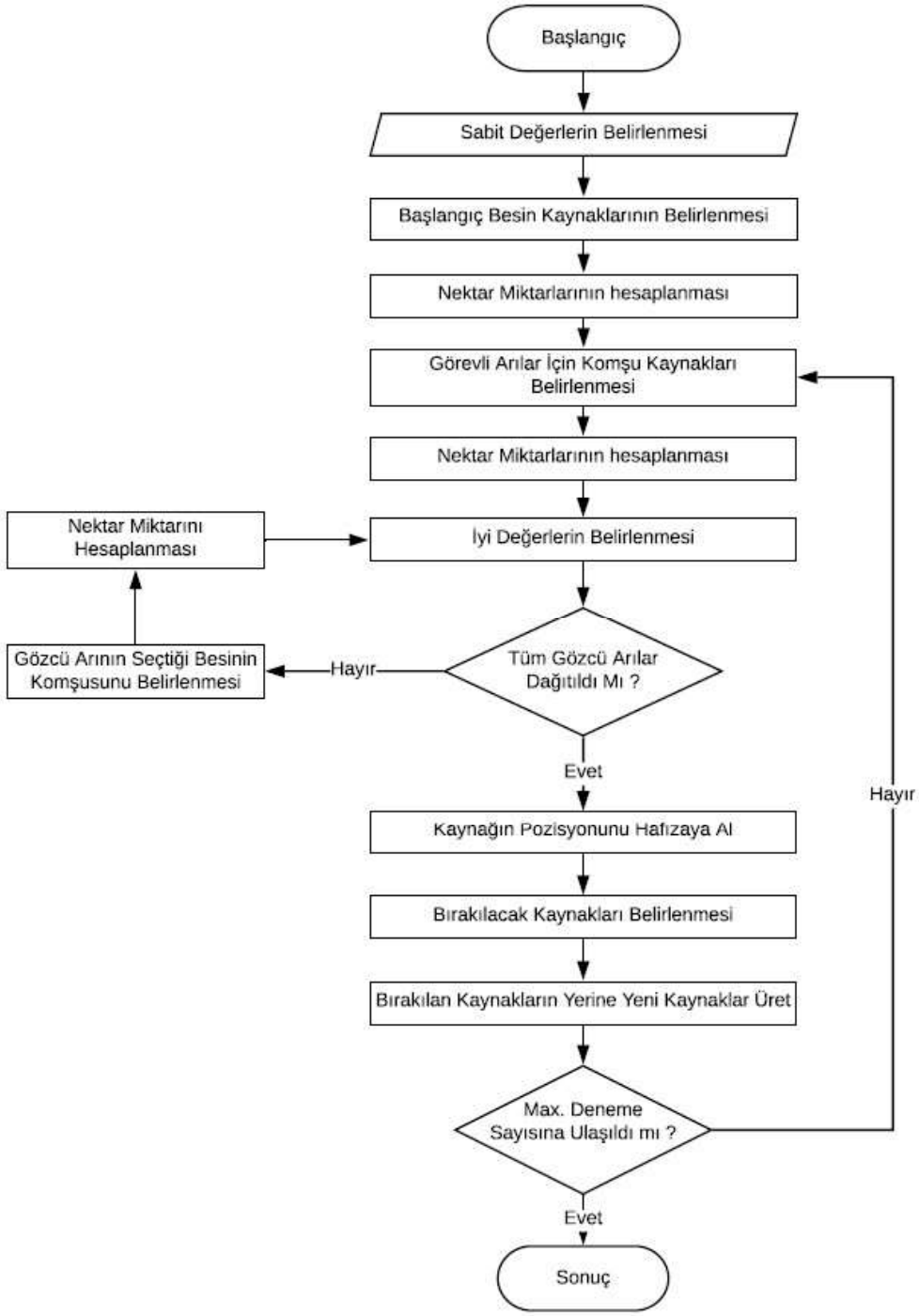
Burada fit_i , i numaralı besinin uygunluk (fitness) değerini, p_i ise i numaralı besinin olasılık değerini ifade etmektedir. Bu aşamada aynı zamanda her bir arı için bir sayaç oluşturulur ve bir arı eğer yollandığı konumda bir önceki gittiği konumdan daha çok besin (uygunluk) elde etmediyse sayaç değeri 1 arttırılır.

Algoritmanın üçüncü bölümüne gelindiğinde; gözcü işçi arılar önceden belirlenmiş olan besin olasılık değerleri p_i göz önünde bulundurularak, olasılık değeri düşük olana az görevli işçi arı yollayacak şekilde görevli arıları denklem (3.17) kullanarak yönlendirir. İzci işçi arılar ise her bir arı için başlatılmış olan sayaçların belirlenmiş limit (L) değere ulaşmasıyla ortaya çıkar. Bir kaynak çevresinde dolaşarak başlangıçtaki uygunluk değerinden daha iyi bir sonuç elde edememesi ile sayaç değerinin limit değeri aşması durumunda bu besin kaynağının tükendiği var sayılır ve işçi arılar kaynak konumunu unutarak denklem (3.16) yardımıyla arama uzayında rastgele dolaşır. Bir besin kaynağı bulduğunda tekrar görevli işçi arıya dönüşerek besin taşıma görevine döner. Bu işlem max. deneme sayısı (iterasyon sayısı) kadar tekrarlanır.

ABC algoritması sözlü kod (pseudo kod) örneği aşağıdaki gibidir [29].

Arı sayısı belirlenir ve besin konumları rastgele atanır X_0 (i=1,2,...,n)
Limit, parametre sayısı belirlenir, sabit başlangıç değerleri belirlenir
Besin fayda değerleri (fitness) hesaplanır.
t=1
while *(t < max. deneme sayısı)*
/*İşçi arı evresi*/
for *her besin için, n*
Yeni besin kaynağı saptanır, komşular arasından. (3.17)
Uygunluk değerleri hesaplanır, en iyisi seçilir.
end for
Herbir besin kaynağının olasılığı hesaplanır.
/*Gözcü arı evresi*/
for *her besin için, n*
P oranına göre arayıcı arılar *besin kaynaklarına yollanır.*
Bu alanlarda civardaki yeni besin kaynakları bulunur. (3.17)
Yeni besin kaynaklarının uygunluk değerleri hesaplanır. En iyi olan tespit edilir.
end for
/*Kaşif arı evresi*/
if herhangi bir işçi arı, kaşif arı olduğunda
Rastgele bir alana yeni kaynak bulması için yollanır. (3.16)
t=t+1
end while
return "en iyi değere sahip X_i"

ABC algoritması genel işleyişi Şekil 3.16'deki akış diyagramında verilmiştir [29].



Şekil 3.16: ABC Algoritması Akış Diyagramı [29].

3.3.3 Gri Kurt Optimizasyon Algoritması (GWO)

Gri kurt optimizasyon algoritması (Grey Wolf Optimization, GWO) kurt sürülerinden ilham alınarak avlanma davranışlarının matematiksel olarak bilgisayar ortamında benzetimi ile oluşturulmuştur [30].

Gri kurtlar sürü halinde yaşarlar ve sürü içerisindeki davranışları gözlemlendiğinde bireyler arasında hiyerarşik bir yapı olduğu anlaşılır. Bu hiyerarşik yapıya göre lider kurt alfa (α) olarak adlandırılır ve uyku yeri, kalkma zamanı, avlanma gibi konularda sürüye öncülük eder. İkinci seviyedeki kurtlar beta (β) olarak adlandırılır, alfa kurdun yardımcılığını üstlenirler ve alfadan gelen emirlerin sürüye aktarılmasını sağlarlar. Delta (δ) ise üçüncü güçlü yapıdaki kurtlardır ve alfa ile betadan emir alırlar. Delta kurtlar nöbetçilik, hasta bakıcılık gibi işleri üstlenir. Omega (ω) ise sürünün en alt tabaka kurtlarını oluşturur, özel olarak üstlendikleri bir görev yoktur. Şekil 3.17'de görüldüğü gibi yukarıdan aşağıya doğru azalan güçte bir sosyal hiyerarşiye sahiptirler.



Şekil 3.17: Gri kurt sürü hiyerarşisi ve sürü.

Gri kurt optimizasyon algoritması bu yapıyı matematiksel yöntemler ile bilgisayar ortamına taşımış ve bir kurt sürüsünün avını yakalama davranışları benzetimler aracılığı ile matematiksel olarak modellenmiştir. Başlangıçta arama uzayında rastgele konumlandırılan ve uygunluk değerleri (fitness) hesaplanan kurt bireyleri hiyerarşik yapıya uyularak uygunluk değerlerine göre sınıflandırılırlar. Uygunluk değeri (fitness) en iyi olan konum alfa kurt (X_α), ikinci en iyi olan beta kurt (X_β), üçüncü en iyi olan delta kurt (X_δ) olarak belirlenir. Geriye kalan kurtlar (konumlar) omega sınıfına dahildirler.

GWO algoritmasının ikinci evresinde avın sürü tarafından çevrelenmesi benzetimi denklem (3.19), (3.20) ve (3.21) kullanılarak yapılır. Bu sırada liderlik vasfı olan kurtlar öncülüğünde av sarılmış olur.

$$D_a = \left|C_1 X_a - X_t^i\right|, \qquad D_\beta = \left|C_2 X_\beta - X_t^i\right|, \qquad D_\delta = \left|C_3 X_\delta - X_t^i\right| \tag{3.19}$$

$$X_1 = X_a - A_1 D_a, \qquad X_2 = X_\beta - A_2 D_\beta, \qquad X_3 = X_\delta - A_3 D_\delta \tag{3.20}$$

$$X_{t+1}^i = \frac{X_1 + X_2 + X_3}{3} \tag{3.21}$$

Denklem (3.19)'da C_1, C_2, C_3 parametreleri denklem (3.22) yardımıyla bulunur. X_t^i kurdun o anki konumunu ifade eder. D_a, D_β, D_δ parametreleri ise kurdun alfa, beta ve delta kurtlara olan uzaklığının "C" katsayıları ile çarpılmış halini ifade eder. Denklem (3.20)'deki A_1, A_2, A_3 denklem (3.23) ile hesaplanır. Böylece denklem (3.21) kullanılarak kurdun yeni konumu X_{t+1}^i belirlenir.

$$C = 2\,r \tag{3.22}$$

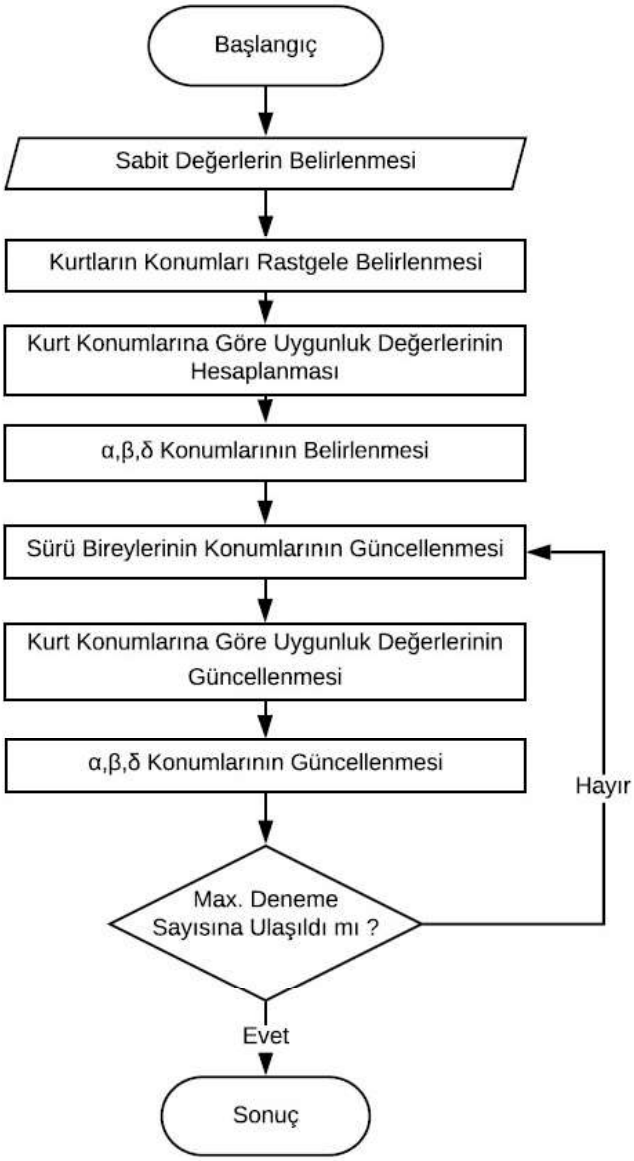
$$A = 2\,a\,r - a \tag{3.23}$$

Burada r, [0,1] aralığında rastgele bir sayıdır. a ise [0,2] aralığında iterasyon sayısı (maksimum deneme sayısı) ile doğrusal orantılı olarak azalan bir sayıdır. Bu durumda A, [-2a, 2a] aralığında değişen rastgele bir değerdir. Eğer |A| < 1 ise kurdun yeni konumu ava yakınlaşır ve bu evre saldırı olarak adlandırılır. Eğer |A| > 1 ise bu durumda kurdun konumu avdan uzaklaşarak yeni av arar ve bu evreye de arama evresi denir [30, 64, 65]. Toplamda dört evreden oluşan bu süreç iterasyon sayısı kadar tekrarlanır ve GWO algoritmasını oluşturur.

Aşağıda GWO algoritması sözde kod (pseudo kod) örneği verilmiştir [30].

Kurt sayısı belirlenir ve konumları rastgele atanır X_i (i=1,2,...,n)
Sabit başlangıç değerleri belirlenir
Kurt konumlarının uygunluk değerleri (fitness) hesaplanır.
Xα=Konum uygunluk değeri en iyi kurt
Xβ=Konum uygunluk değeri ikinci en iyi kurt
Xδ=Konum uygunluk değeri üçüncü en iyi kurt
t=1
while *(t < max. deneme sayısı)*
  for *her kurt için, n*
    Kurt konumları güncellenir. (3.21)
  end for
  a, A *ve* C *parametrelerini güncelle*
  Herbir kurt konumunun uygunluk değerleri hesaplanır.
  Xα, Xβ, Xδ değerleri güncellenir
  t=t+1
end while
return X*a*

GWO algoritmasının genel işleyişini özetleyen akış diyagramı aşağıdaki Şekil 3.18'de verilmiştir [30].



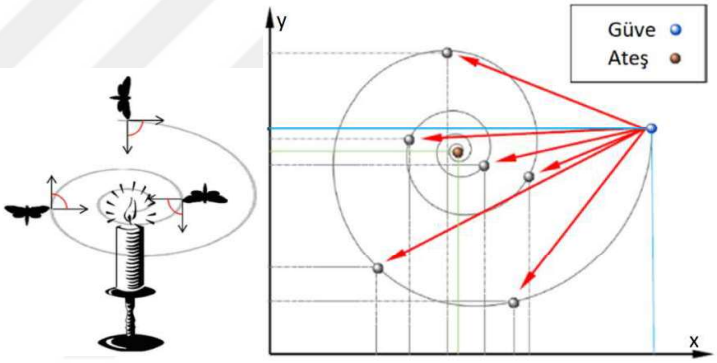
Şekil 3.18: GWO akış diyagramı [30].

3.3.4 Güve-Ateş Optimizasyon Algoritması (MFO)

Güve-ateş optimizasyonu (Moth-Flame optimization, MFO), güve kelebeklerinin ateş etrafında dönen davranışlarından esinlenilerek geliştirilmiştir [31]. Güve kelebekleri ateş kaynağına ulaşmak için çevresinde dönerek spiral şeklinde (hypereliptic) bir yörünge izlerler. Şekil 3.19 de bir güve kelebeği görseli ve Şekil 3.20'de ise bir güve kelebeğinin ateşe yaklaşırken izlediği yol örneklemesi görsel olarak verilmiştir. Güvenin temsili spiral şeklindeki yolu üzerindeki konum vektörleri kırmızı oklar ile gösterilmiştir [67].



Şekil 3.19: Güve Kelebeği.

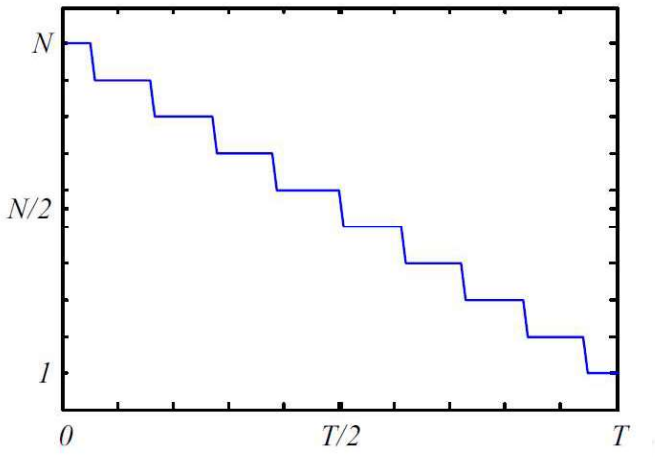


Şekil 3.20: Güve Kelebeği Yörüngesi [67].

Popülasyon temelli bu algoritma başlangıcında öncelikle güve kelebeği sayısı yani popülasyon birey sayısı ve iterasyon sayısı belirlenir. Ardından güvelere arama uzayı sınırları içerisinde rastgele olacak şekilde konum atamaları yapılır. Konumları belirlenen güvelerin amaç fonksiyonuna uygulanmasıyla uygunluk değerleri (fitness) bulunur. Bu aşamada arama uzayı içerisinde bulunan ateş sayısının belirlenmesi denklem (3.24) ile yapılır.

$$Ateş\ Sayısı = round\left((N-l)\frac{N-1}{T}\right) \tag{3.24}$$

Burada round fonksiyonu ondalık sayıları tam sayıya yuvarlama görevini üstlenir, N güve sayısını, l : o anki iterasyon numarasını, T : en büyük iterasyon sayısını temsil eder. Bu denkleme göre iterasyon sayısına bağlı olarak ateş sayısının değişimi Şekil 3.21'de grafik olarak verilmiştir.



Şekil 3.21: Ateş sayısının *l*'ye bağlı olarak değişimi [31].

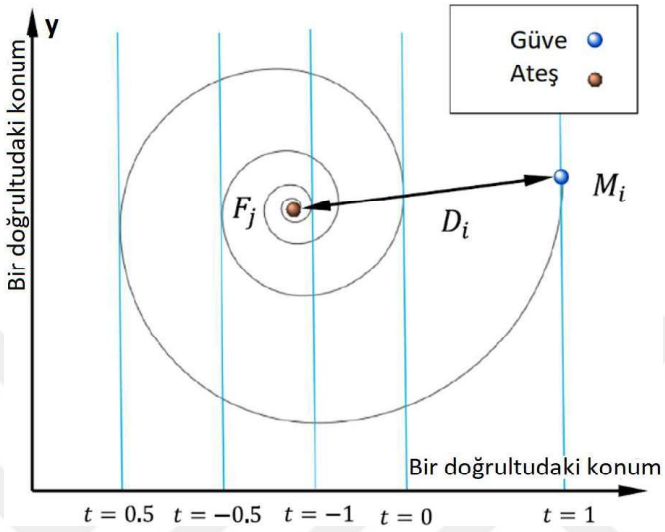
Uygunluk değerlerine göre en iyiden en kötüye sıralanan güve konumları ateş konumu olarak atanır. Bu ateş konumları temel alınarak mevcut güve konumları, denklem (3.25) ve (3.26) kullanılarak güncellenir.

$$D_i = |F_j - M_i| \tag{3.25}$$

$$S(M_i, F_j) = D_i\, e^{bt} \cos(2\pi t) + F_j \tag{3.26}$$

Denklem (3.25)'de D_i güve ile ateş arasındaki direkt uzaklığı ifade ederken, F_j ateş konumunu, M_i güve konumunu temsil etmektedir. Denklem (3.26)'da $S(M_i, F_j)$ ifadesi güvenin izleyeceği yörünge üzerindeki yeni konumunu ifade etmektedir. b parametresi spiral şeklinin ifadesi için gerekli sabit katsayıdır. t parametresi [-1,1] arasında değişen belirlenen noktanın ateş konumuna yakınlığını ifade eden parametredir. t'ye bağlı olarak güve konumunun ateşe yakınlığının değişimi iki boyutlu temsili görsel olarak Şekil 3.22'de verilmiştir. t, 1'den -1'e doğru yaklaştıkça güvenin konumu da ateşe giderek yaklaşır. Güvelerin yeni konumları güncellendikten sonra uygunluk değerleri hesaplanır ve ateş sayısı (3.24) ile güncellenir. Ateş sayısı her güncellendiğinde güve sayısı ateş sayısından fazla gelir. Dolayısı ile çevresinde dolaşabilecek ateş bulamayan güveler oluşur. Bu güveler güncellenmiş olan ateş popülasyonu içerisinde uygunluk derecesi en kötü olan ateş ile eşleştirilir. Böylece arama uzayında yeni keşif noktaları oluşma ihtimali arttırılarak daha geniş bir alan taranmış olur. Bu süreç belirlenen iterasyon sayısı boyunca tekrarlanarak devam eder. Son iterasyona gelindiğinde tek bir ateş konumu kalmış olur ve bütün güveler aynı ateş konumu etrafında gezerek hassas şekilde bu

alanda arama yapmaları sağlanır. Algoritma çıktısı olarak uygunluk derecesi en iyi ateş konumu dönülür [31, 66, 67].

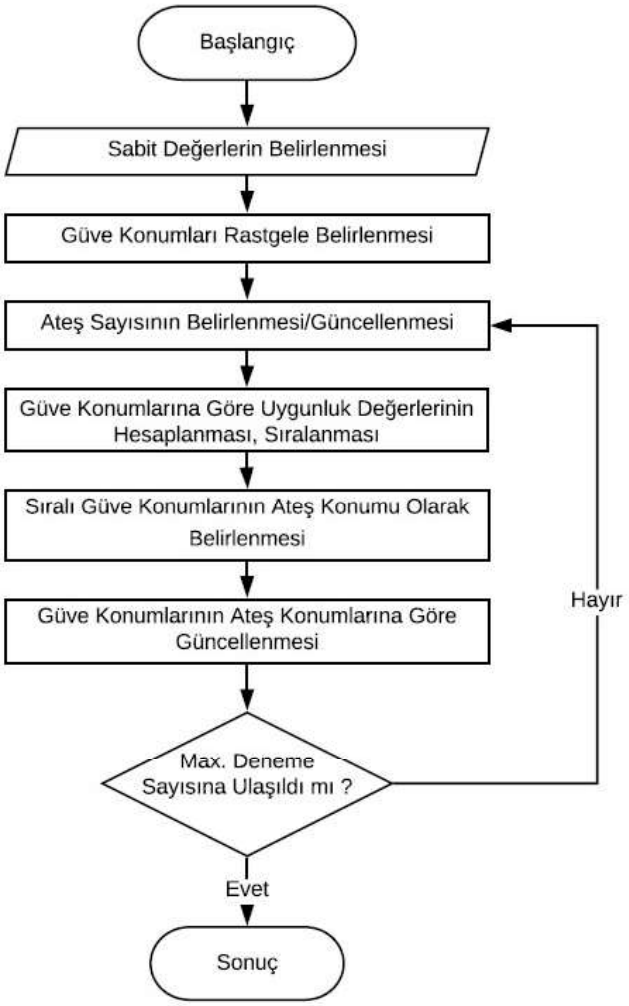


Şekil 3.22: *t*'ye bağlı M_i değişimi [31].

Aşağıda MFO algoritmasının sözde kodu (Pseudo kod) verilmiştir [31].

Güve sayısı belirlenir ve konumları rastgele atanır X_i *(i=1,2,...,n)*
Sabit başlangıç değerleri belirlenir
t=1
while *(t < max. deneme sayısı)*
    Ateş sayısı güncellenir (3.24)
    Güve konumlarının uygunluk değerleri (fitness) hesaplanır
    Uygunluk (fitness) değerlerine göre güve konumları sıralanır
    En iyi konumdan başlanarak güve konumları ateş konumu olarak atanır
    for *her güve için, n*
        Güve konumları güncellenir. (3.26)
    end for
t=t+1
end while
return F_{son} "sona kalan ateş konumu"

MFO algoritmasının genel işleyişinin akış diyagramı Şekil 3.23'de verilmiştir [31].

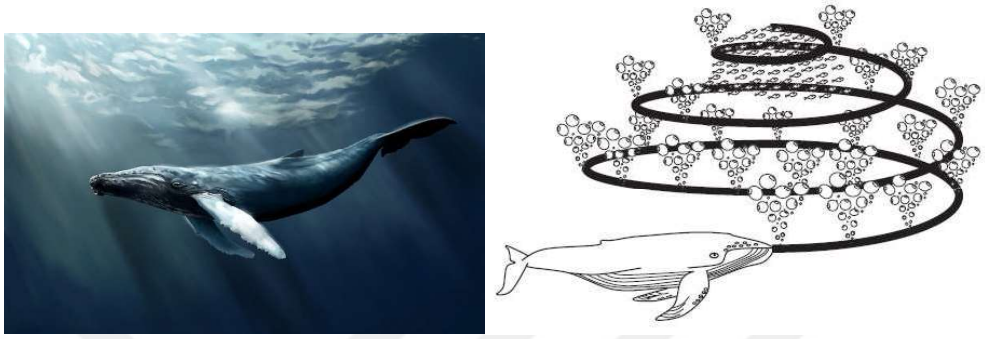


Şekil 3.23: MFO Akış Diyagramı.

3.3.5 Balina Optimizasyon Algoritması (WOA)

Balina optimizasyonu algoritması (Whale optimization Algorithm, WOA), kambur balinaların avlanma davranışlarından esinlenilerek, matematiksel yöntemlerle bilgisayar ortamında oluşturulmuş sezgisel bir optimizasyon yöntemidir. Kambur balinaların en çok tercih ettikleri besin türü küçük balık sürüleridir [32]. Kambur balinalar küçük balık sürülerini hedef aldıklarında sadece kendilerinin kullandığı, bilinen başka hiçbir canlı türünün kullanmadığı bir avlanma davranışı sergilerler. "Kabarcık ağı" diye isimlendirilen bu yöntem balinaların avına alttan yükselerek spiral çizerken kabarcıklar çıkarmasıdır. Bu yöntem ile sürü halindeki küçük balıkların panikleyerek bir araya toplanmasını sağlarlar. Böylece balinalar avına saldırmadan

önce sürü toplu halde bulunur ve balina daha çok fayda sağlar. Şekil 3.24'te kambur balina ve kabarcık ağı görselleri verilmiştir [69].



Şekil 3.24: Kambur balina ve kabarcık ağı.

Ancak kambur balinalar her zaman kabarcık ağı yöntemi ile avlanmazlar. Zaman zaman, besin ile karşılaşılan ortam şartlarına bağlı olarak balinanın doğrudan avına yöneldiği de gözlemlenmektedir. Bu durum balina optimizasyonu algoritmasında bir olasılık hesabı ile benzetilmiştir. Başlangıç parametre ve balina sayılarının belirlenmesi sonrasında balinalara arama uzayı içerisinden rastgele konumlar atanır. Her bir besin konumunun fayda değerleri amaç fonksiyonu kullanılarak hesaplanır ve rastgele olarak olasılık değerleri (Denklem 3.28'de p değişkeni olasılığı temsil eder.) atanır. Fayda olasılığı 0.5'den yüksek olan hedeflerde balina, kabarcık ağı metodu ile avlanır. Ancak olasılığı 0.5'den düşük olan hedeflerde bu yöntem uygulanmaz. Kabarcık ağı avlanma yöntemini uygulamayan balina avına olan uzaklığına bağlı bir ihtimalle (Denklem 3.28'de A değişkeni ile temsil edilmiştir.) direkt olarak avına yakınlaşacaktır ya da doğrudan hedef değiştirerek arama uzayı içerisindeki rastgele bir konuma atanacaktır [32, 70]. Bu avlanma yönelimleri denklem (3.27) ve (3.28)'de matematiksel olarak ifade edilmiştir.

$$D = |C\, X_{best} - X_t| \tag{3.27}$$

$$X_{t+1} = \begin{cases} \begin{cases} X_{best} - A\, D\,, & A < 1 \\ X_{rand} - A\, D\,, & A \geq 1 \end{cases}, & p < 0.5\ ise\ C = 2\, r \\ D\ e^{bl}\ \cos(2\pi l) + X_{best}\,, & p \geq 0.5\ ise\ C = 1 \end{cases} \tag{3.28}$$

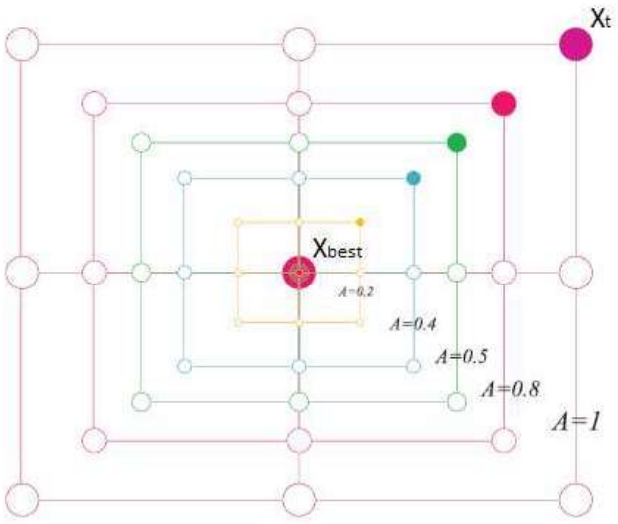
Denklem (3.27)'de D besin ile balina konumu rasındaki direkt uzaklığı; X_{best} uygunluk değeri en iyi besin konumunu; X_t balina konumunu temsil etmektedir. C, denklem (3.29)'da verilmiştir. Burada r_1, [0,1] aralığında rastgele bir sayıdır.

$$C = 2\, r_1 \tag{3.29}$$

Denklem (3.28)'de X_{t+1} balinanın yeni konumunu, b spiral şeklin sağlanması için gerekli katsayı sabitini; l [-1,1] aralığında rastgele bir sayıyı, p ise [0,1] aralığında rastgele bir sayı olan besinin olasılık değerini ifade etmektedir. Burada A, denklem (3.30)'da verilen şekilde hesaplanır. Burada r_2, [0,1] aralığında rastgele bir sayı, a [-1,1] aralığında iterasyon sayısı ile doğru orantılı olarak azalan bir parametredir

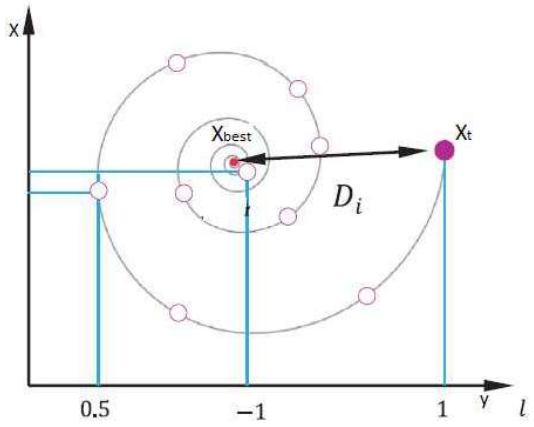
$$A = 2\, a\, r_2 - a \tag{3.30}$$

Denklem (3.30)'daki A parametresinin balinanın yeni konumuna etkisi, Şekil 3.25'te temsili olarak iki boyutlu gösterilmiştir.



Şekil 3.25: Arama uzayında balina hareketine A'nın etkisinin temsili gösterimi [32].

l parametresinin balinanın yeni konumunu belirlerken etkisi Şekil 3.26'da 2 boyutlu olarak gösterilmiştir. l parametresi 1'den uzaklaşarak -1'e yaklaştıkça balinanın yeni konumu da uygunluk derecesi en iyi olan besin konumuna yaklaşır.



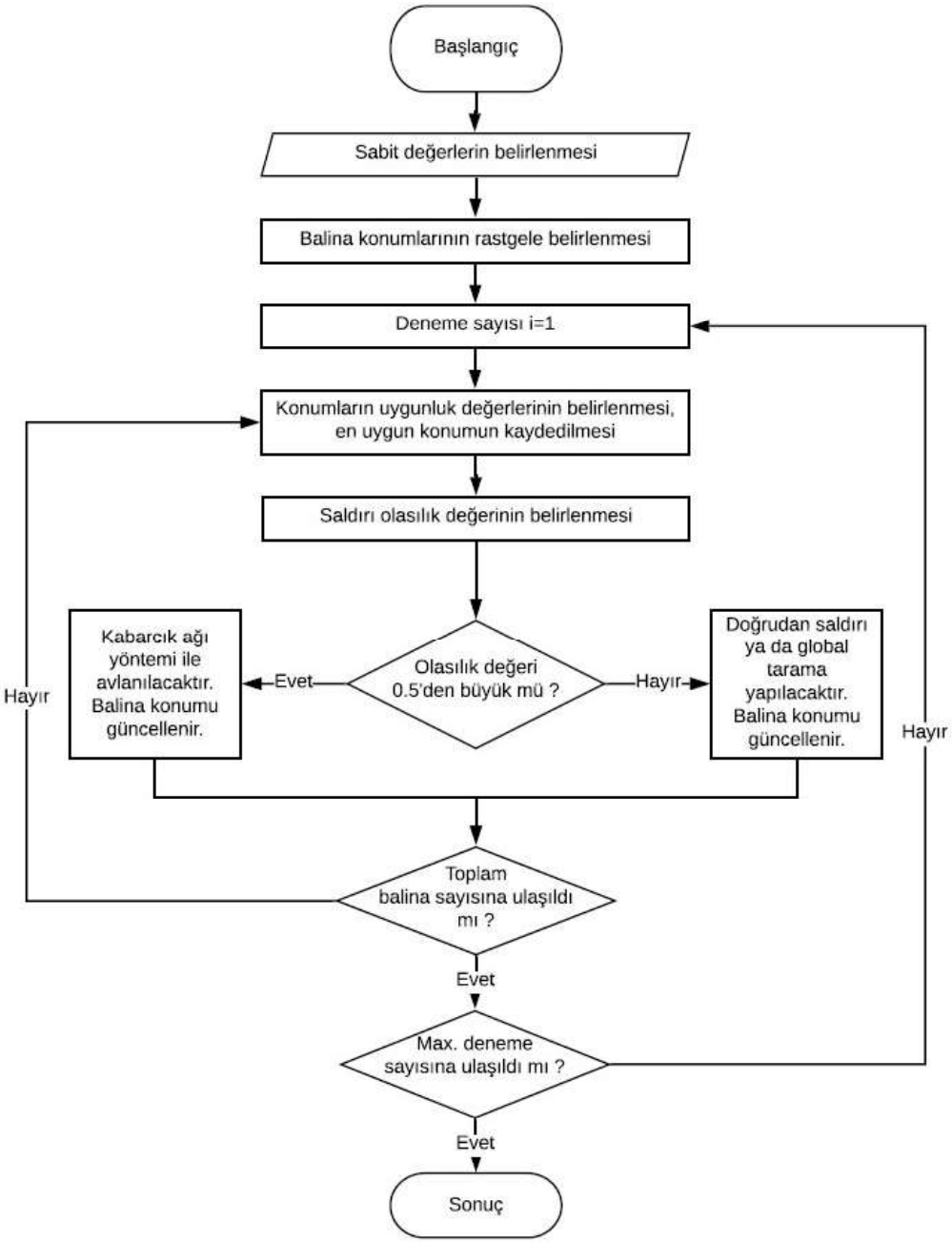
Şekil 3.26: 2 boyutlu arama uzayında balina hareketine *l*'nın etkisi [32].

Bu bilgiler doğrultusunda konumları güncellenen balina konumlarının uygunluk değerleri tekrar hesaplanır ve aynı işlemler iterasyon sayısı kadar tekrar edilerek devam eder.

WOA algoritması sözde kodu (pseudo code) aşağıda verilmiştir [32].

Balina sayısı belirlenir ve konumları rastgele atanır X_i (i=1,2,...,n)
Sabit başlangıç değerleri belirlenir
Balina konumlarının uygunluk değerleri (fitness) hesaplanır
Xo= Bilinen en uygun konum olarak atanır
t=1
***while** (t < max. deneme sayısı)*
  ***for** her baline için, n*
    Uygunluk değeri temel alınarak konum olasılık değeri P hesaplanır
    İf (P>0.5)
      Kabarcık ağı yöntemi ile balina konumu güncellenir. (3.28)
    else
      Rastgele atama yöntemi veya doğrudan saldırı yöntemi ile balina konumu güncellenir. (3.28)
    end if
  end for
  ***for** her baline için, n*
    Uygunluk değerleri hesaplanır
    Daha uygun çözüm varsa Xo güncellenir.
  end for
t=t+1
end while
return Xo

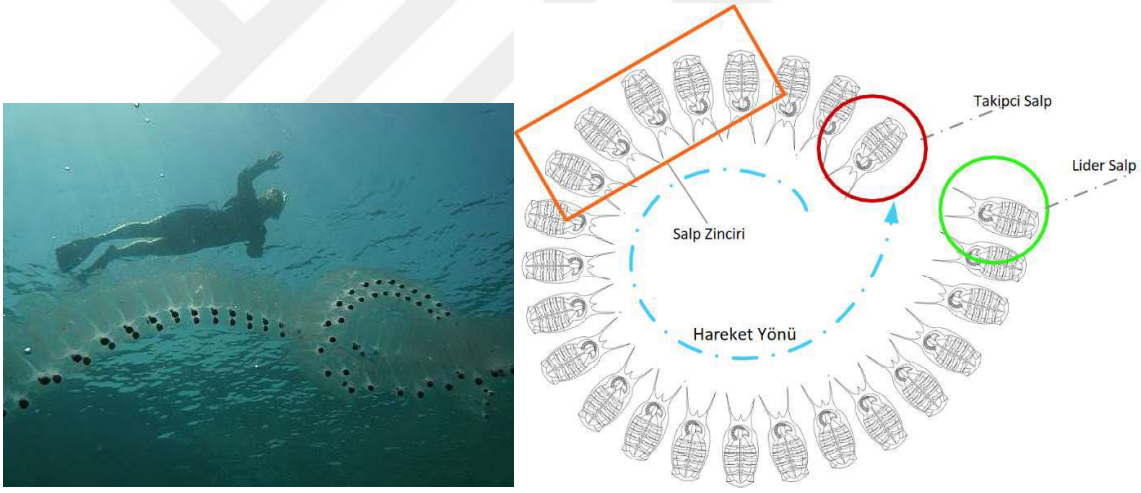
Şekil 3.27'de WOA yönteminin genel işleyişini özetleyen akış diyagramı verilmiştir [32].



Şekil 3.27: WOA akış diyagramı [32].

3.3.6 Salp Sürüsü Optimizasyon Algoritması (SSA)

Salpler, şeffaf renkte deniz canlılarıdır. Yapıları, daha çok bilinen denizanasına benzemektedir. Hareket ederken suyu vücutlarının içinden geçirerek pompalarlar. Ulaşılması zor yerlerde yaşaması, laboratuvar ortamında incelenmesini zorlaştırdığı için hakkındaki bilgi kısıtlı kalmıştır. Ancak bilim insanları davranışlarını kısıtlıda olsa incelemeyi başarabilmişlerdir. Sürü halinde yaşayan bu canlılar avlanmaları sırasında bir zincir oluştururlar. Sürü halinde hareket kabiliyetlerini arttırdıkları ve daha geniş alanı tarayabildikleri için bu yöntemi tercih ettikleri tahmin edilmektedir. Aşağıdaki Şekil 3.28'de salp ve salp sürüsünün oluşturduğu salp zinciri verilmiştir. Salp sürüsü algoritması (Salp Swarm Algorithm, SSA) bu canlıların avlanma davranışlarının benzetim yöntemiyle matematiksel modelinin çıkarılması sonucu elde edilmiş bir optimizasyon yöntemidir [33, 71].



Şekil 3.28: Salp Sürüsü

Salp sürüsü algoritması, sürü içerisindeki hiyerarşik yapının oluşturulması ile başlar. Sürü içerisinde, zincirin başında lider salp bulunur. Zincirin devamındaki diğer salpler lider olan salpı takip eder ve sürünün hangi yöne gideceğine lider karar verir.

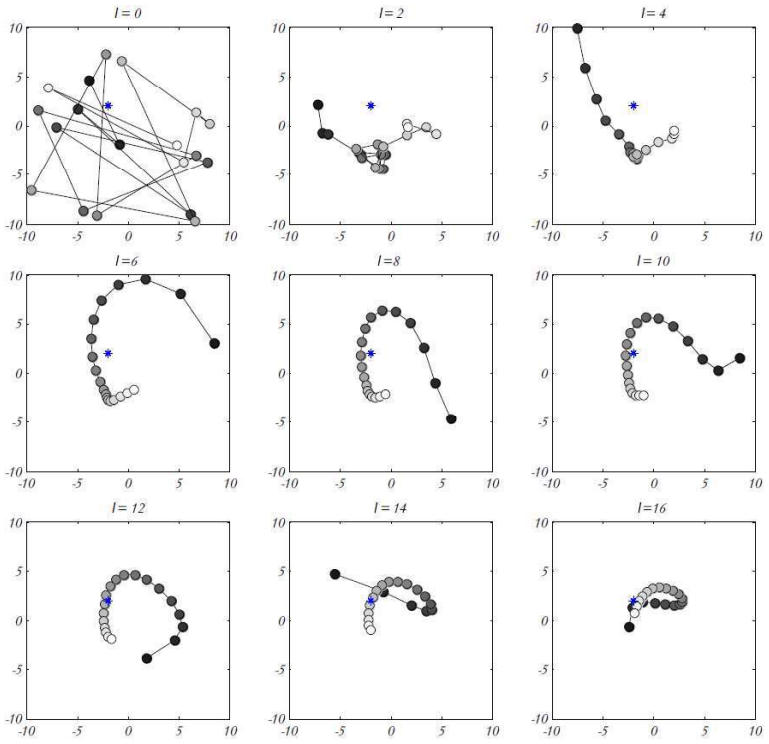
Algoritmada akış başladığında öncelikle salp sayısı ve iterasyon (tekrar) sayısı belirlenir. Ardından salplerin başlangıç konumları arama uzayı içerisinden rastgele atanır. Konumların uygunluk (fitness) değerleri başlangıç konumlarının amaç fonksiyonuna uygulanması ile hesaplanır ve sıralanır. Sıralanmış bu konumlardan uygunluk değeri en iyi olan besin konumu olarak belirlenir ve bundan sonraki süreçte lider olan salpin konumu besin konumu baz alınarak denklem (3.31) ile güncellenir [33, 71, 72].

$$X^{1(lider)} = \begin{cases} X_{besin} + c_1\left((ub - lb)\, c_2 + lb\right), & c_3 \geq 0 \\ X_{besin} - c_1\left((ub - lb)\, c_2 + lb\right), & c_3 < 0 \end{cases} \quad (3.31)$$

Denklem (3.31)'de $X^{1(lider)}$ liderin yeni konumunu; X_{besin} besin konumunu; c_2, c_3 : [0,1] aralığında rastgele sayıları; *ub* arama uzayının üst limitlerini; *lb* arama uzayının alt limitlerini temsil eder. c_1 ise bu algoritmadaki en önemli parametredir. Lider salpın tarama alanını limitleyen değerdir ve yeni keşif mi yoksa hassas konum saptaması mı yapacağını belirler. $c_1 = 2e^{-(4l/L)}$ şeklinde ifade edilir. *l* o anki iterasyon sayısı, *L* ise max. iterasyon sayısını temsil eder. Lider dışındaki diğer salplerin konumları ise denklem (3.32) ile güncellenir.

$$X^i_{yeni} = \frac{X^i + X^{i-1}}{2} \quad (3.32)$$

Salp konumları güncellendikten sonra tekrar uygunluk değerleri hesap adımına dönülür ve deneme (iterasyon) sayısına ulaşılana kadar bu işlem tekrarlayarak devam eder. Algoritma işleyişi sırasındaki salplerin 2 boyutlu düzlemdeki temsili konumları Şekil 3.29'da verilmiştir. Burada popülasyon bireylerinin konumları *l*=0, *l*=2, *l*=4, *l*=6, *l*=8, *l*=10, *l*=12, *l*=14, *l*=16 anında gösterilerek salp sürüsünün temsili hareketi gözlenebilir.

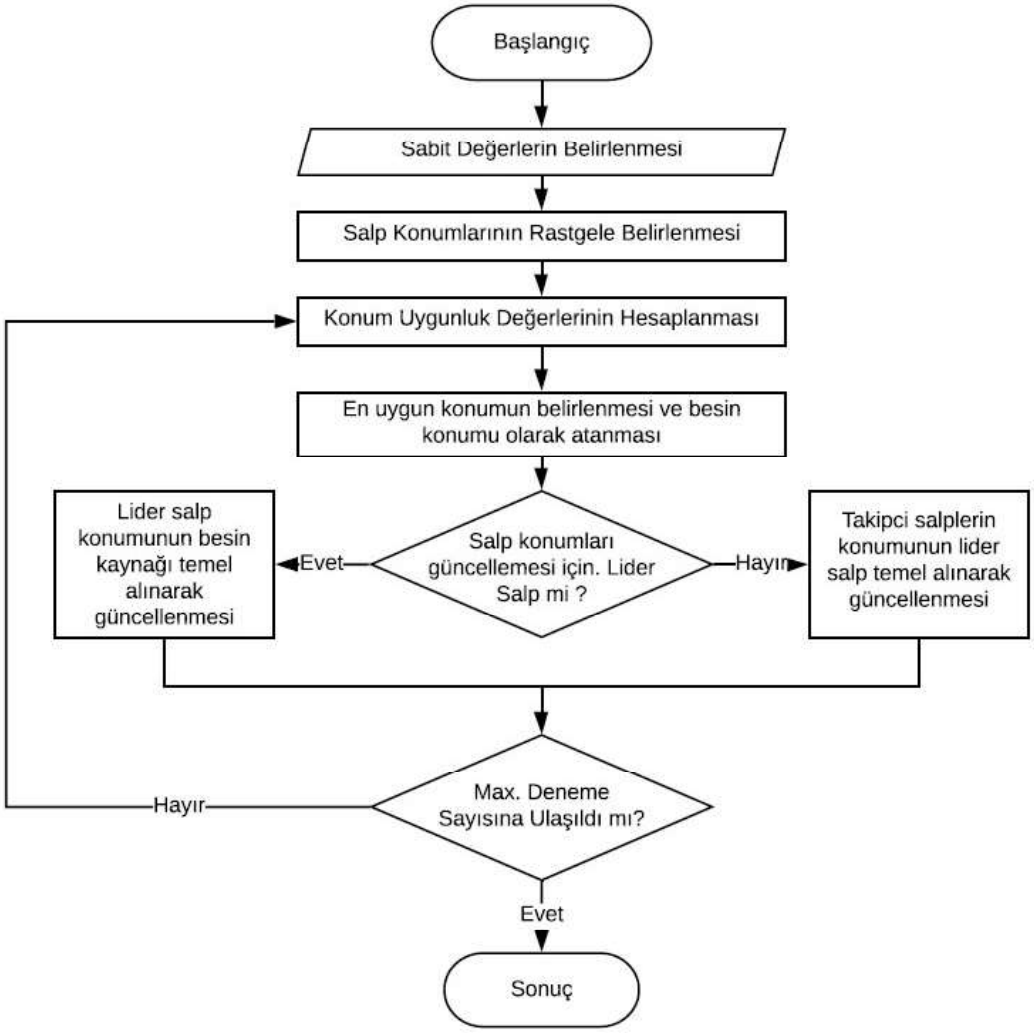


Şekil 3.29: SSA Algoritma sıraısnda salp sürüsünün hareketi (2 boyutlu) [33].

SSA yöntemi sözde kodu (pseudo kod) aşağıda verilmiştir [33].

Salp sayısı belirlenir ve konumları rastgele atanır X_i (i=1,2,...,n)
Sabit başlangıç değerleri belirlenir
Salp konumlarının uygunluk değerleri (fitness) hesaplanır
Xo= Bilinen en uygun konum olarak atanır (Besin konumu), t=1
while *(t < max. deneme sayısı)*
  for *her salp için, n*
    İf lider salp ise
      Besin konumu temel alınarak lider salp konumu güncellenir (3.31)
    else
      Lider salp konumu temel alınarak takipci salp konumları güncellenir (3.32)
    end if
  end for
  for *her salp için, n*
    Uygunluk değerleri hesaplanır
    Daha uygun çözüm varsa $X^{1(lider)}$ güncellenir.
  end for
t=t+1
end while
return $X^{1(lider)}$

SSA algoritmasının genel işleyişini özetleyen akış diyagramı Şekil 3.30'da verilmiştir [33].



Şekil 3.30: SSA akış diyagramı

3.3.7 Sinüs-Kosinüs Optimizasyon Algoritması (SCA)

Sinüs-Kosinüs Algoritması (Sine-Cosine Algorithm, SCA), matematiksel ifade olan sinüs ve kosinüs fonksiyonlarının salınım yapmasından faydalanılarak oluşturulmuş bir optimizasyon algoritmasıdır [34]. Belirlenen arama uzayında sinüs ve kosinüs fonksiyonu salınımları sayesinde yapılan tarama ile elde edilen en iyi değerler saptanır.

Algoritma akışında öncelikle iterasyon sayısı (deneme sayısı, tekrar sayısı), arama uzayı sınırları gibi sabit değerler ve kaç örnekleme ajanı ile çalışılacağı belirlenir; ardından ajan konumları sınırları belli olan arama uzayında rastgele konumlara atanır. Ajan konumlarının uygunluk değerleri (fitness) amaç fonksiyonu kullanılarak hesaplanır ve birbirleri ile karşılaştırılarak en iyi uygunluk değerine sahip nokta hedef (P_t) nokta olarak belirlenir. Her bir ajanın yeni konumları, bu hedef noktasını temel alarak denklem (3.33) ile güncellenir.

$$X_{t+1}^{i} = \begin{cases} X_t^i + r_1 \ \sin(r_2) \ |r_3 \ P_t - X_t^i|, & r_4 < 0.5 \\ X_t^i + r_1 \cos(r_2) \ |r_3 \ P_t - X_t^i|, & r_4 \geq 0.5 \end{cases} \tag{3.33}$$

Denklem (3.33)'de X_t^i, i numaralı ajanın t anındaki konumunu; X_{t+1}^i ajanın t+1 anındaki konumunu; P_t hedef konumunu sembolize eder. r_1 ise deneme (iterasyon) sayısı ile ters orantılı olarak azalan, [a,0] (a, max. dalga genliğidir.) aralığında yapılan salınımın genliğini belirleyen değişkendir ve denklem (3.34)'deki gibi ifade edilir. Burada T maksimum deneme sayısını (iterasyon sayısı) temsil eder. Rastgele sayılar olan r_2, r_3 ve r_4'ün ifadeleri aşağıda denklem (3.35), (3.46) ve (3.47)'de verilmiştir. Burada rand() fonksiyonu [0.1] aralığında rastgele bir sayı üretmektedir.

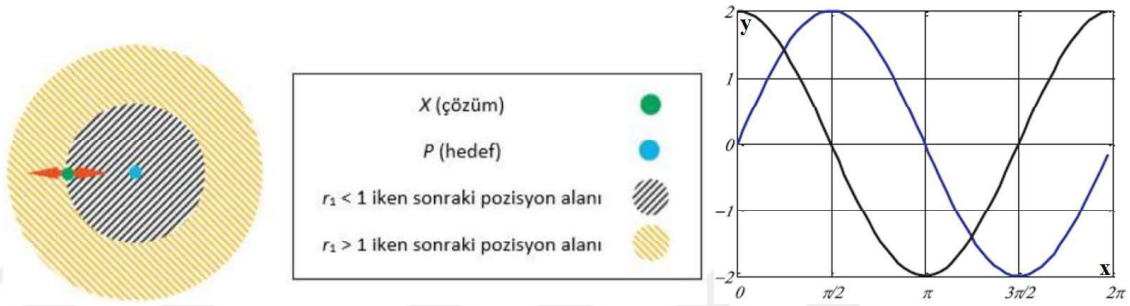
$$r_1 = a - t\frac{a}{T} \tag{3.34}$$

$$r_2 = 2\,\pi\, rand() \tag{3.35}$$

$$r_3 = 2\, rand() \tag{3.36}$$

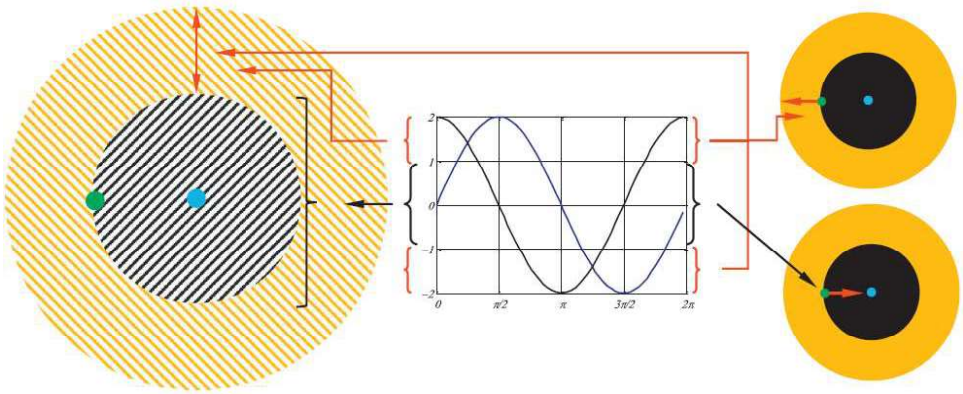
$$r_4 = rand() \tag{3.37}$$

Şekil 3.31 ve 3.32'de r_1 parametre değişiminin X_{t+1}^i konum değerlerine etkisi temsili iki boyutlu bir görsel ile anlatılmıştır. Buradaki görsel örnekte a sabiti 2 olarak kabul edilmiş ve sinüs, kosinüs dalgaları verilmiştir.



Şekil 3.31: r_1 değerinin X_{t+1}^i konumuna etkisi ve 2 genlikli sinüs-kosinüs fonksiyonları [34, 73, 74].

r_1>1 olduğu durumda ajanların yeni konumları P konumundan uzaklaşır ve global keşif için yeni konumlar aranır. r_1<1 olduğu durumlarda ise P konumuna yaklaşılarak sömürme işlemi yapılır ve en iyi değerin bulunması için hassas ayarlamalar yapılır. r_2 değeri yeni konumun P konumuna yakınlığı ya da P konumunda uzaklaşacağını belirler ve [0,2π] değer aralığındadır. r_3, hedef için rastgele bir ağırlığı tanımlar. Hedef (P) konumunun yeni konum belirlemedeki etkisini ayarlar. r_4, [0,1] aralığında değişir ve %50 olasılıkla hareketin sinüs ya da kosinüs tabanlı olma durumunu belirler [34, 73, 74].

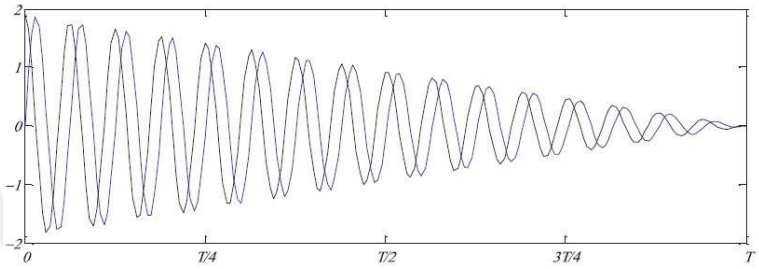


Şekil 3.32: r_1 değerinin X_{t+1}^i konumuna etkisi [34, 73, 74].

Konumların güncellenmesinden sonra tekrar uygunluk (fitness) hesaplama adımına geçilir ve her uygunluk hesaplama adımında hedeften daha iyi uygunluk değerine sahip bir konum bulunursa; hedef konum P, bulunan bu daha iyi değerlere sahip konum ile değiştirilir. Bu döngü

deneme (iterasyon) sayısına ulaşılana kadar tekrarlayarak devam eder ve algoritma sonunda hedef değer (*P*) bulunan en iyi sonuçlara sahip konumu içerir.

Şekil 3.33'de bir parçacığın (ajanın) SCA algoritması sırasında arama uzayında izleyeceği yol görsel olarak verilmiştir.

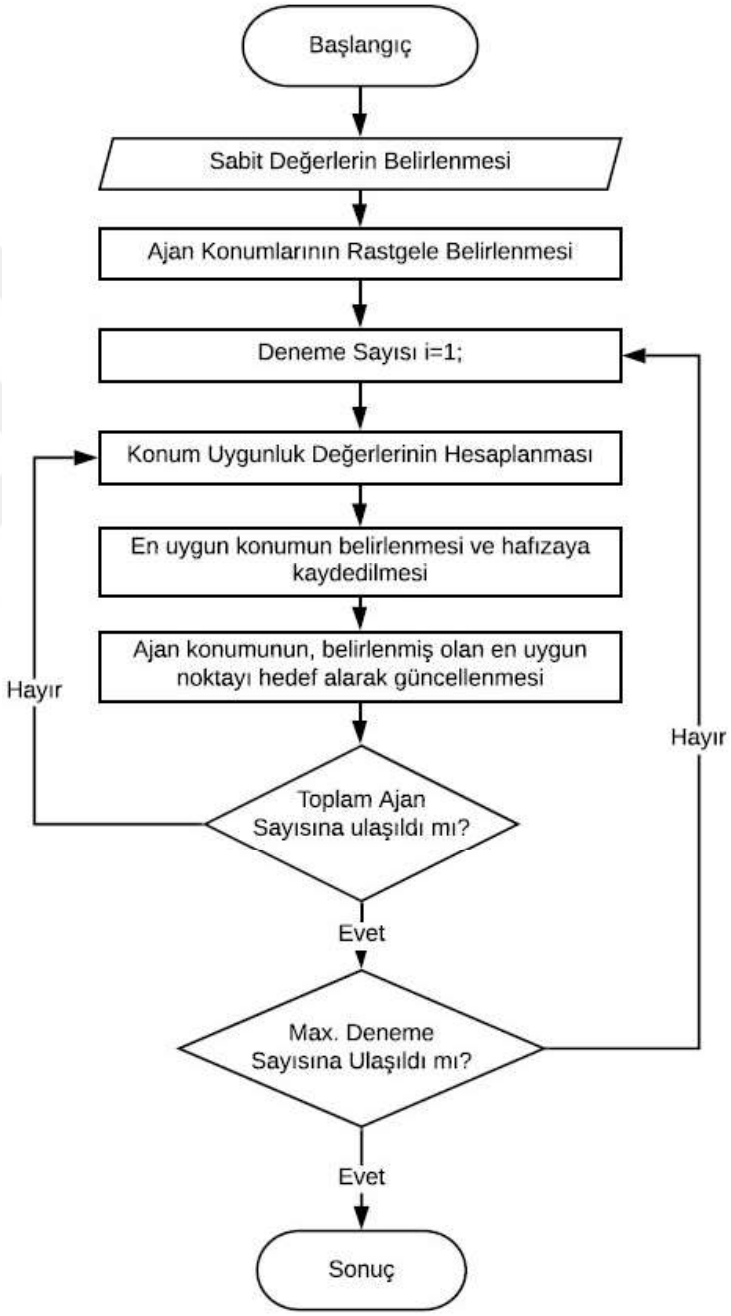


Şekil 3.33: SCA simülasyonu sırasında fonksiyon dağılım aralığının zamana(yaklaşım sayısına) bağlı dağılımı (a = 2) [34, 73].

SCA optimizasyon yöntemini özetleyen sözde kod (poseudo kod) aşağıdaki gibidir [34].

Keşif ajanı sayısı belirlenir ve konumları rastgele atanır X_i *(i=1,2,...,n)*
Sabit başlangıç değerleri belirlenir
Ajan konumlarının uygunluk değerleri (fitness) hesaplanır
P= Bilinen en uygun konum olarak atanır (Besin konumu)
t=1
while *(t < max. deneme sayısı)*
  for *her ajan için, n*
    Xo konumu temel alınarak ajan konumu güncellenir (3.33)
  end for
  for *her ajan için, n*
    Uygunluk değerleri hesaplanır
    Daha uygun çözüm varsa P güncellenir.
  end for
t=t+1
end while
return *P*

SCA optimizasyon yönteminin genel işleyişi aşağıdaki Şekil 3.34'de akış diyagramı olarak verilmiştir [34].



Şekil 3.34: SCA akış diyagramı [34].

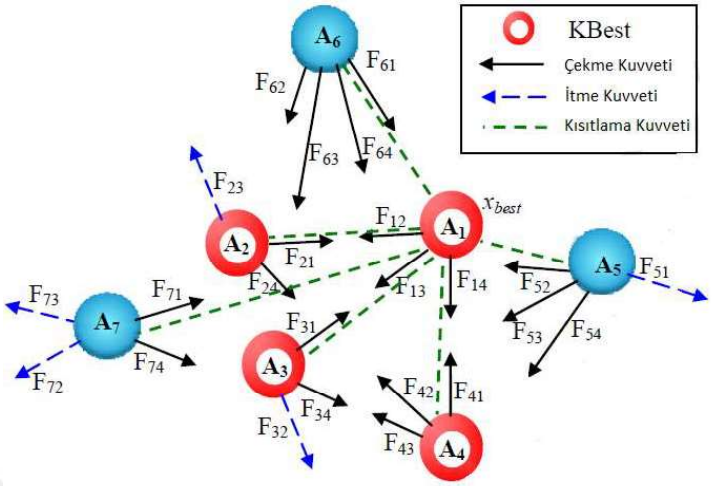
3.3.8 Atom Arama Optimizasyon Algoritması (ASO)

Atom Arama Optimizasyonu (Atom Search Optimization, ASO) algoritması fizik kurallarından esinlenilerek oluşturulmuş bir optimizasyon algoritmasıdır [35, 55, 56]. Maddenin yapısı kaynaklı olarak fizik kuralları gereğince atomlar arası oluşan itme ve çekme kuvvetleri mevcuttur. Bu kuvvetler simülasyonlar sırasında atomlar arasındaki uzaklığa bağlı olarak ifade edilmek istendiğinde genellikle Lennard-Jones potansiyel enerji kuvvetleri denkleminin kullanıldığı bilinmektedir [75]. Bu algoritmada da atomlar arasındaki potansiyel enerji etkileşimini uzaklık ilişkisinden faydalanarak ifade eden Lennard-Jones matematik modeli kullanılmıştır. Ek olarak atomlar arasındaki kovalent bağ ilişkisinden dolayı ikinci bir kuvvet daha atomlara etkiyecektir.

$$F_{net} = F_i + G_i \tag{3.38}$$

Denklem (3.38)'de F_i atomlar arası potansiyel enerji kuvveti; G_i atomlar arası kovalent bağ etkileşim kuvvetini; F_{net} atomlar arası net kuvveti temsil etmektedir.

Aynı şekilde bu çalışmadaki simülasyon sırasında arama uzayındaki ajan konumlarının birbirine yakınlaşma ve uzaklaşmalarına sebep olacak kuvvet hesabı denklem (3.38) ile hesaplanmıştır. Bir atoma (arama ajanına) etkiyen F_i hesaplanırken atomun komşularından K tane en uygun değere sahip atom (Şekil 3.35'te K tane en uygun değere sahip atom K_{best} ile temsil edilmiştir.) ele alınacak ve bu atomların uyguladığı kuvvet hesaplanacaktır. K değeri, atom sayısından başlayarak iterasyon sayısı ile ters orantılı olarak azalacak şekilde belirlenen bir parametredir ve [1,atom sayısı] aralığında değerler alır. Kısıtlayıcı kuvvet olan kovalent bağ kaynaklı kuvvet G_i hesaplanırken modeli basitleştirmek için sadece en iyi uygunluk değerine sahip atom ile olan etkileşimi hesaplanır [55]. Belirtilen kuvvetlerin 2 boyutlu temsili gösterimi Şekil 3.35'te verilmiştir.



Şekil 3.35: Atomlar ve aralarındaki kuvvetler [35].

Bu optimizasyon algoritmasında öncelikle başlangıç değerleri ve kaç atom (arama ajanı) olacağı belirlenir. Ardından atom konumları arama uzayı içerisinden rastgele atanır, uygunluk değerleri hesaplanır ve uygunluk değeri en iyi olan konum kaydedilir. Sonrasında atom konumlarının güncellenmesi için K adet en iyi komşu belirlenir ve komşu atomlar arasındaki potansiyel enerji kaynaklı kuvvetler (F_{ij}, j. en iyi atomun i. atom üzerine kuvvetidir.) denklem (3.39) ve denklem (3.40) ile hesaplanır. Atoma uygulanan net kuvvetin hesaplanabilmesi için kısıtlayıcı kovalent bağ kuvvetinin de (G_i) denklem (3.41) ve denklem (3.42) ile hesaplanması gerekir. Ardından bulunan bu iki kuvvet toplanarak net kuvvete ulaşılır.

$$F_{ij} = -\eta(t)\left[2\left(h_{ij}(t)\right)^{-13} - \left(h_{ij}(t)\right)^{-7}\right] \tag{3.39}$$

$$\eta(t) = a\left(1 - \frac{t-1}{1}\right)^{3} e^{-\frac{20t}{T}} \tag{3.40}$$

$$G_i(t) = \lambda(t)\ (X^d_{best}(\mathrm{t}) - X^d_i(\mathrm{t})) \tag{3.41}$$

$$\lambda(t) = \beta\, e^{-\frac{20t}{T}} \tag{3.42}$$

Denklem (3.39)'da atomlar arası potansiyelden kaynaklanan kuvvet eşitliği verilmiştir. Bu denklemde η(t) derinlik parametresi denklem (3.40) ile hesaplanır. $h_{ij}(t)$ parametresi atom türünün karakteristik özellikleri ve konumlarına bağlı olarak hesaplanabilen bir değişkendir [55, 56]. α derinlik ağırlığı, t o an gerçekleştirilen deneme (iterasyon) numarası, T toplamda gerçekleştirilecek olan deneme (iterasyon) sayısıdır.

Denklem (3.41)'deki $X_{best}(t)$ uygunluk değeri en iyi olan atom konumunu; $X_i(t)$ o an konum güncellemesi için kuvvet hesabı yapılan atomun konumunu; $\lambda(t)$ Lagrangian çarpanını; β ise Lagrangian çarpan ağırlığıdır.

Bir atom için net kuvvetin hesaplanmasından sonra Newton yasalarından faydalanarak denklem (3.43) kullanılıp ivmelenme hesaplanır.

$$a_i = \frac{F_i + G_i}{m_i} \tag{3.43}$$

Denklem (3.43) ve (3.44)'deki m_i değişkeni atom kütlesidir. Dolayısıyla ivme hesapları yapılırken atom kütlesine ihtiyaç duyulur. Denklem (3.45) ve (3.46) ile atom kütlesi hesaplanır. Bu algoritmada atomun gerçek kütlesinden bahsedilemeyeceği için temsili olarak uygunluk değeri büyük olana daha büyük bir kütle gelecek şekilde benzetilmiş bir model sunulmuştur.

$$a_i(t) = \frac{F_i(t)}{m_i(t)} + \frac{G_i(t)}{m_i(t)}$$

$$= -\left(1 - \frac{t-1}{1}\right)^3 e^{-\frac{20t}{T}} \sum_{j \in Kbest} \frac{rand_j \left[2\left(h_{ij}(t)\right)^{-13} - \left(h_{ij}(t)\right)^{-7}\right]}{m_i(t)} \tag{3.44}$$

$$\frac{\left(X_j^d(t) - X_i^d(t)\right)}{\left\|\overrightarrow{X_i^d}(t), \overrightarrow{X_j^d}(t)\right\|_2} + \beta\, e^{-\frac{20t}{T}} \frac{X_{best}^d(t) - X_i^d(t)}{m_i(t)}$$

Burada *d* konumun kaçıncı boyutu olduğunu belirtir. Atom ivmelenmesinin bulunabilmesi için denklem (3.39) ve (3.41)'de verilen F_i ve G_i parametreleri denklem (3.43)'de yerine konulduğunda denklem (3.44) elde edilir. Buradaki $m_i(t)$ denklem (3.45) ve (3.46) ile bulunur.

$$M_i(t) = e^{-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \tag{3.45}$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^{N} M_j(t)} \tag{3.46}$$

Denklem (3.45) ve (3.46)'da Fit_i, ivmesi hesaplanan atomun amaç fonksiyonuna sokulması ile elde edilen uygunluk değeridir (fitness). Fit_{best} mevcut atom konumları içerisinde en iyi (min) uygunluk değeri; Fit_{worst} ise mevcut atom konumları arasında en kötü (max) uygunluk değeridir.

Atom ivmesi bulunduktan sonra denklem (3.47) ile atomun yeni hızı hesaplanır ve denklem (3.48) kullanılarak da atomun yeni konumu saptanır.

$$V_i(t+1) = rand\ v_i(t) + a_i(t) \tag{3.47}$$

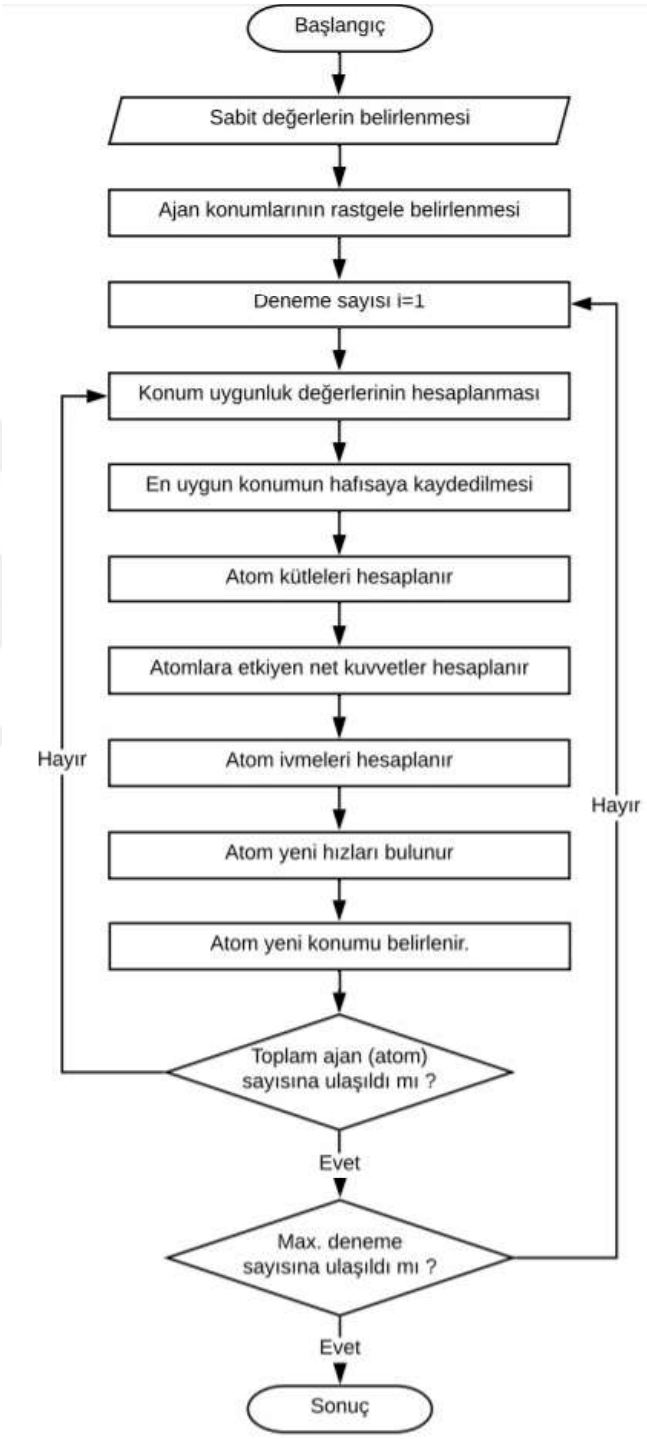
$$X_i(t+1) = X_i(t) + v_i(t+1) \tag{3.48}$$

Atomların yeni konumlarının belirlenmesinin ardından uygunluk (fitness) değerlerinin hesaplanması adımına geri dönülür ve bu akış deneme (iterasyon) sayısına ulaşılana kadar tekrarlayarak devam eder. Hafızaya kaydedilmiş olan uygunluk değeri en iyi (minimum) olan konum X_{bes} algoritmanın dönüş (sonuç) değeridir.

ASO algoritmasının sözde kod örneği (pseudo kod) aşağıca verilmiştir [35].

Keşif ajanı sayısı belirlenir ve konumları rastgele atanır X_i (i=1,2,...,n)
Sabit başlangıç değerleri belirlenir
Ajan (atom) konumlarının uygunluk değerleri (fitness) hesaplanır
X_{best}= Bilinen en uygun konum olarak atanır
t=1
***while** (t < max. deneme sayısı)*
  ***for** her ajan için, n*
    Atom konumlarının uygunluk değerleri (fitness) hesaplanır
    Xo'dan daha uygun sonuç çıkarsa X_{best} güncellenir
    Atom kütlesi hesaplanır (3.46)
    Atoma etkiyen net kuvvet bulunur. (3.38)
    Atom ivmesi hesaplanır, a (3.43)
    Atom hızı hesaplanır, V (3.47)
    Atom konumu güncellenir X_i (3.48)
  end for
t=t+1
end while
return X_{best}

ASO algoritmasının genel işleyişi Şekil 3.36'daki akış diyagramında verilmiştir [35].



Şekil 3.36: ASO akış diyagramı.

3.4 SEZGİSEL OPTİMİZASYON ALGORİTMALARININ FOPID GÜÇ SANTRAL KONTROLÖRÜNE UYGULANMASI

Bu tezde gaz türbinli güç santralinin kontrolünde FOPID kontrolör kullanılmış ve kontrolörün optimizasyonu önceki bölümlerde detaylıca bahsedilen sekiz adet optimizyon algoritmaları ile yapılmıştır. Bu metasezgisel optimizasyon algoritmalarının çalışma prensipleri gereği sistem modeline uygulanmaları sırasında elde edilen değerlerin karşılaştırılabilmesi için amaç fonksiyonuna ihtiyaç duyulur. Literatürde yaygın olarak kullanılan 4 amaç fonksiyonu vardır [76, 77, 78, 79]. Bunlar mutlak hatanın toplamı (Integral Absolute Error, IAE), hata karelerinin toplamı (Integral Square Error, ISE), zaman ağırlıklı hata karelerinin toplamı (Integral Time-weighted Square Error, ITSE), zaman ağırlıklı mutlak hata toplamı (Integral Time-weighted Absolute Error, ITAE) olarak adlandırılan amaç fonksiyonları olup; aşağıdaki denklemlerde verilmişlerdir.

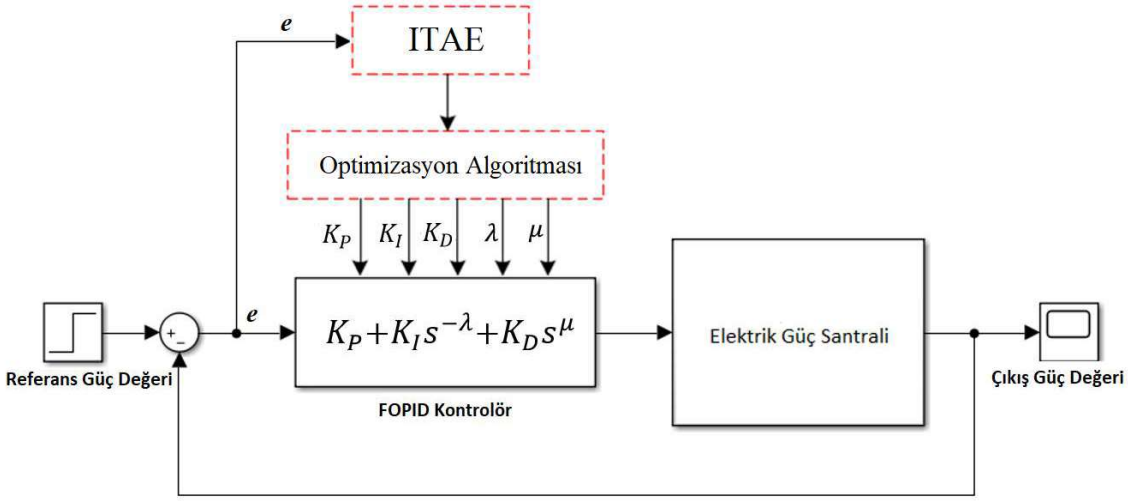
$$IAE = \int_0^T |e| \; dt \tag{3.49}$$

$$ISE = \int_0^T e^2 \; dt \tag{3.50}$$

$$ITAE = \int_0^T t \, |e| \; dt \tag{3.51}$$

$$ITSE = \int_0^T t \, e^2 \; dt \tag{3.52}$$

Burada *e* hatayı (error) ifade etmektedir ve bu tezde santral çıkış değeri ile referans değer arasındaki farka eşittir. Şekil 3.37'de *e* (hata), optimizasyon sistem modeli üzerinde gösterilmiştir.



Şekil 3.37: FOPID kontrolör ile denetlenen gaz türbinli elektrik güç santral modeli.

Optimizasyon algoritmalarının sistem modeline uygulanması sırasında e (hata) sinyalini sıfıra yakın olacak şekilde minimum seviyede tutmak ve sistem çıkış sinyalinin yerleşme süresinin kısa olmasını sağlayabilmek için ITAE hesaplama yöntemi kullanılmıştır. Algoritmaların işleyişi sırasında öncelikle her bir ajan için FOPID kontrolörün başlangıç parametreleri, kullanılan algoritma prensiplerine uygun şekilde belirlenir. Ardından parametreler sistem modelinde yerine yazılarak ITAE hesabı yapılır ve uygunluk değerleri (fitness) bulunur. Bu işlem arama ajanlarının hepsine uygulanır (popülasyon sayısı kadar). Uygunluk değerleri belirlenmiş olan popülasyon bireylerinin konumları algoritma işletilerek güncellenir. Bu işlem iterasyon sayısı süresince tekrarlayarak devam eder ve farklı parametre grupları sistem modeli üzerinde denenmiş olur. İterasyonların sonucunda ITAE değeri en düşük olan ajan (parametre grubu) yani uygunluk (fitness) değeri en düşük olan parametre grubu (K_P, K_I , K_D, λ, μ) algoritma çıktısı olarak belirlenir.

Burada amaç fonksiyonu sonucunda en düşük değeri veren parametre değerlerinin bulunması algoritmaların asıl amacıdır. Şekil 3.37'de gaz türbinli elektrik güç santral modeline uygulanan FOPID kontrolör ve parametreleri (K_P, K_I , K_D, λ, μ), optimizasyon algoritmasının sisteme uygulanışı ve algoritmanın istenen sonuca ulaşabilmesi için amaç fonksiyonunun (ITAE), hata kolu (e) ile ilişkisi görsel olarak verilmiştir.

Bu tezde simülasyon referans güç değeri olarak 50sn +1, 50sn -1 uygulanmıştır. Böylelikle sistemin hem pozitif hem negatif alanda verdiği tepkiler ölçülerek, +1 den -1 e geçerkenki ani değişime verdiği tepki gözlenebilmiştir.

- Tezde ele alınan sekiz optimizasyon algoritması için tek tek sonuçlar verilecek olursa, ilk olarak FOPID kontrolör için PSO simülasyonu Tablo 3.3'te verilmiş parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.3: FOPID optimizasyonu için PSO simülasyon parametre değerleri.

Parametre	**Değer**
Parçacık Sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]
c1, c2	2, 2

Tablo 3.3'den hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 3.8943 K_I = 3.1747 K_D =0.77962 λ = 1.0209 μ = 1.2364

- FOPID kontrolör için ABC simülasyonu Tablo 3.4'teki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.4: FOPID optimizasyonu için ABC simülasyon parametre değerleri.

Parametre	**Değer**
Besin Sayısı (Kaşif Arı Sayısı)	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]
Limit (L)	80

Tablo 3.4'den hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 2.74 K_I = 1.9575 K_D = 0.5182 λ = 1.0110 μ = 1.31

- FOPID kontrolör için GWO simülasyonu Tablo 3.5'teki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.5 : FOPID optimizasyonu için GWO simülasyon parametre değerleri.

Parametre	**Değer**
Kurt Sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]

Tablo 3.5'den hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 2.7202 K_I = 2.2442 K_D = 0.53742 λ = 1.0095 μ = 1.2383

- FOPID kontrolör için MFO simülasyonu Tablo 3.6'daki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.6: FOPID optimizasyonu için MFO simülasyon parametre değerleri.

Parametre	**Değer**
Alev Sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]
b	1

Tablo 3.6'dan hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 1.4895 K_I = 1.4916 K_D = 0.2174 λ = 1.0116 μ = 1.1059

- FOPID kontrolör için WOA simülasyonu Tablo 3.7'deki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.7: FOPID optimizasyonu için WOA simülasyon parametre değerleri.

Parametre	**Değer**
Parçacık Sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]
b	1

Tablo 3.7'den hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 0.7225 K_I = 1.3633 K_D = 0.7399 λ = 1.0144 μ = 0.3308

- FOPID kontrolör için SSA simülasyonu Tablo 3.8'deki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.8: FOPID optimizasyonu için SSA simülasyon parametre değerleri.

Parametre	**Değer**
Salp Sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]

Tablo 3.8'den hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 1.2776 K_I = 1.3483 K_D = 0.4471 λ = 1.1975 μ = 0.5463

- FOPID kontrolör için SCA simülasyonu Tablo 3.9'daki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.9: FOPID optimizasyonu için SCA simülasyon parametre değerleri.

Parametre	**Değer**
Başlangıç noktası sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]
a	2

Tablo 3.9'dan hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 0.0209 K_I = 1.2237 K_D = 1.5781 λ = 1.2356 μ = 0.0962

- FOPID kontrolör için ASO simülasyonu Tablo 3.10'daki parametreler kullanılarak gerçekleştirilmiştir.

Tablo 3.10: FOPID optimizasyonu için ASO simülasyon parametre değerleri.

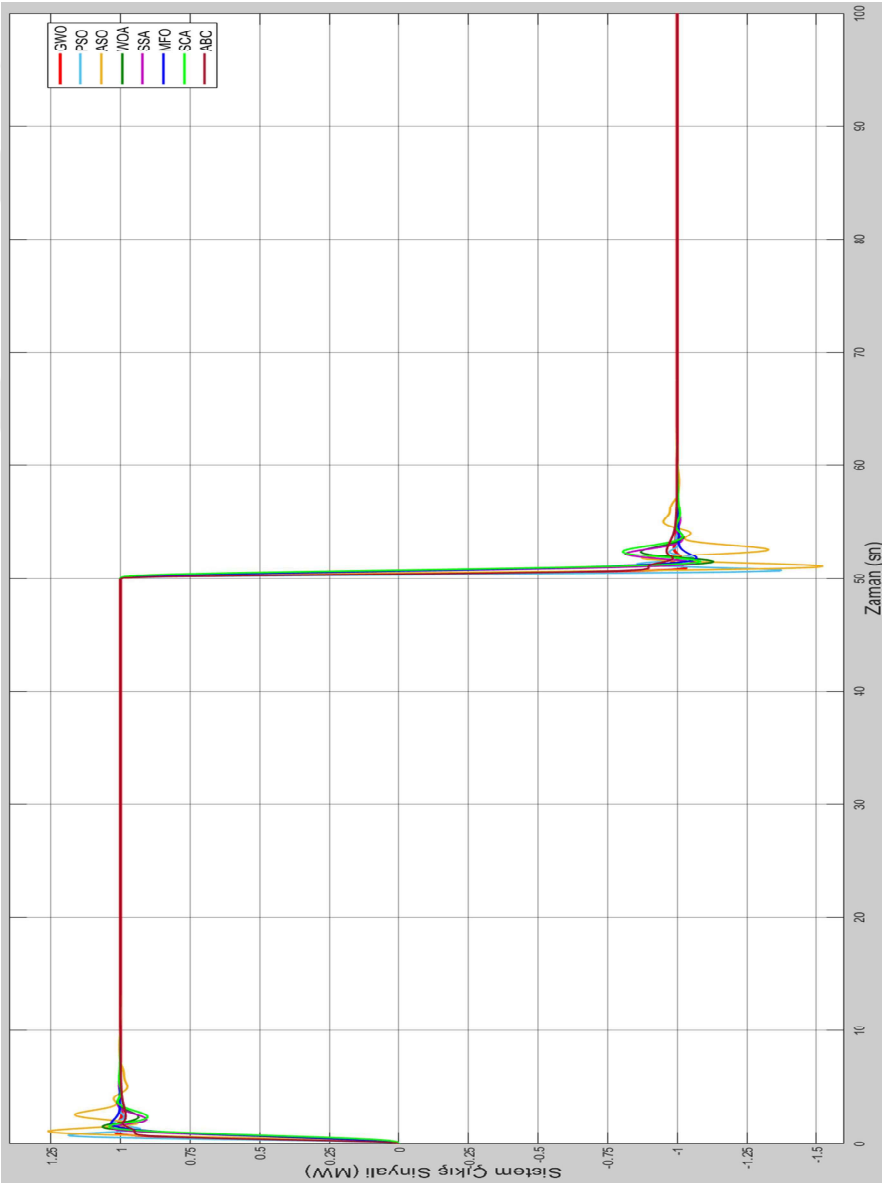
Parametre	**Değer**
Atom Sayısı	50
Deneme Sayısı (iterasyon)	30
K_P, K_I, K_D, λ, μ alt limitleri	[0.001, 0.001, 0.001 0,001 0,001]
K_P, K_I, K_D, λ, μ üst limitleri	[20, 20, 20, 2, 2]
α (Derinlik ağırlığı)	50
β (Lagrangian çarpanı ağırlığı)	0.2

Tablo 3.10'dan hareketle elde edilen FOPID parametreleri şu değerlerdedir:

K_P = 1.3061 K_I = 3.4435 K_D = 1.1787 λ = 1.4377 μ = 0.5512

4 BULGULAR

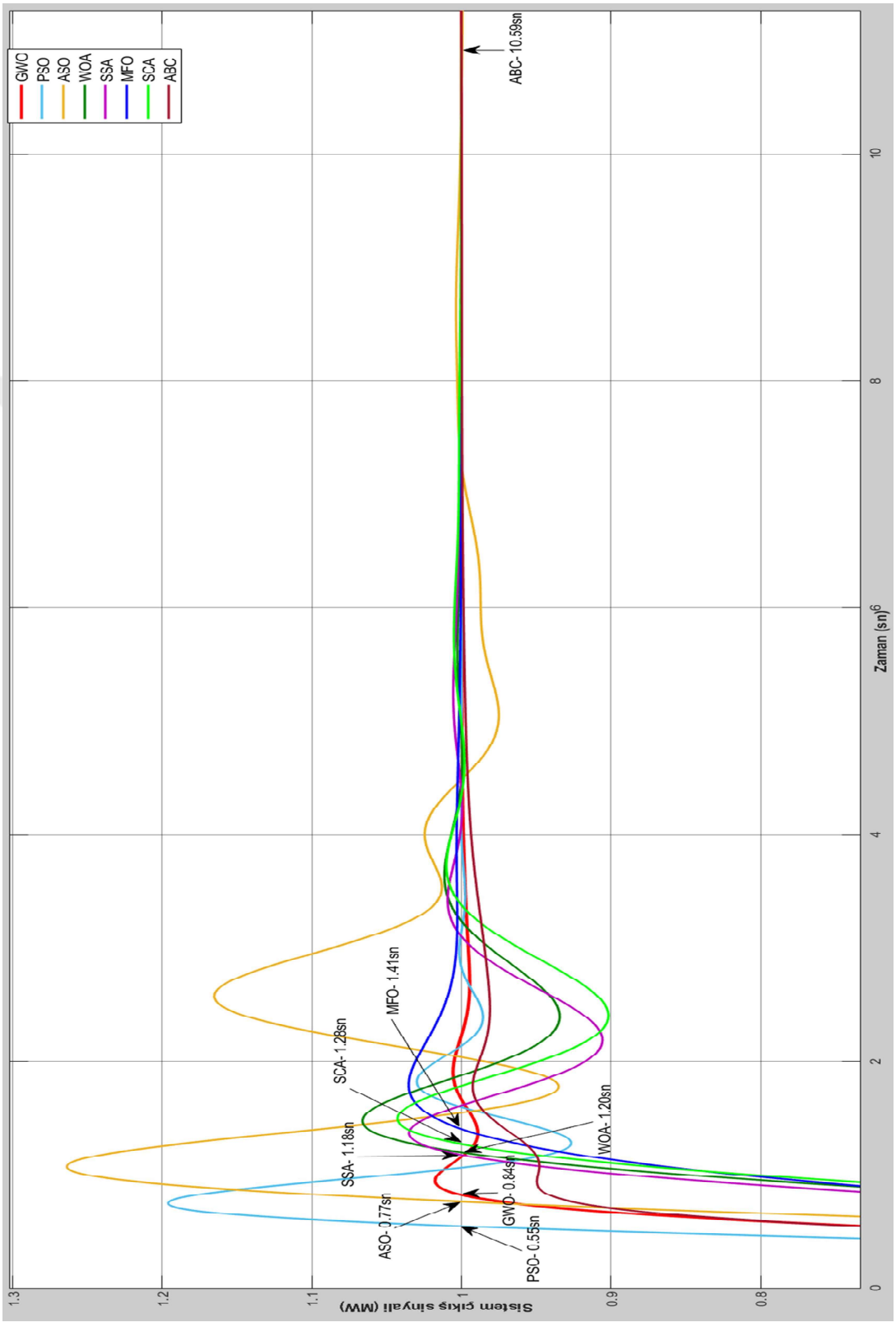
Optimizasyon algoritmalarının sonuçlarına göre kontrolör parametre değerleri belirlenmiştir. Kontrol sistemine referans değer olarak 0-50 sn arası +1MW, 50-100 sn -1MW değeri uygulanmıştır. Bu referans giriş simülasyonların gerçekleştirildiği gaz türbinli santralin kontrol sistemine uygulandığında farklı optimizasyon yöntemleri ile tasarlamış olduğumuz FOPID kontrolörlerin denetlediği santral sistemi çıkış sinyalleri Şekil 4.1'de verilmiştir.



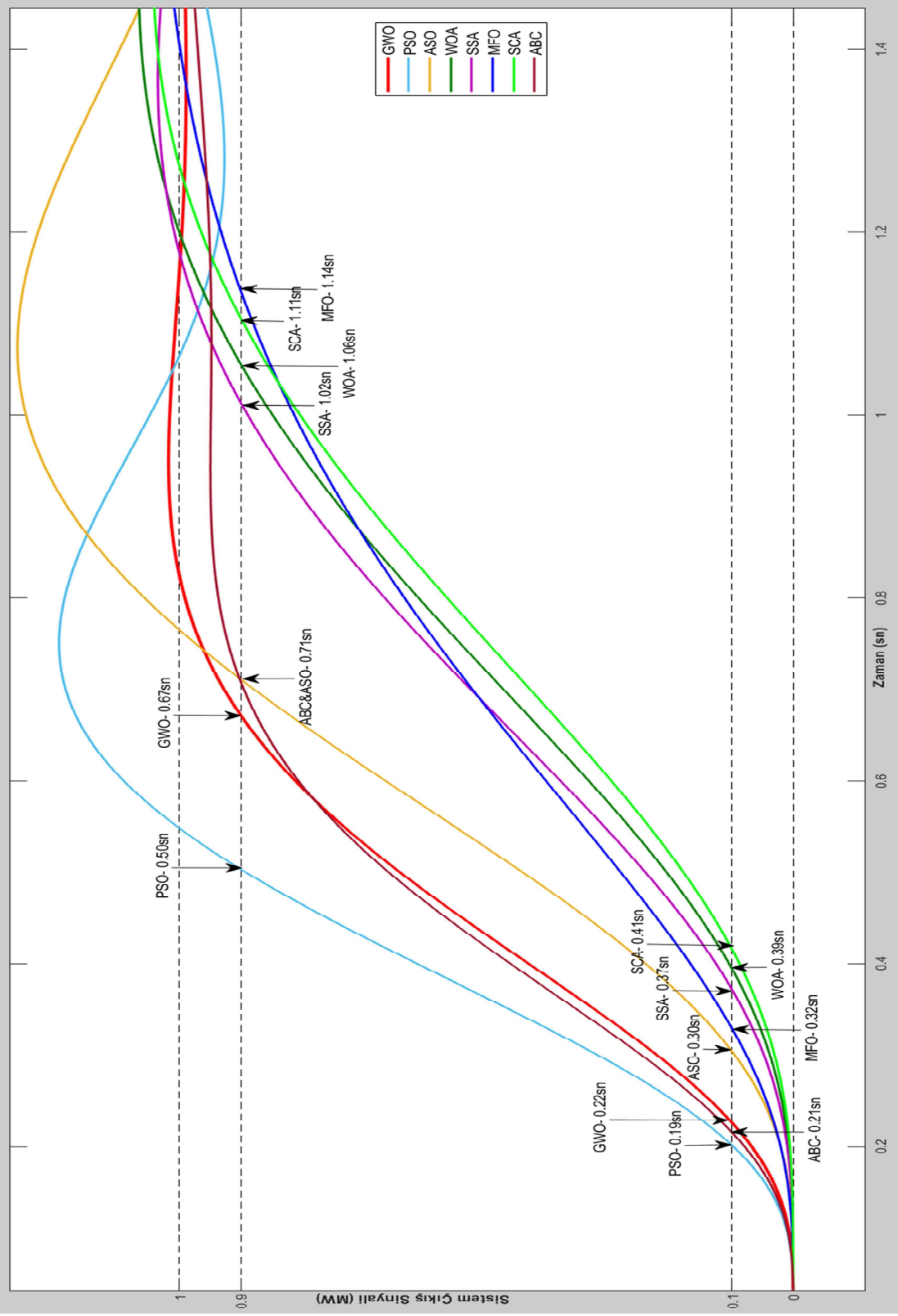
Şekil 4.1: 0-100sn aralığında santral sistemi çıkış sinyali.

Optimize edilen FOPID kontrolörlerin performans kıyaslamasını net şekilde yapabilmek için yükselme süresi, yerleşme süresi ve yüzde en büyük aşım kriterleri kullanılmıştır. Bu kriterler değerlendirilirken referans değerin hem pozitif hem negatif olarak uygulanması farklı koşullarda sistem davranışının gözlemlenebilmesine imkan tanımıştır.

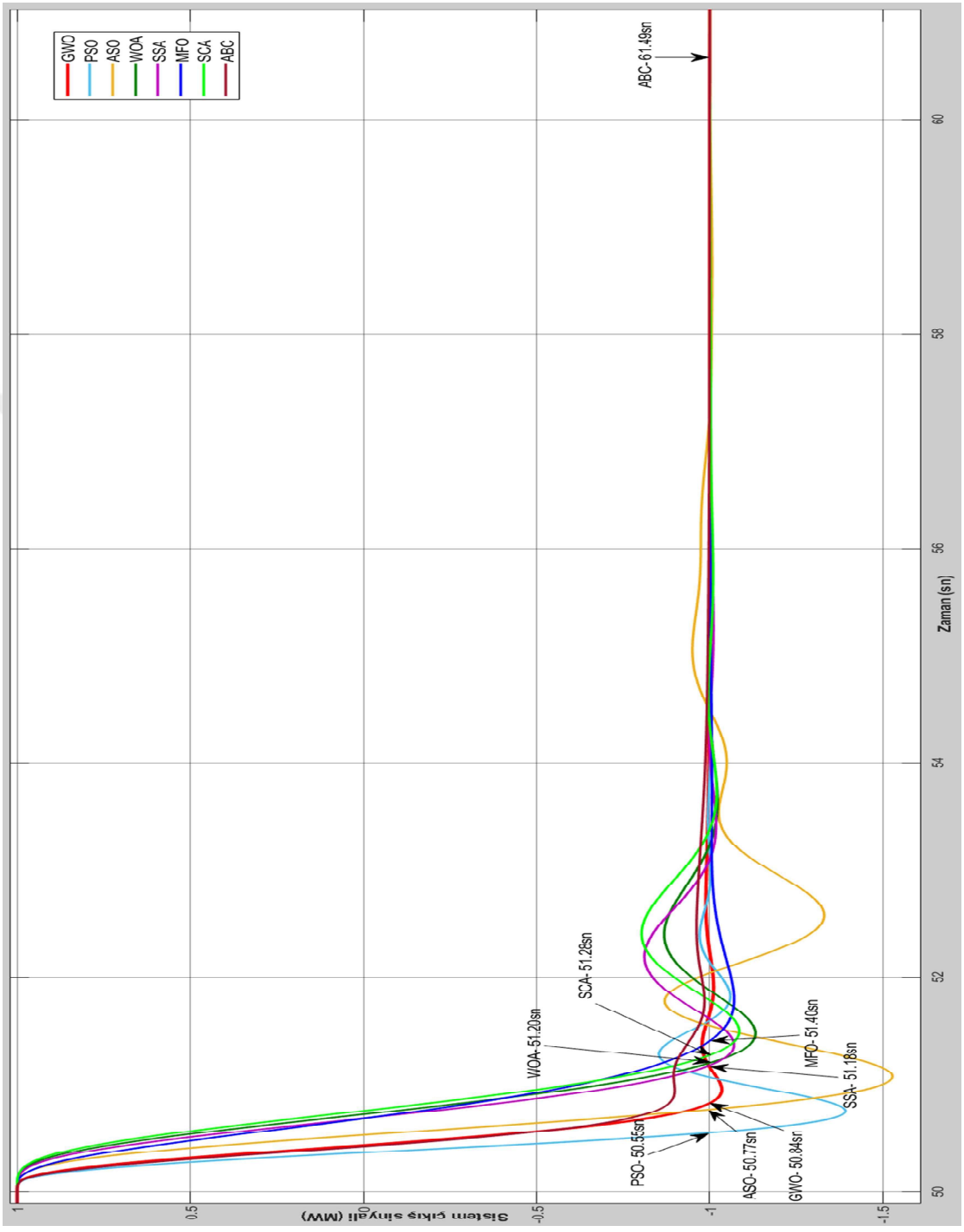
FOPID kontrolör uygulanmış basitleştirilmiş gaz türbinli elektrik güç santral modeli çıkış işaretlerinin yükselme süresi karşılaştırmaları yaygın olarak kullanılan iki yaklaşım kriteriyle incelenmiştir. Bunlardan biri sistem çıkış işaretinin kalıcı durum değerinin %0'ından %100'üne ulaşıncaya kadar geçen süreyi kabul ederken; ikincisi ise %10'undan %90'ına ulaşana kadar geçen süreyi yükselme süresi olarak kabul etmektedir [40]. Hem pozitif bölgede hem de negatif bölgede sekiz farklı optimizasyon algoritması ile tasarımı yapılmış olan FOPID kontrolörlerin uygulandığı sistemin çıkış işaretleri için bu iki yaklaşım kullanılarak yükselme sürelerinin karşılaştırması yapılmıştır. Pozitif bölgede ele alınan sistemin çıkış işareti için yükselme süresi karşılaştırmaları yapılırken Şekil 4.2 ve Şekil 4.3'ten faydalanılmıştır. Şekil 4.2'de verilmiş grafikte optimize edilmiş sistemin pozitif bölgedeki çıkış işaretlerinin %0'ından %100'üne ulaşma süreleri oklar ile gösterilmiştir. Buna göre yükselme süreleri küçükten büyüğe doğru sıralanınca PSO, ASO, GWO, SSA, WOA, SCA, MFO ve ABC algoritmaları ile optimize edilmiş FOPID kontrolörler kullanılan gaz türbinli elektrik güç santral sisteminin çıkış işaretlerinde sırasıyla saniye cinsinden 0.55, 0.77, 0.84, 1.18, 1.20, 1.28, 1.41 olduğu görülmektedir. Şekil 4.3'de verilmiş grafikte optimize edilmiş sistemin pozitif bölgedeki çıkış işaretlerinin %10'undan %90'ına ulaşma süreleri oklar ile gösterilmiştir. Buna göre yükselme süreleri küçükten büyüğe doğru sıralanınca PSO, ASO, GWO, ABC, SSA, WOA, SCA, MFO algoritmaları ile optimize edilmiş FOPID kontrolörler kullanılan gaz türbinli elektrik güç santral sisteminin çıkış işaretlerinde sırasıyla saniye cinsinden 0.31, 0.41, 0.45, 0.50, 0.65, 0.67, 0.70, 0.82 olduğu görülmektedir. Negatif bölgede ele alınan sistemin çıkış eğrisi için yükselme süresi karşılaştırmaları yapılırken Şekil 4.4 ve Şekil 4.5'ten faydalanılmıştır. Şekil 4.4'de verilmiş grafikte optimize edilmiş sistemin negatif bölgedeki çıkış işaretlerinin %0'ından %100'üne ulaşma süreleri oklar ile gösterilmiştir. Buna göre yükselme süreleri küçükten büyüğe doğru sıralanınca PSO, ASO, GWO, SSA, WOA, SCA, MFO, ABC algoritmaları ile optimize edilmiş FOPID kontrolörler kullanılan gaz türbinli elektrik güç santral sisteminin çıkış işaretlerinde sırasıyla saniye cinsinden 0.55, 0.77, 0.84, 1.18, 1.20, 1.28, 1.40, 11.49 olduğu görülür. .



Şekil 4.2: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yükselme zamanları (%0-%100)



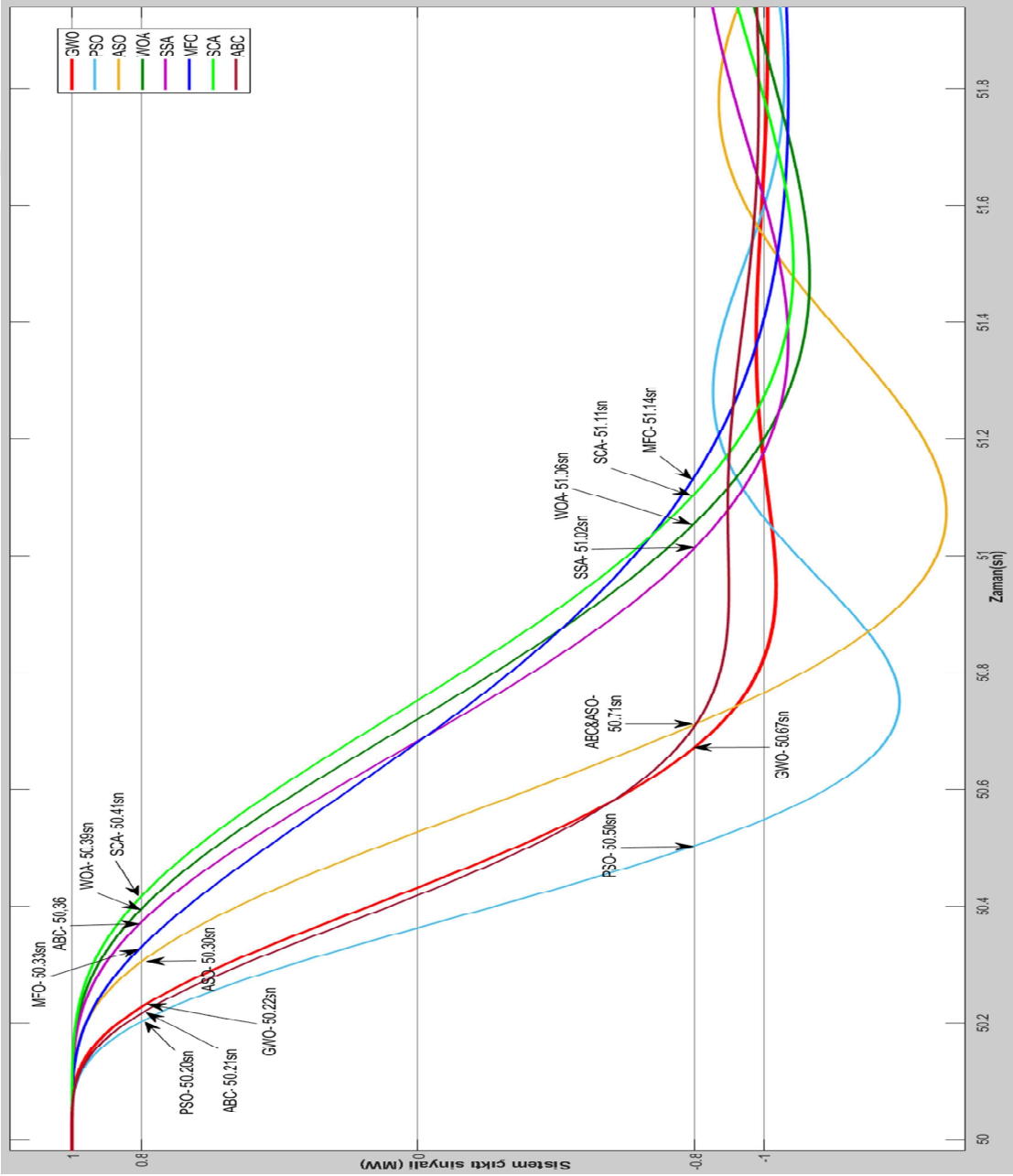
Şekil 4.3 Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yükselme zamanları (%10-%90)



Şekil 4.4: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yükselme zamanları (%0-%100)

Şekil 4.5'de verilmiş grafikte optimize edilmiş sistemin negatif bölgedeki çıkış işaretlerinin %10'undan %90'ına ulaşma süreleri oklar ile gösterilmiştir. Buna göre yükselme süreleri küçükten büyüğe doğru sıralanınca PSO, ASO, GWO, ABC, SSA, WOA, SCA, MFO

algoritmaları ile optimize edilmiş FOPID kontrolörler kullanılan gaz türbinli elektrik güç santral sisteminin çıkış işaretlerinde sırasıyla saniye cinsinden 0.30, 0.41, 0.45, 0.50, 0.66, 0.67, 0.70, 0.81 olduğu görülür. Yükselme süresi karşılaştırmalarının daha net yapılabilmesi için Şekil 4.2, Şekil 4.3, Şekil 4.4 ve Şekil 4.5'te gösterilmiş olan değerler Tablo 4.1 ve Tablo 4.2'de toplu halde verilmiştir.

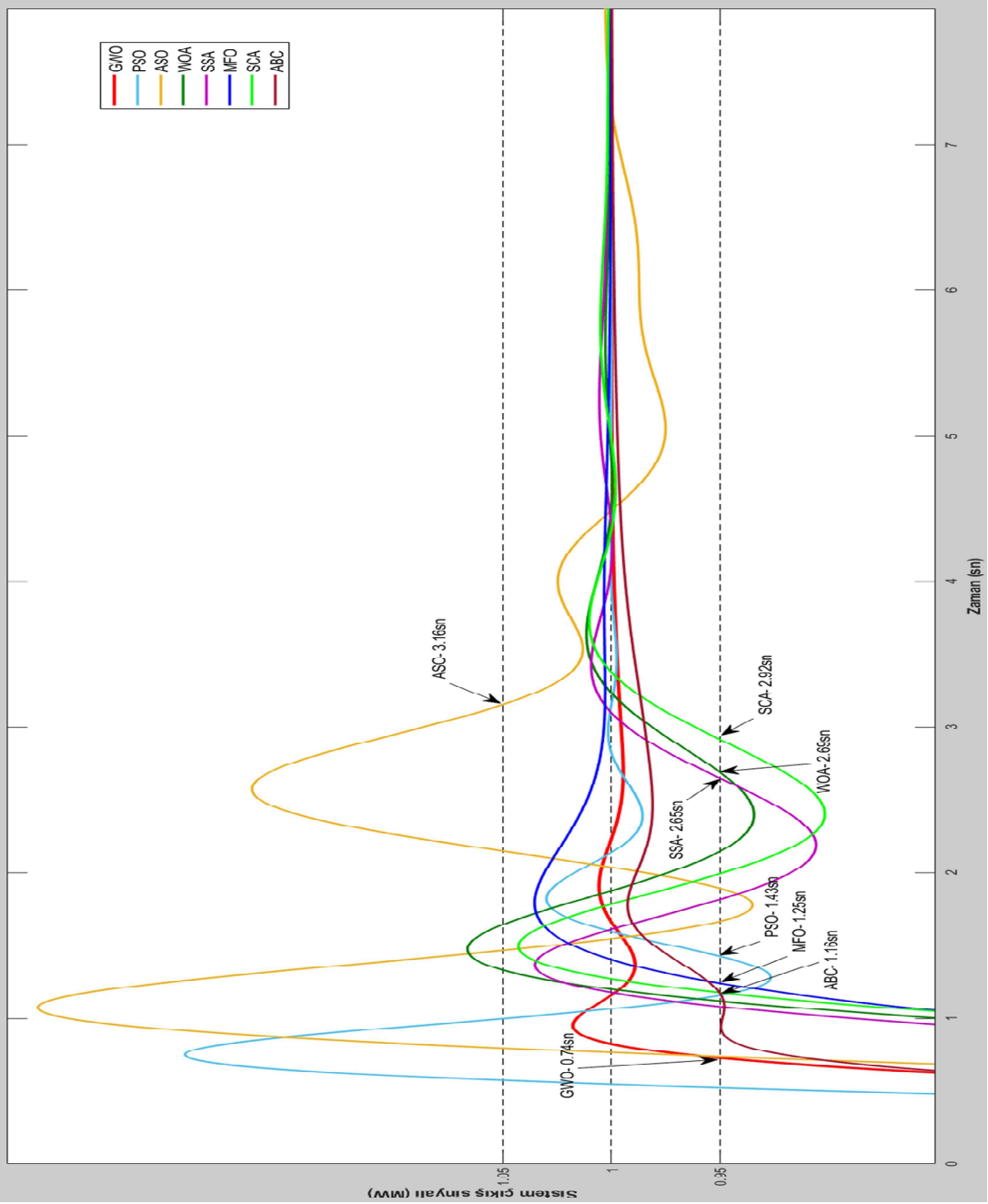


Şekil 4.5: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yükselme zamanları (%10-%90)

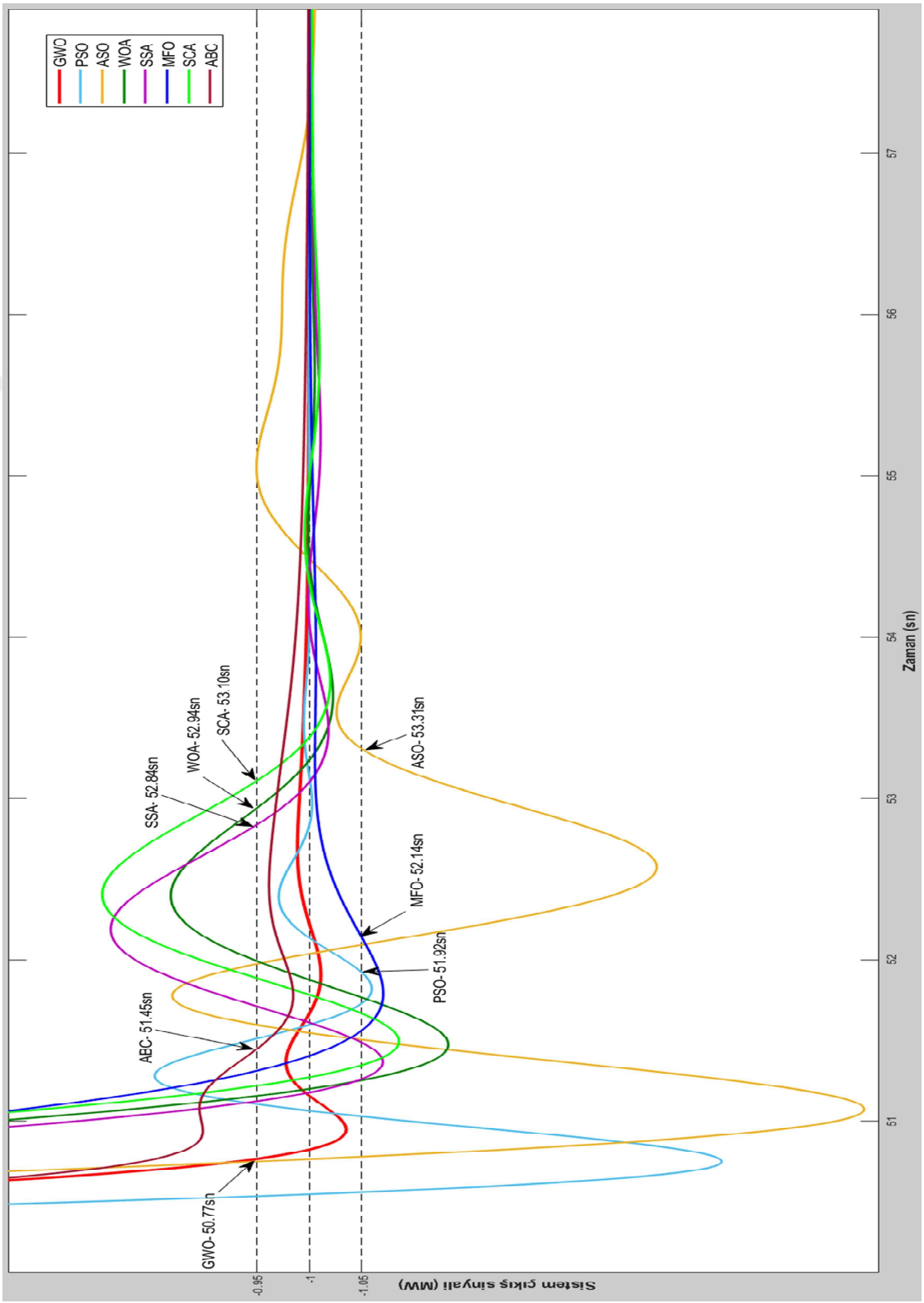
Yerleşme süreleri değerlendirilirken yaygın kullanılan %5 ve %2'lik tolerans aralıkları esas alınmıştır, böylece farklı tolerans seviyelerinde PSO, ASO, GWO, SSA, WOA, SCA, MFO, ABC algoritmaları ile optimize edilmiş FOPID kontrolörler kullanılan gaz türbinli elektrik güç santral sisteminin çıkış işaretleri, yerleşme süreleri açısından karşılaştırılmıştır. İlk olarak %5'lik tolerans bandı ele alındığında; pozitif bölgede üst sınır 1.05 MW, alt sınır 0.95 MW negatif bölgede ise üst sınır -0.95, alt sınır -1.05 olarak belirlenmiştir. %2'lik tolerans bandında ise pozitif bölgede üst sınır 1.02 MW, alt sınır 0.98 MW ve negatif bölgede üst sınır -0.98 MW, alt sınır -1.02 MW olarak belirlenmiştir. Bu yaklaşımlardan yola çıkılarak sistem çıkış işaretlerinin yerleşme süreleri Şekil 4.6 ve Şekil 4.8'de pozitif bölgede, Şekil 4.7 ve Şekil 4.9'da negatif bölgede oklar ile gösterilmiştir.

Pozitif bölgede %5'lik bant aralığı değerlendirildiğinde Şekil 4.6'da verilmiş olan çıkış işaretlerinin yerleşme süreleri küçükten büyüğe doğru sıralandığında ve buna göre sistemin optimize edildiği algoritmaların sıralaması yapıldığında GWO (0.74 sn), ABC (1.16 sn), MFO (1.25 sn), PSO (1.43 sn), SSA (2.65 sn), WOA (2.69 sn), SCA (2.92 sn), ASO (3.16 sn) dizilimi elde edilmiştir. Negatif bölgede %5'lik bant aralığı değerlendirildiğinde Şekil 4.7'da verilmiş olan çıkış işaretlerinin yerleşme süreleri küçükten büyüğe doğru sıralandığında ve buna göre sistemin optimize edildiği algoritmaların sıralaması yapıldığında GWO (0.77 sn), ABC (1.45 sn), PSO (1.92 sn), MFO (2.14 sn), SSA (2.84 sn), WOA (2.94 sn), SCA (3.10), ASO (3.31 sn) dizilimi elde edilmiştir. Pozitif bölgede %2'lik bant aralığı değerlendirildiğinde Şekil 4.8'da verilmiş olan çıkış işaretlerinin yerleşme süreleri küçükten büyüğe doğru sıralandığında ve buna göre sistemin optimize edildiği algoritmaların sıralaması yapıldığında GWO (0.78 sn), ABC (1.50 sn), PSO (1.97 sn), MFO (2.26 sn), SSA (2.88 sn), WOA (2.99 sn), SCA (3.15 sn), ASO (5.39 sn) dizilimi elde edilmiştir. Negatif bölgede %2'lik bant aralığı değerlendirildiğinde Şekil 4.9'da verilmiş olan çıkış işaretlerinin yerleşme süreleri küçükten büyüğe doğru sıralandığında ve buna göre sistemin optimize edildiği algoritmaların sıralaması yapıldığında GWO (1.44 sn), MFO (2.55 sn), PSO (2.57 sn), SSA (2.97 sn), SCA (3.25 sn), ABC (3.54 sn), WOA (3.83 sn), ASO (6.63 sn) dizilimi elde edilmiştir.Yerleşme süresi kıyaslamalarının net şekilde yapılabilmesi için çıkış işaretlerinden elde edilen yerleşme süresi verileri Tablo 4.1 ve Tablo 4.2'de toplu halde verilmiştir. Bu tezde incelenen sekiz optimizasyon algoritması ile tasarlanmış FOPID kontrolörlerin uygulandığı sistemlerin çıkış işaretleri incelendiğinde ABC algoritması ile optimize edilmiş kontrolörün uygulandığı sistem çıkış işaretinin geçici rejimde, kalıcı hal değerini aşmadığı için aşırı sönümlü (overdamped) gibi bir özellik gösterdiği diğer

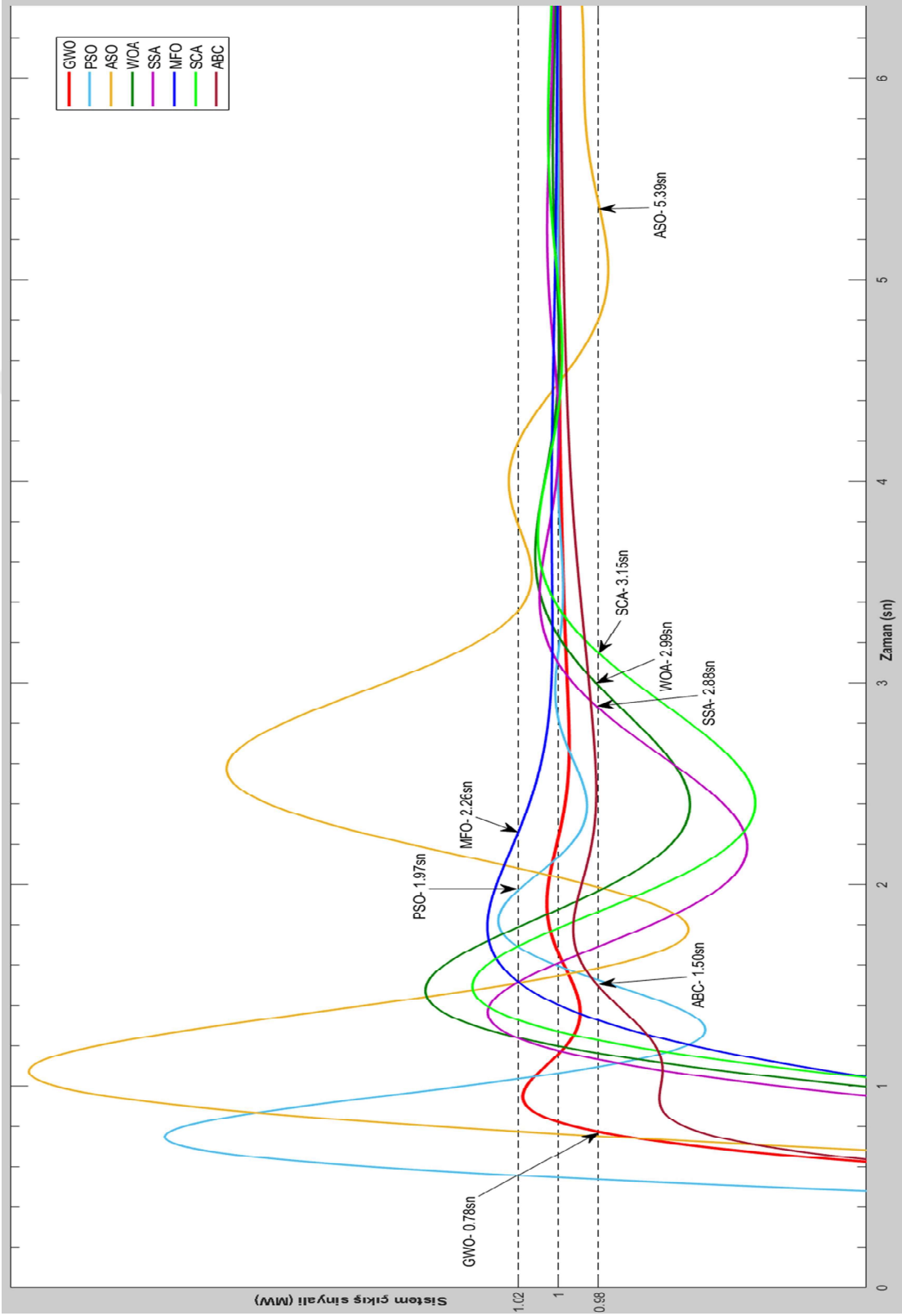
yedi yöntemin ise geçici rejimde, kalıcı durum değerini aştığı için sönümlü (underdamped) özellik gösterdiği anlaşılmaktadır [1, 2, 4, 40]. Bunların yanı sıra GWO algoritması pozitif bölgede aşım yapmasına rağmen yerleşme süreleri için tanımlanmış %2 ve %5'lik tolerans bandına bir kez girdikten sonra tekrar çıkmadığı için literatürde en iyi sönümlü ya da optimum sönümlü (optimally dumped) olarak da adlandırılan davranış göstermektedir [80, 81].



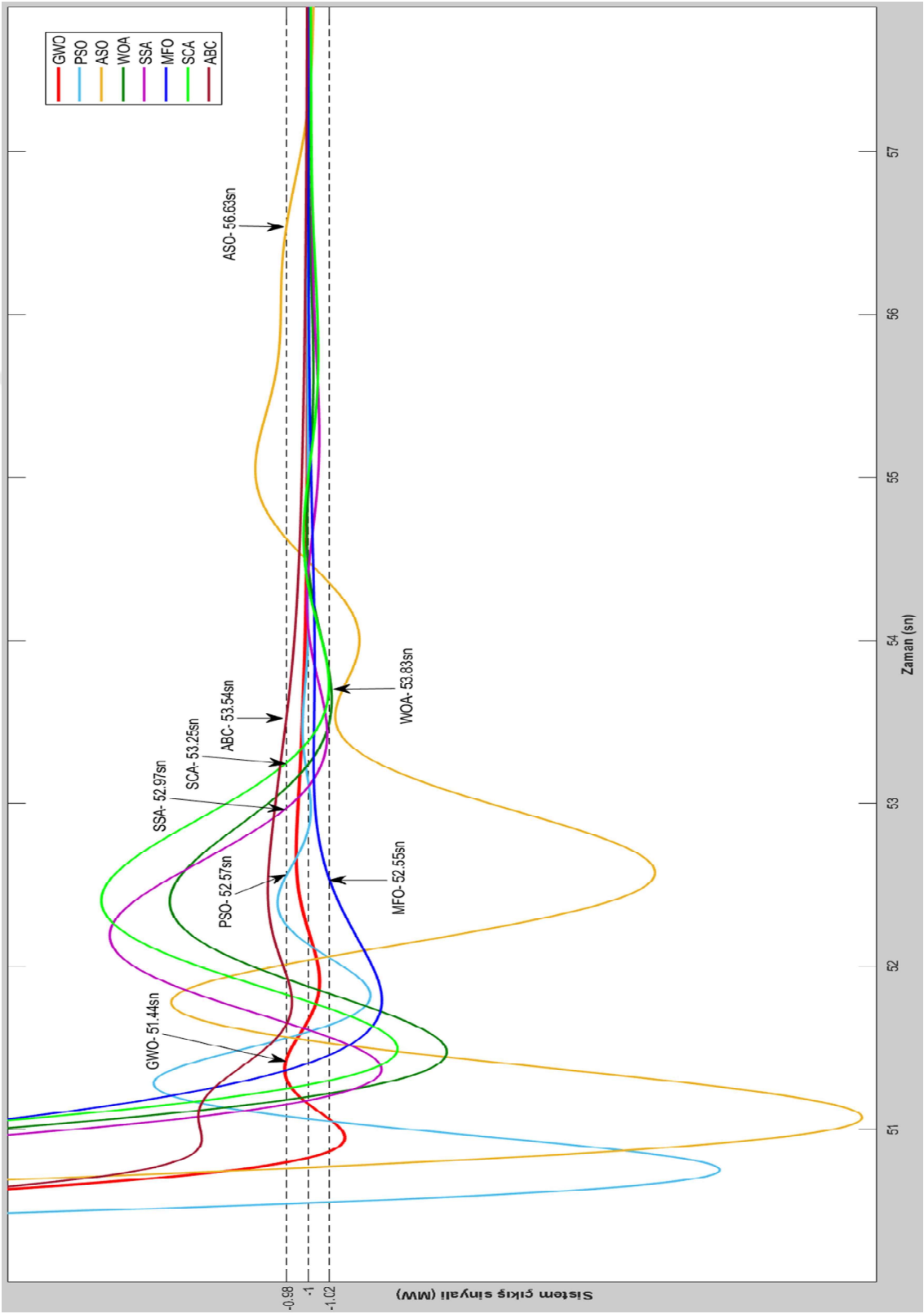
Şekil 4.6: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yerleşme zamanları (%5)



Şekil 4.7: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yerleşme zamanları (%5)



Şekil 4.8: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin pozitif bölgedeki çıkış işaretleri ve yerleşme zamanları (%2)



Şekil 4.9: Sezgisel optimizasyon algoritmaları tabanlı FOPID kontrolör ile denetlenen elektrik güç santral sisteminin negatif bölgedeki çıkış işaretleri ve yerleşme zamanları (%2)

Dördüncü bölüm olan bulgular başlığı altında verilmiş grafiklerden toplanan bilgiler derlenerek aşağıda verilmiş olan Tablo 4.1 ve Tablo 4.2 oluşturulmuştur.

Tablo 4.1: Pozitif bölge geçici hal kriter ve değerleri (0-50sn)

Optimizasyon Algoritması	Yükselme süresi %0-%100	Yükselme süresi %10-%90	Yerleşme süresi %2	Yerleşme süresi %5	Yüzde en büyük aşım
PSO-FOPID	0.55 sn (0-0,55)	0,31 sn (0,19-0,50)	1.97 sn	1,43 sn	% 19,62
ABC-FOPID	10,59 sn (0-10,59)	0,50 sn (0,21-0,71)	1.50 sn	1,16 sn	% 0
GWO-FOPID	**0,84 sn (0-0,84)**	**0,45 sn (0,22-0,67)**	**0.78 sn**	**0,74** sn	**% 1,76**
MFO-FOPID	1,41 sn (0-1,41)	0,82 sn (0,32-1,14)	2.26 sn	1,25 sn	% 3,52
WOA-FOPID	1,20 sn (0-1,20)	0,67 sn (0,39-1,06)	2.99 sn	2,69 sn	% 6,65
SSA-FOPID	1,18 sn (0-1,18)	0,65 sn (0,37-1,02)	2.88 sn	2,65 sn	% 3,51
SCA-FOPID	1,28 sn (0-1,28)	0,70 sn (0,41-1,11)	3.15 sn	2,92 sn	% 4,26
ASO-FOPID	0,77 sn (0-0,77)	0,41 sn (0,30-0,71)	5.39 sn	3,16 sn	% 26,40

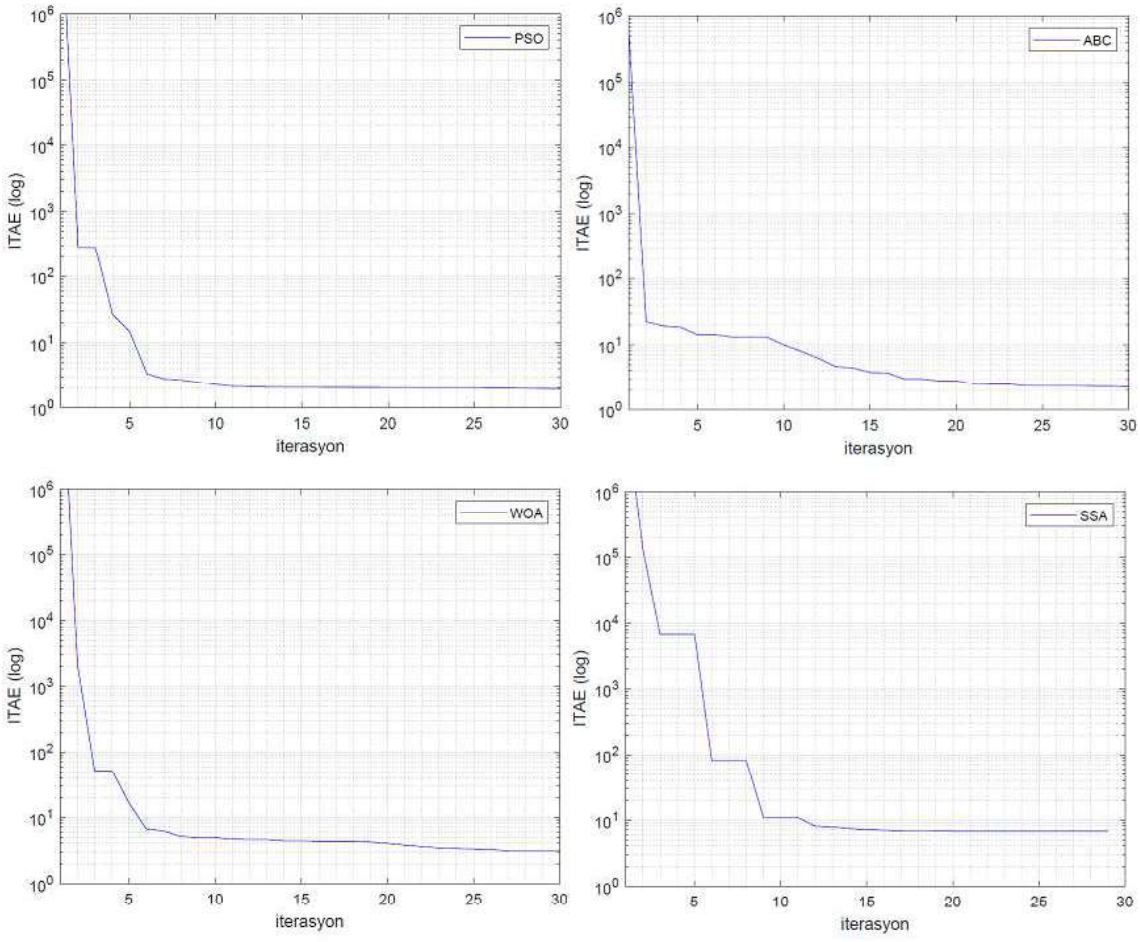
Tablo 4.2: Negatif bölge geçici hal kriter ve değerleri (50-100sn)

Optimizasyon Algoritması	Yükselme süresi %0-%100	Yükselme süresi %10-%90	Yerleşme süresi %2	Yerleşme süresi %5	Yüzde en büyük aşım
PSO-FOPID	0,55 sn (50-50,55)	0,30 sn (50,20- 50,50)	2.57 sn	1,92 sn	% 39,3
ABC-FOPID	11,49 sn (50-61,49)	0,50 sn (50,21- 50,71)	3.54 sn	1,45 sn	% 0
GWO-FOPID	**0,84 sn (50-50,84)**	**0,45 sn (50,22- 50,67)**	**1.44 sn**	**0,77 sn**	**% 3,51**
MFO-FOPID	1,40 sn (50-51,40)	0,81 sn (50,33- 51,14)	2.55 sn	2,14 sn	% 7,1
WOA-FOPID	1,20 sn (50-51,20)	0,67 sn (50,39- 51,06)	3.83 sn	2,94 sn	% 13,0
SSA-FOPID	1,18 sn (50-51,18)	0,66 sn (50,36- 51,02)	2.97 sn	2,84 sn	% 6,82
SCA-FOPID	1,28 sn (50-51,28)	0,70 sn (50,41- 51,11)	3.25 sn	3,10 sn	% 8,4
ASO-FOPID	0,77 sn (50-50,77)	0,41 sn (50,30- 50,71)	6.63 sn	3,31 sn	% 51,2

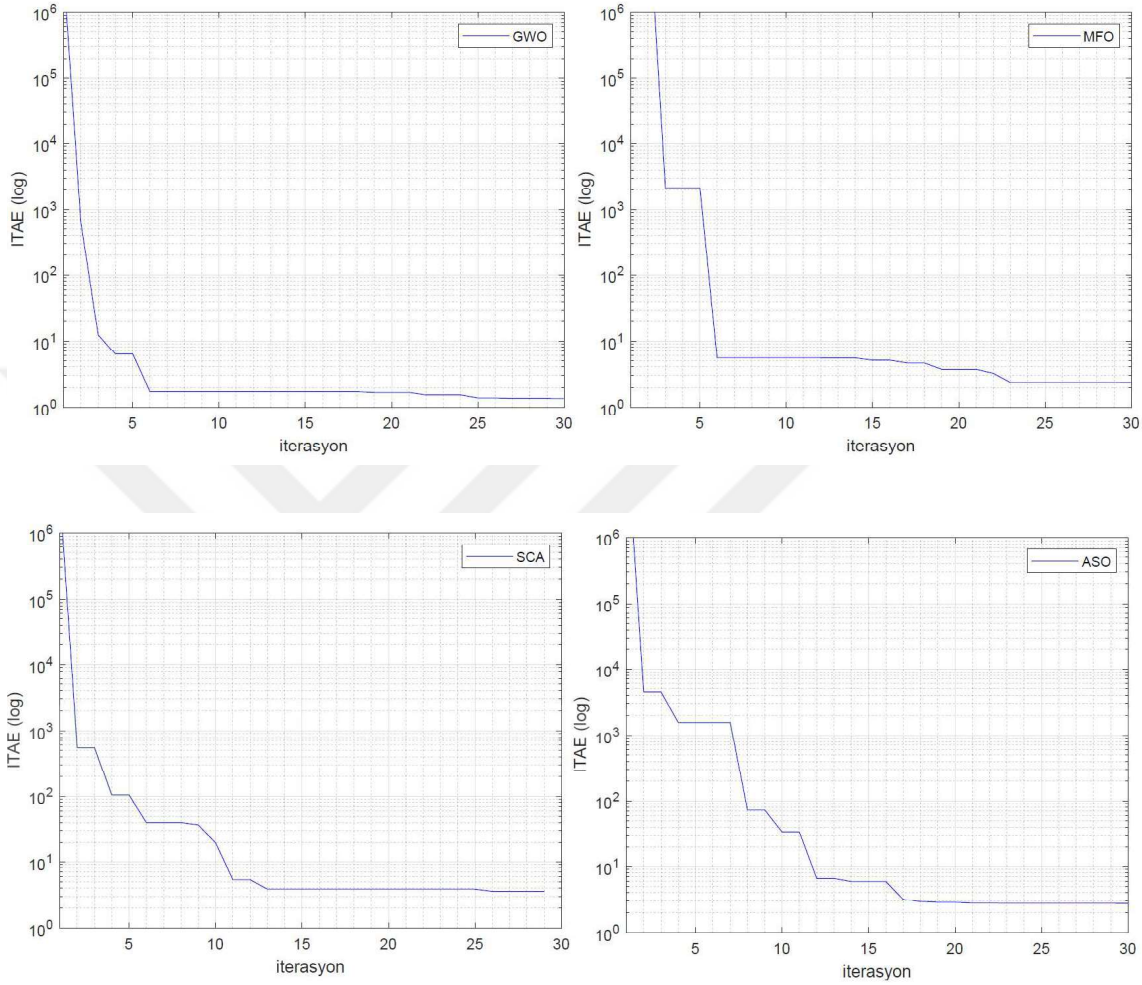
Burada tablolar incelendiğinde GWO algoritması ile optimize edilmiş olan FOPID kontrolöre sahip gaz türbinli elektrik santrali için cevap eğrisinin yerleşme süresi açısından hem pozitif hem de negatif bölgede diğerlerine kıyasla daha iyi sonuçlar verdiği görülmektedir. Aşım miktarları değerlendirildiğinde hem pozitif hem de negatif bölgede ABC algoritması ile ede edilen FOPID kontrolör parametrelerinin uygulandığı sistem çıkış eğrisi değerlerinin diğerlerinden daha üstün olduğu, en az aşım değerine ABC (%0) algoritması kullanıldığında ulaşıldığı görülmektedir. GWO algoritması ile optimize edilmiş olan FOPID kontrolöre sahip gaz türbinli elektrik santal sistemi için cevap eğrisinin yüzde en büyük aşım değeri pozitif bölgede %1,76, negatif bölgede %3,51 ile ikinci en az aşıma sahip olduğu görülmektedir. Bu durumda aşım istenmediği durumlarda ABC algoritması ile elde edilen FOPID kontrolör parametrelerinin daha uygun olduğu görülmektedir. Yükselme süresi kriteri açısından Tablo 4.1 ve Tablo 4.2 incelendiğinde, hem pozitif hem de negatif bölgede PSO algoritması ile

optimize edilen sistem çıkış işaretine ait değerlerin diğerlerine göre daha küçük olduğu görülmektedir.

Bu tezde kullanılmış olan sekiz optimizasyon algoritmasının çalışması sırasında algoritmanın optimize edilmiş değerlere yaklaşım eğrileri yani amaç fonksiyonu ITAE sıfıra yaklaşırken ki durumları Şekil 4.10 ve Şekil 4.11'de verilmiştir. Her bir iterasyonda ITAE değerini gösteren, Şekil 4.10 ve Şekil 4.11'deki yaklaşım eğri grafikleri incelendiğinde ITAE amaç fonksiyonu değerinin 0'a en hızlı ve en çok yaklaşan algoritmanın GWO olduğu görülebilmektedir. ITAE değerinin sıfıra yakınlaşması sistem çıkış eğrisinin, kontrolör referans eğrisine benzerliğinin arttığını göstermektedir. Bu durumda ITAE değerinin sıfıra ne kadar hızlı yaklaştığı, optimizasyon algoritmasının amacına ne kadar hızlı ulaştığının bir göstergesidir.



Şekil 4.10: Optimizasyon algoritmaları yaklaşım eğrileri (PSO, ABC, WOA, SSA)



Şekil 4.11: Optimizasyon algoritmaları yaklaşım eğrileri (GEO, MFO, SCA, ASO)

5 TARTIŞMA VE SONUÇ

Günümüzde bir geminin otomatik dümen kontrol sisteminin istenilen hedefe en kısa yoldan ulaşması için; hava şartları, akıntı vb. birçok etkene rağmen rotadan sapmaması beklenir. Aynı durum otomobillerdeki hız sabitleyici sistemler için de geçerlidir. Otomobil ile yokuş yukarı ya da aşağı giderken fiziki koşullardan etkilenmeden aracın hızını sabit tutması beklenir. Benzer durum bu tez çalışmasında ele alınan gaz türbinli güç santrali için de geçerlidir. Santralin beslediği yük değişimlerinin bir sonucu olan santral çıkış frekansındaki değişimler istenmemekte, frekansın sabit olması ve tüketilen elektriksel güç ile üretilen elektriksel güç farkının sıfır olması beklenmektedir. Değişkenlik gösteren yük karşısında elektrik santral sistemi ve elektrik hattının zarar görmemesi için güç santrali kontrolörünün değişimlere hızlı şekilde cevap vererek, belirlenen tolerans bandının dışına çıkmaması, çıkarsa da en kısa sürede tekrar tolerans bandına ulaşması ve güç seviyelerini dengede tutması gerekmektedir. Bu gibi kritik denetleme işlemlerinin hassas bir şekilde yapılabilmesi için iyi optimize edilmiş kontrol sistemlerinin kullanılması gerekmektedir.

Literatürdeki çalışmalar incelendiğinde kontrol sistemlerinin performansının arttırılması için klasik PID kontrolörler yerine FOPID kontrolörler kullanılabileceği anlaşılmaktadır. Aynı zamanda parametre optimizasyonunun da kontrolör performansını direkt etkilediği bilinmektedir. Bu tezde FOPID kontrolörün optimizasyonunda kullanmak için modern metasezgisel optimizasyon algoritmalarından PSO, ABC, GWO, MFA, WOA, SSA, SCA, ASO incelenmiş, örnek model olarak ele alınan FOPID kontrolör içeren basitleştirilmiş gaz türbinli elektrik güç santral modeline uygulanmışlardır. Bu çalışmalar sırasında elde edilen bilgilere göre oluşturulan Tablo 4.1 ve Tablo 4.2'de verilen değerler kontrolöre uygulanan 50 sn süresince +1MW ve 50 sn süresince -1MW ile elde edilmiştir. Bu durumda FOPID kontrolör ile denetlenen sistemin pozitif bölgedeki çıkış eğrileri ile negatif bölgedekiler kendi aralarında kıyaslanmış ve hem pozitif hem de negatif bölgedeki zaman tanım bölgesi geçici rejim parametreleri açısından incelenmiştir.

Burada Tablo 4.1 ve Tablo 4.2 göz önünde bulundurularak ilk olarak yükselme süresi kriteri ele alındığında hem pozitif bölgede hem de negatif bölgede en kısa yükselme süresi PSO algoritması ile optimize edilmiş olan kontrolörün bulunduğu sistem çıkış eğrisine aitken sırasıyla ikinci ve üçüncü en kısa sürelere sahip olan kontrolörlerin kullanıldığı sistemlerin çıkış eğrileri ASO ve GWO algoritmalarıdır. Bu üç algoritma ile optimize edilen kontrolörün

kullanıldığı sistem çıkış eğrilerinin yükselme süreleri hem %0-%100 hem de %10-%90 yaklaşımlarına göre değerlendirildiğinde aynı sıralamaya sahiptirler. Diğer yandan üçünün de (PSO, ASO, GWO) yükselme süreleri birbirlerine oldukça yakındır. Bunların dışında ele alınan diğer algoritmalar ile optimize edilen kontrolörlerin kullanıldığı sistemlerin çıkış eğrilerine bakıldığında yükselme sürelerinin bu üçüne kıyasla daha fazla olduğu anlaşılabilmektedir.

Yerleşme süresi kriteri açısında sistem çıkış eğrileri incelendiğinde %2'lik ve %5'lik tolerans bandı kullanılarak pozitif ve negatif bölgede en kısa yerleşme süresine GWO ile optimize edilmiş olan FOPID kontrolörün uygulandığı sistem sahiptir. Pozitif bölgede %2, %5 ve negatif bölgede %5'lik bant aralığındaki çıkış eğrileri incelendiğinde GWO ile optimize edilmiş FOPID kontrolörün uygulandığı sistem çıkış eğrisinin bu bant aralığına girdikten sonra tekrar çıkmadığı ancak biraz aşım yaptığı böylece litaratürde optimally damped olarak adlandırılan bir davranış sergilediği görülmektedir.

Son olarak bu tezde kullanılan üçüncü değerlendirme kriteri olan aşım miktarına gelindiğinde, hem pozitif hem de negatif bölgede en fazla aşımın ASO ile tasarlanan FOPID kontrolörün uygulandığı sistem çıkış eğrisi olduğu dikkat çekerken; en az aşımın, ABC algoritması ile optimize edilmiş FOPID kontrolörün uygulandığı sistem çıkış işaretinde görülmüş ve hatta hiç aşım yapmadığı anlaşılmıştır. PSO algoritması ile optimize edilmiş FOPID kontrolörün kullanıldığı sistem çıkış eğrisinde yükselme süreleri diğer algoritmalar ile optimize edilenlere kıyasla performans açısından önde gelenlerden iken; yüzde en büyük aşım değerinin oldukça büyük olması (pozitif bölgede:19.62 sn , negatif bölgede:39.30 sn) PSO algoritması uygulanan sistem performansında olumsuz bir gösterge olmuştur. Yerleşme süresi kriterinde diğerlerine göre üstün performansı ile öne çıkan GWO algoritması ile optimize edilmiş olan FOPID kontrolörün uygulandığı sistem çıkış işareti ele alındığında pozitif bölgedeki aşım miktarlarının %2 ve %5'lik tolerans bantları içerisinde kaldığı, negatif bölgede %3.51'lik yüzde en büyük aşım değeri ile %5'lik bant aralığını aşmadığı ancak %2'lik bant içerisinde kalmadığı görülmektedir. Aşım miktarı istenmeyen ve yerleşme süresinin bir miktar daha uzun olabileceği durumlarda ABC algoritması ile optimize edilmiş FOPID kontrolörün uygulandığı sistem tercih edilebilir. Bu tezde kullanılmış olan sekiz algoritma içinde GWO ve ABC algoritmaları öne çıkmaktadır.

Şekil 4.10 ve Şekil 4.11'de verilmiş olan grafiklerde optimizasyon algoritmalarının yaklaşım eğrileri incelendiğinde GWO algoritmasının ilk iterasyonlarda doğrudan ITAE amaç

fonksiyonu değerini 0'a yaklaştırma eğiliminde olduğu görülmektedir. İterasyonlar tamamlandığında hatanın 0'a çok yaklaştığını ve diğer algoritmalara göre ITAE değerini daha çok azaltabildiği anlaşılabilmektedir. Bu gözlemler doğrultusunda GWO algoritmasının bu tezde ele alınan diğer 7 algoritmadan daha hızlı ve daha iyi sonuç verdiğinin bir diğer göstergesidir.

Simülasyonlar sonucunda görülmüştür ki GWO ile optimize edilmiş FOPID kontrolör gaz türbinli elektrik santraline uygulandığında oluşan yük değişimlerine vereceği hızlı cevaplar ile yakıt tüketimindeki dengesizlikleri azaltabileceği, verim artışı sağlayabileceği, santral çıkışındaki frekans değişimlerinden dolayı oluşabilecek istenmeyen harmonikleri minimum seviyeye indirerek operasyonel riskleri azaltabileceği öngörülmektedir.

FOPID kontrolörün optimizasyonu için metasezgisel algoritmaların tercih edilmesi birçok avantajı beraberinde getirmektedir. FOPID kontrolörlerin aslında bir genelleme olmasından kaynaklı "λ, μ" parametrelerinin değişken olması PID, PI, PD kontrolör çeşitlerinin de kullanılma olasılıklarının aynı anda değerlendirilmesine imkan tanır. Örneğin bir sistemde PI kontrolör kullanılması daha uygunsa optimizasyon algoritma çıktılarında μ değeri 0'a yakın bir değer alacaktır. Aynı şekilde PID kontrolörün kullanılması uygun ise λ, μ değerleri 1'e yakın değerler alacaktır (PID kontrolör için λ=1, μ=1, PI kontrolör için λ=1, μ=0). Sezgisel algoritmalar ile FOPID kontrolörler birlikte kullanıldığında kazanç, integral ve türev etkilerinden oluşabilecek bütün kontrolörlerin sisteme uygunlukları test edilmiş olur. Böylelikle hangi kontrolörün kullanılmasının daha uygun olduğu metasezgisel optimizasyon algoritması tarafından tespit edilebilir. Bu durum sisteme uygulanacak olan kontrolörün yükselme süresi, yerleşme süresi, aşım miktarı kriterlerine göre en iyi performansa ulaşmasına yardımcı olur.

Sonuç olarak; FOPID kontrolörün optimizasyonu, örnek olarak ele alınan basitleştirilmiş gaz türbinli elektrik güç santrali modeli için gri kurt optimizasyon algoritması (GWO) veya yapay arı kolonisi (ABC) algoritması ile yapıldığında başarılı sonuçlar alınmıştır. Her iki algoritma ile tasarlanan kontrolörlerin uygulandığı sistem girişindeki değişimlere verdiği cevap işaretinin geçici hal değerlendirme kriterleri göz önünde bulundurulduğunda, basitleştirilmiş gaz türbinli elektrik santrali modeline uygun ve yüksek performanslı bir kontrolör tasarlanabildiği anlaşılmaktadır. Algoritma amaç fonksiyonlarının iterasyonlar sırasındaki sıfıra yaklaşım

eğrileri değerlendirildiğinde GWO algoritmasının diğer algoritmalardan daha hızlı yaklaşım gösterdiği ve sıfıra en yakın değere ulaştığı görülmüştür.

İleriye dönük çalışmalar olarak, yeni bir metasezgisel optimizasyon algoritması oluşturulması, makine öğrenmesi yöntemlerinin kontrol sistemleri optimizasyonuna uygulanması gibi çalışmalar yapılması düşünülmektedir.

6 KAYNAKLAR

[1] Dorf R. ve Bishop R., 2017, Modern Control Systems, Boston: University of California, pp. 5-9.

[2] Lewis F., 1992, Applied Optimal Control & Estimation & Digital Design & Implementation, The University of Texas at Arlington, pp. 4-9.

[3] Mayr O., 1970, The Origins of Feedback Control, Cambridge: The Massachusetts Institute of Technology, pp. 53-115.

[4] Bishop R., 2000, The Machatronics Hand Book, Texas: The University of Texas at Austin, pp. 18-25.

[5] Bennett S., 2000, The Past of PID Controllers, *IFAC Proceedings Volume,* cilt 33, no. 4, pp. 1-11.

[6] Minorsky N., 1922, Directional Stability of Automatically Steered Bodies, *Journal of the American Society for Naval Engineers,* cilt 34, no. 2, pp. 280-309.

[7] Bennett S., 1984, Nicolas Minorsky and the Automatic Steering of Ships, *IEEE Control Systems Magazine,* cilt 4, no. 4.

[8] Mahmood A. ve Mohammed B., 2019, Digital Fractional Order PID Controller Design And Realization, University of Mosul, Mosul, https://www.researchgate.net/publication/332333112.

[9] Machado J., Kiryakova V. ve Mainardi F., 2011, Recent history of fractional calculus, *Communications in Nonlinear Science and Numerical Simulation,* cilt 16, no. 3, pp. 1140-1153.

[10] Machado J. ve Kiryakova V., 2019, Recent history of the fractional calculus: data and statistics, *Basic Theory,* pp. 1-22.

[11] Podlubny I., 2003, Fractional-Order Systems and FOPID-Controllers, *Inst. Exp. Phys., Slovak Acad. Sci.,* cilt 44, no. 1.

[12] Valerio D. ve Costa J., 2005, Ziegler-Nichols type tuning rules for fractional PID controllers, *Proceedings of IDETC/CIE 2005*, Long Beach, California, USA.

[13] Basu A., Mohanty S. ve Sharma R., 2017, Designing of the PID and FOPID Controllersusing Conventional Tuning Techniques, ITM University.

[14] Ziegler J. ve Nichols N., 1942, Optimum settings for automatic controllers, Transactions of the ASME.

[15] Kumar A. ve Garg K., 2015, Comparison of Ziegler-Nichols, CohenCoon and fuzzy logic controllers for heat exchanger model, *International Journal of Science, Engineering and Technology Research,* cilt 4, no. 6.

[16] Astrom K. ve Hagglund T., 1995, PID Controllers: Theory, Design and Tuning, The Instrumentation, Systems and Automation Society (ISA).

[17] Monje C., Vinagre B., Felui V. ve Chen Y., 2006, On AutoTuning of Fractional Order FOPID Controllers, *2nd IFAC Proceedings Volume*, Porto.

[18] Chen Y., Dou H., Vinagre B. ve Monje C., 2006, Robust Tuning Method for Fractional Order PI Controllers, *2nd IFAC Proceedings Volume*, Porto.

[19] Valerio D. ve Costa S. J., 2006, Tuning-Rules for Fractional PID Controllers, *2nd IFAC Proceedings Volume*, Porto.

[20] Caponetto R. ve Fortuna L., 2002, Parameter Tuning of a Non Integer Order PID Controller, University of Catania,System and Control Group, Catania, İtalya.

[21] Keyser R., Muresal C. ve Ionescu C., 2018, Autotuning of a Robust Fractional Order PID Controller, *IFAC PapersOnLine 51-25*.

[22] Gül O. ve Tan N., 2019, Application of Fractional-Order Voltage Controller in Building-Integrated Photovoltaic and Wind Turbine System, *Measurement and Control,* pp. 1-14.

[23] Malek H., 2014, Control of Grid-Connected Photovoltaic Systems Using Fractional Order Operators, Doktora, Elektrik Müh., Utah State University.

[24] Göktürk E., 2009, Kesirli PID Tasarım Yöntemi ve Klasik PID ile Karşılaştırmalar, Yüksek Lisans, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul.

[25] Topljakov A., Alagöz B., Yeroğlu C., Gonzalez E., Hosseinnia H. ve Petlenkov E., 2018, FOPID controllers and their industrial applications: a survey of recent Results, *IFAC PapersOnLine,* cilt 51, no. 4, pp. 25-30.

[26] Budak E., 2018, Multi-Objective Optimization Based Fractional Order PID Controller Design, Yüksek Lisans, İstanbul Teknik Üniversitesi, İstanbul.

[27] Usta A., 2019, Meta-Sezgisel Optimizasyon Algoritmalarinin Grup Asansör Kontrol Optimizasyon Probleminin Çözümündeki Performanslarinin Değerlendirilmesi, ,Yüksek Lisans, Karabük Üniversitesi.

[28] Kennedy J. ve Eberhart R., 1995, Particle Swarm Optimization, *Proceedings of ICNN'95* , Perth, WA, Australia.

[29] Karaboğa D., 2005, An Idia Based on Honey Bee Swarm For Numerical Optimization, Erciyes University, Kayseri.

[30] Mirjalili S., Mirjalili S. ve Lewis A., 2014, Grey Wolf Optimizer, cilt 69, pp. 46-61.

[31] Mirjalili S., 2015, Moth-Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm, *Knowledge-Based Systems,* cilt 89, pp. 228-249.

[32] Mirjalili S. ve Lewis A., 2016, The Whale Optimization Algorithm, *Advances in Engineering Software,* cilt 89, pp. 51-67.

[33] Mirjalili S., Gandomi A. H., Mirjalili S., Saremi S., Faris H. ve Mirjalili S., 2017, Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems, *Advances in Engineering Software,* cilt 114, pp. 163-191.

[34] Mirjalili S., 2016, A Sine Cosine Algorithm for Solving Optimization Problems, *Knowledge-Based Systems,* cilt 96, pp. 120-133.

[35] Zhao W., Wang L. ve Zhang Z., 2018, A Novel Atom Search Optimization for Dispersion Coefficient Estimation in Groundwater, *Future Generation Computer Systems,* cilt 91, pp. 601-610.

[36] Holland J., 1975, Adaptation in natural and artificial systems, University of Michigan Press, p. 211.

[37] McCall J., 2005, Genetic Algorithms for Modelling and Optimisation, *Journal of Computational and Applied Mathematics,* cilt 184, p. 205–222.

[38] Çiftkaya B., 2010, Elektrik üretiminde kullanılan gaz türbinlerinin incelenmesi ve benzetimi, Doktora, Fen Bilimleri Enstitüsü,İstanbul Teknik Üniversitesi.

[39] Arıkuşu Y., Bayhan N. ve Tiryaki H., 2019, Comparison of Conventional and Modern Controllers for a Gas Turbine Power Plant, *Electrica,* cilt 2019;09, pp. 146-157.

[40] Ogata K., 2010, Modern Control Engineering, Pearson, pp. 567-641.

[41] http://www.dataforth.com, Dataforth Corporation, [Çevrimiçi].

[42] Karahan M., 2019, Dört rotorlu bir insansız hava aracının modellenmesi ve PID kontrolcü tasarlanması, Yüksek Lisans, Fen Bilimleri Enstitüsü, TOBB Üniversitesi.

[43] Yıldız N., 2019, Makine öğrenimi tabanlı PID optimizasyonu ile çift aktif köprülü doğru akım güç çeviricilerinin denetimi, Yüksek Lisans, Fen Bilimleri Enstitüsü, Pamukkale Üniversitesi, Denizli.

[44] Arpacı H., 2018, Yapay Sinir Ağları ile Kesir Dereceli PID Denetleyici Katsayılarının Ayarlanması, cilt 9, no. 156-173.

[45] www.teias.gov.tr, TEIAS, [Çevrimiçi]. Erişim: https://webapi.teias.gov.tr/file/080af670-d5cb-4e53-b188-11a187da2037.

[46] Tiryaki H., 2013, Modern Kontrol Yöntemlerinin Yük Dağıtım Sistemlerinde Uygulanması, Doktora, Fen Bilimleri Enstitüsü, İstanbul Üniversitesi, İstanbul.

[47] Sezer K.S., Bayhan N. ve Tiryaki H., 2020, Elektrik Santrali için Metasezgisel Algoritmalar ile Kesir Dereceli PID Kontrolör Tasarımı, *EFIS 2020 proceedıngs*, Zonguldak.

[48] Kocaarslan İ., Tiryaki H., Bal E. ve Gümüş K., 2012, Modern Optimizasyon Yöntemleri (ABC, PSO) ile Yük-Frekans Kontrolü, Bursa.

[49] Rowen W., 1995, https://ieeexplore.ieee.org/document/472832, IEEE, London, UK, Dynamic response characteristics of heavy duty gas turbines and combined cycle systems in frequency regulating duty.

[50] Demirören A., Zeynelgil L., 2004, Elektrik Enerji Sistemlerinin Kararlılığı Kontrolü ve Çalışması, Istanbul: Birsen Yayınevi.

[52] MATLAB 2020a Reference Manual, 2020. (Lisans numarası: 40914345)

[53] Tepljakov A., Petlenkov E. ve Belikov J., 2011, FOMCON: Fractional-Order Modeling and Control Toolbox for MATLAB,*18th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2011*, Gliwice, Poland.

[54] Tepljakov A., Petlenkov E. ve Belikov J., 2011, FOMCON: a MATLAB Toolbox for Fractional-order System Identification and Control,*18th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2011*, Tallinn, Estonia.

[55] Zhao W., Wang L. ve Zhang Z., 2018, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowledge-Based Systems,* cilt 163, pp. 283-304.

[56] Ekinci S., Demirören A., Zeynelgil H. ve Hekimoğlu B., 2020, An opposition-based atom search optimization algorithm for automatic voltage regulator system, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi,* cilt 35, no. 3, pp. 1141 - 1158.

[57] Eberhart R. ve Kennedy J., 1995, A new optimizer using particle swarm theory, *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science,* pp. 39-43.

[58] Van den Bergh F. ve Engelbrecht A., 2002, A new locally convergent particle swarm optimiser, *IEEE Internationalconference on systems, man and cybernetics,* cilt 3.

[59] Poli R., 2008, Analysis of the publications on the applications of particle swarm optimisation, *Journal of Artificial Evolution and Application.*

[60] Li X. ve Deb K., 2010, Comparing lbest PSO niching algorithms using different position update rules, %1 içinde *IEEE congress on evolutionary computation*.

[61] Karaboga D. ve Akay B., 2009, A comparative study of artificial bee colony algorithm, *Applied mathematics and computation 214.1,* pp. 108-132.

[62] Karaboga D. ve Basturk B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of global optimization 39.3,* pp. 459-471.

[63] Karaboga D. ve Akay B., 2010, Proportional-integral-derivative controller design by using artificial bee colony, harmony search, and the bees algorithms, *Journal of Systems and Control Engineering 224.7,* pp. 869-883.

[64] Karaboga D. ve Basturk B., 2008, On the performance of artificial beevcolony (ABC) algorithm, *Applied soft computing 8.1,* pp. 687-697.

[65] Koç İ. , Baykan O. ve Babaoğlu I. , 2018, Gri kurt optimizasyon algoritmasına dayanan çok seviyeli imge eşik seçimi, *Politeknik Dergisi,,* cilt 21, no. 4, pp. 841-847.

[66] Doğan C. , 2019, Balina optimizasyon algoritması ve gri kurt optimizasyon algoritması kullanılarak yeni hibrit optimizasyon algoritmasının geliştirilmesi, Yüksek Lisans, Erciyes Üniversitesi, Kayseri.

[67] Shehab M., Abualigah L., AlHamad H. ve Alabool H., 2019, Moth–flame optimization algorithm: variants and applications, *Neural Computing and Applications*.

[68] Sayed G. ve Hassanien A., 2018, A hybrid SA-MFO algorithm for function optimization and engineering design problems, *Complex & Intelligent Systems,* cilt 4, p. 195–212.

[69] Rana N., Latiff M., Abdulhamid S. ve Chiroma H., 2020, Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments, *Neural Computing and Applications*.

[70] Mohammed H., Umar S. ve Rashid T., 2019, A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm, *Computational Intelligence and Neuroscience,* cilt 2019, p. 25.

[71] Hekimoğlu B., Ekinci S., Demiray V., Doguruci R. ve Yıldırım A., 2018, Speed control of dc motor using PID controller tuned by salp swarm algoritm,*International Engineering and Natural Sciences Conference*.

[72] Kumari S. ve Shankar G., 2018, A Novel Application of Salp Swarm Algorithm in Load Frequency Control of Multi-Area Power System,*2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*.

[73] Demir G. ve Tanyıldızı E., 2017, Optimizasyon Problemlerinin Çözümünde Sinüs Kosinüs Algoritması (SKA)'nın Kullanılması, *Fırat Üniv. Müh. Bil. Dergisi,* cilt 29, pp. 225-236.

[74] Ekinci S., 2019, Optimal design of power system stabilizer using sine cosine algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University,* cilt 34, no. 3, pp. 1329-1350.

[75] Lennard-Jones J.E., 1924, On the determination of molecular field, Proceedings of the Royal society of LondonA: Mathematical, Physical and Engineering Sciences, The Royal Society, 106(738). 463-477

[76] Almabrok A., Psarakis M. ve Dounis A., 2018, Fast Tuning of the PID Controller in An HVAC System Using the Big Bang–Big Crunch Algorithm and FPGA Technology, *algorithms,* cilt 11, p. 146.

[77] Hajisalm A. ve Altaş İ., 2014, Optimization of PID Controller Parameters in Wind/PV Energy Systems Using PSO and GA,*Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, Bursa.

[78] Kılıç E. ve Özdemir M., 2018, Güç sistemlerindeki optimum otomatik gerilim regülasyonu için çoklu amaç fonksiyonunun belirlenmesi, *DÜMF Mühendislik Dergisi,* cilt 10, no. 1, pp. 1-12.

[79] Tabakoğlu B., 2009, Üç alanli güç sisteminde genetik algoritma ile otomatik üretim kontrolü, Yüksek Lisans, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul.

[80] Yang H.C ve Allstot D.J., 1990, Considerations of Fast Settling Operational Amplifiers, IEEE Transections on circuits and systems, vol.37, no.3

[81] Walling J.S., Shekhar S. ve Allstot D.J., 2008, Wide Band CMOS Amplifier Desing: Time Domain Consideretaions, IEEE Transections on circuits and systems, Vol.55, No:7