

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



SEZGİSEL PID KONTROLÖR TASARIMI VE DC MOTOR
SİMÜLASYONU

Abdulmutalip YILDIRIM

Yüksek Lisans Tezi

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

AĞUSTOS 2021

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yazılım Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

SEZGİSEL PID KONTROLÖR TASARIMI VE DC MOTOR SİMÜLASYONU

Tez Yazarı

Abdulmutalip YILDIRIM

Danışman

Prof. Dr. Soner ÖZGEN

İkinci Danışman

Prof. Dr. Erkan TANYILDIZI

AĞUSTOS 2021

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yazılım Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı:	Sezgisel PID Kontrolör Tasarımı ve DC Motor Simülasyonu
Yazarı:	Abdulmutalip YILDIRIM
İlk Teslim Tarihi:	02.07.2021
Savunma Tarihi:	16.08.2021

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Prof. Dr. Soner ÖZGEN Fırat Üniversitesi, Fen Fakültesi	<i>İmza</i> Onayladım
İkinci Danışman:	Prof. Dr. Erkan TANYILDIZI Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım
Başkan:	Dr. Öğr. Üyesi Ahmet Arif AYDIN İnönü Üniversitesi, Mühendislik Fakültesi	Onayladım
Üye:	Doç. Dr. Murat KARABATAK Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza
Doç. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Sezgisel PID Kontrolör Tasarımı ve DC Motor Simülasyonu” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

16.08.2021

Abdulmutalip YILDIRIM



ÖNSÖZ

Günümüzde meta sezgisel algoritmalar matematiksel model gerektirmeyen birçok mühendislik problemlerin çözümünde kullanılmaktadır. PID kontrolör sanayi ve endüstriyel gibi alanlarda yaygın olarak kullanılmaktadır. Yaygın olarak kullanılmasının nedenlerinden biri az sayıda ayarlanabilir parametrelerinin olmasından kaynaklanmaktadır. Bu parametrelerin ayarlanmasında çeşitli yöntemler bulunmaktadır. Bu çalışmada ASO, PSO, HHO, GWO meta sezgisel algoritmalar ile IAE, ISE, ITAE, ITSE uygunluk fonksiyonlarına göre en uygun PID parametrelerinin ayarlanması amaçlanmıştır. Ayarlanan PID parametreler Matlab simülink DC motorun transfer fonksiyonu üzerinden test edilmiştir. Meta sezgisel algoritmalar kullanılarak ayarlanan PID parametrelerinin klasik yöntemlerden ZN yöntemine göre daha iyi sonuç verdiği görülmüştür. IAE ve ITSE uygunluk fonksiyonuna göre PSO algoritması, ISE ve ITAE uygunluk fonksiyonuna göre ise HHO algoritmasının daha iyi sonuç verdiği görülmektedir.

Çalışmalarında Yüksek Lisans danışmanlığımı üstelenerek her zaman bana zaman ayıran, sorularımı cevaplayan Sayın Prof. Dr. Soner ÖZGEN ve Prof. Dr. Erkan TANYILDIZI hocalarıma çok teşekkür ederim.

Ayrıca tezimin hazırlanmasında maddi ve manevi desteğini eksik etmeyen aileme teşekkürü borç bilirim.

Abdulmutalip YILDIRIM
ELAZIĞ, 2021

İÇİNDEKİLER

	Sayfa
ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ÖZET	vi
ABSTRACT	vii
ŞEKİLLER LİSTESİ	viii
TABLolar LİSTESİ	x
SİMGELER	xi
KISALTMALAR	xii
1. GİRİŞ	1
2. MATERYAL VE METOT	4
2.1. PID Kontrolör Sistemi	4
2.2. DC Motor Modeli	6
2.3. Meta Sezgisel Algoritmalar	7
2.3.1. Parçacık Sürü Optimizasyon (PSO) Algoritması	7
2.3.2. Gri Kurt Optimizasyon (GWO) Algoritması	10
2.3.3. Atom Arama Optimizasyon (ASO) Algoritması	15
2.3.4. Harris Şahinleri Optimizasyon (HHO) Algoritması	22
2.4. Uygunluk Fonksiyonu	28
3. BULGULAR VE TARTIŞMA	30
4. SONUÇLAR.....	45
ÖNERİLER	46
KAYNAKLAR	47
ÖZGEÇMİŞ	

ÖZET

Sezgisel PID Kontrolör Tasarımı ve DC Motor Simülasyonu

Abdulmutalip YILDIRIM

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Ağustos 2021, Sayfa: xii + 49

Optimizasyon problemlerin çözümünde farklı teknikler kullanılmaktadır. Bu tekniklerden biri olan metasezgisel optimizasyon algoritmaları karmaşık mühendislik problemlerin çözümünde yaygın olarak tercih edilmektedir. Literatürde bulunan bir optimizasyon algoritması her mühendislik problemi için en ideal çözümü gerçekleştirememektedir. Bu nedenle bir problemin en ideal çözümü bulmak için farklı optimizasyon algoritmasının gerçekleştirilmesi ile bulunabilir.

Bu çalışmada endüstriyel uygulamalarda yaygın olarak PID kontrolörün parametreleri klasik matematiksel yöntem yerine sezgisel PID kontrolör tasarımı yapılmış ve literatürde bulunan DC motorun özellikleri transfer fonksiyonuna uygulanmıştır. Meta sezgisel optimizasyon algoritmaları olarak Parçacık Sürü Optimizasyonu (PSO), Gri Kurt Optimizasyonu (GWO), Atom Arama Optimizasyonu (ASO) ve Harris Şahinleri Optimizasyonu (HHO) tercih edilmiştir. Bu optimizasyon algoritmalarının aynı koşullarda IAE, ISE, ITAE ve ITSE uygunluk fonksiyonu kriterine göre hesaplanan PID parametrelerinin performans değerleri algoritmalar için kendi aralarında ve klasik yöntemlerden Ziegler – Nichols (ZN) birim basamak cevabı yöntemine göre karşılaştırılması yapılmıştır. IAE, ISE, ITAE ve ITSE uygunluk fonksiyonu kriterine göre yapılan çalışmalarda meta sezgisel algoritmalar ZN yöntemine göre daha iyi sonuç verdiği görülmüştür. ITSE uygunluk fonksiyonu kriterine göre yapılan çalışmalarda algoritmaların diğer uygunluk fonksiyonlarına göre daha iyi sonuç verdiği görülmüştür. IAE ve ITSE uygunluk fonksiyonuna göre yapılan çalışmada PSO algoritması, ISE ve ITAE uygunluk fonksiyonuna göre yapılan çalışmada ise HHO algoritmasının daha iyi değerlere ulaştığı görülmektedir.

Anahtar Kelimeler: PID Kontrolör, Metasezgisel Algoritmalar, PSO, ASO, GWO, HHO

ABSTRACT

Metaheuristic PID Controller Design and Simulation of DC Motor

Abdulmutalip YILDIRIM

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

August 2021, Pages: xii + 49

Different techniques are used to solve optimization problems. Metaheuristic optimization algorithms, one of these techniques, are widely preferred in solving complex engineering problems. An optimization algorithm in the literature cannot realize the ideal solution for every engineering problem. For this reason, it can be found by performing different optimization algorithms to find the most ideal solution to a problem.

In this study, the parameters of the PID controller, which are common in industrial applications, were designed instead of the classical mathematical method, and the characteristics of the DC motor in the literature were applied to the transfer function. Particle Swarm Optimization (PSO), Gray Wolf Optimization (GWO), Atom Search Optimization (ASO) and Harris Hawks Optimization (HHO) were preferred as metaheuristic optimization algorithms. The performance values of the PID parameters calculated according to the IAE, ISE, ITAE and ITSE fitness function criteria of these optimization algorithms under the same conditions were compared among themselves for the algorithms and according to the Ziegler – Nichols (ZN) unit step response method, which is one of the classical methods. In studies conducted according to IAE, ISE, ITAE and ITSE fitness function criteria, it was seen that metaheuristic algorithms gave better results than the ZN method. In studies conducted according to ITSE fitness function criteria, it has been seen that algorithms give better results than other fitness functions. It is seen that the PSO algorithm in the study performed according to the IAE and ITSE fitness function, and the HHO algorithm in the study carried out according to the ISE and ITAE fitness function reach better values.

Keywords: PID Controller, Metaheuristic Algorithms, PSO, ASO, GWO, HHO

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. PID genel yapısı	4
Şekil 2.2. Geçici durum cevabı.....	5
Şekil 2.3. DC motorun eşdeğer devresi	6
Şekil 2.4. PSO algoritması ile en iyi çözüme ulaşma	8
Şekil 2.5. PSO parametrelerin vektörel gösterimi	9
Şekil 2.6. PSO algoritmasına ait akış diyagramı	10
Şekil 2.7. Gri kurt hiyerarşisi (baskınlık yukarıdan aşağıya azalır).....	11
Şekil 2.8. Gri kurtların 2D ve 3D uzayındaki konum vektörleri ve olası sonraki konumları	12
Şekil 2.9. GWO'da konum güncelleme	13
Şekil 2.10. Gri kurtların ava saldırma ve av arama durumu	14
Şekil 2.11. PSO algoritmasına ait akış diyagramı	15
Şekil 2.12. L-J potansiyel eğrisi	16
Şekil 2.13. Atomların kuvvet eğrisi.....	16
Şekil 2.14. Farklı η değerlerine karşılık gelen <i>Fijdt</i> fonksiyonun davranışı	17
Şekil 2.15. K=5 için <i>KBest</i> içeren bir atom sisteminin kuvvetleri	20
Şekil 2.16. 3 boyutlu bir uzayda bir hedef etrafında 5 atomun sürüklenme hareketi	21
Şekil 2.17. ASO algoritmasına ait akış diyagramı.....	22
Şekil 2.18. HHO'nun avlanma aşamaları.....	23
Şekil 2.19. Kaçan avın enerjisinin iterasyon sayısına bağlı değişimi	25
Şekil 2.20. Aşamalı hızlı dalışlar ile yumuşak kuşatma durumunda genel vektörlere örnek	27
Şekil 2.21. Aşamalı hızlı dalışlar ile sert kuşatma durumunda genel vektörlere örnek	27
Şekil 3.1. Sezgisel algoritmada kullanılan PID blok şeması	30
Şekil 3.2. ZN yöntemine göre ayarlanan PID parametrelerin DC motor sisteminin açısal hız tepkisi	31
Şekil 3.3. İterasyon sayısına bağlı olarak algoritmaların IAE uygunluk fonksiyonu değerinin değişimi	34
Şekil 3.4. IAE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi	35
Şekil 3.5. İterasyon sayısına bağlı olarak algoritmaların ISE uygunluk fonksiyonu değerinin değişimi	36
Şekil 3.6. ISE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi	37
Şekil 3.7. İterasyon sayısına bağlı olarak algoritmaların ITAE uygunluk fonksiyonu değerinin değişimi ..	38
Şekil 3.8. ITAE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi	39
Şekil 3.9. İterasyon sayısına bağlı olarak algoritmaların ITSE uygunluk fonksiyonu değerinin değişimi ...	40
Şekil 3.10. ITSE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi	41

Şekil 3.11. ASO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısıl hız tepkisi.....	43
Şekil 3.12. HHO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısıl hız tepkisi.....	43
Şekil 3.13. GWO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısıl hız tepkisi.....	44
Şekil 3.14. PSO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısıl hız tepkisi.....	44



TABLÖLAR LİSTESİ

	Sayfa
Tablo 2.1 Kapalı çevrim kontrol sistemi denetleyicileri etkileri	5
Tablo 2.2. Yaygın kullanılan uygunluk fonksiyonları	29
Tablo 3.1. DC motor özellikleri	30
Tablo 3.2. Ziegler – Nichols birim basamak cevabı yöntemine göre PID parametrelerin hesaplanması.....	31
Tablo 3.3. ZN yöntemine göre PID parametrelerin performans değerleri	31
Tablo 3.4. Sezgisel algoritmaların başlangıç konumları	32
Tablo 3.5. Sezgisel algoritmaların ortak parametreleri	33
Tablo 3.6. Sezgisel algoritmalara ait özel parametreler	33
Tablo 3.7. IAE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi	34
Tablo 3.8. IAE uygunluk fonksiyonuna göre PID performans değerleri	35
Tablo 3.9. ISE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi	36
Tablo 3.10. ISE uygunluk fonksiyonuna göre PID performans değerleri	37
Tablo 3.11. ITAE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi	38
Tablo 3.12. ITAE uygunluk fonksiyonuna göre PID performans değerleri	39
Tablo 3.13. ITSE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi	41
Tablo 3.14. ITSE uygunluk fonksiyonuna göre PID performans değerleri.....	42

SİMGELER

A	: GWO algoritmasında katsayı vektörü
C	: GWO algoritmasında katsayı vektörü
c_1, c_2	: PSO algoritmasında hızlanma katsayıları
E	: HHO algoritmasında kaçan avın enerjisi
E_0	: HHO algoritmasında kaçan avın başlangıç kaçış enerjisi
$Fit_{best}(t)$	: ASO algoritmasında maksimum uygunluk değerine sahip atom
Fit_{worst}	: ASO algoritmasında minimum uygunluk değerine sahip atom
G_{best}	: PSO algoritmasında tüm iterasyonda ve parçacıklarda en uygun değeri olan parçacık
K	: Oransal kazanç
K_d	: PID denetimin türev değeri
K_p	: PID denetimin orantı değeri
K_i	: PID denetimin integral değeri
K_T	: Motor Tork sabiti
L	: Endüktans
LF	: Levy flight tabanlı model fonksiyonu
$L-J$	: Lennard-Jones potansiyeli
P_{best}	: PSO algoritmasında en uygun değeri olan parçacık
R	: Omik Direnç
r_1, r_2	: PSO algoritmasında rastgele sayılar
T_i	: İntegral zaman sabiti
T_d	: Türev zaman sabiti
V_i^k	: PSO algoritmasında parçacıkların hızı
v	: HHO algoritmasında rastgele sayı
X_i	: Parçacıkların konumu
ΔX	: HHO algoritmasında tavşan ile harris şahinlerinin mevcut konumu arasındaki fark
X_m	: Harris şahin popülasyonunun ortalama konumu
w	: PSO algoritmasında eylemsizlik ağırlığı
ε	: ASO algoritmasında potansiyel derinlik
σ	: ASO algoritmasında çarpışma çapı
η	: ASO algoritmasında derinlik fonksiyonu
α	: ASO algoritmasında derinlik ağırlığı
β	: ASO algoritmasında çarpan ağırlığı
a_i	: ASO algoritmasında atomun hızlanma katsayısı
r_5	: HHO algoritmasında rastgele sayı
u	: HHO algoritmasında rastgele sayı

KISALTMALAR

ABC	: Yapay arı kolonisi algoritması
AC	: Alternatif akım
AQM	: Yönlendirici ve anahtar
ASO	: Atom arama optimizasyonu
DC	: Doğru akım
GWO	: Gri kurt optimizasyonu
HHO	: Harris şahinleri optimizasyonu
IAE	: Hatanın mutlak değerlerinin toplamı
ISE	: Hataların karelerinin toplamı
ITAE	: Zaman ağırlıklı mutlak hataların toplamı
ITSE	: Zaman ağırlıklı hataların karelerinin toplamı
PID	: Oransal integral türev denetim
PSO	: Parçacık sürü optimizasyonu
SCA	: Sinüs kosinüs algoritması
LB	: Sezgisel algoritmalarda arama alanının alt sınır değeri
UB	: Sezgisel algoritmalarda arama alanının üst sınır değeri
ZN	: Ziegler – Nichols

1. Giriş

Günümüzde kontrol sistemleri ve optimizasyon yöntemleri endüstriyel, otomasyon gibi birçok alanda kullanılmaktadır. Birçok alanda kontrol sistemlerinde PID kontrolör en çok kullanılan kontrolördür. PID kontrolörün tercih edilmesi nedenleri arasında birçok neden vardır. PID denetleyicinin yaygın kullanılan denetleyici türlerinden biri olmasının sebepleri arasında az sayıda ayarlanacak parametresinin olması, her bir parametrenin sistem çıkışına fiziksel olarak etkisinin gözlemlenebilmesi ve kullanıcı tarafından uzmanlık gerektirmeden manuel olarak ayarlanabilme imkanının olması sayılabilir [1]. Uygun PID parametrelerinin belirlenmesi sistemin enerji tüketimi, kararlı çalışması ve işleyişi bakımından önemlidir. Kontrol parametrelerinin belirlenmesinde günümüzde farklı yöntemler kullanılmaktadır. Bu yöntemlerden en fazla kullanılan matematiksel yöntemler ve meta sezgisel yöntemlerdir. Matematiksel yöntemler karmaşık ve zaman gerektirir. En uygun çözümü sunmayabilir. Meta sezgisel yöntemler ise doğadan esinlenerek matematiksel olarak modellenen kolay uygulanabilen ve en uygun çözümü bulmayı hedefler. Meta sezgisellikte vurgu, çözüm uzayının en umut verici bölgelerinin derinlemesine araştırılmasıdır. Bu yöntemler tipik olarak karmaşık komşuluk arama kurallarını, bellek yapılarını ve çözümlerin rekombinasyonlarını birleştirir. Bu yöntemlerle üretilen çözümlerin kalitesi, genellikle klasik buluşsal yöntemlerle elde edilenden çok daha yüksektir [2]. Meta sezgisel algoritmaların optimizasyon problemlerinde kullanımlarında global ve yerel arama teknikleri olmak üzere iki kullanımı vardır. Yerel arama, çözüm kümesinin belirli bir bölgesindeki en iyi çözümleri bulmayı amaçlarken, global arama genel en iyi çözümü hedefler. Yerel aramada elde edilen en iyi nokta, çözüm kümesindeki en iyi noktayı garanti etmez [3].

Meta sezgisel yöntemlerde bir problemin çözümü için bir veya birden fazla optimizasyon algoritması kullanılabilir. Kullanılan bir optimizasyon algoritması bir problem için en iyi çözümü garanti etmemektedir. Sezgisel algoritmalar karar verici tarafından daha basit algılanır. Optimizasyon problemleri doğrudan kesin çözüm üretmediğinden ve sadece kesin çözümü bulma işleminin bir parçası olmasından dolayı sezgisel algoritmalara ihtiyaç duyulmaktadır [4]. Meta sezgisel yöntemlerdeki algoritmalar genellikle birlikte belirli bir amaç için hareket eden sürü halindeki canlıların davranışlarından esinlenerek modellenmiştir. Sezgisel yöntemlere dayalı algoritmalar kullanılarak yapılan bazı çalışmalar aşağıda verilmiştir.

Anbumani vd. [5] 2017 yılında yaptıkları çalışmada Gri kurt optimizasyonu algoritması (GWO) kullanılarak bir ısı değişimi için kontrolör tasarımı önerdi. PID kontrolörünün ayarlanması GWO algoritması tarafından yapıldı ve simülasyon, ısı değiştiricinin birinci dereceden transfer fonksiyonu modeli için gerçekleştirildi.

Sabry vd. [6] 2019 yılında yaptıkları çalışmada Gri kurt optimizasyonu algoritması (GWO) kullanılarak TCP ağlarında, Active Queue Management (AQM), uç yönlendiriciler tarafından ağ

tıkanıklığını önlemek ve paket kaybını önlemek, yönlendirici kuyruk uzunluğunu düzenlemek için gri kurt optimizasyonu algoritması (GWO) kullandılar. Önerilen GWO-PID denetleyicisinin kararlı olduğunu ve değişken gidiş-dönüş süresi ve ağ trafiği ile dinamik ortamlarda daha hızlı yanıt elde ettiğini tespit ettiler.

Zhi vd. [7] 2019 yılında yaptıkları çalışmada Yapay Arı Kolonisi Algoritmasını (ABC) kullanarak tek fazlı invertör için PID kontrol parametrelerini optimize etmiştir.

Bisht vd. [8] 2020 yılında yaptıkları çalışmada hibrit elektrikli araçlarda motorun gaz kelebeği konumunu kontrol ederek hız kontrolü gerçekleştirdiler. Kontrolörün parametrelerini optimize etmek için Gri Kurt Optimizasyonu (GWO) ve Parçacık Sürü Optimizasyonu (PSO) yöntemini kullandılar.

Nie vd. [9] 2020 yılında yaptıkları çalışmada Nükleer enerji endüstrisinde basınç dengeleyicinin çalışmasını stabilize etmek, basınç dengeleyicinin basınç ve su seviyesi kontrol sistemini kontrol etmek için PID kontrolörü kullandılar. Matlab programlama ve simulink simülasyonunu birleştiren PSO algoritması, PID denetleyicisinin parametrelerini ayarlamak için önerilmiştir.

Li vd. [10] 2020 yılında yaptıkları çalışmada su yosunları sıcaklıktan büyük ölçüde etkilendiğinden PSO ve PID kullanarak su yosunlarının sıcaklık kontrolü yapmışlar.

Zhu vd. [11] 2020 yılında yaptıkları çalışmada rüzgar türbinlerinin valf kontrollü hidrolik yükleme sisteminin matematiksel modeli oluşturulmuş ve matlabda simülasyonu edilmiştir. Hidrolik yükleme sisteminin kapalı döngü kontrol sistemini modellemek ve simüle etmek için PID kontrol yöntemi ve parçacık sürüsü algoritması kullandılar.

Literatürde yapılan çalışmalar genellikle matlab ortamında çalışmaların simülasyonu gerçekleştirilmiştir ve bir ya da birden çok algoritma kullanılmıştır.

Bu çalışmada Parçacık Sürü Optimizasyonu (PSO), Gri Kurt Optimizasyon(GWO), Atom Arama Optimizasyon (ASO) ve Harris Şahinleri Optimizasyon (HHO) olmak üzere dört farklı meta sezgisel algoritma kullanılarak en uygun PID parametreleri belirlemek amacıyla matlab ortamında yayınlanan çalışmalarda bulunan DC motor özelliklerinin transfer fonksiyonu ile simülasyonu gerçekleştirilmiştir. Aynı koşullarda IAE, ISE, ITAE ve ITSE uygunluk fonksiyonuna göre algoritmaların performans kıyaslaması yapılmış ve klasik yöntemlerden Ziegler – Nichols (ZN) birim basamak cevabı yöntemine göre karşılaştırılması yapılmıştır. Belirlenen performans değerlerine göre DC motor spin kaplama, konum kontrol, hız kontrolü gibi çeşitli uygulamalarda kullanılabilir.

İkinci bölümde materyal ve metotlar yer almaktadır. Bu bölümde matlab simülasyon çalışmasında kullanılan sistemin modeli ve parametreleri anlatılmıştır. Simülasyonda kullanılan PID ve DC motor yapısı, sezgisel algoritmalarda performans değerlendirmesi için kullanılan

uygunluk fonksiyonları, sezgisel algoritmaların yapıları, matematiksel modellemeleri ve oluşturulması ele alınmıştır.

Üçüncü bölümde sistemin simülasyon çalışması yapılmıştır. Klasik yöntemlerden Ziegler – Nichols (ZN) birim basamak cevabı yöntemine göre PID parametrelerinin değerleri hesaplanmıştır. Çalışmada kullanılan sezgisel algoritmaların başlangıç konumları, algoritmalara özel ve genel parametreler belirlenmiştir. Bulunan PID parametrelerinin performans değerleri ayrıntılı belirtilmiş ve genel performansları tablo ve şekillerle belirtilmiştir.

Dördüncü bölümde ise PID parametrelerin hesaplanmasında kullanılan ZN yöntemi ile sezgisel yöntemin karşılaştırılması yapılmıştır. Sezgisel algoritmaların ve uygunluk fonksiyonların performans değerlendirmesi ve karşılaştırılması yapılmıştır.

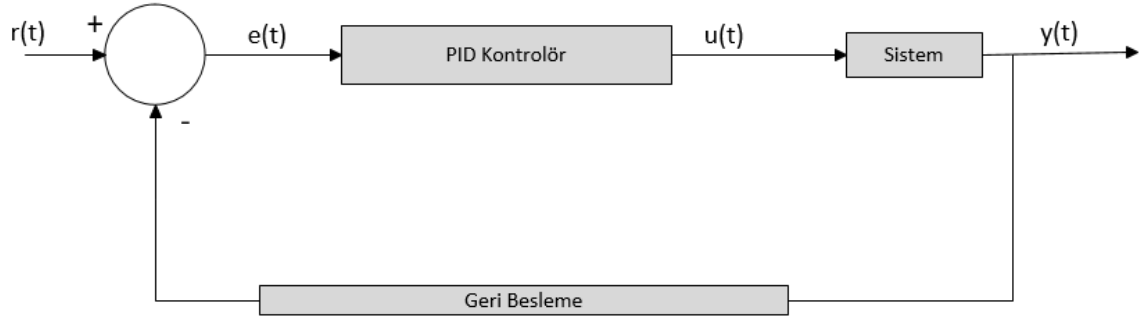


2. MATERYAL VE METOT

Bu bölümde matlab simülasyon ortamında kullanılan PID, DC motor sistem model parametreleri ile sürü zekasına dayalı Parçacık Sürü Algoritması, Gri Kurt Algoritması, Atom Arama Algoritması, Harris Şahinleri Algoritması algoritmaları ve modellenmesi ele alınmıştır.

2.1. PID Kontrolör Sistemi

PID az sayıda parametre içermesi nedeniyle endüstriyel ve otomasyon gibi alanlarda diğer karmaşık denetleyicilerden daha yaygın tercih edilmektedir. Genel yapısı Şekil 2.1’de verilmiştir. PID kontrolör K_p , K_i ve K_d üç kazanç parametresi içermektedir. Bu parametrelerin her biri sistem üzerinde farklı etkilere sahiptir. Bir kontrol uygulamasında bu parametrelerden biri, ikisi veya her üçü birden kullanılabilir. Kontrolör tasarlanırken hangi parametrelerden hangi oranlarda kullanılacağı kararlaştırılır [12]. Bu parametreler her farklı sistem için farklı değerler almaktadır.



Şekil 2.1. PID genel yapısı

PID kontrolörün matematiksel yapısı Denklem 2.1’de verilmiştir.

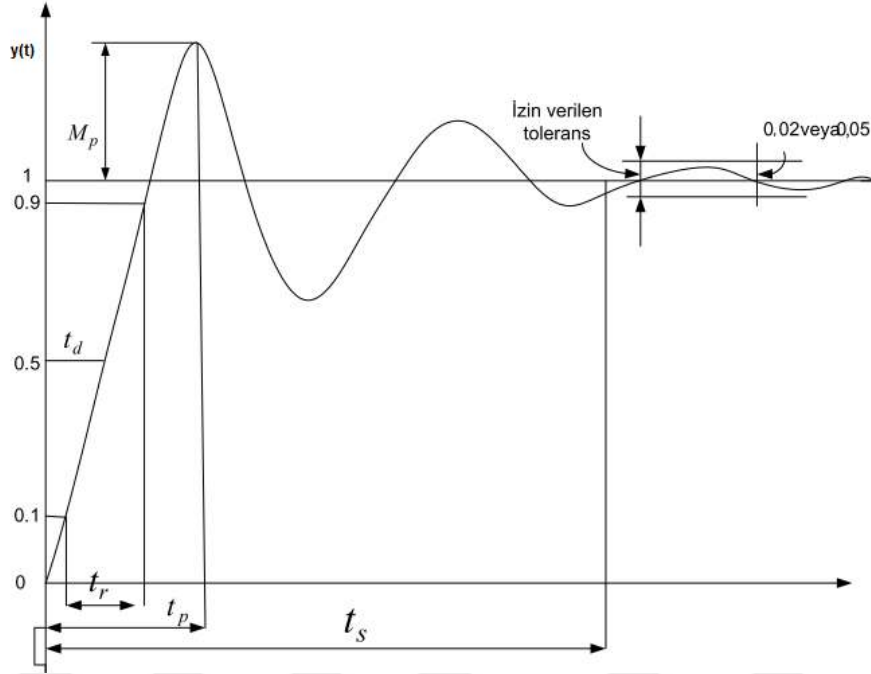
$$u(t) = K \left[e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (2.1)$$

Burada $u(t)$ kontrol değişkeni, $e(t)$ zamana bağlı etkin hata fonksiyonudur. $r(t)$ referans giriş değerini ve $y(t)$ çıkış değerini temsil eder. Denklem 2.1’deki terimler sırasıyla; Oransal terim, İntegral terim ve Türevsel terimdir. Kontrolör parametreleri sırasıyla; K Oransal kazanç, T_i İntegral zaman parametresi ve T_d Türev zaman parametresi şeklinde tanımlanır.

Kontrol edilecek sistemin çıkışı ile girişe uygulanan referans değerinin farkı hata olarak hesaplanır. hata, oransal K_p parametresiyle, hatanın türevi K_d parametresiyle, hatanın integrali K_i parametresiyle çarpılarak elde edilen değerlerin toplamı sisteme giriş olarak uygulanır. Sistemin

çıkışı tekrar referans değeri ile karşılaştırılarak yeni bir hata değeri hesaplanır ve kontrolcünün üreteceği yeni çıkış sisteme giriş olarak uygulanır. Bu işlem hata sıfır olana kadar devam eder [13].

PID kontrolörün basamak girişine karşı geçici durum cevabı Şekil 2.2’de verilmiştir.



Şekil 2.2. Geçici durum cevabı [14]

t_d : Gecikme zamanı, cevabın son değerinin yarısına ilk defa ulaşması için geçen zamandır [14].

t_r : Yükselme zamanı, sistem cevabının sistem giriş değerinin %10’dan %90’a ulaşması için geçen süredir.

t_p : Tepe zamanı, sistem cevabının ilk defa maksimum olduğu süredir.

M_p : Maksimum aşma, sistem cevabı ile sistem giriş değerinin arasındaki fark olarak açıklanmaktadır.

t_s : Oturma zamanı, sistemin ilk dengeye ulaştığı süredir.

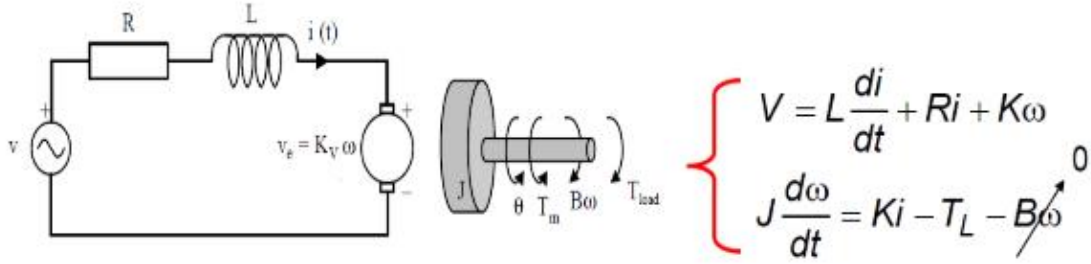
Kararlı durum hatası: Kontrolör sistem çıkışını referans giriş değerine yaklaştırmaya çalışır. Fakat bazı durumlarda bu iki değer arasında bir fark olur ve bu farka kalıcı durum hatası denilmektedir [15] K_p , K_i ve K_d parametrelerin durum cevabına etkisi Tablo 2.1’de belirtilmiştir.

Tablo 2.1 Kapalı çevrim kontrol sistemi denetleyicileri etkileri [15]

	Yükselme Zamanı	Aşma	Yerleşme Zamanı	Kararlı Durum Hatası
K_p	Azalır	Artar	Küçük Oranda Değişir	Azalır
K_i	Azalır	Artar	Artar	Elimine Edilir
K_d	Küçük Oranda Değişir	Azalır	Azalır	Küçük Oranda Değişir

2.2. DC Motor Modeli

Günümüzde, elektrik motorları hemen hemen tüm endüstriyel alanda çeşitli yükleri sürmek amacıyla kullanılmaktadır. Dünyada kullanılan milyarlarca elektrik motorları, çeşitli uygulamalar için farklı güçlerde bulunmaktadır. Büyük güçlü motorlar daha büyük iş makinalarında ve küçük güçlü motorlar ise ev aletlerinde kullanılmaktadır [16]. Elektrik motorları genellikle DC motorlar ve AC motorlar olarak iki gruba ayrılabilir. Her iki motor çeşitlerinin bir birine göre avantajları ve dezavantajları vardır. Bu tezde DC motor kullanıldığından DC motorun genel yapısı ve işlevi açıklanmaktadır. DC motor sistemi, genellikle DC motorun armatür kontrolü olan hız ve ayar konumunu kontrol etmek için tercih edilen bir motor türüdür. Armatür kontrollü bir DC motor, statik alan akımını sabit tutmak için düzenleme armatürü ile kontrol edilir [17]. Basit bir DC motor, elektrik ve manyetik alanı kullanarak mil üzerine tork uygular. Sabit mıknatıslı DC motorlar daha yüksek kalkış momenti ihtiyaçlarına iyi bir hız kontrolü sağlarlar ve değişken hız ihtiyacı olan endüstriyel uygulamalarda AC motorlardan daha iyi performans sergilerler. DC motorların batarya veya fotovoltaik gibi kaynaklarla da çalışabilmesi DC motorların kullanım alanlarını arttırmıştır [18]. Şekil 2.3'te DC motorun eşdeğer devresi verilmiştir.



Şekil 2.3. DC motorun eşdeğer devresi [19]

V : Gerilim

R : Endüvi Sargısı Omik Direnç

L : Endüvi Sargısı Endüktans

$i(t)$: Endüvi Akım

$V_e(t)$: Endüvi Gerilimi

J : Rotor Atalet Momenti

B : Viskoz Sönümlleme

K_T : Motor Tork sabiti

K_v : Zıt Elektromotor sabiti

Transfer fonksiyonu bir sistemin giriş ile çıkış arasındaki dinamik ilişkiyi matematiksel olarak modeller. Bir DC motor hız kontrolörü için sistem girişi motora uygulanan gerilim, sistem çıkışı ise motor milinin açısal hızı (rad/sn) olacak şekilde bir transfer fonksiyonu oluşturulmuştur [20]. DC motorun transfer fonksiyonu Denklem 2.2’de verilmiştir.

$$\frac{\omega}{V} = \frac{K_T}{(Js+B)(Ls+R)+K_T K_v} \quad (2.2)$$

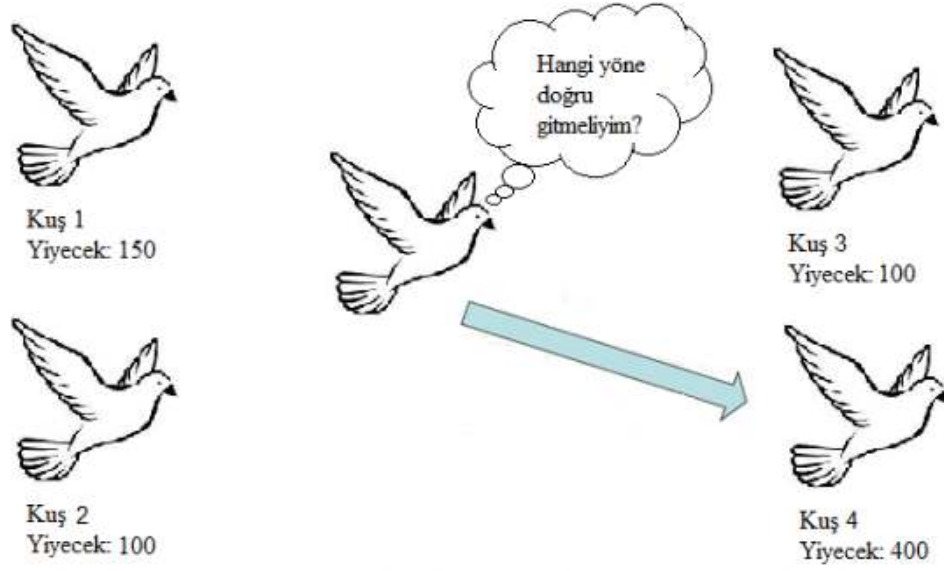
2.3. Meta Sezgisel Algoritmalar

Bu çalışmada dört farklı literatürde çok fazla kullanılan optimizasyon algoritmalarından PSO, GWO, ASO ve HHO algoritmaları değerlendirilmiştir.

2.3.1. Parçacık Sürü Optimizasyon (PSO) Algoritması

Sezgisel yöntemlerden biri olan PSO tekniği ilk olarak kuş ve balık sürülerinin hareketlerinden esinlenerek doğrusal olmayan sayısal problemlere en iyi sonuçlar bulmak için 1995–1996 yıllarında sosyolog-psikolog James Kennedy ve elektrik mühendisi Russel Eberhart tarafından ortaya atılmıştır [4].

Kuşlar doğal ortamda yerini bilmediği yiyeceği rastgele arayıp bulma davranışları gözlemlenerek oluşturulmuştur. Şekil 2.4’te bir kuş sürüsünün tespit edilen yiyecek kaynaklarından yiyecek miktarının daha fazla olduğu yöne doğru eğilimleri resmedilmiştir. Buradan anlaşılacağı üzere, sürüdeki tüm kuşlar yiyecek miktarının daha fazla olduğu optimum noktada toplanırlar [1]. PSO algoritması uygunluk fonksiyonu hesaplamasında cebirsel işlemlere ihtiyaç duymayan bir sürü zekasına dayalı algoritmadır. Bu algoritma çeşitli ve karmaşık mühendislik problemlerinde uygulanabilmektedir. Kuşların hareket uzayında, yerini bilmedikleri bir nesneyi rastgele gelişen hareketler sayesinde arayıp bulmaları, bir problemin çözümüne benzetilir. Kuşlar nesneyi ararken o nesneye en yakın olan kuşu takip ederler. PSO’nun temel olarak dayandığı nokta, gruptaki bireyler arasında gerçekleşen veri paylaşımını geliştirmektir. Bu yöntemde bireylere parçacık adı verilir. Bu parçacıklardan oluşan grup da sürü olarak adlandırılır [22].



Şekil 2.4. PSO algoritması ile en iyi çözüme ulaşma [1]

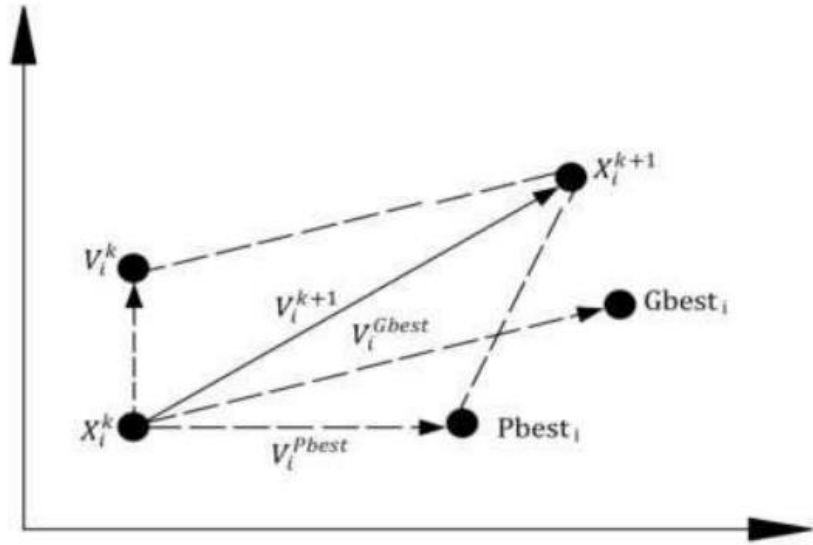
Sürü içerisindeki her bir kuş bir cevabı temsil eder. Aynı zamanda her bir kuş bilinmeyen sayısı kadar cevap üretir. Verilen cevaplar o kuşun o andaki pozisyonu anlamını taşır [23]. Algoritma rastgele oluşturulan parçacıklarla başlar ve parçacıkların her biri problem için bir çözümü ifade etmektedir. İlk iterasyonda her bir parçacığın en iyisi olan “pbest” kendisine eşit olmaktadır. Buradan elde edilen en iyi değere karşılık gelen parçacık (sürünün en iyisi) “gbest” olarak seçilir [24]. Bu parçacıklar güncellenerek uygunluk fonksiyonunda en uygun değerleri bulmaya çalışır. Her parçacıklardan en uygun değeri olan parçacık P_{best} olarak ifade edilmektedir. Tüm parçacık ve her iterasyon sonunda en uygun değeri olan parçacık en iyi çözüm G_{best} olarak ifade edilir. Parçacıkların hızları ve konumlarının güncellenmesi Denklem 2.3 ve 2.4 ‘te verilmiştir.

$$V_i^{(k+1)} = w * V_i^k + c_1 * r_1 * (x_{i,best}^k - x_i^k) + c_2 * r_2 * (x_{gbest}^k - x_i^k) \quad (2.3)$$

$$x_i^{(k+1)} = x_i^k + V_i^{(k+1)} \quad (2.4)$$

c_1 ve c_2 hızlanma katsayıları olup sabit sayılardır. 0.2 ile 2 arasında değer almaktadır. r_1 ve r_2 rastgele sayılardır. 0 ile 1 arasında değeri değişmektedir. Her iterasyonda yenilenir. w eylemsizlik ağırlığı olarak tanımlanır ve 0.1 ile 1 arasında değer almaktadır.

PSO’da parçacıklar, her bir iterasyonda çoklu arama alanındaki konumlarını değiştirirler. PSO’da arama alanındaki parçacıklar hareketi Şekil 2.5’te gösterilmiştir [25].



Şekil 2.5. PSO parametrelerin vektörel gösterimi [25]

x_i^k ve x_i^{k+1} sırasıyla sürüdeki parçacığın anlık konumunu ve bir sonraki konumu, V_i^k ve V_i^{k+1} sürüdeki parçacığın anlık hızını ve bir sonraki hızı, V_i^{Pbest} ve V_i^{Gbest} P_{best} parçacığın hızı ve G_{best} parçacığın hızını ifade etmektedir.

P_{best} ve G_{best} parçacıklarının hızları Denklem 2.5 ve 2.6'da verilmiştir.

$$Pbest_i^k = \begin{cases} x_{i,best}^{k-1} & \text{if } f(x_i^k) \geq f(x_{i,best}^{k-1}) \\ x_i^k & \text{else} \end{cases} \quad (2.5)$$

$$Gbest^k = \min(f(x_{1,best}^k), f(x_{2,best}^k), f(x_{3,best}^k), \dots, f(x_{n,best}^k)) \quad (2.6)$$

PSO algoritması aşağıdaki adımlara göre çalışır.

Adım 1.Başlangıç parametrelerini ve parçacık sayısını belirle.

Adım 2.Parçacık konum değerini rastgele oluştur.

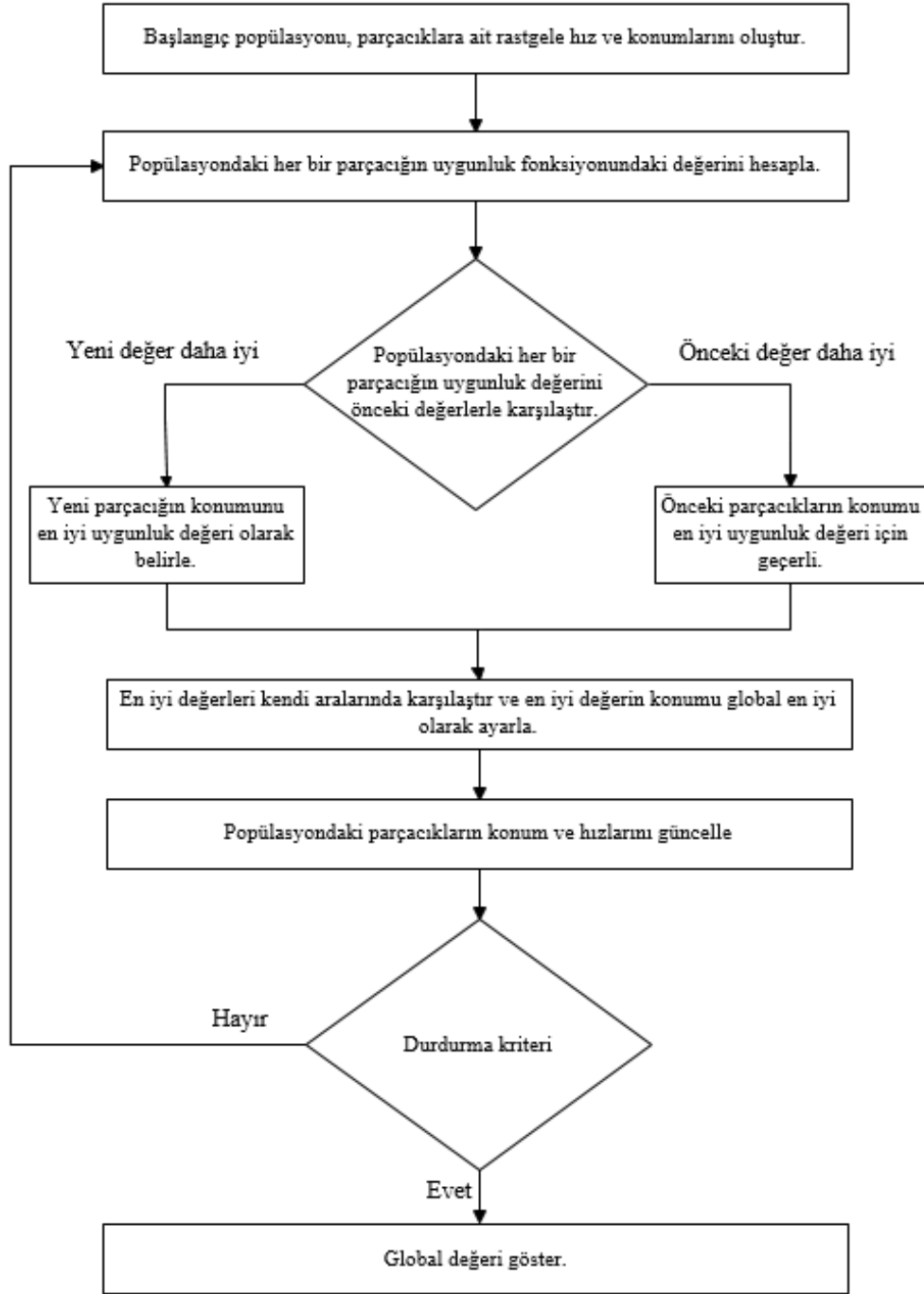
Adım 3.Sürüdeki her bir parçacık için uygunluk fonksiyonundaki değeri hesapla.

Adım 4.Sürüdeki uygunluk fonksiyonu sonucunda yerel (P_{best}) ve küresel (G_{best}) en iyi parçacık değerini hesapla.

Adım 5.Sürüdeki parçacıkların yeni konum ve hızlarının değerini Denklem 2.3 ve 2.4'e göre güncelle.

Adım 6.Algoritma tekrarlama şartını sağlanana kadar üçüncü adımdan itibaren tekrarla.

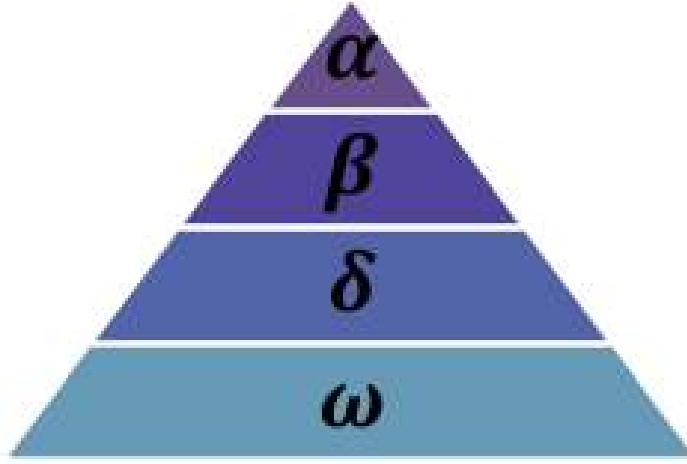
PSO algoritmasına ait akış diyagramı Şekil 2.6'da gösterilmiştir.



Şekil 2.6. PSO algoritmasına ait akış diyagramı

2.3.2. Gri Kurt Optimizasyon (GWO) Algoritması

Gri Kurt Optimizasyon(GWO) Algoritması araştırmacı Mirjalili vd. [26] tarafından 2014 yılında gri kurtların doğadaki avlanma ve toplumsal yaşam biçimlerinden esinlenerek geliştirilen sürü zekasına dayalı meta sezgisel optimizasyon algoritmadır. Algoritmada alfa(α), beta(β), delta(δ) ve omega(ω) olarak dört çeşit kurt bulunmaktadır. Bu durum hiyerarşik olarak Şekil 2.7’de gösterilmiştir.



Şekil 2.7. Gri kurt hiyerarşisi (baskınlık yukarıdan aşağıya azalır) [26]

Alfa kurtlar, kurt sürüsünde lider konumdadır. Beta kurtlar, alfa kurtlara yardımcı olan ikincil kurtları ifade etmektedir. Omega kurtlar alfa ve beta kurtlar tarafından seçilmektedir. Alfa, beta veya omega kurtlar sınıfına dahil olmayan kurtlar delta kurtlar olarak adlandırılır [27].

GWO’da en iyi çözüm α olarak adlandırılır. İkinci ve üçüncü en iyi çözüm ise sırasıyla β ve delta (δ) olarak adlandırılır. Geriye kalan aday çözümler ise omega (ω) olarak adlandırılır. Avlanma (optimizasyon) α , β ve δ tarafından yönlendirilir [28]. Denklem 2.7 ve 2.8 algoritmada avı çevreleme ve konumu güncellemek için kullanılır.

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (2.7)$$

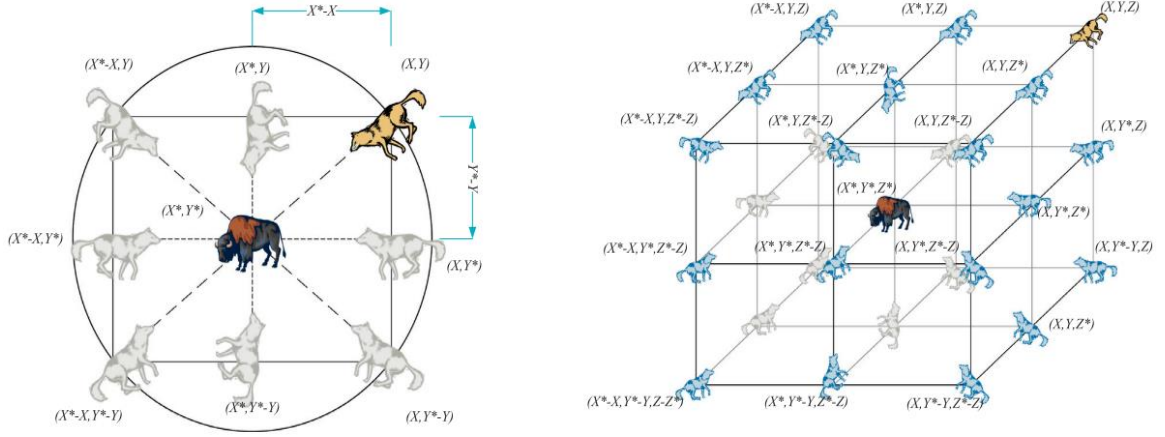
$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2.8)$$

t değeri iterasyonu, $\vec{A}$ ve $\vec{C}$ katsayı vektörlerini ifade etmektedir. $\vec{X}_p$ avın konumu, $\vec{X}$ bir gri kurdun konumu ifade eden vektörlerdir. $\vec{A}$ ve $\vec{C}$ katsayı vektörlerinin hesaplanması Denklem 2.9 ve 2.10’da belirtilmiştir.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (2.9)$$

$$\vec{C} = 2\vec{r}_2 \quad (2.10)$$

$\vec{a}$ ’nın bileşenleri yinelemeler sırasında lineer olarak 2’den 0’a düşürülür. $\vec{r}_1$ ve $\vec{r}_2$ [0, 1] arasındaki rastgele bir vektördür. $\vec{r}_1$ ve $\vec{r}_2$ rastgele vektörleri, kurtların Şekil 2.8’de gösterilen 2D ve 3D uzayındaki noktalardan herhangi bir yere ulaşmasına izin verir [29]



Şekil 2.8. Gri kurtların 2D ve 3D uzayındaki konum vektörleri ve olası sonraki konumları [29]

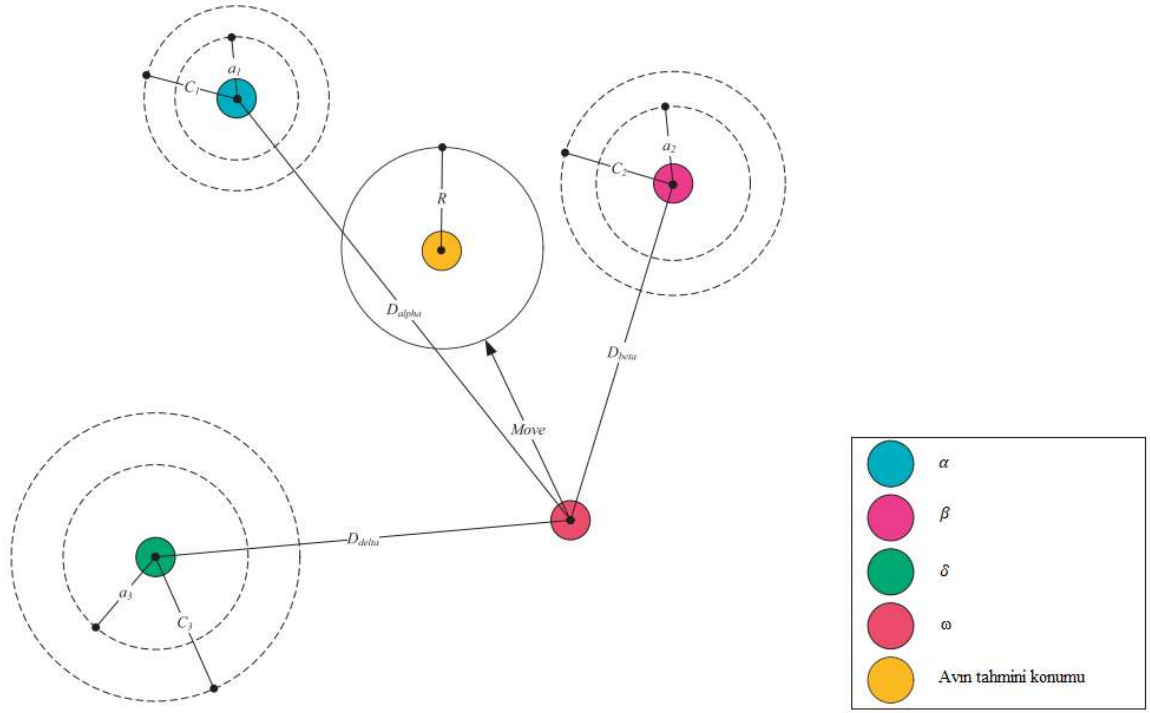
Gri Kurt Algoritmasında alfa, beta ve delta kurtlarının avın mevcut konumu hakkında daha iyi bilgisi olduğu varsayılmaktadır. Avlanma alfa kurt liderliğinde yapılır ve gerektiğinde diğer kurtların katılımı sağlanabilir. Alfa(α), beta(β), delta(δ) kurtlarının konumları Denklem 2.11, 2.12 ve 2.13'te belirtildiği gibi güncellenebilir.

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (2.11)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (2.12)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (2.13)$$

$\vec{X}_\alpha$ alfa kurtlarının, $\vec{X}_\beta$ beta kurtlarının, $\vec{X}_\delta$ delta kurtlarının konumu ifade etmektedir. $\vec{C}_1$, $\vec{C}_2$ ve $\vec{C}_3$ rastgele vektörler, $\vec{X}$ mevcut çözümün konumu göstermektedir [28]. Şekil 2.9'da bir arama alanında alfa, beta ve deltaya göre nasıl güncellendiğini gösterir. Alfa, beta ve delta avın konumunu tahmin eder ve diğer kurtlar avın etrafındaki konumlarını rastgele günceller.



Şekil 2.9. GWO'da konum güncelleme [26]

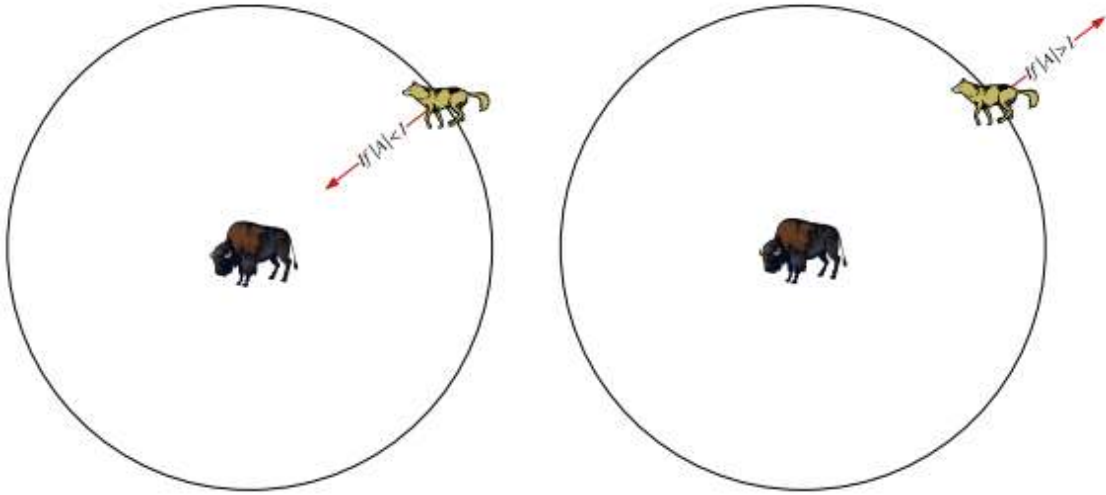
Her bir iterasyonda α , β ve δ kurtlarının konumları güncellenir. Mevcut çözümlerin son konumları Denklem 2.14 ve 2.15'te verilmiştir [26]

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \vec{D}_\delta\end{aligned}\tag{2.14}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}\tag{2.15}$$

Burada $\vec{A}_1$, $\vec{A}_2$ ve $\vec{A}_3$ rastgele vektörleri, $\vec{X}(t+1)$ avın yeni konumu, t ise iterasyon sayısını ifade etmektedir.

GWO' da hedefteki avın konumu tespit edildikten sonra hedefe saldırı işlemi gerçekleştirir. Saldırı işlemi hedefteki avın yorulup hareketsiz kaldıktan sonra gerçekleşir. Matematiksel model olarak Denklem 2.9'da belirtilen $\vec{A}$ değerine bağlı olarak saldırı gerçekleşir. $\vec{A}$ Değeri 2'den 0'a doğru azaltılır. Eğer $\vec{A}$ değeri 1'den büyükse kurtlar ava saldırırmaz avdan uzaklaşır ve yeni bir av aramaya başlar, $\vec{A}$ değeri 1'den küçük ise kurtlar ava saldırır [27]. Karar verme durumu Şekil 2.10'da belirtilmiştir. İşlem maksimum iterasyon sayısına ulaşıncaya kadar veya uygunluk fonksiyonunda en iyi çözümü bulana kadar devam eder.



Şekil 2.10. Gri kurtların ava saldırma ve av arama durumu [30]

GWO algoritması aşağıdaki adımlara göre çalışır.

Adım 1. Başlangıç parametrelerini ve kurt popülasyon sayısını belirle.

Adım 2. Her bir kurt için uygunluk fonksiyon değerini hesapla.

Adım 3. En iyi üç uygunluk değerleri seçip bunları α , β ve δ kurt olarak kaydet.

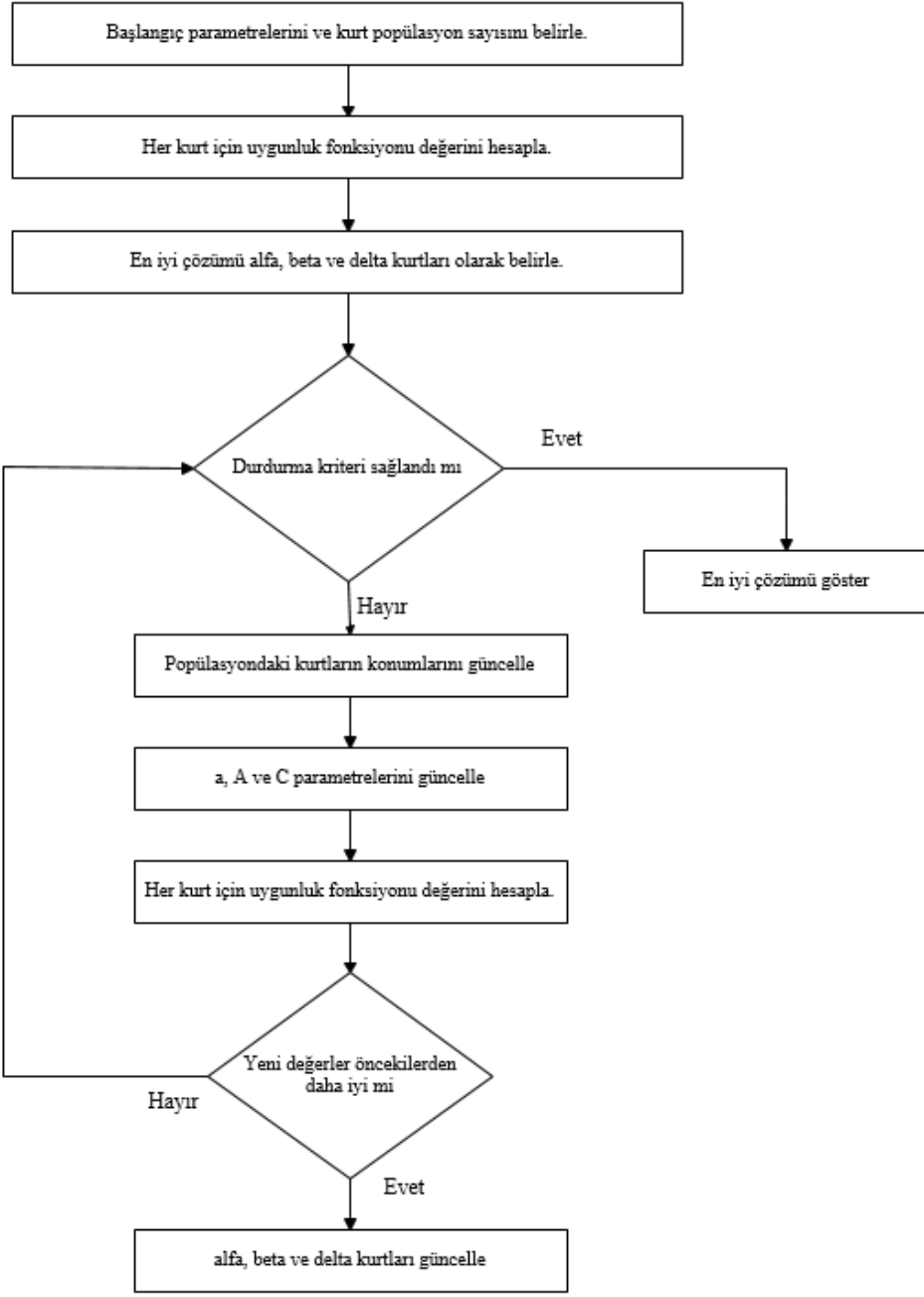
Adım 4. Kurtların konumlarını Denklem 2.15 ile güncelle.

Adım 5. a, A ve C parametrelerini güncelle ve her bir kurdun uygunluk değerini hesapla.

Adım 6. $\vec{X}_\alpha$, $\vec{X}_\beta$, ve $\vec{X}_\delta$ parametrelerini hesapla.

Adım 7. Algoritma tekrarlama şartını sağlanana kadar dördüncü adımdan itibaren tekrarla.

GWO algoritmasına ait akış diyagramı Şekil 2.11’de gösterilmiştir.



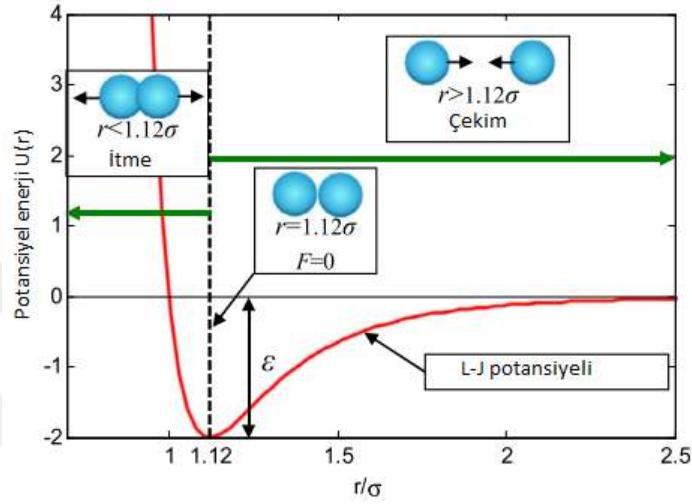
Şekil 2.11. PSO algoritmasına ait akış diyagramı

2.3.3. Atom Arama Optimizasyon (ASO) Algoritması

Atom Arama Algoritması Zhao vd [31] tarafından atomların temel moleküler dinamiklerden esinlenerek oluşturulmuştur. Başlangıçta sıvı için önerilen Lennard-Jones (L-J) potansiyeli bir çift atom arasındaki etkileşim kuvvetine yaklaşan basit bir matematiksel modeldir. i. ve j. atomlar arasındaki L-J potansiyeli genellikle Denklem 2.16'daki gibi tanımlanır.

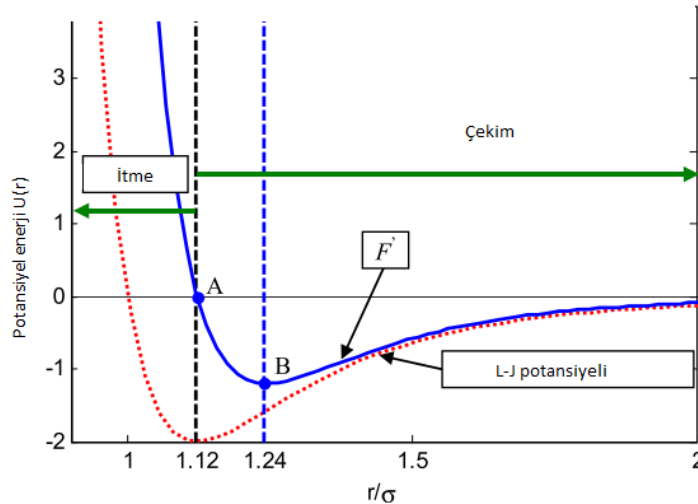
$$U(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \quad (2.16)$$

Burada ε etkileşimin gücünü temsil eden potansiyel derinliği, σ çarpışma çapını gösteren uzunluk ölçeğini, $r_{ij} = x_j - x_i$ konum vektörünü, $x_j = (x_{j1}, x_{j2}, x_{j3})$ j. atomun konum vektörünü temsil etmektedir. $x_i = (x_{i1}, x_{i2}, x_{i3})$ ise i. atomunun konum vektörüdür. $(\frac{\sigma}{r})^{12}$ ve $(\frac{\sigma}{r})^6$ sırasıyla itici ve çekici etkileşimleri temsil eder. L-J potansiyel eğrisinde çekim ve itme bölgeleri Şekil 2.12’de gösterilmektedir [31].



Şekil 2.12. L-J potansiyel eğrisi [31]

L-J potansiyelinden kaynaklanan etkileşim kuvvetinde atomların kuvvet eğrisi Şekil 2.13’te verilmiştir. Şekil 2.13’te gösterildiği gibi atomlar, itme veya çekmeden her zaman belirli bir aralıkta değişen göreceli bir mesafeyi ve dengeleme mesafesine ($r = 1.12 \sigma$) çekiminkinden çok daha büyüktür. Bu nedenle bu model optimizasyon problemlerini doğrudan ele almak için kullanılamaz [31].



Şekil 2.13. Atomların kuvvet eğrisi [31]

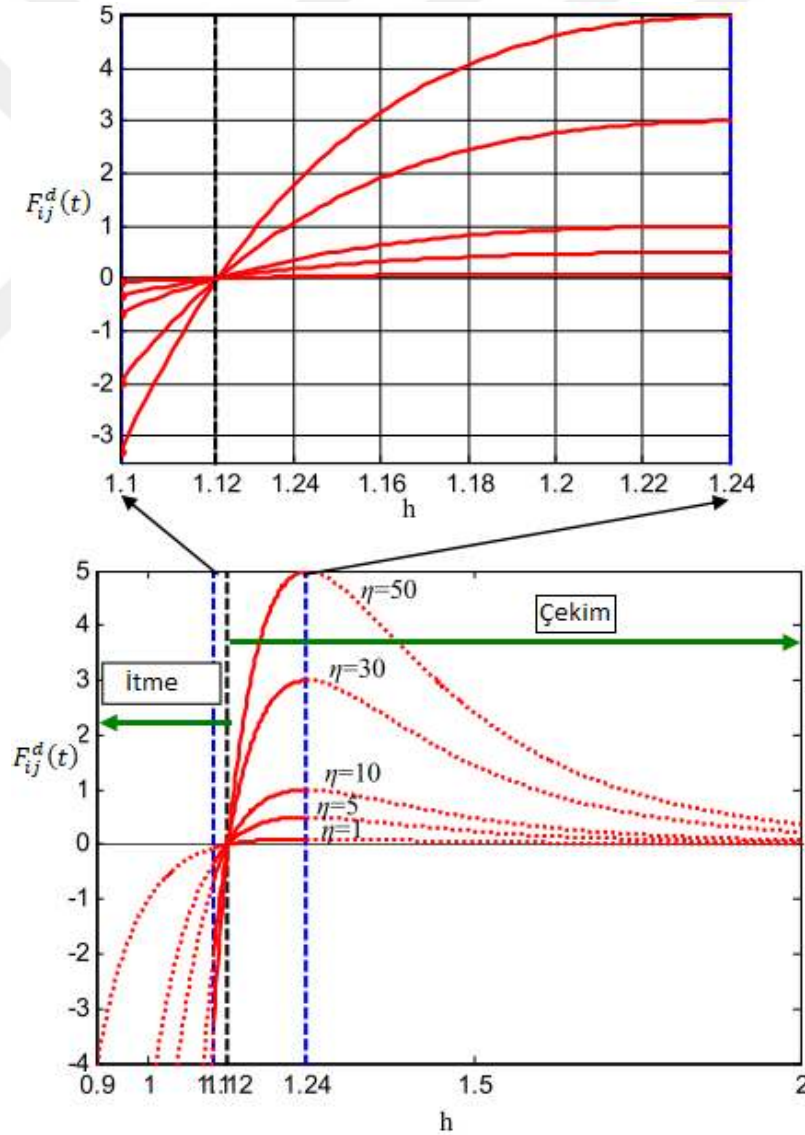
Optimizasyon problemlerini çözmek için L-J potansiyeli denklemin revize edilmiş hali Denklem 2.17'deki gibi tanımlanmıştır.

$$F_{ij}^d(t) = -\eta(t)[2(h_{ij}(t))^{13} - (h_{ij}(t))^7] \quad (2.17)$$

burada $\eta(t)$ itme bölgesini veya çekim bölgesini ayarlamak için kullanılan derinlik fonksiyonu olarak tanımlanır. Bu fonksiyon Denklem 2.18'deki gibi tanımlanmıştır.

$$\eta(t) = \alpha(1 - \frac{t-1}{T})^3 e^{-\frac{20t}{T}} \quad (2.18)$$

α derinlik ağırlığı, t mevcut iterasyon sayısı, T maksimum iterasyon sayısıdır. $F_{ij}^d(t)$ 'nin 0,9 ile 2 arasında değişen h 'a karşılık gelen η fonksiyonun davranışı Şekil 2.14'te verilmiştir [31].



Şekil 2.14. Farklı η değerlerine karşılık gelen $F_{ij}^d(t)$ fonksiyonun davranışı [31]

ASO'da keşfi artırmak için daha küçük bir fonksiyon değerine sahip itmenin bir alt sınırı $h = 1.1$ 'e ve daha büyük bir fonksiyon değerine sahip bir üst çekim limiti $h = 1.24$ 'e ayarlanır. Bu nedenle h Denklem 2.19'daki gibi tanımlanır.

$$h_{ij}(t) = \begin{cases} h_{min} & \frac{r_{ij}}{\sigma(t)} < h_{min} \\ \frac{r_{ij}}{\sigma(t)} & h_{min} \leq \frac{r_{ij}}{\sigma(t)} \leq h_{max} \\ h_{max} & \frac{r_{ij}}{\sigma(t)} > h_{max} \end{cases} \quad (2.19)$$

burada h_{min} ve h_{max} ölçeklenmiş mesafenin, sırasıyla, alt ve üst sınırlarını temsil eder [32]. $\sigma(t)$ uzunluk ölçeği ise Denklem 2.20'deki gibi tanımlanır.

$$\sigma(t) = \left\| x_{ij}(t), \frac{\sum_{j \in Kbest} x_{ij}(t)}{K(t)} \right\|_2 \quad (2.20)$$

Bir atom popülasyonunun bir alt kümesi olan $Kbest$, en iyi fonksiyon uygunluk değerlerine sahip ilk K atomlarından oluşur. h_{min} ve h_{max} Denklem 2.21 ve 2.22'deki gibi tanımlanır.

$$h_{min} = g_0 + g(t) \quad (2.21)$$

$$h_{max} = u \quad (2.22)$$

sürüklenme faktörü olarak $g(t)$, algoritmayı keşif aşamasından sömürü aşamasına geçirebilir ve Denklem 2.23'teki gibi tanımlanır.

$$g(t) = 0.1 \times \sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) \quad (2.23)$$

ASO'da, arama alanı içindeki her atomun konumu, kütlesi ile ölçülen bir çözümü temsil eder. Aralarındaki mesafeye göre tüm atomlar, birbirlerini çekecek ya da itecektir. Daha hafif olan atomlar daha ağır olanlara doğru hareket etmeye teşvik edecektir. Daha ağır atomların hızı daha küçüktür, bu da onları yerel alanlarda daha iyi çözümler aramasına neden olur. Daha hafif atomlar daha büyük bir hıza sahip olduğundan bu durum tüm arama alanında geniş çözüm bulabilmektedir [31]. Atom arama optimizasyon algoritmasına göre, d . boyuttaki i . atoma etki eden dinamik olarak değişen komşu atomların uyguladığı itme ve çekme kuvvetlerinin vektörel toplamı olan toplam etkileşim kuvvetleri Denklem 2.24'teki gibi verilmiştir [33].

$$F_i^d(t) = \sum_{j \in Kbest} rand_j F_{ij}^d(t) \quad (2.24)$$

$rand_j$ [0,1] arasında rastgele bir sayı, bir atom popülasyonunun bir alt kümesi olan $Kbest$, en iyi fonksiyon uygunluk değerlerine sahip ilk K atomlarından oluşur. ASO'daki her atomun en iyi atomla bir kovalent bağına sahip olduğunu varsayalım. Böylelikle her bir atom, en iyi atomdan

gelen bir kısıtlama kuvveti ile hareket eder. Bu durumda i . Atomun kısıtlaması Denklem 2.25'teki gibi yazılabilir [31].

$$\theta_i(t) = [|x_i(t) - x_{best}(t)|^2 - b_{i,best}^2] \quad (2.25)$$

$x_{best}(t)$ en iyi atomun t . iterasyonundaki konumunu, $b_{i,best}$ ise en iyi atom ile i . atom arasındaki sabit bağ uzunluğunu tanımlar. Dolayısıyla kısıtlama kuvveti Denklem 2.26'daki gibi yazılabilir [31].

$$G_i^d(t) = -\lambda(t)\nabla\theta_i^d(t) = -2\lambda(t)(x_i^d(t) - x_{best}^d(t)) \quad (2.26)$$

Burada $\lambda(t)$ Langragian çarpanı olup $2\lambda \rightarrow \lambda$ Denklem 2.26'da yerine yazıp tekrar düzenlenirse kısıtlama kuvveti Denklem 2.27'deki gibi yazılabilir.

$$G_i^d(t) = \lambda(t)(x_{best}^d(t) - x_i^d(t)) \quad (2.27)$$

$\lambda(t)$ Langragian çarpanı Denklem 2.28'de tanımlanmıştır.

$$\lambda(t) = \beta e^{-\frac{20t}{T}} \quad (2.28)$$

Burada β çarpan ağırlığıdır. i . atomun t . iterasyondaki kütlesi Denklem 2.29 ve 2.30 gibi hesaplanabilir.

$$M_i(t) = e^{-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \quad (2.29)$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \quad (2.30)$$

Burada $m_i(t)$, fonksiyonun uygunluk değeri ile en basit seviyede ölçülebilen, t . yinelemedeki i . atomun kütlesidir. $Fit_{worst}(t)$ ve $Fit_{best}(t)$ sırasıyla t . iterasyondaki minimum ve maksimum uygunluk değerine sahip atomlardır. $Fit_i(t)$ ise i . atomun t . iterasyondaki uygunluk değeri fonksiyonudur [33]. $Fit_{worst}(t)$ ve $Fit_{best}(t)$ Denklem 2.31 ve 2.32 gibi tanımlanır.

$$Fit_{best}(t) = \min Fit_i(t) \quad i \in \{1, 2, \dots, N\} \quad (2.31)$$

$$Fit_{worst}(t) = \max Fit_i(t) \quad i \in \{1, 2, \dots, N\} \quad (2.32)$$

i . atomun $(t+1)$. iterasyondaki hızı ve konumu Denklem 2.33 ve 2.34 gibi tanımlanır.

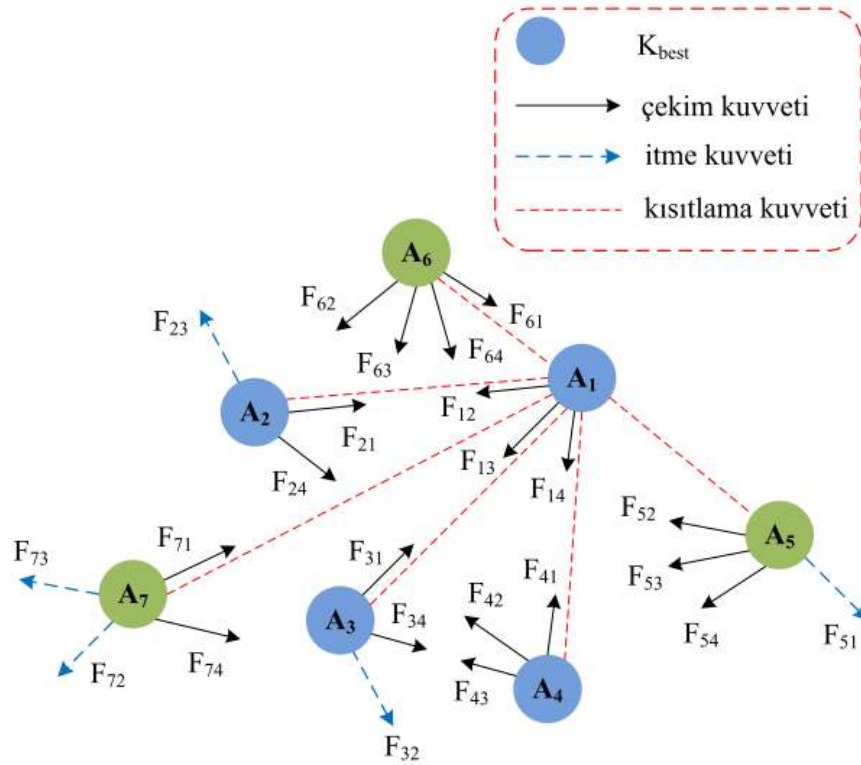
$$v_i^d(t+1) = rand_i^d v_i^d(t) + a_i^d(t) \quad (2.33)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (2.34)$$

Burada d değişkenin boyut sayısı, t mevcut iterasyon sayısı, a_i [0-1] arasında rastgele oluşturulan atomun hızlanma katsayısı, i atom indeksi, v_i atomun i . hızı ve x_i ise atomun i . konumunu tanımlar. ASO algoritmasında her bir atomun K komşuları kadar daha iyi uygunluk değerlerine sahip çok sayıda atomla etkileşime girmesi gerektirir. Bu durum keşiflerin ilk aşamasında keşfi geliştirmek için gereklidir. Algoritmada sömürü işlemini artırmak için yinelemenin son aşamasında atomların K komşuları kadar en az daha iyi uygunluk değerine yakın değerde bazı atom ile etkileşime girmesi gerekir. Bu durumda K Denklem 2.35'teki gibi kademeli bir şekilde azaltılır [31].

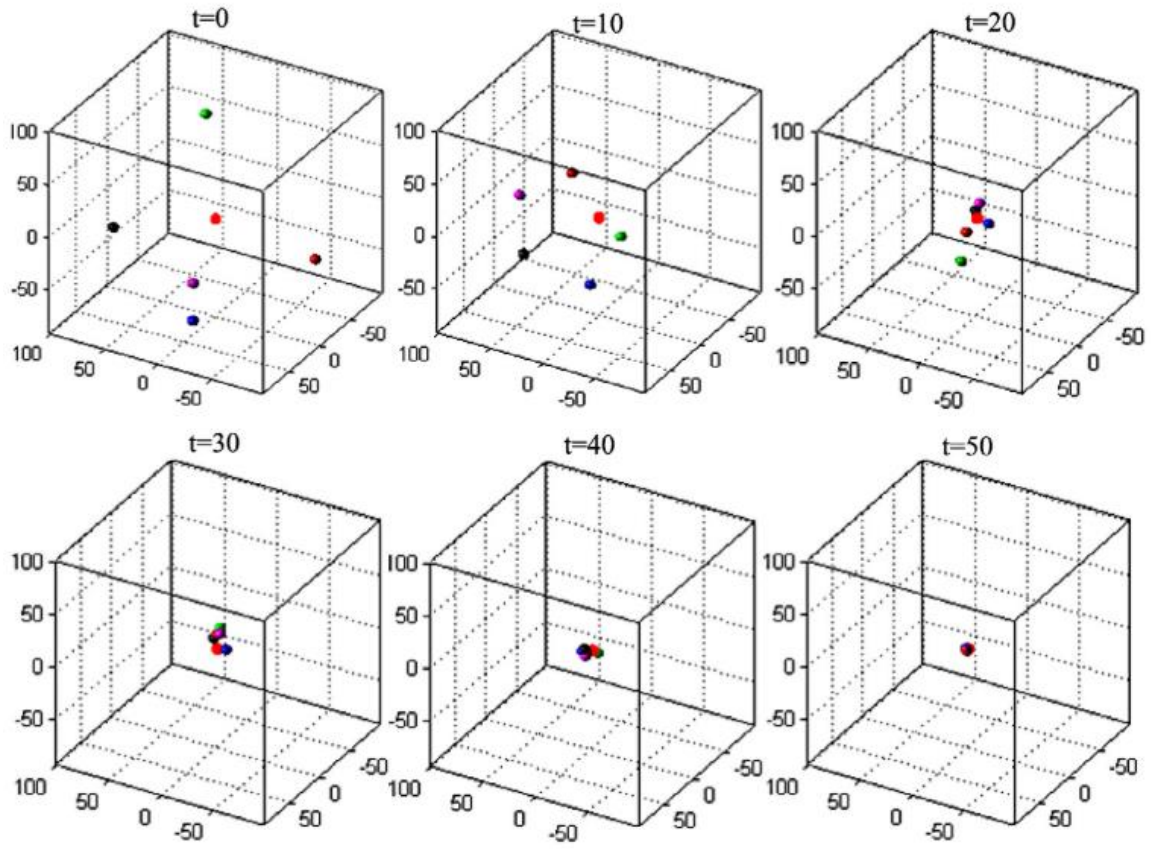
$$K(t) = N - (N - 2) \times \sqrt{\frac{t}{T}} \quad (2.35)$$

Burada T ve t sırası ile maksimum iterasyon sayısı ve mevcut iterasyon sayısı, N sistemdeki toplam atom sayısıdır. Bir atom popülasyonunun kuvvetleri Şekil 2.15'te gösterilmektedir. En iyi uygunluk değerlerine sahip ilk 5 atom K_{Best} olarak kabul edilir.



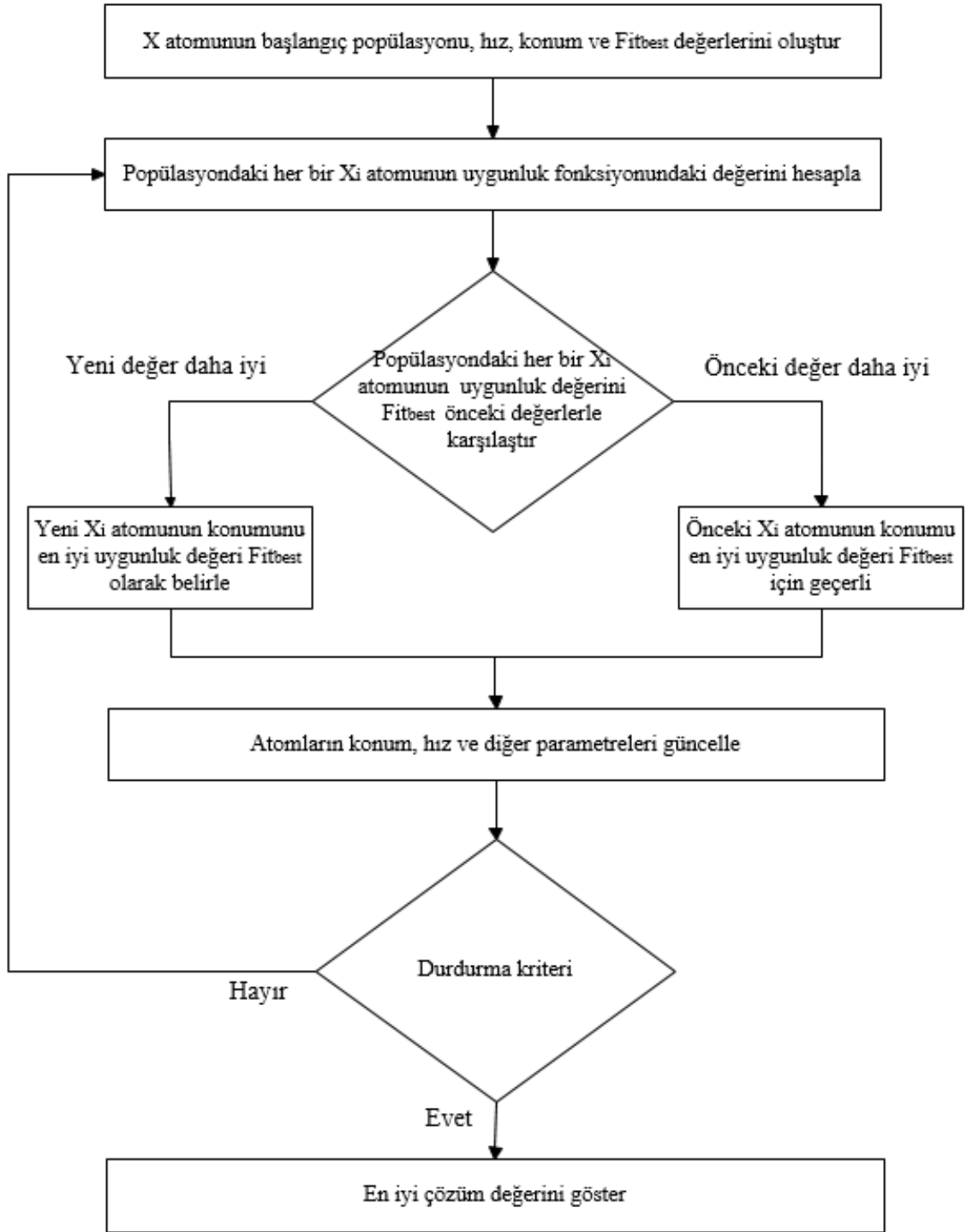
Şekil 2.15. $K=5$ için K_{Best} içeren bir atom sisteminin kuvvetleri [33]

Bu matematiksel model ile atomların nasıl hareket ettiğini incelemek için bir simülasyon yapıldı. 3 boyutlu bir uzayda bir hedef etrafında 5 atomun sürü hareketi Şekil 2.16'da gösterilmektedir. Burada 5 farklı renkli top 5 farklı atomu temsil etmektedir ve kırmızı nokta, her atomun ulaşmak istediği istenen hedefi temsil etmektedir. Başlangıçta 5 atomun pozisyonları, arama alanında rastgele oluşturulmuştur.



Şekil 2.16. 3 boyutlu bir uzayda bir hedef etrafında 5 atomun sürüklenme hareketi [31]

ASO algoritmasına ait akış diyagramı Şekil 2.17’de gösterilmiştir.



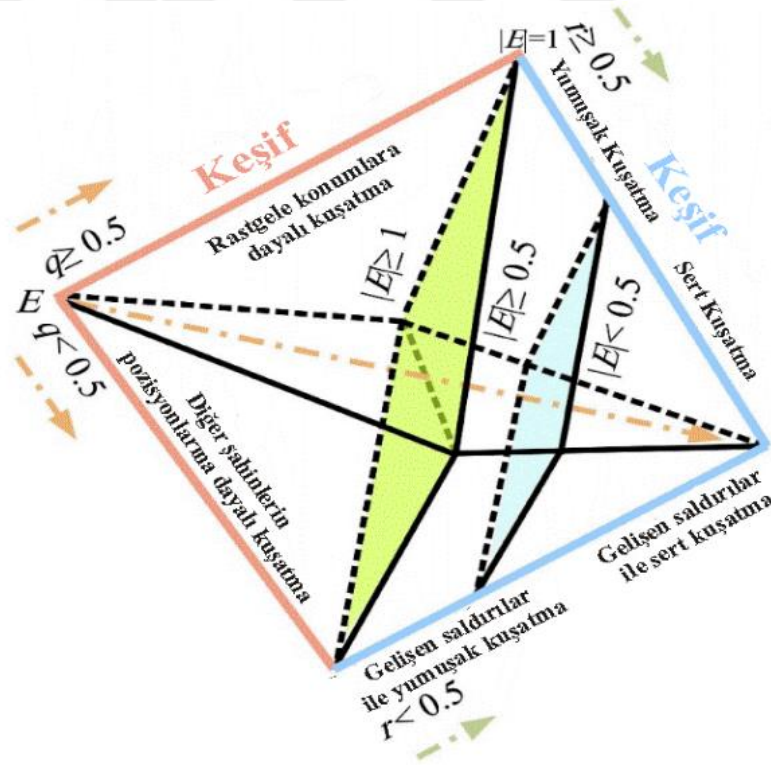
Şekil 2.17. ASO algoritmasına ait akış diyagramı

2.3.4. Harris Şahinleri Optimizasyon (HHO) Algoritması

Harris şahinleri, doğada bilinen en zeki avcı kuşlardan birisidir [34]. Harris şahini ABD’de Arizona’nın güney yarısında bulunan bir şekilde sabit gruplarda hayatta kalan iyi bilinen bir yırtıcı kuştur. Harris’in şahinleri, koşulların dinamik doğasına ve bir avın kaçış biçimlerine bağlı olarak çeşitli kovalama tarzları gösterebilir. En iyi şahin (lider) avına eğilip kaybolduğunda bir değiştirme

taktiği oluşturur ve kovalamaca ava katılan diğer üyelerinden biri tarafından devam ettirilir. Bu durum farklı biçimlerde gözlemlenebilir. Buradaki amaç kaçan tavşanın kafasını karıştırmak içindir. Bu işbirlikçi taktiklerin temel avantajı, Harris şahinlerinin tespit edilen tavşanı yoruluncaya kadar takip edebilmesidir, bu da tavşanın savunmasızlığını artırır. Böylece kaçan avı şaşkına çevirerek avın savunma yeteneklerini geri kazandırmaz ve takım kuşatmasında avla karşı karşıya kalan çoğu zaman en güçlü ve deneyimli şahinlerden biri yorgun tavşanı zahmetsizce yakalayıp onu diğer takım üyeleri ile paylaşır [35]. Harris şahinleri optimizasyon algoritması (HHO), 2019 yılında Heidari vd [35] tarafından geliştirilmiş meta sezgisel popülasyon tabanlı bir optimizasyon algoritmadır.

Harris şahinlerinin avlanma stratejilerinden esinlenerek HHO matematiksel olarak modellenebilir. HHO'nun avlanmadaki aşamaları Şekil 2.18'de gösterilmiştir. HHO, popülasyon tabanlı, gradyan içermeyen bir optimizasyon tekniğidir. Dolayısıyla uygun bir formülasyona tabi olarak herhangi bir optimizasyon problemine uygulanabilir [35].



Şekil 2.18. HHO'nun avlanma aşamaları [3]

A- Keşif Aşaması

Harris şahinlerinin doğasını düşünersek, güçlü gözleriyle avını izleyebilir ve tespit edebilirler ancak bazen av kolaylıkla göremez. Bu nedenle şahinler belki birkaç saat sonra bir avı tespit etmek için çöl alanda bekler, alanı gözlemler ve izler. HHO'da Harris şahinleri bazı yerlerde rastgele tüneler ve iki stratejiye göre bir avı tespit etmeye karar verir. Bu, eşit olasılıklı iki tüneme

stratejisine dayanır. İlk tüneme stratejisinde, tüneme yeri diğer şahinlerin ve tavşanın konumlarına dayanır, böylece şahinler avlarına saldırırken aile üyelerine yakın olabilirler. İkinci tüneme stratejisinde ise rastgele seçilen uzun ağaçlara tüneler, böylece şahin ailesinin av menzili içinde rastgele yerleşirler [36]. Arama stratejileri Denklem 2.36 ve 37’de belirtilen q değerinin [0-1] arasında rastgele olarak belirlenen değerine göre karar verir [35].

$$X(t + 1) = X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| \quad q \geq 0.5 \quad (2.36)$$

$$X(t + 1) = (X_{tavşan}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) \quad q < 0.5 \quad (2.37)$$

Burada $X(t + 1)$ harris şahinlerinin her iterasyonda bir sonraki konum değeri, $X_{tavşan}$ tavşanın konumu, $X(t)$ harris şahinin konumu, r_1, r_2, r_3 ve r_4 değeri 0 ile 1 arasında değişen rastgele sayılardır. LB ve UB sırası ile her iterasyonda güncellenen alt ve üst limit sınır değerleridir. $X_{rand}(t)$ mevcut popülasyonda seçilen rastgele bir harris şahin ve $X_m(t)$ mevcut harris şahin popülasyonun ortalama konumunu temsil etmektedir [35]. Harris şahinlerin ortama konumlarını belirlemek için Denklem 2.38 kullanılır.

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (2.38)$$

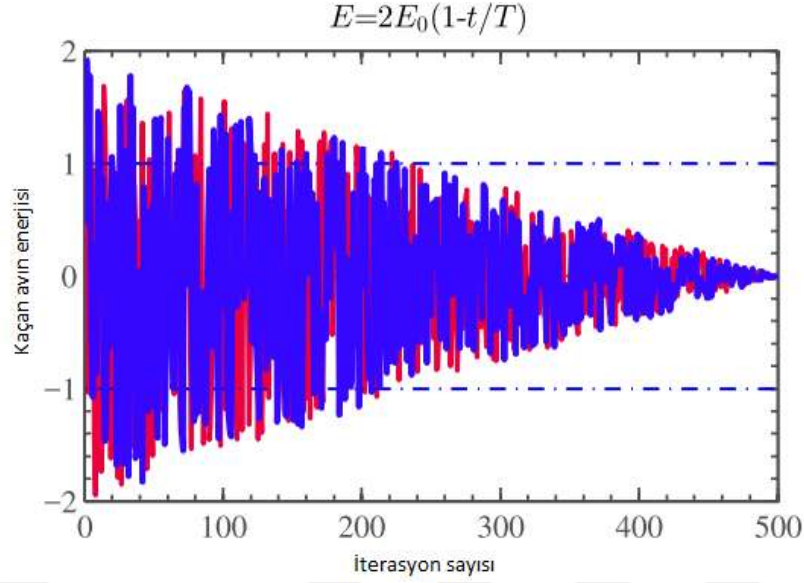
Burada N toplam harris şahinlerin sayısı, $X_i(t)$ ise t . iterasyondaki her harris şahinin konumunu göstermektedir.

Keşif Aşamasından Sömürü Aşamasına Geçiş

Keşif işleminden sonra avın kaçış enerjisine bağlı olarak harris şahinleri farklı saldırı taktikleri geliştirebilir. Avın enerjisi kaçma sırasında azalır ve bir avın kaçış enerjisi Denklem 2.39’daki gibi temsil edilir.

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (2.39)$$

Burada E kaçan avın enerjisi gösterir. T ve t sırası ile maksimum iterasyon sayısı ve mevcut iterasyon sayısını, E_0 ise -1 ile 1 arasında her iterasyonda rastgele değişen kaçan avın başlangıç kaçış enerjisini temsil etmektedir. $|E| \geq 1$ olduğu durumda harris şahinleri herhangi bir avın konumunu tespit etmek için arama yüzeyinin farklı bölgelerini araştırmaktadır. $|E| < 1$ olduğu durumda ise harris şahinleri izledikleri ava saldırmakta ve sömürü aşamasındadır. İterasyon sayısına bağlı olarak kaçan avın enerjisi Şekil 2.19’da gösterilmiştir [35].



Şekil 2.19. Kaçan avın enerjisinin iterasyon sayısına bağlı değişimi [35].

B- Sömürü aşaması

Harris şahinleri, önceki aşamada tespit edilen amaçlanan ava saldırarak sürpriz saldırıyı gerçekleştirir. Bununla birlikte avlar genellikle tehlikeli durumlardan kaçmaya çalışır. Bu nedenle gerçek durumlarda farklı takip stilleri ortaya çıkar. Avın kaçma davranışlarına ve Harris şahinlerinin kovalamaca stratejilerine göre saldırı aşamasını modellemek için HHO'da dört olası strateji önerilmiştir [35].

Yumuşak kuşatma: Harris şahinleri tavşanı daha bitkin hale getirmek için yumuşak bir şekilde etrafını sarar, avın enerjisini azaltmaya çalışır ($r \geq 0.5$ ve $|E| \geq 0.5$) ve ardından ava saldırı işlemini gerçekleştirir. Bu davranış Denklem 2.40 ve 2.41'deki kurallara göre modellenmiştir [35].

$$X(t+1) = \Delta X(t) - E|JX_{tavşan}(t) - X(t)| \quad (2.40)$$

$$\Delta X(t) = X_{tavşan}(t) - X(t) \quad (2.41)$$

Burada $\Delta X(t)$ tavşan ile harris şahinlerinin t . iterasyondaki mevcut konumu arasındaki fark, r kaçan avın avlanma şansı, $J = 2(1 - r_5)$ doğal tavşan hareketlerini simüle etmek için tavşanın kaçış boyunca zıplama gücünü temsil etmektedir. r_5 ise 0 ile 1 arasında her iterasyonda rasgele değişen bir sayıdır.

Sert kuşatma:

$r \geq 0.5$ ve $|E| < 0.5$ olduğu durumda avın enerjisi oldukça düşüktür. Harris şahinleri bu durumda avı neredeyse hiç kuşatmadan saldırmaktadır. Bu durum Denklem 2.42'de gösterilmiştir [35].

$$X(t + 1) = X_{tavşan}(t) - E|\Delta X(t)| \quad (2.42)$$

Aşamalı hızlı dalışlarla yumuşak kuşatma:

Bu aşamada av kaçmak için yeterli enerjiye sahiptir. Harris şahini ise sürpriz sıçramadan önce hala yumuşak kuşatma yapmaktadır. Bu süreç önceki strateji adımından daha akıllıdır. Şahinler yumuşak kuşatmaya başlamadan önce bir sonraki hamlesine Denklem 2.43'e göre karar verir [37].

$$Y = X_{tavşan}(t) - E|JX_{tavşan}(t) - X(t)| \quad (2.43)$$

Denklem 2.43'teki bir hareketin olası sonucunu bir önceki dalışla karşılaştırarak bunun iyi bir dalış olup olmayacağını tespit ederler. Makul olmazsa (avın daha aldatıcı hareketler yaptığını gördüklerinde), tavşana yaklaşırken düzensiz, ani ve hızlı dalışlar da yapmaya başlarlar. Karar verme aşamasında Denklem 2.44'teki kuralı kullanarak levy flight (LF) tabanlı modelle göre dalış yapacaktır [35].

$$Z = Y + S \times LF(D) \quad (2.44)$$

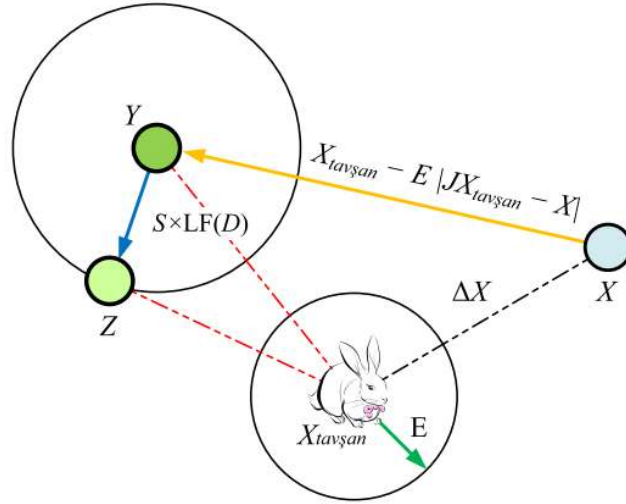
S , $1 \times D$ boyutunda rastgele bir vektör, D problem boyutunu temsil etmektedir. LF fonksiyonu ise Denklem 2.45'teki gibi hesaplanır.

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin \frac{\pi \beta}{2}}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (2.45)$$

Burada β değeri 1.5 olan bir sabittir. u ve v ise değeri 0 ile 1 arasında değişen rastgele bir sayıdır. Yumuşak kuşatmadaki harris şahinlerin konumlarını güncellemek için Denklem 2.46 kullanılır [35].

$$X(t + 1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (2.46)$$

Denklem 2.43 ve 2.44 kullanılarak Y ve Z bulunur. Bu aşamanın basit bir gösterimi Şekil 2.20'de sunulmuştur. Her adımda, sadece en iyi Y ya da Z bir sonraki konum olarak seçilmektedir [38].



Şekil 2.20. Aşamalı hızlı dalışlar ile yumuşak kuşatma durumunda genel vektörlere örnek [39]

Aşamalı hızlı dalışlarla sert kuşatma: Bu aşamada av kaçmak için yeterli enerjiye sahip değildir. Harris şahini, avını yakalamak için sürpriz sıçramadan önce sert bir kuşatma yapar. Sert kuşatma durumu Denklem 2.47 kullanılarak bulunur [37].

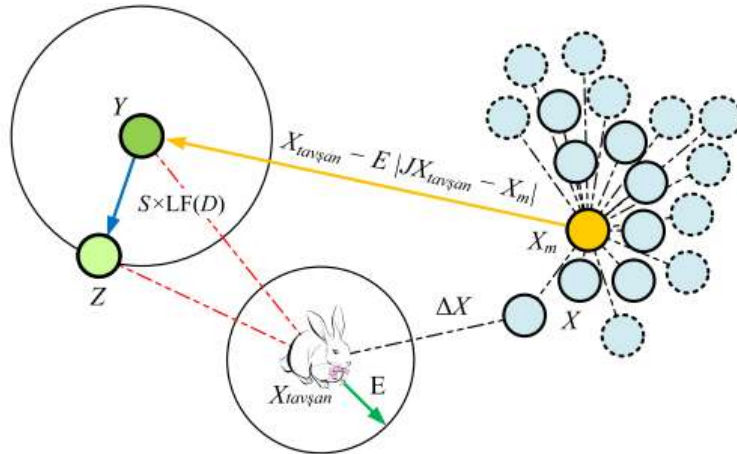
$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (2.47)$$

Y ve Z Denklem 2.48 ve 2.49 kullanılarak bulunur.

$$Y = X_{tavşan}(t) - E |JX_{tavşan}(t) - X_m(t)| \quad (2.48)$$

$$Z = Y + S \times LF(D) \quad (2.49)$$

Burada $X_m(t)$ Eşitlik 2.38 ile elde edilir. Bu aşamanın basit bir gösterimi Şekil 2.21'de sunulmuştur. Her adımda, sadece en iyi Y ya da Z bir sonraki konum olarak seçilmektedir [39].



Şekil 2.21. Aşamalı hızlı dalışlar ile sert kuşatma durumunda genel vektörlere örnek [39]

HHO algoritmasının sözde kodu aşağıdaki gibidir [35].

Başlangıç parametrelerini ve uygunluk fonksiyonu değerlerini başlat

while (durdurma kriteri sağlandı mı?)

Harris şahinlerinin uygunluk fonksiyonu değerlerini hesapla

$X_{tavşan}$ en iyi tavşan konumu olarak ayarla

for (Her şahinin X_i konumu için)

E_0 ve J güncelle

E değerini denklem 2.39'u kullanarak güncelle

if ($|E| \geq 1$) ▷ Keşif aşaması

Konumları denklem 2.36 ve 2.37 kullanarak güncelle

if ($|E| < 1$) ▷ Sömürü aşaması

if ($r \geq 0.5$ ve $|E| \geq 0.5$) ▷ Yumuşak kuşatma

Konumları denklem 2.40 kullanarak güncelle

else if ($r \geq 0.5$ ve $|E| < 0.5$) ▷ Sert kuşatma

Konumları denklem 2.42 kullanarak güncelle.

else if ($r < 0.5$ ve $|E| \geq 0.5$) ▷ Aşamalı hızlı dalışlarla yumuşak kuşatma

Konumları denklem 2.46 kullanarak güncelle

else if ($r < 0.5$ ve $|E| < 0.5$) ▷ Aşamalı hızlı dalışlarla sert kuşatma

Konumları denklem 2.47 kullanarak güncelle

return $X_{tavşan}$

2.4. Uygunluk Fonksiyonu

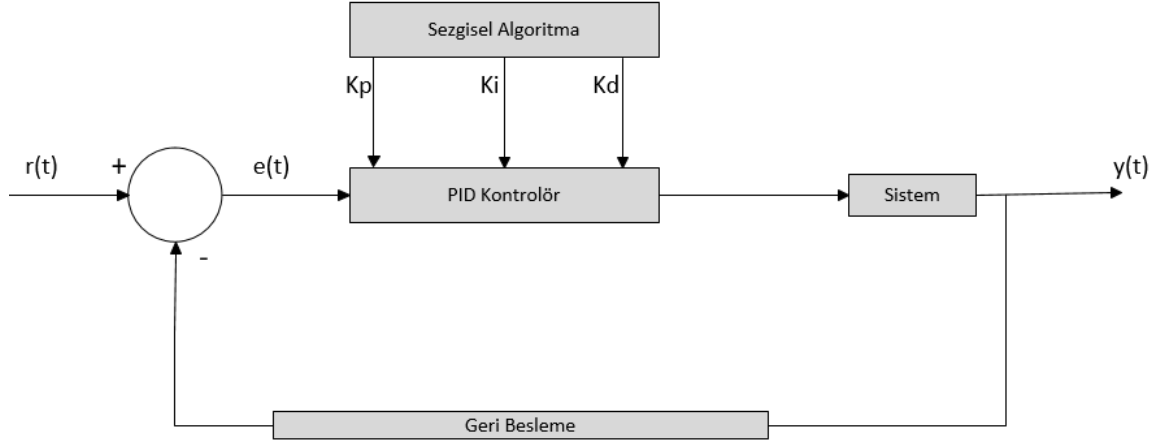
PID kontrolör için literatürde çok fazla uygunluk fonksiyonu bulunmaktadır. Bu uygunluk fonksiyonların her birinin birbirlerine göre avantajı ve dezavantajı vardır. Örneğin IAE ve ISE kriterlerinin dezavantajı yerleşme zamanlarının uzun olmasıdır. Çünkü ISE performans kriterinde tüm hatalar zamandan bağımsızdır [21]. Literatürde kontrolör için en çok kullanan uygunluk fonksiyonları integral of absolute error (IAE), integral of time weighted absolute error (ITAE), integral of squared error (ISE) ve integral of time weighted squarred error (ITSE) fonksiyonlardır. Bu fonksiyonları Tablo 2.2'de belirtilmiştir. Tablo 2.2'de $r(t)$ sistem girişindeki referans değerini, $y(t)$ sistem çıkış değerini, $e(t)$ sistem giriş ile çıkış arasındaki hatayı belirtmektedir.

Tablo 2.2. Yaygın kullanılan uygunluk fonksiyonları

Fonksiyon Adı	Denklem
IAE	$\int_0^{\infty} r(t) - y(t) dt = \int_0^{\infty} e(t) dt$
ISE	$\int_0^{\infty} e^2(t) dt$
ITAE	$\int_0^{\infty} t e(t) dt$
ITSE	$\int_0^{\infty} t e^2(t) dt$

3. BULGULAR VE TARTIŞMA

Bu çalışma Matlab 2020b simülinkde giriş referans değeri $r(t)=1$, PID bloğun filtre katsayısı $N=50000$, simulinkte durdurma zamanı $T=2$ saniye seçilerek çalışmanın simülasyonu yapılmıştır. Simülasyona ait PID denetiminin genel blok şeması Şekil 3.1’de verilmiştir.



Şekil 3.1. Sezgisel algoritmada kullanılan PID blok şeması

PID kontrolör için literatürde kullanılan Tablo 3.1’de belirtilen DC motor özellikleri kullanılmıştır.

Tablo 3.1. DC motor özellikleri [39]

Parametre Tanımı	Sembol	Değer	Birim
Armatür direnci	R	0.4	Ω
Armatür endüktansı	L	2.7	Henry
Motor tork sabiti	K_T	0.015	N.m/A
Zıt elektromotor sabiti	K_v	0.05	V.s
Viskoz sönümlleme katsayısı	B	0.0022	N.m.s/rad
Atalet momenti	J	0.0004	Kg. m^2

Tablo 3.1’de belirtilen DC motorun transfer fonksiyonu Denklem 3.1’de belirtilmiştir.

$$G(s) = \frac{15}{1.08s^2 + 6.1s + 1.63} \quad (3.1)$$

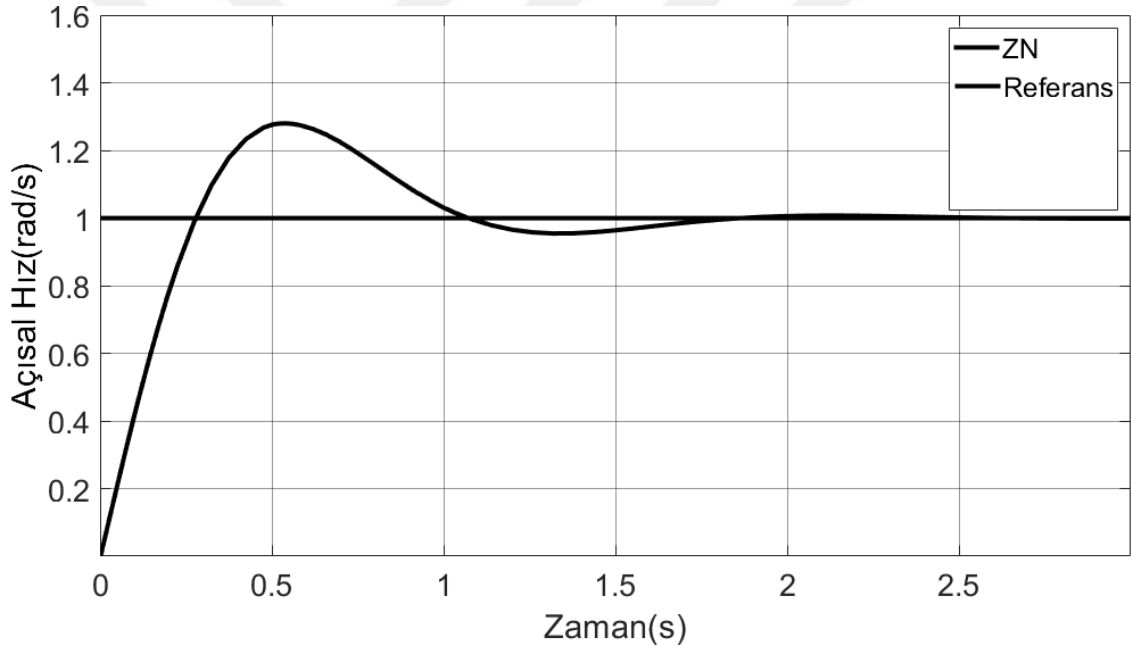
Klasik yöntemlerden Ziegler – Nichols (ZN) birim basamak cevabı yöntemine göre Tablo 3.1’de belirtilen DC motorun sistem çıkış değeri $K=9.19$, ölü zaman değeri $L=0.195$, zaman sabiti değeri

$T=4.745$ olarak hesaplanmıştır. Ziegler – Nichols birim basamak cevabı yöntemine göre PID kontrolörün parametrelerinin hesaplanması Tablo 3.2’de verilen denklemlere göre hesaplanır.

Tablo 3.2. Ziegler – Nichols birim basamak cevabı yöntemine göre PID parametrelerin hesaplanması

K_p	K_i	K_d
$K_p = \frac{1.2}{K * L/T}$	$T_i = 2 * L$ $K_i = K_p / T_i$	$T_d = L/2$ $K_d = K_p * T_d$

Tablo 3.2’ye göre PID parametreleri $K_p = 3.18$, $K_i = 8.15$, $K_d = 0.31$ olarak hesaplanmıştır. Hesaplanan PID parametresine göre DC motorun zamana göre açısal hız tepkisi Şekil 3.2’de verilmiştir.



Şekil 3.2. ZN yöntemine göre ayarlanan PID parametrelerin DC motor sisteminin açısal hız tepkisi

ZN yöntemine göre PID parametrelerine ait yükselme zamanı, yerleşme zamanı, maksimum aşma değeri Tablo 3.3’te belirtilmiştir.

Tablo 3.3. ZN yöntemine göre PID parametrelerin performans değerleri

Yükselme Zamanı (s)	Yerleşme Zamanı (s)	Maksimum Aşma(%)
0.215	1.64	28.1

Şekil 3.2 ve Tablo 3.3'teki değerlere göre yükselme ve yerleşme zamanı ile maksimum aşma değeri yüksek olduğu görülmektedir. Klasik yöntemden farklı olarak uygun PID parametre değerleri belirlemek için Parçacık Sürü Optimizasyonu (PSO), Gri Kurt Optimizasyonu (GWO), Atom Arama Optimizasyonu (ASO) ve Harris Şahinleri Optimizasyonu (HHO) meta sezgisel algoritmalar kullanılmıştır. Çalışma i7 9. nesil işlemci, 24 gb ram, 1 tb ssd özellikteki laptop ile matlab 2020b ortamında simulink model kullanılarak yapılmıştır. Çözücü olarak ode15s seçilmiştir. Matlab'da problem tipleri için farklı çözücüler ve bu çözücüler ile ilgili doğruluk düzeyleri bulunmaktadır [40]. Algoritmaların performans karşılaştırılmasının eşit koşullarda olabilmesi için her algoritmada iterasyon sayısı, popülasyon sayısı, parçacıkların konumu gibi genel kriterler her algoritma ve uygunluk fonksiyonu için aynı koşullarda yapılmıştır. Başlangıç konumları ve diğer ortak parametreler sırası ile Tablo 3.4 ve 3.5'teki gibi belirtilmiştir.

Tablo 3.4. Sezgisel algoritmaların başlangıç konumları

Parçacık	K_p	K_i	K_d
1	5.1322	4.4206	13.8610
2	29.3538	27.1680	2.5082
3	1.4510	28.0987	27.1172
4	29.2076	6.9271	16.0391
5	15.1332	11.8830	13.8428
6	10.8866	16.1198	6.3122
7	16.6040	25.2077	15.3028
8	22.1950	24.6534	2.8309
9	8.6584	9.6317	13.1056
10	19.6741	3.4861	6.4978
11	17.7239	1.1415	4.3737
12	6.8319	29.3838	24.1572
13	16.3048	8.4092	6.1377
14	18.5599	4.1483	9.3575
15	0.3697	14.1485	0.001
16	6.9004	23.8663	20.2961
17	26.0492	10.5039	15.9072
18	25.0089	5.9216	16.7124
19	27.1561	20.1179	0.2927
20	10.2719	5.1187	10.4132

Tablo 3.5. Sezgisel algoritmaların ortak parametreleri

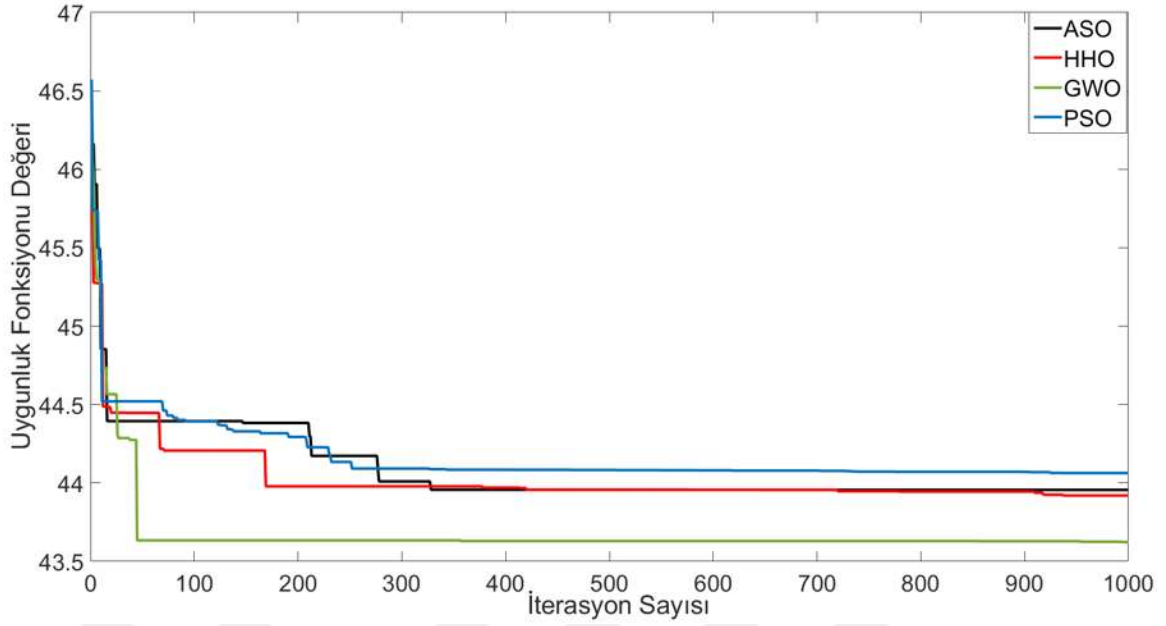
Parametre Tanımı	Sembol	Değer
Arama alanının alt sınırı	LB	0.001
Arama alanının üst sınırı	UB	30
İterasyon sayısı	-	1000
Popülasyon sayısı	-	20

Tablo 3.4'teki sezgisel algoritmaların başlangıç konumları rastgele belirlenmiş ve çalışmada kullanılan tüm algoritmalar ve uygunluk fonksiyonları için bu başlangıç konumları kullanılmıştır. Tablo 3.5'teki arama alanlarının alt ve üst sınır değerleri Ziegler – Nichols birim basamak cevabı yöntemine göre hesaplanan K_p , K_i ve K_d parametrelerinin üzerindeki değere göre [0.001-30] arasında belirlenmiştir. Sezgisel algoritmalara ait özel parametreler Tablo 3.6'daki gibi belirtilmiştir.

Tablo 3.6. Sezgisel algoritmalara ait özel parametreler

Algoritma Adı	Parametre Tanımı	Sembol	Değer
ASO	Çarpan ağırlığı	β	0.2
	Derinlik ağırlığı	α	50
PSO	Eylemsizlik ağırlığı	w	1
	Hızlanma katsayıları	c_1	1.5
		c_2	2
HHO	LF fonksiyon sabiti	β	1.5
GWO	-	-	-

Her bir sezgisel algoritma IAE, ISE, ITAE ve ITSE uygunluk fonksiyonuna göre Tablo 3.4-6'da belirtilen koşullarda 10 defa çalıştırılmıştır. Uygunluk fonksiyonunda sayısal değeri en düşük olan algoritma en iyi uygunluk değeri kabul edilmiştir. Tablo 2.2'deki IAE uygunluk fonksiyonuna göre sezgisel algoritmaların iterasyon sayısına bağlı olarak uygunluk fonksiyonun değerinin değişimi Şekil 3.3'te verilmiştir.



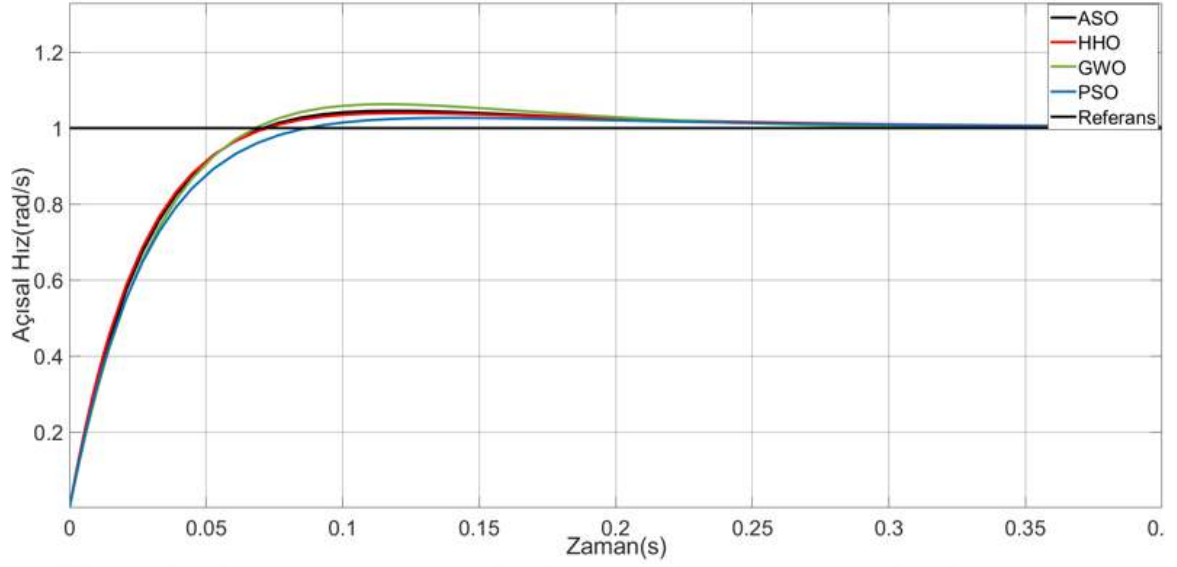
Şekil 3.3. İterasyon sayısına bağlı olarak algoritmaların IAE uygunluk fonksiyonu değerinin değişimi

Şekil 3.3'teki grafikteki değerlere göre meta sezgisel algoritmaların iterasyon sayısına bağlı olarak en az uygunluk fonksiyonu değerine ulaşma hızları yaklaşık ilk 50 inci iterasyonda GWO algoritması diğer algoritmalarla göre en düşük uygunluk fonksiyonu değerine ulaşmıştır. HHO algoritması 185 inci, PSO algoritması 250 inci ve ASO algoritması 320 inci iterasyonda yaklaşık en düşük uygunluk fonksiyonu değerine ulaşmıştır. IAE uygunluk fonksiyonuna göre algoritmaların PID parametrelerinin değerleri ve simülasyonun çalışma süresi Tablo 3.7'de belirtilmiştir.

Tablo 3.7. IAE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi

Algoritma Adı	K_p	K_i	K_d	Simülasyon Süresi (s)
ASO	28.0163	2.19800	2.79010	6071
HHO	26.9381	9.40718	2.92461	11942
GWO	29.9996	0.00184351	2.56251	5517
PSO	21.6528	4.83200	2.69250	5413

Tablo 3.7'deki değerlere göre HHO algoritmasının simülasyon süresi diğer algoritmalarla iki kat fazla olduğu görülmektedir. IAE uygunluk fonksiyonun PID parametrelerinin değerlerine göre algoritmaların DC motorun zamana göre açısal hız tepkisi Şekil 3.4'te verilmiştir.



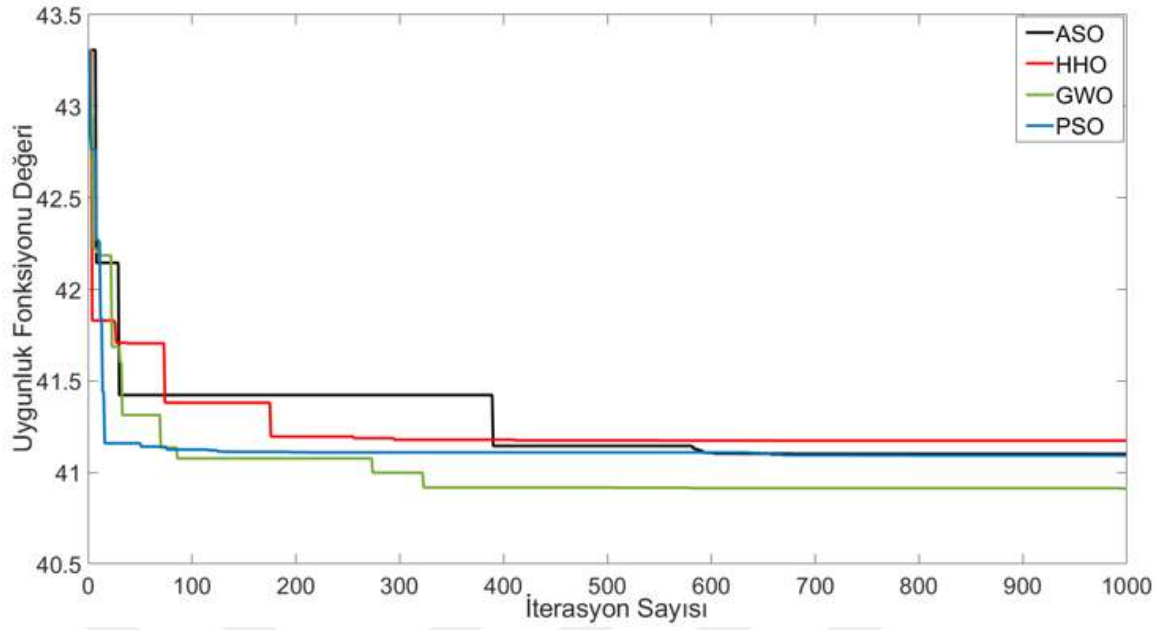
Şekil 3.4. IAE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi

IAE uygunluk fonksiyonuna göre algoritmaların PID parametrelerinin yükselme zamanı, yerleşme zamanı, maksimum aşma değeri Tablo 3.8’de belirtilmiştir.

Tablo 3.8. IAE uygunluk fonksiyonuna göre PID performans değerleri

Algoritma Adı	Yükselme Süresi (s)	Yerleşme Süresi (s)	Maksimum Aşma(%)
ASO	0.0455	0.216	4.58
HHO	0.0451	0.224	3.96
GWO	0.0461	0.224	6.31
PSO	0.0512	0.200	2.68

Şekil 3.4 ve Tablo 3.8’deki değerlere göre tüm algoritmaların hesaplanan PID parametrelerinde maksimum aşma görülmektedir. Maksimum aşma ve yerleşme süresi performans değerleri dikkate alındığında maksimum aşmanın en az olduğu algoritmalar sırası ile PSO, HHO, ASO ve GWO algoritması, yerleşme süresinin en az olduğu algoritmalar ise sırası ile PSO, ASO algoritmasıdır. PSO algoritması en az yerleşme süresi ve maksimum aşma değeri ile çalışmadaki diğer algoritmalarla göre en iyi PID parametrelerinin değerlerini hesaplamıştır. Sezgisel algoritmalar kullanılarak IAE uygunluk fonksiyonu ile hesaplanan PID parametrelerinin performans değerleri ZN yöntemine göre daha iyi değerler hesaplamıştır. Tablo 2.2’deki ISE uygunluk fonksiyonuna göre sezgisel algoritmaların iterasyon sayısına bağlı olarak uygunluk fonksiyonunun değerinin değişimi Şekil 3.5’te verilmiştir.



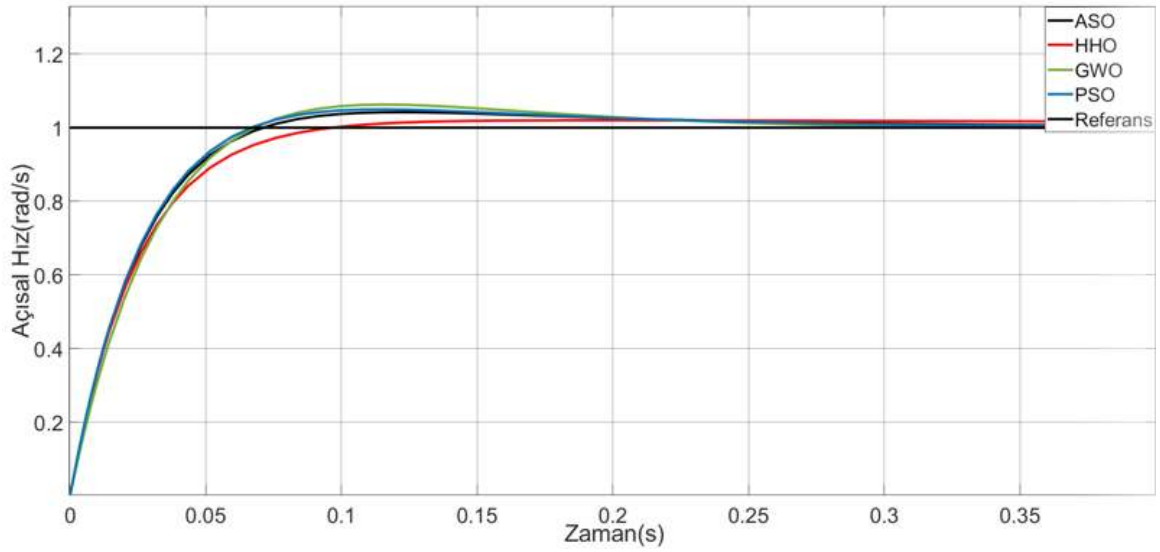
Şekil 3.5. İterasyon sayısına bağlı olarak algoritmaların ISE uygunluk fonksiyonu değerinin değişimi

Şekil 3.5'teki grafikteki değerlere göre meta sezgisel algoritmaların iterasyon sayısına bağlı olarak en az uygunluk fonksiyonu değerine ulaşma hızları yaklaşık ilk 310 uncu iterasyonda GWO algoritması diğer algoritmalara göre en düşük uygunluk fonksiyonu değerine ulaşmıştır. PSO algoritması 50 inci HHO algoritması 185 inci, ve ASO algoritması 600 inci iterasyonda yaklaşık en düşük uygunluk fonksiyonu değerine ulaşmıştır. ISE uygunluk fonksiyonuna göre algoritmaların PID parametrelerinin değerleri ve simülasyonun çalışma süresi Tablo 3.9'da belirtilmiştir.

Tablo 3.9. ISE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi

Algoritma Adı	K_p	K_i	K_d	Simülasyon Süresi (s)
ASO	27.4298	10.8550	2.9240	6798
HHO	19.2137	20.2780	2.9332	12401
GWO	29.9827	0.0225826	2.56713	5580
PSO	29.9973	9.4030	2.9241	5832

Tablo 3.9'daki değerlere göre HHO algoritmasının simülasyon süresi diğer algoritmalarından iki kat fazla olduğu görülmektedir. ISE uygunluk fonksiyonun PID değerlerine göre algoritmaların DC motorun zamana göre açısal hız tepkisi Şekil 3.6'da verilmiştir.



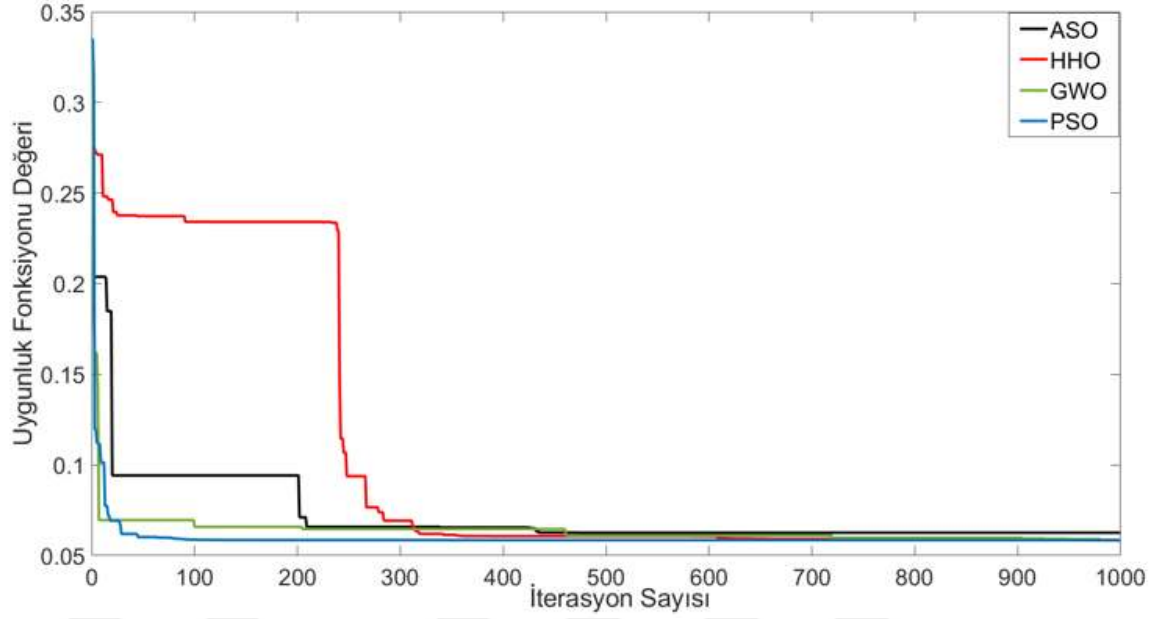
Şekil 3.6. ISE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi

ISE uygunluk fonksiyonuna göre algoritmaların PID parametrelerinin yükselme zamanı, yerleşme zamanı, maksimum aşma değeri Tablo 3.10'da belirtilmiştir.

Tablo 3.10. ISE uygunluk fonksiyonuna göre PID performans değerleri

Algoritma Adı	Yükselme Süresi (s)	Yerleşme Süresi (s)	Maksimum Aşma(%)
ASO	0.0448	0.229	4.18
HHO	0.0507	0.080	1.98
GWO	0.0460	0.224	6.28
PSO	0.4330	0.226	4.96

Şekil 3.6 ve Tablo 3.10'daki değerlere göre tüm algoritmaların hesaplanan PID parametrelerinde maksimum aşma görülmektedir. Maksimum aşma ve yerleşme süresi performans değerleri dikkate alındığında maksimum aşmanın en az olduğu algoritmalar sırası ile HHO, ASO, PSO ve GWO algoritması, yerleşme süresinin en az olduğu algoritmalar ise sırası ile HHO, GWO, PSO ve ASO algoritmasıdır. HHO algoritması en az yerleşme süresi ve maksimum aşma değeri ile çalışmadaki diğer algoritmalarla göre en iyi PID parametrelerinin değerlerini hesaplamıştır. ISE ve IAE uygunluk fonksiyonları ile yapılan hesaplamalarda algoritmaların PID performans değerleri karşılaştırıldığında ISE uygunluk fonksiyonu kullanarak yapılan hesaplamada daha iyi değerlere ulaşılmıştır. Tablo 2.2'deki ITAE uygunluk fonksiyonuna göre sezgisel algoritmaların iterasyon sayısına bağlı olarak uygunluk fonksiyonunun değerinin değişimi Şekil 3.7'de verilmiştir.



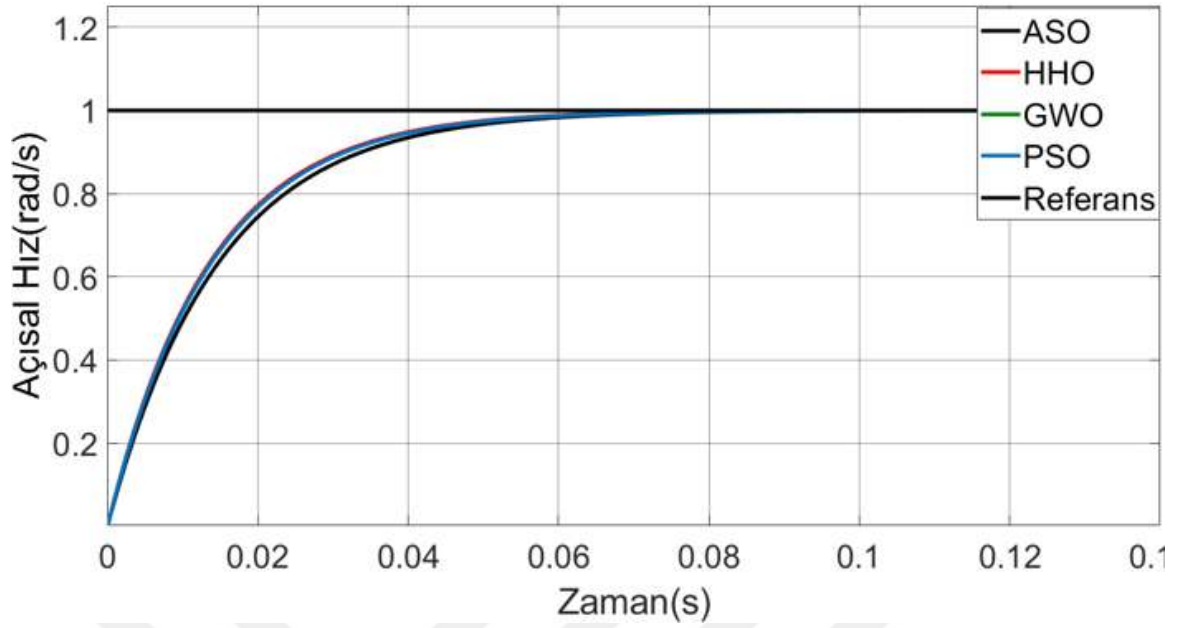
Şekil 3.7. İterasyon sayısına bağlı olarak algoritmaların ITAE uygunluk fonksiyonu değerinin değişimi

Şekil 3.7’deki grafikteki değerlere göre meta sezgisel algoritmaların iterasyon sayısına bağlı olarak en az uygunluk fonksiyonu değerine ulaşma hızları yaklaşık ilk 50 inci iterasyonda PSO algoritması diğer algoritmalara göre en düşük uygunluk fonksiyonu değerine ulaşmıştır. HHO algoritması 400 inci, GWO ve ASO algoritması 450 inci iterasyonda yaklaşık en düşük uygunluk fonksiyonu değerine ulaşmıştır. ITAE uygunluk fonksiyonun göre algoritmaların PID parametrelerinin değerleri ve simülasyonun çalışma süresi Tablo 3.11’de belirtilmiştir.

Tablo 3.11. ITAE uygunluk fonksiyonuna göre PID parametrelerinin değerleri ve simülasyon süresi

Algoritma Adı	K_p	K_i	K_d	Simülasyon Süresi (s)
ASO	27.7164	7.39450	4.89670	8253
HHO	29.9838	7.98077	5.31221	15731
GWO	29.3650	7.92333	5.24263	8764
PSO	29.3606	7.92870	5.24290	7877

Tablo 3.11’deki değerlere göre HHO algoritmasının simülasyon süresi diğer algoritmalarından yaklaşık olarak iki kat fazla olduğu görülmektedir. ITAE uygunluk fonksiyonun PID değerlerine göre algoritmaların DC motorun zamana göre açısal hız tepkisi Şekil 3.8’de verilmiştir.



Şekil 3.8. ITAE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi

ITAE uygunluk fonksiyonuna göre algoritmaların PID parametrelerinin yükselme zamanı, yerleşme zamanı, maksimum aşma değeri Tablo 3.12’de belirtilmiştir.

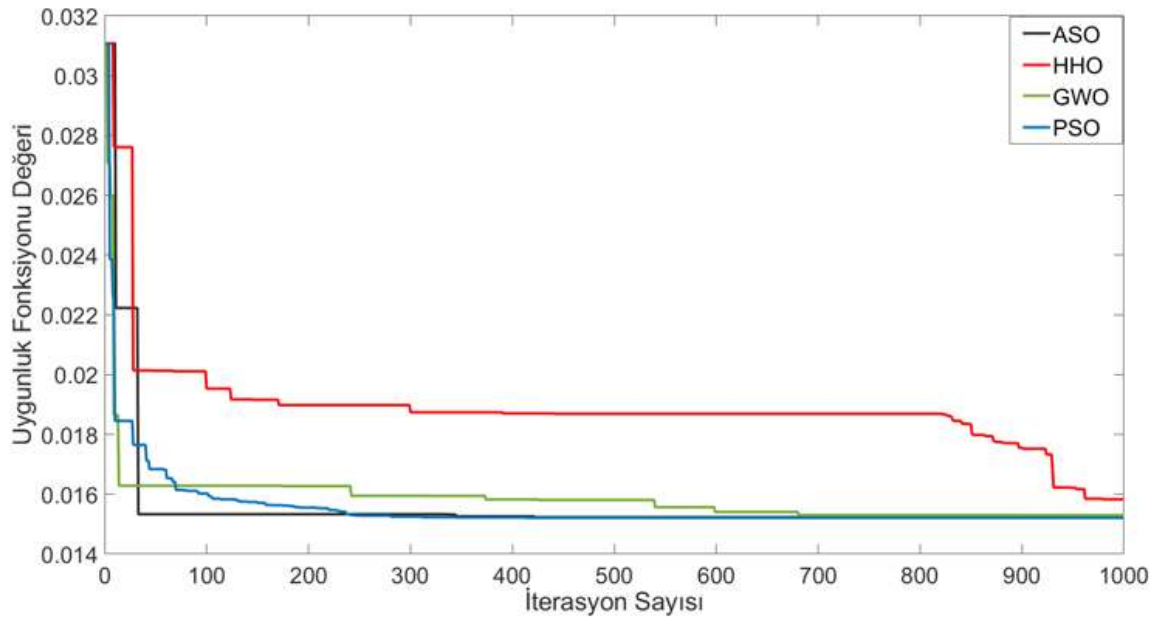
Tablo 3.12. ITAE uygunluk fonksiyonuna göre PID performans değerleri

Algoritma Adı	Yükselme Süresi (s)	Yerleşme Süresi (s)	Maksimum Aşma(%)
ASO	0.0322	0.0574	0.00567
HHO	0.0297	0.0530	0
GWO	0.0302	0.0540	0
PSO	0.0302	0.0540	0

Şekil 3.8 ve Tablo 3.12’deki değerlere göre tüm algoritmaların hesaplanan PID parametrelerinde neredeyse maksimum aşma görülmemiştir. Yerleşme ve yükselme süresi performans değerleri dikkate alındığında yerleşme süresinin en az olduğu algoritmalar sırası ile HHO, PSO ya da GWO ve ASO algoritması, yükselme süresinin en az olduğu algoritmalar ise sırası ile HHO, GWO ya da PSO ve ASO algoritmasıdır. HHO algoritması en az yerleşme ve yükselme süresi değeri ile çalışmadaki diğer algoritmalarla göre en iyi PID parametrelerinin değerlerini hesaplamıştır. ITAE ve ISE uygunluk fonksiyonları ile yapılan hesaplamalarda algoritmaların PID performans değerleri karşılaştırıldığında ITAE uygunluk fonksiyonu kullanarak yapılan hesaplamada daha iyi değerlere ulaşılmıştır. Tablo 3.1’deki motor özellikleri ve ITAE

uygunluk fonksiyonu kullanarak Ekinci vd. [39] 2020 yılında yayınlanan çalışmasında Sinüs Kosinüs Optimizasyon (SCA), Gri Kurt Optimizasyon(GWO), Atom Arama Optimizasyon (ASO) ve Harris Şahinleri Optimizasyon (HHO) algoritmaları kullanarak parçacık sayısı 50, iterasyon sayısı 40, HHO için LF fonksiyon sabiti $\beta = 1.5$, çalışma sayısı 25, arama alanlarının alt ve üst sınır değerleri 0.001 ve 20 koşullarında PID parametreleri belirlemiştir. Bu koşullarda hesaplanan en iyi algoritmanın HHO olduğu PID performans değerleri için yükselme süresi 0.0568, yerleşme süresi 0.1003, maksimum aşmanın sıfır olduğu çalışmasında belirtmiştir. Bu tezde HHO algoritması ile yapılan ITAE uygunluk fonksiyonuyla hesaplanan PID parametrelerinin performans değerleri ile Ekinci vd. [39] yaptığı çalışma karşılaştırıldığında bu tezde yapılan çalışmanın Tablo 3.12'deki HHO algoritmasının değerlerinin daha iyi olduğu görülmektedir.

Tablo 2.2'deki ITSE uygunluk fonksiyonuna göre sezgisel algoritmaların iterasyon sayısına bağlı olarak uygunluk fonksiyonun değerinin değişimi Şekil 3.9'da verilmiştir.



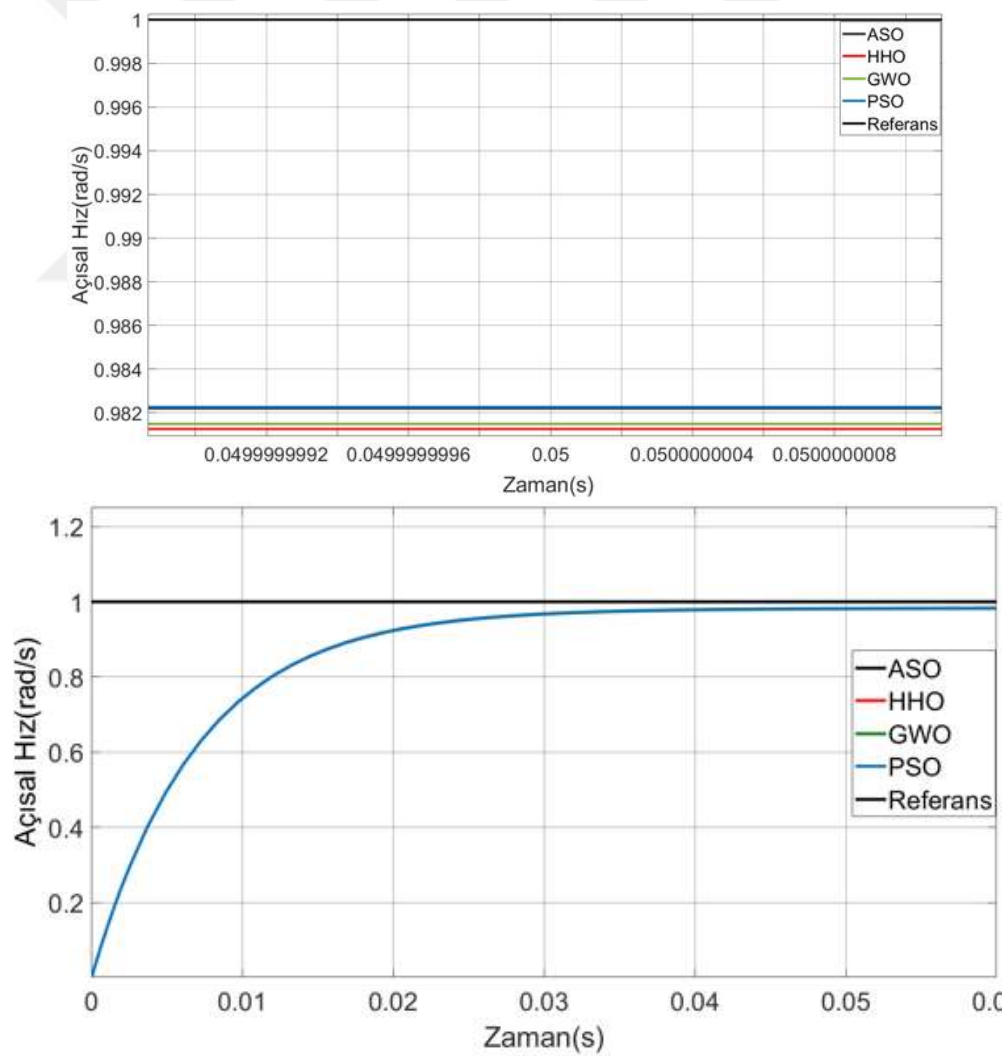
Şekil 3.9. İterasyon sayısına bağlı olarak algoritmaların ITSE uygunluk fonksiyonu değerinin değişimi

Şekil 3.9'daki değerlere göre meta sezgisel algoritmaların iterasyon sayısına bağlı olarak en az uygunluk fonksiyonu değerine ulaşma hızları yaklaşık ilk 35 inci iterasyonda ASO algoritması diğer algoritmalara göre en düşük uygunluk fonksiyonu değerine ulaşmıştır. PSO algoritması 250 inci, GWO algoritması 680 inci, HHO algoritması 960 inci iterasyonda yaklaşık en düşük uygunluk fonksiyonu değerine ulaşmıştır. ITSE uygunluk fonksiyonun göre algoritmaların PID parametrelerinin değerleri ve simülasyonun çalışma süresi Tablo 3.13'te belirtilmiştir.

Tablo 3.13. ITSE uygunluk fonksiyonuna göre PID parametrelerinin deęerleri ve simülasyon süresi

Algoritma Adı	K_p	K_i	K_d	Simülasyon Süresi (s)
ASO	29.8277	9.2536	9.99550	9167
HHO	28.0256	24.5267	9.99556	18854
GWO	28.8279	11.9668	9.99506	8905
PSO	30	10.9990	9.98710	9307

Tablo 3.13'teki deęerlere göre HHO algoritmasının simülasyon süresi dięer algoritmalarından iki kat fazla olduęu görölmektedir. ITSE uygunluk fonksiyonun PID deęerlerine göre algoritmaların DC motorun zamana göre açısal hız tepkisi Şekil 3.10'da verilmiştir.



Şekil 3.10. ITSE uygunluk fonksiyonuna göre algoritmaların DC motor sisteminin açısal hız tepkisi

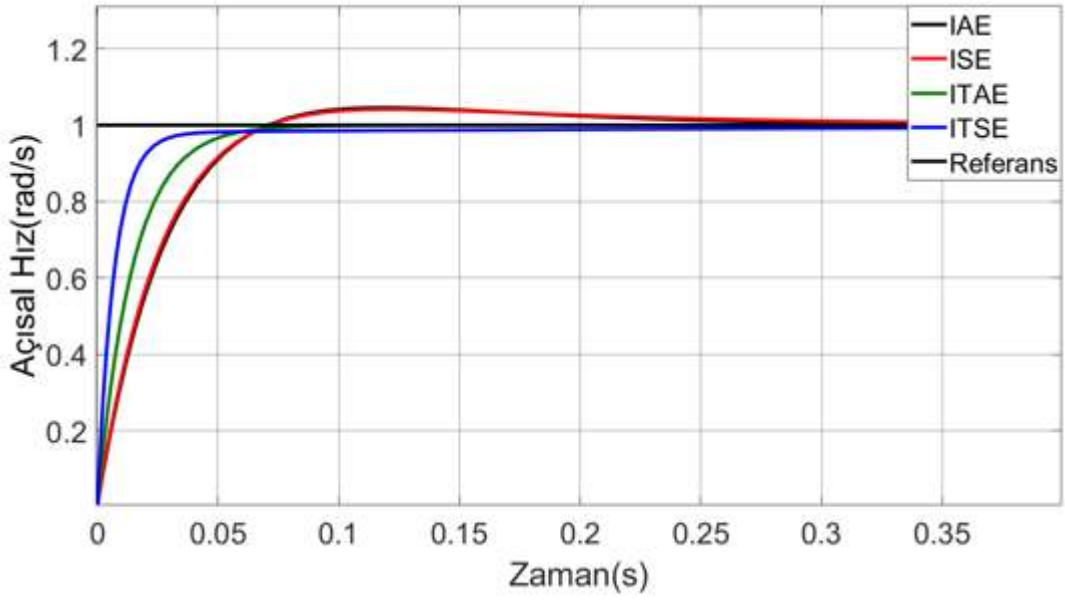
ITSE uygunluk fonksiyonuna göre algoritmaların PID parametrelerinin yükselme zamanı, yerleşme zamanı, maksimum aşma değeri Tablo 3.14'te belirtilmiştir.

Tablo 3.14. ITSE uygunluk fonksiyonuna göre PID performans değerleri

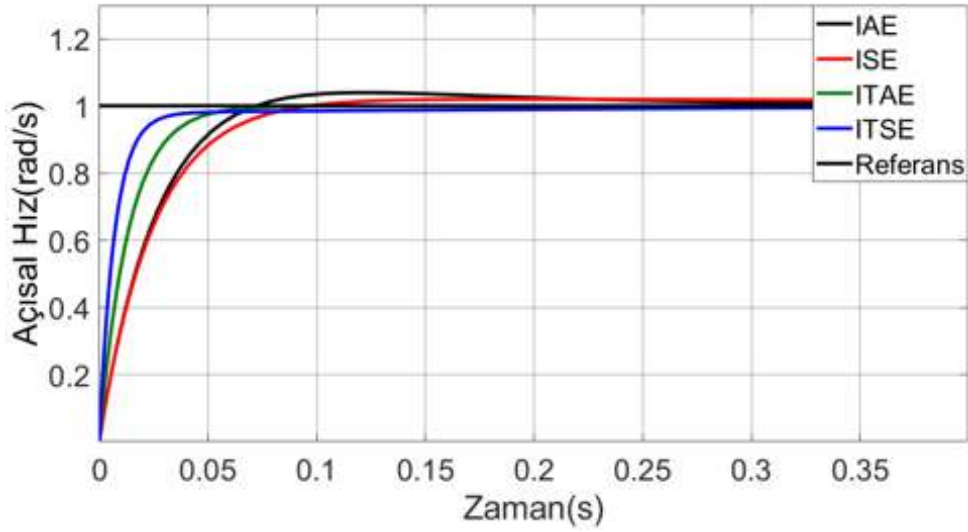
Algoritma Adı	Yükselme Süresi (s)	Yerleşme Süresi (s)	Maksimum Aşma(%)
ASO	0.0168	0.0420	0
HHO	0.0169	0.0446	0
GWO	0.0168	0.0437	0
PSO	0.0168	0.0417	0

Şekil 3.10 ve Tablo 3.14'teki değerlere göre tüm algoritmaların hesaplanan PID parametrelerinde maksimum aşma görülmemiştir fakat $t=0.05$ saniyede tüm algoritmalar için yaklaşık olarak yüzde 1.8 kalıcı durum hatası olduğu görülmektedir. PID parametrelerini belirlerken kalıcı durum hatasının oluşması istenmeyen bir durumdur fakat izin verilen tolerans aralığı kalıcı durum hatası için yüzde 2 değer aralığındadır. Bu nedenle yüzde 1.8 kalıcı durum hatası izin verilen tolerans aralığındadır. Yerleşme ve yükselme süresi performans değerleri dikkate alındığında yerleşme süresinin en az olduğu algoritmalar sırası ile PSO, ASO, GWO ve HHO algoritması, yükselme süresi PSO, GWO ve ASO algoritmalarının eşit olup değeri en az olduğu algoritmalar. PSO algoritması en az yerleşme ve yükselme süresi değeri ile çalışmadaki diğer algoritmalarla göre en iyi PID parametrelerinin değerlerini hesaplamıştır. ITSE ve ITAE uygunluk fonksiyonları ile yapılan hesaplamalarda algoritmaların PID performans değerleri karşılaştırıldığında ITSE uygunluk fonksiyonu kullanarak yapılan hesaplamada daha iyi değerlere ulaşılmıştır. PID parametrelerinin hesaplanmasında kullanılan algoritmaların IAE, ISE, ITAE ve ITSE uygunluk fonksiyonlarına göre DC motorun zamana göre açısal hız tepkisi Şekil 3.11 - Şekil 3.14'te verilmiştir.

Şekil 3.11 incelendiğinde ASO algoritması için; IAE ve ISE uygunluk fonksiyonlarının değişimlerinin daha yavaş olduğu ve referans değerini önemli ölçüde aştığı görülmektedir. ITAE ve ITSE fonksiyonları karşılaştırıldığında ise ITSE'nin daha hızlı yükseldiği ve ITAE'nin nispeten daha yavaş yükseldiği görülmektedir. ITSE'nin hızlı yükselmesi ve referans değerini aşmaması uygulamada aranan özellikleri temsil etmektedir. Şekil 3.12'de HHO algoritması için aynı uygunluk fonksiyonlarının benzer cevaplar verdiği görülmektedir. Ancak burada, IAE ve ISE aşmalarının daha düşük olduğu gözlenmektedir. Ayrıca, HHO algoritması için IAE ve ISE aşmaları birbirleriyle karşılaştırıldığında ISE'nin daha az aşma yaptığı görülür. Bu sonuç, uygunluk fonksiyonlarının algoritmaya bağlı olarak farklı davranışlar sergilediğini ortaya koymaktadır.

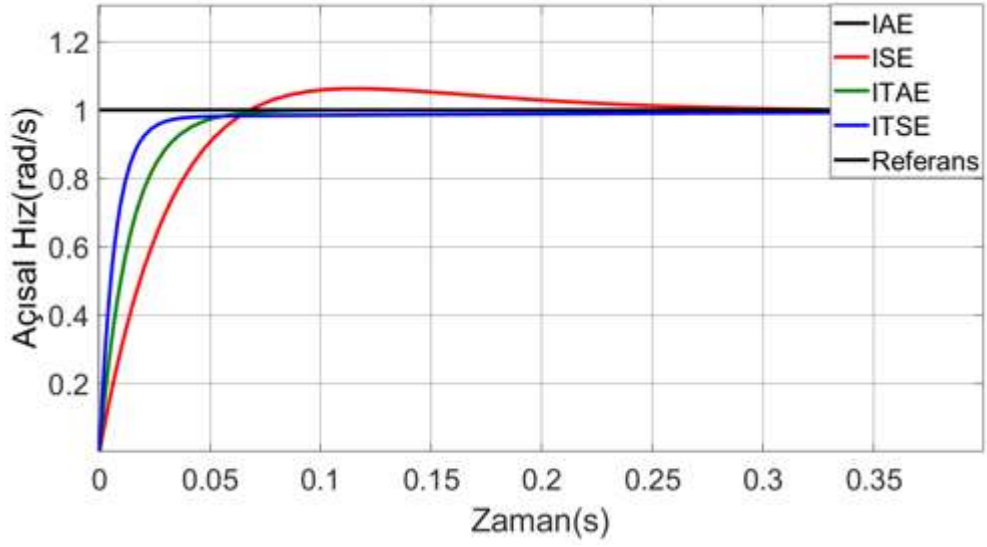


Şekil 3.11. ASO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısal hız tepkisi

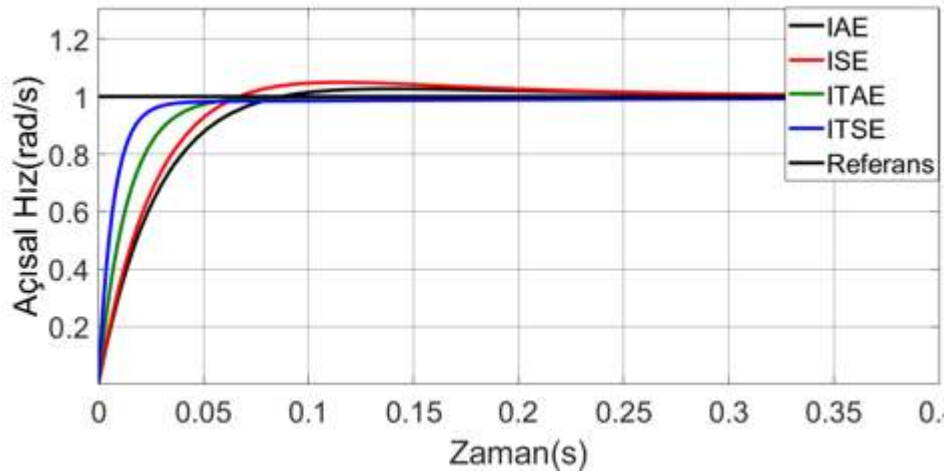


Şekil 3.12. HHO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısal hız tepkisi

GWO ve PSO algoritmalarının uygunluk fonksiyonu cevapları Şekil 3.13 ve Şekil 3.14’de verilmiştir. Burada da IAE ve ISE fonksiyonlarının referans değeri aştığı görülmektedir. Ancak bu kez GWO algoritması için IAE ve ISE fonksiyonlarının tamamen aynı cevabı verdiği görülür. PSO algoritması için HHO’nun tam tersine, IAE’nin aşmasının ISE’nin aşmasına göre daha düşük olduğu görülür. Bununla birlikte ITAE ve ITSE fonksiyonlarının neredeyse kullanılan algoritmalarından bağımsız davrandığı görülmüştür. Bu ikisi arasındaki fark sayısal değerlerin incelenmesiyle anlaşılabilmektedir.



Şekil 3.13. GWO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısal hız tepkisi



Şekil 3.14. PSO algoritması ile uygunluk fonksiyonuna göre DC motorun zamana göre açısal hız tepkisi

Şekil 3.11-14 incelendiğinde PID performans değerlerinin yükselme, yerleşme süresi ve maksimum aşma için ITSE ve ITAE uygunluk fonksiyonları ile yapılan hesaplamalarda IAE ve ISE uygunluk fonksiyonlarına göre çalışmada kullanılan tüm sezgisel algoritmalar için daha iyi değerler hesapladığı görülmektedir. ITSE ve ITAE uygunluk fonksiyonları karşılaştırıldığında ise PID performans değerlerinin yükselme, yerleşme süresi için ITAE uygunluk fonksiyonuna göre çalışmada kullanılan tüm sezgisel algoritmalar için daha iyi değerler hesapladığı görülmektedir.

4. SONUÇLAR

Bu çalışmada DC motor kontrolü için dört farklı sezgisel optimizasyon algoritması kullanılarak kapalı çevrim ve endüstriyel alanlarda yaygın olarak kullanılan PID kontrolörün, en uygun parametreleri belirlenmesi amaçlanmıştır. Çalışma matlab simülasyon ortamında literatürde bulunan DC motor özelliklerinin transfer fonksiyonu kullanılarak yapıldı. PID parametreleri belirlemede PSO, GWO, ASO ve HHO sezgisel optimizasyon algoritmaları ile IAE, ISE, ITAE, ITSE uygunluk fonksiyonlara göre simülasyonu gerçekleştirilmiştir. Bu algoritmaların performanslarını eşit koşullarda karşılaştırmak için belirlenen parçacık sayısı, parçacıkların başlangıç konumları, iterasyon sayısı ve arama alanının alt ve üst sınır değerleri eşit belirlenmiş ve Tablo 2.2’de belirtilen uygunluk fonksiyonlar ile gerçekleştirilmiştir.

Simülasyonu gerçekleştirilen sezgisel algoritmaların tümünün PID performans değerleri klasik yöntemlerden ZN birim basamak cevabı yöntemine göre daha iyi sonuç verdiği görülmüştür. IAE ve ISE uygunluk fonksiyonları kullanılarak yapılan çalışmalarda tüm algoritmalarda maksimum aşma görülmüştür. IAE ve ISE uygunluk fonksiyonları karşılaştırıldığında en az maksimum aşma ve yerleşme zamanı değerine ISE uygunluk fonksiyonu ile yapılan hesaplamada ulaşılmıştır. IAE uygunluk fonksiyonu kullanılarak yapılan çalışmada en az maksimum aşma ve yerleşme zamanı değerine PSO algoritması, ISE uygunluk fonksiyonu kullanılarak yapılan çalışmada ise en az maksimum aşma ve yerleşme zamanı değerine ise HHO algoritması ulaşmıştır. ITAE ile IAE ve ISE uygunluk fonksiyonları ile yapılan çalışmalar karşılaştırıldığında en az maksimum aşma ve yerleşme zamanı değerine tüm algoritmalar için ITAE uygunluk fonksiyonu ile yapılan çalışma daha iyi sonuç vermiştir. ITAE uygunluk fonksiyonu ile yapılan çalışmada yükselme ve yerleşme zamanı en az değerine HHO algoritması ulaşmıştır. ITAE ile ITSE uygunluk fonksiyonları ile yapılan çalışmalar karşılaştırıldığında ITSE uygunluk fonksiyonları ile yapılan çalışmalarda maksimum aşma durumu oluşmamış ayrıca yükselme ve yerleşme zamanı en az değerine ITSE uygunluk fonksiyonları ile yapılan çalışmalarda ulaşılmıştır. ITSE uygunluk fonksiyonu ile yapılan çalışmada yükselme ve yerleşme zamanı en az değerine PSO algoritması ulaşmıştır. Algoritmaların simülasyon çalışma zamanı en fazla olan HHO algoritmasıdır. HHO algoritmasının çalışma zamanı diğer algoritmalarla göre yaklaşık olarak iki kat zaman almaktır.

ÖNERİLER

Tez çalışmasında klasik tam sayı dereceli PID kontrolörleri kullanılmış ve sezgisel optimizasyon algoritmaları ile en uygun katsayılar hesaplanmıştır. Tam sayı dereceli PID kontroller yerine günümüzde kullanılması giderek yaygınlaşmakta olan kesir dereceli kontrolörlerinin ($PI^{\lambda}D^{\mu}$) katsayı ve parametrelerin en uygun şekilde hesaplanması için optimizasyon algoritmaları kullanılabilir.



KAYNAKLAR

- [1] G. Gidemen Cücü, Yenilenen Parçacıklı Sürü Optimizasyonu ile PID Parametrelerinin Ayarlanması, Yüksek Lisans Tezi, İskenderun Teknik Üniversitesi, Mühendislik ve Fen Bilimleri Enstitüsü, 2018.
- [2] G. Laporte, Classical and modern heuristics for the vehicle routing problem, *International Transactions in Operational Research*. 7 (2000) 285–300. [https://doi.org/10.1016/S0969-6016\(00\)00003-4](https://doi.org/10.1016/S0969-6016(00)00003-4).
- [3] Y. Çelik, İ. Yıldız, A.T. Karadeniz, Son Üç Yılda Geliştirilen Metasezgisel Algoritmalar Hakkında Kısa Bir İnceleme, *European Journal of Science and Technology*. (2019) 463–477. <https://doi.org/10.31590/ejosat.638431>.
- [4] S. Akyol, B. Alataş, Güncel Sürü Zekası Optimizasyon Algoritmaları, Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi. 1 (2012). <https://dergipark.org.tr/tr/pub/nufbed/issue/27853/294758> (accessed April 13, 2021).
- [5] K. Anbumani, R. Ranihemamalini, G. Pechinathan, GWO based tuning of PID controller for a heat exchanger process, in: 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), IEEE, 2017: pp. 417–421. <https://doi.org/10.1109/SSPS.2017.8071631>.
- [6] S.S. Sabry, H.S. Obaid, A.M. Al-Chalabi, Grey Wolf Algorithm Based Pid Controller for Congestion Avoidance in Tcp Network, in: 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, 2019: pp. 299–304. <https://doi.org/10.1109/ICICICT46008.2019.8993256>.
- [7] D. Zhi, Optimization of PID Controller for Single Phase Inverter Based on ABC, in: 2019 5th International Conference on Control, Automation and Robotics (ICCAR), IEEE, 2019: pp. 501–504. <https://doi.org/10.1109/ICCAR.2019.8813361>.
- [8] P. Bisht, J. Yadav, Optimal Speed Control of Hybrid Electric Vehicle Using GWO Based Fuzzy-PID Controller, in: 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), IEEE, 2020: pp. 115–120. <https://doi.org/10.1109/ICACCM50413.2020.9212985>.
- [9] W. Nie, Z. Wu, C. Luo, S. Zhang, A Tuning Method for PID Controller Parameters Based on Particle Swarm Optimization (PSO), in: 2020 Chinese Automation Congress (CAC), IEEE, 2020: pp. 497–501. <https://doi.org/10.1109/CAC51589.2020.9327536>.
- [10] Y. Li, Y. Zhang, S. Cui, Y. Liu, X. Lv, Application of optimizing PID Parameters based on PSO in the Temperature Control System of Haematococcus Pluvialis, in: 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), IEEE, 2020: pp. 1850–1853. <https://doi.org/10.1109/ITAIC49862.2020.9338777>.
- [11] Y. Zhu, Y. Sun, G. Chen, Simulation Research on Hydraulic Loading System of Wind Turbine Based on Particle Swarm PID Control, in: 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), IEEE, 2020: pp. 385–388. <https://doi.org/10.1109/TOCS50858.2020.9339746>.
- [12] H. Develi, Ziegler nichols yöntemi ve migo yaklaşımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2005.
- [13] H.O. Erkol, GA ve PSO ile Kontrol Parametrelerinin Optimizasyonu, Karaelmas Fen ve Mühendislik Dergisi. 7 (2017) 179–185.

- [14] M.F. Özlük, H.H. Sayan, Matlab GUI ile DA Motor için PID Denetleyicili Arayüz Tasarımı, İleri Teknoloji Bilimleri Dergisi. 2 (2013) 10–18.
- [15] T. Acıman, PID Kontrol ile DC Motorun Hassas Konum Kontrolü, ,Yüksek Lisans Tezi, Tokat Gaziosmanpaşa Üniversitesi, Fen Bilimleri Enstitüsü, 2019.
- [16] H. Nory, Fırçasız Doğru Akım Motorunun Tasarımı ve Denetimi, ,Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, 2018.
- [17] H. Yazgan, F. Yener, S. Soysal, A. Gür, Comparison Performances of PSO and GA to Tuning PID Controller for the DC Motor, Sakarya University Journal of Science. 23 (2019) 162–174. <https://doi.org/10.16984/saufenbilder.376464>.
- [18] F. Cibo, Doğru Akım Motorunun Bulanık Mantık-PI Kontrolör İle Hız Kontrol Performansının İyileştirilmesi, ,Yüksek Lisans Tezi, Zonguldak Bülent Ecevit Üniversitesi, Fen Bilimleri Enstitüsü, 2020.
- [19] K. Kaplan, M. Kuncan, H. Polat, B. Tepe, H.M. Ertunç, Real Time Position Control of PID and Fuzzy Logic Based DC Motor, Journal of the Institute of Science and Technology. 10 (2020) 900–916. <https://doi.org/10.21597/jist.621724>.
- [20] A. Emiroğlu, T. Yaren, S. Kızır, Kendinden Ayarlamalı Denetleyici ile DA Motor Hız Kontrolü, Journal of Polytechnic. 0900 (2021) 0–3. <https://doi.org/10.2339/politeknik.840519>.
- [21] H. Alruim Alhasan, Parçacık Sürü Optimizasyonu (PSO) Kullanarak Öz Ayarlamalı PID Kontrolör Tasarım, ,Yüksek Lisans Tezi,Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, 2018.
- [22] A.S. Doğru, B. Temel, T. Eren, Comparison of Particle Swarm Optimization and Bat Algorithm Methods in Localization of Wireless Sensor Networks, Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi. 11 (2019) 793–801. <https://doi.org/10.29137/umagd.668724>.
- [23] E. Çetin, Parçacık Sürüsü Optimizasyonu Tabanlı PID Kontrolör İle AA Servomotor Denetimi, ,Yüksek Lisans Tezi, Hitit Üniversitesi, Fen Bilimleri Enstitüsü, 2013.
- [24] A. Ulusoy, M. Güneş, Mobil Robot Kolunun PSO ile Stabilizasyonu, Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi. 22 (2019) 288–297. <https://doi.org/10.17780/ksujes.600310>.
- [25] O. Kapukaya, Adaptif Parçacık Sürü Optimizasyonu ve Karınca Kolonisi Algoritması Kullanarak Öz Ayarlamalı PID Denetleyici Tasarımı, ,Yüksek Lisans Tezi,Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, 2019.
- [26] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software. 69 (2014) 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [27] F.A. Şenel, A.S. Yüksel, F. Gökçe, T. Yiğit, Gri kurt optimizasyon algoritması ile iki boyutlu dizilim yazılımının geliştirilmesi, Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi. 20 (2018) 293–306. <https://doi.org/10.25092/baunfbid.433321>.
- [28] E. Uzlu, Türkiye için gri kurt optimizasyon algoritması ile yapay sinir ağlarını kullanarak enerji tüketiminin tahmini, Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji. 7 (2019) 245–262. <https://doi.org/10.29109/gujsc.519553>.
- [29] I. Koc, O.K. Baykan, I. Babaoglu, Gri Kurt Optimizasyon Algoritmasına Dayanan Çok Seviyeli İmge Eşik Seçimi, Journal of Polytechnic. 21 (2018) 841–847. <https://doi.org/10.2339/politeknik.389613>.

- [30] A. Köybaşı, İ. Yazıcı, Solution of Test Problems with Grey Wolf Optimization Algorithm and Comparison with Particle Swarm Optimization, *Sakarya University Journal of Science*. 24 (2020) 1294–1302. <https://doi.org/10.16984/aufenbilder.788681>.
- [31] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowledge-Based Systems*. 163 (2019) 283–304. <https://doi.org/10.1016/j.knosys.2018.08.030>.
- [32] S. Ekinici, A. Demirören, H.L. Zeynelgil, B. Hekimoğlu, An opposition-based atom search optimization algorithm for automatic voltage regulator system, *Journal of the Faculty of Engineering and Architecture of Gazi University*. 35 (2020) 1141–1157. <https://doi.org/10.17341/GAZIMMFD.598576>.
- [33] E. Eker, M. Kayri, S. Ekinici, Kısıtlı optimizasyon problemlerinin çözümü için atom arama optimizasyon algoritması, *DÜMF Mühendislik Dergisi*. 10 (2019) 841–851. <https://doi.org/10.24012/dumf.540402>.
- [34] L. Lefebvre, P. Whittle, E. Lascaris, A. Finkelstein, Feeding innovations and forebrain size in birds, *Animal Behaviour*. 53 (1997) 549–560. <https://doi.org/10.1006/anbe.1996.0330>.
- [35] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Generation Computer Systems*. 97 (2019) 849–872. <https://doi.org/10.1016/j.future.2019.02.028>.
- [36] S. Ekinici, B. Hekimoglu, A. Demiroren, S. Kaya, Harris Hawks Optimization Approach for Tuning of FOPID Controller in DC-DC Buck Converter, in: 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), IEEE, 2019: pp. 1–9. <https://doi.org/10.1109/IDAP.2019.8875992>.
- [37] O. Akdağ, C. Ates, Abdullah;Yeroğlu, Harris Şahini Optimizasyon Algoritması ile Aktif Güç Kayıplarının Minimizasyonu, *Deu Muhendislik Fakultesi Fen ve Muhendislik*. 22 (2020) 481–490. <https://doi.org/10.21205/deufmd.2020226516>.
- [38] S. Ekinici, B. Hekimoglu, E. Eker, Optimum Design of PID Controller in AVR System Using Harris Hawks Optimization, in: 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, 2019: pp. 1–6. <https://doi.org/10.1109/ISMSIT.2019.8932941>.
- [39] S. Ekinici, D. Izci, B. Hekimoglu, PID Speed Control of DC Motor Using Harris Hawks Optimization Algorithm, in: 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE, 2020: pp. 1–6. <https://doi.org/10.1109/ICECCE49384.2020.9179308>.
- [40] <https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html>, Erişim:3 Ağustos 2021.

ÖZGEÇMİŞ

Abdulmutalip YILDIRIM

[REDACTED]	
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]