GÜNCEL METASEZGİSEL ALGORİTMALARLA DENETLEYİCİ OPTİMİZASYONLARI

Emre HACIİSKENDEROĞLU

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÜNCEL METASEZGİSEL ALGORİTMALARLA DENETLEYİCİ OPTİMİZASYONLARI

Emre HACIİSKENDEROĞLU
0000-0001-5724-1153

Prof. Dr. Fahri VATANSEVER
(Danışman)

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2022

TEZ ONAYI

Emre HACIİSKENDEROĞLU tarafından hazırlanan "GÜNCEL METASEZGİSEL ALGORİTMALARLA DENETLEYİCİ OPTİMİZASYONLARI" adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman: Prof. Dr. Fahri VATANSEVER

Başkan	**:**	Prof. Dr. Fahri VATANSEVER 0000-0002-3885-8622 Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü	İmza
Üye	**:**	Doç. Dr. Sait Eser Karlık 0000-0001-5985-210X Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü	İmza
Üye	**:**	Dr. Öğr. Üyesi Ekrem Düven 0000-0003-4957-6126 Bursa Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Mekatronik Mühendisliği Bölümü	İmza

Yukarıdaki sonucu onaylarım

Prof. Dr. Hüseyin Aksel EREN
Enstitü Müdürü
../../….

B.U.Ü. Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

18/09/2022

Emre HACIİSKENDEROĞLU

TEZ YAYINLANMA
FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezin/raporun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma izni Bursa Uludağ Üniversitesi'ne aittir. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları ile tezin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları tarafımıza ait olacaktır. Tezde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığını ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederiz.

Yükseköğretim Kurulu tarafından yayınlanan "**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**" kapsamında, yönerge tarafından belirtilen kısıtlamalar olmadığı takdirde tezin YÖK Ulusal Tez Merkezi / B.U.Ü. Kütüphanesi Açık Erişim Sistemi ve üye olunan diğer veri tabanlarının (Proquest veri tabanı gibi) erişimine açılması uygundur.

Prof. Dr. Fahri VATANSEVER Emre HACIİSKENDEROĞLU

ÖZET

Yüksek Lisans Tezi

GÜNCEL METASEZGİSEL ALGORİTMALARLA DENETLEYİCİ OPTİMİZASYONLARI

Emre HACIİSKENDEROĞLU

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Fahri VATANSEVER

En uygun duruma getirme olan optimizasyon işlemi, mühendislikte en sık kullanılan adımlardan/süreçlerden birisidir. Herhangi bir mühendislik probleminde, belirli koşullar altında mümkün olan alternatiflerin (çözümlerin) içinde en iyisinin seçimi birçok açıdan (verim, maliyet vb.) son derece önemlidir. Matematiksel olarak bakıldığında bu işlem; belirlenen amaç fonksiyonunu istenen tanım aralığında optimum (en uygun) yapan değeri bulmaktır. Bunun için farklı yöntemler (deterministik, rassal vb.) mevcuttur.

Son yıllarda optimizasyon problemlerinin çözümünde metasezgisel algoritmalardan geniş alanda faydalanılmaktadır. Bu doğrultuda da birçok metasezgisel algoritma geliştirilmiş ve geliştirilmeye devam edilmektedir. Evrimsel, doğadan esinlenilen (sürü tabanlı, fizik/kimya tabanlı, biyolojik tabanlı, insan tabanlı, vb.) gibi türleri olan bu algoritmalar özellikle büyük ve çok boyutlu optimizasyon problemlerinin çözümünde son derece başarılı sonuçlar vermektedirler.

Elektrik-Elektronik Mühendisliği alanında denetleyiciler ve bunların tasarımları çok önemlidir. Temel PID denetleyicilerde ilgili katsayıları ayarlamak için klasik yöntemler (Ziegler-Nichols, Cohen-Coon, vb.) mevcuttur. Gerçekleştirilen tez çalışmasında ise PID türü denetleyicilerdeki katsayıların ayarlanması, güncel altı tane metasezgisel algoritma ile gerçekleştirilmiştir. Bu metasezgisel algoritmalar kullanılarak farklı sistemler için P, PI ve PID denetleyici tasarımları, değişik kriterler (mutlak hatanın toplamı, hata karelerinin toplamı, zaman ağırlıklı mutlak hatanın toplamı, zaman ağırlıklı hata karelerinin toplamı ve özel hata fonksiyonu) altında yapılarak performansları karşılaştırmalı olarak değerlendirilmiştir.

Anahtar Kelimeler: PID denetleyici, optimizasyon, metasezgisel algoritma
2022, xv + 101 sayfa.

ABSTRACT

MSc Thesis

CONTROLLER OPTIMIZATIONS WITH CURRENT METAHEURISTIC ALGORITHMS

Emre HACIİSKENDEROĞLU

Bursa Uludag University
Graduate School of Natural and Applied Sciences
Department of Electronic Engineering

Supervisor: Prof. Dr. Fahri VATANSEVER

Optimization is one of the most frequently used steps/processes in engineering. In any engineering problem, the selection of the best among the possible alternatives (solutions) under certain conditions is extremely important in many aspects (efficiency, cost, etc.). From a mathematical point of view, this process is to find the value that makes the determined objective function the optimum (most appropriate) in the desired definition range. There are different methods (deterministic, random, etc.) for this.

In recent years, metaheuristic algorithms have been widely used in the solution of optimization problems. In this direction, many metaheuristic algorithms have been developed and continue to be developed. These algorithms, which have types such as evolutionary, nature-inspired (swarm-based, physics/chemistry-based, biological-based, human-based, etc.) give extremely successful results, especially in solving large and multidimensional optimization problems.

Controllers and their designs are very important in the field of Electrical and Electronics Engineering. Classical methods (Ziegler-Nichols, Cohen-Coon, etc.) are available for setting the relevant coefficients in basic PID controllers. In thesis, the adjustment of the coefficients in the PID type controllers was carried out with six current metaheuristic algorithms. Using these metaheuristic algorithms, P, PI and PID controller designs for different systems were made under different criteria (sum of absolute error, sum of error squares, sum of time-weighted absolute error, sum of time-weighted error squares and special error function) and their performances were evaluated comparatively.

Key words: PID controller, optimization, metaheuristic algorithm
2022, xv + 101 pages.

TEŞEKKÜR

Tez çalışmamın başlangıcından itibaren, araştırmam boyunca sabırlı rehberlikleri, araştırmamı gerçekte olduğu gibi şekillendirmemde bana en iyi düşünceleri sağladığı ve tavsiyeleri için çok değerli hocam Prof. Dr. Fahri VATANSEVER'e teşekkürlerimi sunarım. Son olarak hayatımın her döneminde olduğu gibi, yüksek lisans öğrenimim boyunca maddi ve manevi desteklerini esirgemeyen değerli aileme sonsuz teşekkürlerimi sunarım.

Emre HACIİSKENDEROĞLU
18/09/2022

İÇİNDEKİLER

	Sayfa
ÖZET	vi
ABSTRACT	vii
TEŞEKKÜR	viii
SİMGELER ve KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xiv
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	8
2.1. Sistem Tanımlama	8
2.2. Kontrol Tasarımı için Gereklilikler	8
2.3. Basamak Cevabının Özellikleri	9
2.4. PID Denetleyici Yapısı	11
2.4.1. P Denetleyici	12
2.4.2. PD Denetleyici	13
2.4.3. PI Denetleyici	14
2.4.4. PID Denetleyici	15
2.5. PID Denetleyici Parametre Belirleme	16
2.6. PID Denetleyicileri için Ayar Kuralları	17
2.8. Wang-Juang-Chan Metodu	19
2.9. Cohen-Coon Metodu	19
2.10. Chien-Hrones-Reswick (CHR) Metodu	20
3. MATERYAL ve YÖNTEM	21
3.1. Aritmetik Optimizasyon Algoritması (AOA)	21
3.1.1. Başlatma Aşaması	22
3.1.2. Keşif Aşaması	22
3.1.3. Sömürü Aşaması	24
3.2. Dinamik Diferansiyel Tavlı Optimizasyon Algoritması (DDAO)	27
3.3. Kel Kartal Arama Optimizasyon Algoritması (BES)	33
3.3.1. Kel Kartalın Avlanma Davranışı	33
3.3.2. Bölge Seçimi	34
3.3.3. Arama Aşaması	35
3.3.4. Saldırı Aşaması	36
3.4. Aquila (Kartal) Optimizasyon Algoritması (AO)	39
3.4.1. AO'nun matematiksel modeli	39
3.5. Martı Optimizasyon Algoritması (SOA)	46
3.5.1. Matematiksel Model	46
3.6. Serçe Arama Algoritması (SSA)	51
3.6.1. Biyolojik Özellikler	51
3.6.2. Matematiksel Model ve Algoritma	51
4. BULGULAR ve TARTIŞMA	56
5. SONUÇ	92
KAYNAKLAR	94
ÖZGEÇMİŞ	101

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
α	Arama alanı değişimi
X_{eniyi}	En iyi değer
t_d	Gecikme süresi/zamanı
C_{Iter}	Geçerli yineleme
$e(t)$	Hata işareti
Ti	İntegral zamanı
Ki	İntegral kazanç
E_{SS}	Kararlı hal hatası
$u(t)$	Kontrol veya sürme işareti
M_P	Maksimum aşım
M_{Iter}	Maksimum yineleme
Kc	Oransal kazanç
P_{ort}	Ortalama değer
r	Rastgele değer
Kd	Türev kazanç
Td	Türev zamanı
t_s	Yerleşme süresi/zamanı
S^k	Yineleme sayısı
t_r	Yükselme süresi/zamanı

Kısaltmalar	Açıklama
AOA	Aritmetik Optimizasyon Algoritması (Arithmetic Optimization Algorithm)
CHR	Chien-Hrones-Reswick Metodu
DDAO	Dinamik Diferansiyel Tavlı Optimizasyon Algoritması (Dynamic Differential Annealed Optimization)
ISE	Hata karelerinin toplamı (Integral square error)
IAE	Mutlak hatanın toplamı (Integral absolute error)
ITSE	Zaman ağırlıklı hata karelerinin toplamı (Integral time square error)
ITAE	Zaman ağırlıklı mutlak hatanın toplamı (Integral time absolute error)
BES	Kel Kartal Arama Algoritması (Bald Eagle Search Optimization Algorithm)
AO	Aquila (Kartal) Optimizasyon Algoritması (Aquila Optimizer)
SOA	Martı Optimizasyon Algoritması (Seagull Optimization Algorithm)
P	(Oransal) denetleyici
PD	(Oransal - türev) denetleyici
PI	(Oransal - integral) denetleyici
PID	(Oransal - integral – türev) denetleyici
SSA	Serçe Arama Algoritması (Sparrow Search Optimization Algorithm)

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.2.	Kapalı çevrim kontrol sistemi……..	9
Şekil 2.3.	Zaman ve frekans bölgesi (Diethorn 1994'ten değiştirilerek alınmıştır)……..	9
Şekil 2.4.	Birim basamak cevabı (Zumbahlen 2006'dan değiştirilerek…….....	10
Şekil 2.5.	Oransal geri besleme kontrol sistemi……	12
Şekil 2.6.	PD geri besleme kontrol sistemi ($Kd = Kc\tau D$)……..	13
Şekil 2.7.	PI kontrol sistemi.....	14
Şekil 2.8.	PID denetleyici tasarımı.....	15
Şekil 3.1.	AOA'nın arama aşamaları (Abualigah ve ark. 2020'den değiştirilerek alınmıştır)……	21
Şekil 3.2.	Kullanılan operatörlerinin optimum alana yaklaşma modeli (Abualigah ve ark. 2020'den değiştirilerek alınmıştır)…….	23
Şekil 3.3.	Aritmetik optimizasyon algoritması akış diyagramı……	26
Şekil 3.4.	Çift fazlı çelik üretim sürecinin açıklaması (Ghafil ve ark. 2020'den değiştirilerek alınmıştır)……	27
Şekil 3.5.	Farklı enerji seviyeleri için arama alanındaki farklı aday çözümleri……...	29
Şekil 3.6.	Diferansiyel tavlı dövme işlemi(Ghafil ve ark. 2020'den değiştirilerek alınmıştır)……	29
Şekil 3.7.	Dinamik diferansiyel tavlama optimizasyonu akış diyagramı…….	32
Şekil 3.8.	Kel kartalın avlanma sırasındaki davranışı(Alsattar 2020'den değiştirilerek alınmıştır)……	33
Şekil 3.9.	BES avlanma aşamasında izlediği yol (Alsattar 2020'den değiştirilerek alınmıştır)……	34
Şekil 3.10.	Doğaçlama seçim aşaması (Alsattar 2020'den değiştirilerek alınmıştır)……..	35
Şekil 3.11.	Kel kartal arama optimizasyonu akış diyagramı……..	38
Şekil 3.12.	Kartal dikey eğimli süzülme davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)……	39
Şekil 3.13.	Kartal kısa süzülme kontür uçuşu davranışı (Abualigah ve ark. değiştirilerek alınmıştır)……	40
Şekil 3.14.	AO'nın spiral şeklindeki davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)……	41
Şekil 3.15.	Kartal yavaş alçalma davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)……	42
Şekil 3.16.	Aquila (Kartal) optimizasyonu akış diyagramı…….....	45
Şekil 3.17.	Arama aracıları arasında çarpışmadan kaçınma (Dhiman ve ark. 2018)……..	46
Şekil 3.18.	Arama aracıları en iyi komşuya doğru hareketi (Dhiman ve ark. 2018)…….	47
Şekil 3.19.	En iyi arama aracısına yakınsama (Dhiman ve ark. 2018)……...	48
Şekil 3.20.	Martının ava spiral saldırı davranışı (Dhiman ve ark. 2018)…….....	48
Şekil 3.21.	Martı optimizasyon algoritması akış diyagramı……...	50
Şekil 3.22.	Serçe arama algoritması akış diyagramı……...	55
Şekil 4.1.	Tasarlanan yazılım aracındaki süreç için temel akış şeması…….	56

Şekil 4.2.	PID parametrelerinin bulunmasındaki akış şeması……..	57
Şekil 4.3.	Sistem-1 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	58
Şekil 4.4.	Sistem-1 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	59
Şekil 4.5.	Sistem-1 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları……	59
Şekil 4.6.	Sistem-1 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları……	60
Şekil 4.7.	Sistem-2 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	61
Şekil 4.8.	Sistem-2 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	62
Şekil 4.9.	Sistem-2 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları……	63
Şekil 4.10.	Sistem-2 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları……	64
Şekil 4.11.	Sistem-3 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	65
Şekil 4.12.	Sistem-3 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	66
Şekil 4.13.	Sistem-3 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları……	67
Şekil 4.14.	Sistem-3 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları……	67
Şekil 4.15.	Sistem-4 Özel fonksiyon kriterine göre karşılaştırmalı birim basamak cevapları……	69
Şekil 4.16.	Sistem-4 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	69
Şekil 4.17.	Sistem-4 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları……	70
Şekil 4.18.	Sistem-4 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları……	71
Şekil 4.19.	Sistem-4 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları……	72
Şekil 4.20.	Sistem-4 PID denetleyici tasarımında AOA algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi……	73
Şekil 4.21.	Sistem-4 PID denetleyici tasarımında AOA algoritmasındaki çözüm no sayısının sistem cevaplarına etkisi……..	73
Şekil 4.22.	Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi……	74
Şekil 4.23.	Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki maksimum alt yineleme iterasyon sayısının sistem cevaplarına etkisi…….	75
Şekil 4.24.	Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki T0 paramatresinin sistem cevaplarına etkisi……	76
Şekil 4.25.	Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki a paramatresinin sistem cevaplarına etkisi……..	77

Şekil 4.26. Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki *Npop* paramatresinin sistem cevaplarına etkisi…… 78
Şekil 4.27. Sistem-4 PID denetleyici tasarımında BES algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi……. 79
Şekil 4.28. Sistem-4 PID denetleyici tasarımında BES algoritmasındaki *Npop* paramatresinin sistem cevaplarına etkisi…… 80
Şekil 4.29. Sistem-4 PID denetleyici tasarımında AO algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi……. 81
Şekil 4.30. Sistem-4 PID denetleyici tasarımında AO algoritmasındaki çözüm no sayısının sistem cevaplarına etkisi……... 82
Şekil 4.31. Sistem-4 PID denetleyici tasarımında SOA algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi……. 83
Şekil 4.32. Sistem-4 PID denetleyici tasarımında SOA algoritmasındaki arama aracı sayısının sistem cevaplarına etkisi…… 84
Şekil 4.33. Sistem-4 PID denetleyici tasarımında SSA algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi……. 85
Şekil 4.34. Sistem-4 PID denetleyici tasarımında SSA algoritmasındaki arama aracı sayısının sistem cevaplarına etkisi…… 86
Şekil 4.35. Sistem-5 Özel Fonksiyon kriterine göre karşılaştırmalı birim basamak cevapları…… 87
Şekil 4.36. Sistem-5 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları…… 88
Şekil 4.37. Sistem-5 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları…… 89
Şekil 4.38. Sistem-5 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları…… 89
Şekil 4.39. Sistem-5 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları…… 90
Şekil 4.40. Sistem-5 için elde edilen ITAE Değerleri…… 91

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 1.1.	Kapalı çevrim PID kontrolü için literatür taraması …..…….	4
Çizelge 2.1.	Bağımsız olarak artan parametrenin etkisi……..	16
Çizelge 2.2.	Geçici tepki yöntemine göre Kp, Ti, Td ve PID parametreleri…..	19
Çizelge 2.3.	PID katsayılarının Wang-Juang-Chan yöntemiyle elde edilmesi...	19
Çizelge 2.4.	PID katsayılarının Cohen-Coon metodu ile elde edilmesi……..	19
Çizelge 2.5.	PID katsayılarının CHR yöntemiyle elde edilmesi…….	20
Çizelge 4.1.	Sistem-1 için algoritma parametre ayarlaması……	57
Çizelge 4.2.	Sistem-1 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	57
Çizelge 4.3.	Sistem-1 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	57
Çizelge 4.4.	Sistem-1 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar……..	59
Çizelge 4.5.	Sistem-1 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar……...	60
Çizelge 4.6.	Sistem-2 için algoritma parametre ayarlaması……	61
Çizelge 4.7.	Sistem-2 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	61
Çizelge 4.8.	Sistem-2 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	62
Çizelge 4.9.	Sistem-2 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar……..	62
Çizelge 4.10.	Sistem-2 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar……..	63
Çizelge 4.11.	Sistem-3 için algoritma parametre ayarlaması……	64
Çizelge 4.12.	Sistem-3 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	65
Çizelge 4.13.	Sistem-3 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	65
Çizelge 4.14.	Sistem-3 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar……..	66
Çizelge 4.15.	Sistem-3 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar……..	67
Çizelge 4.16.	Sistem-4 için algoritma parametre ayarlaması……	68
Çizelge 4.17.	Sistem-4 için özel fonksiyon tasarım kriterine göre karşılaştırmalı sonuçlar……...	68
Çizelge 4.18.	Sistem-4 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	69
Çizelge 4.19.	Sistem-4 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar……..	70
Çizelge 4.20.	Sistem-4 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar……..	70
Çizelge 4.21.	Sistem-4 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar……..	71

Çizelge 4.22. AOA algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi……. 72
Çizelge 4.23. AOA algoritmasındaki çözüm no sayısının Sistem-4 PID denetleyici tasarımına etkisi…… 73
Çizelge 4.24. DDAO algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi……. 74
Çizelge 4.25. DDAO algoritmasındaki maksimum alt yineleme iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi…….. 75
Çizelge 4.26. DDAO algoritmasındaki $T0$ paramatresinin Sistem-4 PID denetleyici tasarımına etkisi…… 75
Çizelge 4.27. DDAO algoritmasındaki a paramatresinin Sistem-4 PID denetleyici tasarımına etkisi…… 76
Çizelge 4.28. DDAO algoritmasındaki $Npop$ paramatresinin Sistem-4 PID denetleyici tasarımına etkisi…… 77
Çizelge 4.29. BES algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi……. 78
Çizelge 4.30. BES algoritmasındaki $Npop$ paramatresinin Sistem-4 PID denetleyici tasarımına etkisi…… 79
Çizelge 4.31. AO algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi…… 80
Çizelge 4.32. AO algoritmasındaki çözüm no sayısının Sistem-4 PID denetleyici tasarımına etkisi…… 81
Çizelge 4.33. SOA algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi……. 82
Çizelge 4.34. SOA algoritmasındaki arama aracı sayısının Sistem-4 PID denetleyici tasarımına etkisi…… 83
Çizelge 4.35. SSA algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi……. 84
Çizelge 4.36. SSA algoritmasındaki arama aracı sayısının Sistem-4 PID denetleyici tasarımına etkisi…… 85
Çizelge 4.37. Sistem-5 için algoritma parametre ayarlaması…… 86
Çizelge 4.38. Sistem-5 için özel fonksiyon tasarım kriterine göre karşılaştırmalı sonuçlar……... 87
Çizelge 4.39. Sistem-5 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar……... 87
Çizelge 4.40. Sistem-5 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar 88
Çizelge 4.41. Sistem-5 için ITSE tasarım kriterine göre karşılaştırmalı Sonuçlar…….. 89
Çizelge 4.42. Sistem-5 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar.. 90
Çizelge 4.43. ITAE Kriteri Uygunluk Fonksiyonu Değerleri……... 91

1. GİRİŞ

Mühendislik, ekolojik bir dengeyi ve üzerinde yaşanacak güvenli bir gezegeni korurken, doğanın güçlerini anlamak ve insanlığın yararına kullanmakla ilgilenir. Kontrol mühendisliği ise ilgili sistemi anlamak ve istenen çıktı cevabını sistemdeki engellemelere rağmen bulmaya çalışmaktadır.

Kontrol; ayarlama, düzenleme, yönetme ve kumanda etme anlamına gelir. Kontrol sistemi ise bir sistemi veya kendisini yönetme (kumanda etme) için kurulan sisteme denir ve kontrol edilen sistemlerde birden fazla giriş veya çıkış olabilir. Kontrol edilebilirlik ise belirli bir sistemi kabul edilen sınırlar içinde kararlı/istikrarlı/stabil halde tutmaya denir (Kiam ve ark., 2005).

En sık kullanılan kontrol yöntemleri Oransal-İntegral-Türev (PID) denetleyici, kutup yerleştirme, servo kontrol, gözetleyici kontrol ve amaç fonksiyonu kontrolüdür. Sistemleri kararlı hale getirmek için kullanılan PID denetleyici; bir süreç içerisinde istenilen nokta ile değişen değer arasındaki farkı bir hata değeri olarak hesaplar ve bu hata değerini en aza düşürmeye çalışır. PID kontrol yöntemleri P, PI, PD ve PID olarak kullanılır (Zhong ve ark., 2007).

Uygulamalarda sistemlerin kontrolü son derece önemlidir. Bu amaçla uygun denetleyicilerin tasarlanması gerekmektedir. En popüler denetleyicilerin başında PID'ler gelmektedir ve bunların tasarımı için geleneksel yöntemler mevcuttur. PID denetleyiciler endüstride yaygın olarak kullanılmaktadır ve bunlar endüstriyel kontrol sistemlerinin önemli bir bölümünü oluşturmaktadır. Bu nedenle, PID tasarımı ve uygulama metodolojisindeki herhangi bir gelişme, endüstriyel kontrol sistemleri için önemlidir. Bu durum kendinden ayarlamalı PID kontrolünü konvansiyonel kontrollere önemli bir alternatif haline getirmektedir.

Klasik kontrol yöntemleri; sisteme dahil edilmek suretiyle uygulanabilir, kararlı kontrol sistemleri tasarlanabilir. Parametrik çıktı ve zamansal belirsizlikler sistemi kararsızlığa doğru götürür. Sistem karmaşıklığı ile zaman gecikmesi olan sistemlerde kararlı bir cevaba ulaşabilmek için çeşitli tasarım yöntemleri üzerine araştırmalar yapılmıştır (Wu ve ark., 2018).

Keel ve arkadaşları, günümüz kontrol tasarımında temel sistem parametre modellenmesi için yüksek mertebeden kontrol fonksiyonları tanımlamışlardır. Bant genişliği, kazanç ve faz marjları gibi klasik özellikler için performans yöntemleri grafiksel olarak kolayca belirlenebilmektedir. Bunlar PID kontrol tasarımı da olarak ifade edilmektedir (Keel ve ark., 2008).

Son 40 yıl içerisinde PID denetleyicilerde parametre ayarlamasında kullanılmak için birçok yöntem geliştirilmiştir. Bunlar zaman gecikmeli olan birinci derece sistemin dinamik tepkisini karakterize etmek için kullanılır. Klasik sistem modellemesinde birim basamak girişi uygulanarak zaman gecikmesi, kararlı hal hatası ve zaman sabiti olmak üzere 3 parametreye bakılır (Singh & Garg, 2014) .

Son yıllarda PID katsayılarının ayarlanması için metasezgisel algoritmalardan da sıklıkla faydalanılmaktadır. Bu alanda yapay zeka (AI) uygulamanın bir yolu olarak metasezgisel algoritmaların kullanılmasının bu eksikliklerin üstesinden gelmede oldukça etkili olduğu kanıtlanmıştır. Belirli bir problemi çözmek için metasezgisel algoritmayı seçmenin birkaç nedeni vardır. Bunlar;

- Karmaşık problemlerde daha kolay ve hızlı cevap oluşturmaları,
- Çözüm bulmak için optimizasyon yöntemleri uygulanırken keşif sırasında belirli bir alan içerisinde takılmadan sonuca ulaşmaya çalışmaları şeklinde sıralanabilir (Rahman ve ark., 2017).

Yapılan çalışmalarda metasezgisel algoritmalar; popülasyon tabanlı, evrimsel algoritmalar, tek çözüm tabanlı, doğadan esinlenilen, fizikten ilham alan vb. olmak üzere çeşitli gruplara ayrılmıştır.

Ren ve arkadaşları, PID kontrol ile iki tekerlekli aracın kararlılık ve hareket kontrolü üzerine sonuçlar elde etmişlerdir. İki bağımsız tekerleği birbirine bağlayarak motor ile hareketlendirip dinamik analiz yöntemi ile matematiksel bir model oluşturmuşlardır. Çıkarımı yapılan bir sonuca dayalı olarak otomatik ayarlanan PID kontrol stratejisi model, iki tekerlekli araç (TWV)'yi kararlı hale getiren ve istenen hareket komutlarını takip eden bir hareket kontrol sisteminin uygulanması için önerilmiştir (Ren ve ark., 2008).

Ghosal ve arkadaşları, Karınca Kolonisi Optimizasyonu (ACO), Bakteri Toplayıcılık Optimizasyonu (BFO) algoritması ve Parçacık Sürü Optimizasyonu (PSO) algoritmalarını kullanarak farklı tiplerde PID parametrelerinin optimizasyonu üzerine inceleme yapmışlardır (Ghosal ve ark., 2012).

Loucif ve arkadaşları, robot kontrolü için doğrusal olmayan sistemlerde Balina Optimizasyon Algoritmasını (WOA) kullanarak Simulink/MATLAB geliştirme ortamında PID denetleyici için çalışmalar yapmışlardır ve elde ettikleri simülasyon sonuçları arasında performans kriterlerine göre karşılaştırmalar yaparak en iyi sonucu elde etmeye çalışmışlardır (Loucif ve ark., 2020).

Roni ve arkadaşları tarafından metasezgisel algoritmaların çeşitlerinden olan popülasyon tabanlı ve biyolojik olayları örnek alan optimizasyon yöntemlerinden oluşan bir çalışma yapılmıştır. Biyolojik veya doğa olaylarını taklit eden bu yöntemler, gerçek dünya problemlerini çözmek için çeşitli kontrol çalışmalarında kullanılan optimizasyon yöntemlerindeki son eğilimler ve uygulamalar üzerine bir inceleme sunulmuştur (Roni ve ark., 2022).

Sundari ve arkadaşları tarafından çok seviyeli tank sistemine örnek olarak iki dalgalanma etkileşimli ve etkileşimli olmayan tank sistemi bozulma etkisinin, PID denetleyicisinin metasezgisel algoritma kullanılarak araştırma incelemesi yapılmıştır (Sundari ve ark., 2022).

Shehab ve arkadaşları, Güve Alevi Optimizasyon (MFO) algoritmasının gelecekte tasarlanabilecek olan güç ve enerji sistemi, kontrol yöntemleri, tıbbi görüntüleme uygulamalarındaki birçok problemde kullanabileceği ile ilgili bir çalışma sunmuşlardır (Shehab ve ark., 2020).

Jafari ve arkadaşları, gelecekte havacılık mühendisliği problemlerine uygulamak üzerine araştırma zorlukları için potansiyel çözümler, gelişmiş motor modelleme ve denetleyici için en uygun sonuçları elde etmek amacıyla geliştirilmiş ve değiştirilmiş optimizasyon algoritmalarının uygulanmasını içeren bir araştırma sonucu bulmuşlardır (Jafari ve ark., 2019).

Sheta ve arkadaşları, Parçacık Sürü Optimizasyonu (PSO), Genetik Algoritma (GA) ve Karga Arama Algoritmasını(CSA) kullanarak dört tekerli helikopter hareketlerini kararlı hale getirmek için PID kontrol sistemi tasarlamışlardır (Sheta ve ark., 2021).

Souza ve arkadaşları, üç fazlı bir asenkron motor tarafından kontrol edilen robotik kontrolcü hız kontrolörünü darbe genişlik modülasyonu ile kontrol eden PID denetleyici tasarımı çalışmalarını anlatmışlardır (Souza ve ark., 2021).

Dineva ve arkadaşları, hareketli elektrikli makinelerin tasarımı ve kontrolünde yumuşak sürüş yöntemleri için modelleme üzerine bir inceleme yapmışlardır. İnceleme kapsamında GA, PSO ve Fuzzy Logic olmak üzere üç algoritma için literatür çalışması yapılmış ve son gelişmelerin aktarılması amaçlanmıştır (Dineva ve ark., 2019).

Sahu ve arkadaşları, önerilen Karınca Aslanı Optimizasyon Algoritması (ALO) ile PID denetleyici tabanlı maksimum güç noktası takibi (MPPT) tekniğinde, güneş sisteminin salınım, zaman tepkisi, yerleşme süresi ve maksimum gerilim, akım ve güç değerlerini geleneksel P&O tekniğine göre daha iyi sonuçlar ürettiği göstermişlerdir (Sahu ve ark., 2018).

Sharma ve arkadaşları, bulanık mantık tabanlı denetleyici ile BLDC motorunu kontrol etmek için 2000-2018 yılları arasında yapılan klasik PID denetleyici ile metasezgisel algoritmaları kullanarak yapılan denetleyicileri karşılaştırmışlardır. Çalışmada çeşitli PID denetleyici biçimlerinin yanı sıra farklı DC motor türlerinin uygulamaları incelenmiştir (Oladipo ve ark., 2020).

Çizelge 1.1'de PID kontrol ile ilgili yapılan literatür tarama sonuçları tabloda gösterilmiştir.

Çizelge 1.1. Kapalı çevrim PID kontrolü için literatür taraması

Yazar	Yıl	Yapılan Çalışma
Koivo ve ark.	1981	Mikroişlemci tabanlı kapalı çevrim kontrol sistemi ile Ortalama arter kan basıncının (MABP) istenen aralıkta tutulmuştur.
Kaufman ve ark.	1984	Model referansı uyarlamalı kontrol, simülasyon sonuçlarını gerçek zamanlı verilerle eşleştirmek için tasarlanmıştır.

Çizelge 1.1. Kapalı çevrim PID kontrolü için literatür taraması (devam)

Behbehani ve ark.	1991	Entegre bir tür kendi kendini ayarlayan kontrol stratejisi ile Ortalama arter kan basıncının düzenlenmesi için bir kontrol cihazı tasarlanmıştır.
Payakkawan, ve ark.	2009	PSO tabanlı PID denetleyicisi kullanılarak DC motor hız kontrol regülasyonu yapılmıştır.
Chang ve ark.	2011	Evrimsel programlama algoritması (EP) ile sabit mıknatıslı DC motorun konumunu kontrol eden optimum bir PID denetleyici tasarımı yapılmıştır.
Ansari ve ark.	2011	GA algoritmasını kullanarak PID tabanlı üç fazlı BLDC motorun modellenmesi, kontrolü ve geleneksel yöntemlerle karşılaştırmaları yapılmıştır.
Zhu ve ark.	2012	PMDC motor kontrol sistemi için AFSA algoritması ile çeşitli PID denetleyici tasarımlarının kontrol karşılaştırmaları yapılmıştır.
Gandomi ve ark.	2013	Guguk kuşu(Cuckoo) arama algoritması ile yapısal optimizasyon problemlerinin çözülmesi ve diğer algoritmalarla Guguk kuşu arama algoritmasının karşılaştırmaları yapılmıştır.
Wagner ve Neumann	2013	Hızlı yaklaşım güdümlü evrimsel algoritması (AGE-II)'nin çok amaçlı problemlere uygulandığındaki performans sonuçlarını göstermektedir.
Ibrahim ve Mahmoud	2014	Karınca algoritması ile DC motor kontrolü ve geleneksel PID kontrol sonuçları arasında karşılaştırma yapılmıştır.
Mishra ve ark.	2013	ABC algoritması kullanarak PID parametrelerinin seçimi ile bir DC motorun hız kontrolörü tasarlanmıştır.
Sondhi ve Hote	2015	Kazanç ve faz marjı yöntemine dayalı kesirli dereceli oransal integral (FOPI) denetleyicisi tasarlanmıştır.
Ahmed ve Ozbay	2016	Anahtarlama teorisine dayalı denetleyici tasarımı yapılmıştır.
Jain ve ark.	2017	PSO tabanlı FOPID kullanılarak DC motorun hız kontrol tasarımı yapılmıştır. PID kontrol ile FOPID kontrol kıyaslaması yapılmıştır.
Prommee ve Angkeaw	2017	Bipolar transistörle P,I,D ve PID kontrol tasarımları ve simülasyonları yapılmıştır.
Prabu ve ark.	2017	GA algoritması ile BLDC motorun rotor kontrolü için IAE, MSE performans kriterlerine göre PID denetleyici tasarımı yapılmıştır.
Kok ve ark.	2017	Benzetilmiş Tavlama(SA), Diferansiyel Evrim(DE) algoritmaları ile tasarlanan PID denetleyicisinin, Ziegler-Nichols (ZN) PID denetleyici tasarımı ile karşılaştırmaları yapılmıştır.
Molina ve ark.	2018	DE tabanlı bir DC motorun hız kontrolü için PI tabanlı bir uyarlamalı kontrol tasarımı yapılmıştır.
Su ve ark.	2018	PSO tabanlı Model tahmine dayalı kontrol (MPC) ile klasik PID kontrol simülasyon sonuçları karşılaştırılmıştır.

Çizelge 1.1. Kapalı çevrim PID kontrolü için literatür taraması (devam)

Srivastava ve ark.	2018	GA tabanlı PID kontrol tasarımı yapılan bu çalışmada performans kriterlerine göre karşılaştırmalar yapılmıştır. Elde edilen sonuçlara göre tasarımın kullanabilirliği gösterilmiştir.
Klempka ve Filipowicz	2018	FFA, FA, GA, PSO algoritmaları ile fırçasız doğru akım elektrik motoru (BLDC)'nin PI denetleyici parametrelerinin IAE, ISE, ITAE ve ITSE performans kriterlerine göre sonuçları bulunmuştur.
Silva ve ark.	2019	ARM-M3 mikrodenetleyicisi kullanılarak PI denetleyici kontrol tasarımı ile Matlab ortamındaki PI denetleyici tasarımı karşılaştırmaları yapılmıştır.
Quresh ve ark.	2019	Doğrusal Kuadratik Düzenleyiciyi (LQR) ayarlamak için Genelleştirilmiş diferansiyel evrim algoritması (GDE3) kullanılarak PID parametrelerinin bulunmasındaki performans yöntemleri karşılaştırılmıştır.
Altinoz, O.	2019	NSGA- II, AGE- II algoritmaları PID kontrol parametrelerinin performans kriterlerine göre karşılaştırılmalar yapılmıştır.
Susperregui ve ark	2019	Çift beslemeli bir endüksiyon jeneratörünün şebeke tarafı dönüştürücüsüne (GSC) komuta eden ikinci dereceden bir kayan modlu kontrol (2-SMC) şemasının parametrelerini ayarlamak için MOO algoritması kullanılmıştır.
Shatnawi ve Bayoumi	2019	SA algoritması kullanılarak sabit mıknatıslı fırçasız DC motor (PMBLDCM) için PI/PID kontrol parametreleri hesaplanmıştır. Geleneksel kontrol yöntemleri ile karşılaştırmaları yapılmıştır.
Mohanty ve ark.	2020	PV- entegre HAF'ın performansı, oransal-integral (PI) ve bulanık mantık denetleyicisi (FLC) ile senkron kullanılarak analiz edilip önerilen Güç sistemleri sinyal frekans tahmini(RECKF) tekniği ile karşılaştırmaları yapılmıştır.
Sharma ve ark.	2020	Guguk kuşu arama algoritması (CSA) kullanılarak Bulanık Mantık denetleyicisi ile geleneksel PID denetleyici tasarımı karşılaştırmaları yapılmıştır.
Hernandez-Barragan ve ark.	2020	Uyarlanabilir güncelleme kurallarına sahip veri güdümlü PID denetleyici tasarımı yapılmıştır.

Bu tez çalışmasında, metasezgisel algoritmalardan olan Aritmetik optimizasyon algoritması(AOA), Dinamik diferansiyel tavlı optimizasyon algoritması(DDAO), Aquila (Kartal) optimizasyon algoritması(AO), Kel Kartal optimizasyon algoritması(BES), Martı optimizasyon algoritması(SOA), Serçe arama algoritması(SSA) kullanılarak - farklı kriterler altında - PID parametrelerin optimizasyonu amaçlanmıştır.

Tasarlanan PID denetleyicili sistemlerde ilgili güncel algoritmalar kullanılarak en uygun sistem cevabının elde edilmesi hedeflenmektedir.

Birinci bölümde, konunun genel bir tanımı yapılarak bu çalışmanın önemi, amacı ve yapılmış çalışmalara yer verilmiştir. İkinci bölümde, denetim sisteminin temel öğeleri, denetim sistemi tasarımı, PID denetleyiciler ve sistem tasarımında kullanılan klasik metotlardan bazıları tanıtılmıştır. Üçüncü bölümde, güncel metasezgisel algoritmalar ile ilgili tanımlar yapılmış, kullanılan algoritmalar ayrıntılı olarak açıklanmıştır. Dördüncü bölümde, algoritmalar kullanılarak PID denetleyici parametreleri K_p, K_i, K_d değerleri, yükselme süresi, yerleşme süresi, maksimum aşım ve kararlı hal hatası değerlerinin tespiti yapılmıştır. Beşinci ve son bölümde ise algoritmalardan elde edilen sonuçlar kıyaslanarak tasarlanan sistemler için en uygun sonuçlar belirlenmiştir. Tez çalışmasının kısa bir değerlendirilmesi yapılarak elde edilen sonuçlar yorumlanmıştır.

2. KAYNAK ARAŞTIRMASI

2.1. Sistem Tanımlama

Sistemin kontrol edilebilmesi için öncelikle sistem durumunun tanımlanması gerekir. Sistem tanımlamadaki amaç, sistemin davranışının ve dış etkilerinin ölçümlerinden sistem içerisindeki olaylara matematiksel bir ilişki belirlemeye çalışmaktır. Sisteme iletilen sinyaller, belirli işlemler sonucunda sistemden çıkar. Bu sinyaller, sistemin davranışını derinlemesine anlamak için zaman alanı veya frekans alanı şeklinde temsil edilir. Transfer fonksiyonu, sistemin tüm dinamiklerini yakalayabilen her girdi ile sistem çıktısını modelleyen fonksiyondur. Transfer fonksiyonu Denklem 2.1'deki gibi ifade edilir:

$$T(s) = \frac{Y(s)}{X(s)} \tag{2.1}$$

Burada $X(s)$ ve $Y(s)$ sistem giriş ile çıkışının Laplace dönüşümlerini göstermektedir. Kontrol teorisinde iki farklı kontrol tekniği vardır:

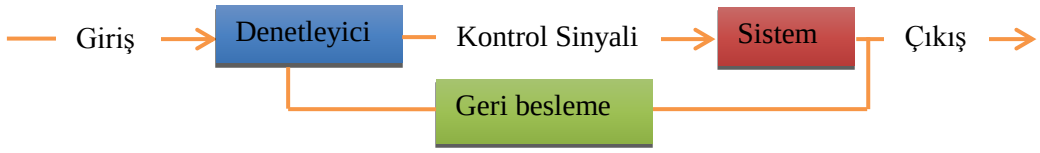
a) Açık çevrim kontrol sistemi: Kontrol eden düzeneğin sistemin çıkışından etkilenmediği, verilen referans değerine göre denetim işlemi yapılan sistemlerdir (Şekil 2.1).

b) Kapalı çevrim kontrol sistemi: Sistemdeki giriş işareti, çıkış işaretine ya da çıkışla orantılı bir işaretle referans arasındaki farka bağlı kontrol sistemidir (Şekil 2.2).

2.2. Kontrol Tasarımı için Gereklilikler

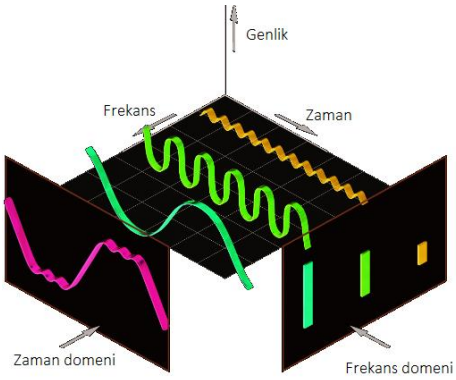
Sinyallerin analiz edildiği iki alan vardır: zaman ve frekans alanı. En büyük aşım, yükselme süresi ve yerleşme süresi gibi kriterler zaman tanım bölge tasarımlarında kullanılırlar. Frekans tanım bölgesi tasarımında ise rezonans tepesi, band genişliği ve göre kararlılık kullanılır.

Şekil 2.1. Açık çevrim kontrol sistemi



Şekil 2.2. Kapalı çevrim kontrol sistemi

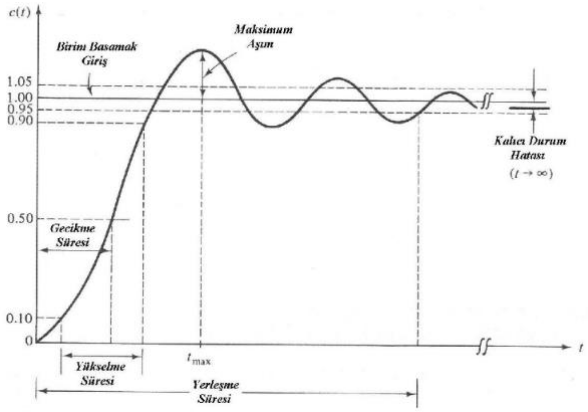
Zaman alanı yalnızca sinyalin zamanla değişen grafiğini gösterir, ancak frekans alanı daha ayrıntıları gösterebilir ve her frekans bandındaki sinyalin bant genişliğini analiz edebilir. Sistemlerde birçok bozucu etki, gerçek sinyalle birleşmektedir. Zaman alanında, yalnızca sinyaldeki salınımlar görülebilir, ancak bu salınımların nereden geldiği anlaşılamaz. Frekans alanında, Şekil 2.3'te gösterildiği gibi, hangi harici sinyallerin orijinal sinyalle karıştığını bulmak için sinyallerin tepe değerlerini temsil eden çubuk grafik elde edilecektir.



Şekil 2.3. Zaman ve frekans bölgesi (Diethorn 1994'ten değiştirilerek alınmıştır)

2.3. Basamak Cevabının Özellikleri

Bir kontrol sisteminde geçici hal cevabının genliği ve süresi belirlenen değerin altında tutulmaya çalışılır ve kontrol sisteminin birim basamak girişine verdiği cevap, birim basamak cevabı olarak adlandırılır. Basamak cevabı, sistemin kararlılığı hakkında bilgi verir. Şekil 2.4'te gösterilen doğrusal bir kontrol sisteminin birim basamak cevabı görülmektedir (Ogata, 2002).



Şekil 2.4. Birim basamak cevabı (Zumbahlen 2006'dan değiştirilerek alınmıştır)

Sisteme ait birim basamak cevabı üzerinde bazı önemli paramatreler vardır:

i. Yükselme süresi: Sistemin istenen durumun %90'ına ulaşması için geçen süredir.

$$t_r \cong \frac{0.8+2.5\zeta}{\omega_n}, \quad 0 < \zeta < 1.0 \tag{2.2}$$

ii. Yerleşme süresi: Sistemin herhangi bir salınım olmaksızın kararlı hale gelmesi için geçen süredir.

$$t_s \cong \frac{3.2}{\zeta\omega_n}, \quad 0 < \zeta < 0.69 \qquad t_s \cong \frac{4.5\zeta}{\omega_n}, \quad \zeta > 0.69 \tag{2.3}$$

iii. Maksimum aşım: Sistem cevabının maksimum değeriyle olması gereken değeri arasındaki farktır.

$$Y_{max} - 1 = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{2.4}$$

iv. Gecikme süresi: Gecikme süresi t_d basamak cevabının son değerinin %50 değerine ulaşma süresi olarak tanımlanır.

$$t_d \cong \frac{1+0.7\zeta}{\omega_n}, \quad 0 < \zeta < 1.0 \tag{2.5}$$

2.4. PID Denetleyici Yapısı

PID denetleyici yapısı dört temel şekilde kullanılmaktadır: P oransal, PI oransal + integral, PD oransal + türev denetleyici ve PID oransal + integral + türev denetleyicidir. Kontrol sistemlernde kullanılan bu temel denetleyiciler, ayrıntılı bir şekilde açıklanacaktır.

PID denetleyici (oransal+integral+türev) olarak bilinen ve endüstri alanında kullanılan kontrol sistemlerini geri besleme ile sistemi kararlı halde tutmaya çalışan mekanizmadır. Denetlemeyi yapan PID denetleyicisi sistemi belirli bir süre içerisinde istenilen değere oturtmayı hedeflemektedir.

PID denetleyici değişkenleri kullanarak süreye göre hatayı minimize etmeye çalışır ve sistemi kararlı hale getirir. Bu denetleyiciler üç tane ayrı parametreye sahiptir: P oran, I integral ve D türevdir.

P oransal olan mevcut hatayı, I integral olan geçmişten gelen toplam hatayı ve D ise sistemin mevcut salınımını belirtmektedir. Bu üç eylem bir sistemin istenilen değerde sabit tutmak için kullanılabilir.

PID kontrol uzun yıllardan beri kullanılan en yaygın kontrol yöntemlerinden biridir. PID kontrol üç parametreyi ayarlayarak tasarlanan sistemin kontrolünü sağlayabilir. Bu parametrelerin uygun şartlarda sistemi kontrol etmesi için belirlenmesi gerekir. Bunu belirlemek kolay değildir ve net bir cevabı yoktur. En uygun sonuç için en az hata, sistemde minimum aşım, kısa sürede hatayı giderme ve sistemde kararlılığı sağlama gibi kriterleri yerine getirecek şekilde ayarlanmalıdır.

Bazı uygulamalar, uygun sistem kontrolünü sağlamak için yalnızca bir veya iki eylemin kullanılmasını gerektirebilir. Bu, diğer parametreleri sıfıra ayarlayarak elde edilir. Böyle sistemlerde sadece P, PI, PD veya PID denetleyicilerinden en uygun sonucu hangisi ile elde ediliyorsa sistem parametreleri ona göre ayarlanmalıdır.

2.4.1. P Denetleyici

Oransal denetleyicideki çıkış işareti, girişindeki işarete sabit bir oran ile bağlıdır. Kontrol işareti kontrol hatasıyla orantılıdır. Bu geri beslemenin en basit şeklidir. K_c oransal kazanç ve $e(t)$ geri besleme hatası olup

$$u(t) = K_c e(t) \tag{2.6}$$

dir. Burada,

$$G_c(s) = K_c \tag{2.7}$$

$e(t)$, referans sinyali $r(t)$ ve çıkış sinyali $y(t)$ arasındaki farktır.

$$e(t) = r(t) - y(t) \tag{2.8}$$

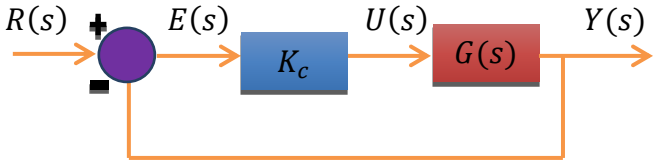
Kapalı çevrim geri besleme kontrol konfigürasyonunun blok şeması Şekil 2.5'te gösterilmiştir. Burada $R(s)$ referans sinyalinin, $E(s)$ ile geri besleme hatasının, $U(s)$ kontrol sinyalinin ve $Y(s)$ ile çıkış sinyalinin Laplace dönüşümlerini göstermektedir.

P oransal denetleyici, sistemi kararlı hale getirmek için kullanılan en temel kontrol yöntemidir. Oransal kontrol kazancı K_c hatadaki her birim değişikliğe denetleyici çıkışında kaç birimlik bir değişim olacağını belirler. Ayrık zamanlı sistemler için kontrol girişi şu şekilde tanımlanır:

$$u(k) = K_c e(k) \tag{2.9}$$

Burada oransal denetleyici,

$$G_c(z) = K_c \tag{2.10}$$



Şekil 2.5. Oransal geri besleme kontrol sistemi

2.4.2. PD Denetleyici

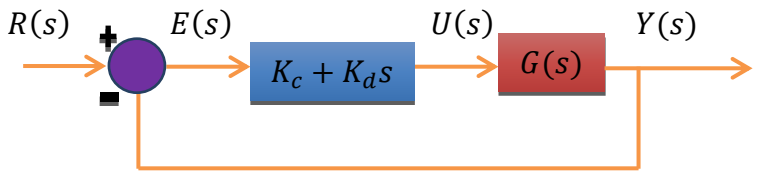
Çoğu sistemde P oransal denetleyici sistemi kararlı hale getirmeye yetmez. Böyle durumlarda sistem sönümünü azaltmak için oransal + türev kontrol yöntemi kullanılır. PD denetleyicisinin transfer fonksiyonunun matematiksel modeli Denklem 2.11'de gösterilmiştir.

$$u(t) = K_c e(t) + K_c \tau_D e(t) \tag{2.11}$$

$e(t) = r(t) - y(t)$'nin $r(t)$ referans sinyali ile $y(t)$ çıkışı arasındaki hata sinyalidir. K_c oransal kazanç ve τ_D türev sabitidir. Denetleyici çıkışının Laplace dönüşümü

$$U(s) = K_c E(s) + K_c \tau_D E(s) \tag{2.12}$$

olup $E(s), e(t)$ hata sinyalinin Laplace dönüşümüdür. Bununla,



Şekil 2.6. PD geri besleme kontrol sistemi($K_d = K_c \tau_D$)

PD denetleyicisi için Laplace transfer fonksiyonu Denklem 2.13'teki gibi ifade edilir:

$$\frac{U(s)}{E(s)} = K_c + K_c \tau_D s \tag{2.13}$$

Bir PD denetleyici için kapalı döngü geri besleme kontrolü Şekil 2.6'da gösterilmiştir. Ayrık zamanlı oransal + türev denetleyici Denklem 2.14'te gösterilmiştir.

$$G_c(z) = K_c T_d \left(\frac{1-z^{-1}}{T}\right) = \frac{K_c T_d}{T}\left(\frac{z-1}{z}\right) \tag{2.14}$$

2.4.3. PI Denetleyici

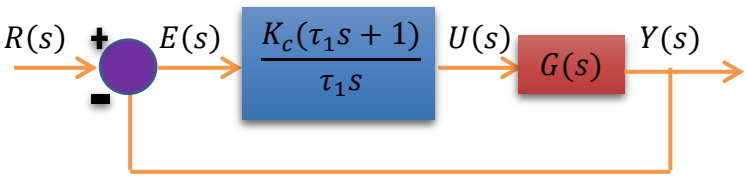
PI, oransal+integral kontrole denir. Bu kombinasyon, ofseti ortadan kaldırarak kararlılığı sağlanmaktadır. Bundan dolayı sıvı işleme endüstrilerinde kullanılan en yaygın denetleyicilerdendir. Diğer denetleyicilerin daha uygun olduğu uygulamalarda bile neredeyse evrensel olarak kullanılır. Sürekli zamanda bir PI denetleyicisinin transfer fonksiyonunun matematiksel modeli gösterilmiştir.

$$u(t) = K_c e(t) + \frac{K_c}{\tau_i}\int_0^t e(t)dt \tag{2.15}$$

$e(t) = r(t) - y(t)$'nin $r(t)$ referans sinyali ile $y(t)$ çıkışı arasındaki hata sinyalidir. K_c oransal kazanç ve T_i integral zaman sabitidir. τ_i parametresi her zaman pozitiftir ve değeri PI denetleyicisi tarafından gerçekleştirilen integral eylemin etkisiyle ters orantılıdır. Denetleyici çıkışının Laplace dönüşümü

$$u(s) = K_c E(s) + \frac{K_c}{\tau_I s} E(s) \tag{2.16}$$

olup $E(s), e(t)$ hata sinyalinin Laplace dönüşümüdür. Bununla,



Şekil 2.7. PI kontrol sistemi

PI denetleyicisi için Laplace transfer fonksiyonu şu şekilde ifade edilir:

$$C(s) = \frac{U(s)}{E(s)} = \frac{K_c(\tau_I s+1)}{\tau_I s} \tag{2.17}$$

Şekil 2.7 PI kontrol sisteminin bir blok şemasını göstermektedir. PI denetleyicinin ayrık zamanlı gösterimleri ise aşağıda verilmektedir:

$$u(k) = K_c\left(e(k) + \frac{T}{T_i}\sum_{i=0}^{k-1} e(i)\right) \tag{2.18}$$

$$G_c(z) = K_c\left(1 + \frac{T}{T_i}\left(\frac{z}{z-1}\right)\right) \tag{2.19}$$

2.4.4. PID Denetleyici

Bir PID denetleyicisi üç kısımdan oluşur ve ideal olarak çıkışı $u(t)$, bu üçünün toplamıdır:

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(\tau) d\tau + K_c \tau_D \frac{de(t)}{dt} \tag{2.20}$$

Burada $e(t) = r(t) - y(t)$, $r(t)$ referans sinyali ile $y(t)$ çıkışı arasındaki geri besleme hata sinyalidir ve τ_D türev kontrol kazancıdır. PID denetleyicilerinin transfer fonksiyonu şu şekildedir:

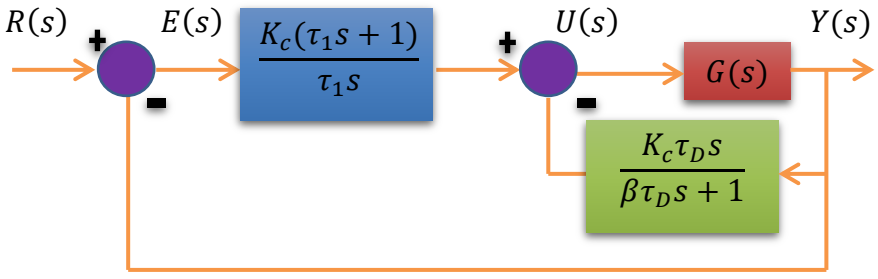
$$\frac{U(s)}{E(s)} = K_c \left(1 + \frac{1}{\tau_{IS}} + \tau_D s\right) \tag{2.21}$$

Tasarım uygun ise τ_D'nin işareti pozitiftir. τ_D'nin işareti negatif ise, türev kontrol terimi ihmal edilir ve bunun yerine bir PI denetleyicisi seçilmelidir.

Bölüm 2.4.2'de belirtilen oransal+türev denetleyiciye benzer şekilde, uygulamaların çoğu için türev denetimi çıkışta yalnızca bir türev filtresiyle uygulanır. Bu nedenle, $U(s)$ kontrol sinyali Denklem 2.22'de ifade edilir:

$$U(s) = K_c \left(1 + \frac{1}{\tau_{IS}}\right) (R(s) - Y(s)) - \frac{K_c \tau_D s}{\beta \tau_D s + 1} Y(s) \tag{2.22}$$

Şekil 2.8'de PID denetleyici yapısının blok şemasını göstermektedir. Çıkış cevabındaki aşım miktarını azaltmak için birim basamak referans değişikliği yapılır.



Şekil 2.8. PID denetleyici tasarımı

Ayrık zamanlı sistemler durumunda, PID kontrol girişi ve denetleyici, fark denklemi ile tanımlanabilir:

$$u(k) = K_c \left(e(k) + \frac{T}{T_i} \sum_{i=0}^{k-1} e(i) + \frac{T_d}{T} [e(k) - e(k-1)] \right) \quad (2.23)$$

$$G_c(z) = K_c \left[1 + \frac{T}{T_i} \left(\frac{z}{z-1} \right) + \frac{T_d}{T} \left(\frac{z-1}{z} \right) \right] \quad (2.24)$$

Kapalı çevrim sisteminde her bir denetleyicinin etkisi K_p, K_i ve K_d kazançları verilerek Çizelge 2.1'de özetlenmiştir (Vatansever & Hacıiskenderoğlu, 2022) .

Çizelge 2.1. Bağımsız olarak artan parametrenin etkisi

Denetleyici Türü	**Kapalı Çevrim Cevabı**	**Yükselme Zamanı**	**Aşım Değeri**	**Yerleşim Zamanı**	**Kararlı Hal Hatası**
Oransal	Kp	Azalır	Artar	Küçük Oranda Artar	Azalır
İntegral	Ki	Küçük Oranda Azalır	Artar	Artar	Büyük Oranda Azalır
Türev	Kd	Küçük Oranda Azalır	Azalır	Azalır	Önemsiz Değişim

2.5. PID Denetleyici Parametre Belirleme

Metasezgisel optimizasyon algoritmaları belirlenen optimizasyon problemini minimize etmek için gereken parametreleri bulurlar. Bu parametrelerin optimizasyonu için bir fonksiyon tanımlamak gerekir. Tanımlanan bu fonksiyona "amaç fonksiyonu" denir. Yaygın olarak kullanılan dört tür amaç fonksiyonu vardır (Campo, 2012).

- Mutlak Hatanın Toplamı

$$IAE = \int_0^\infty |e(t)|\, dt \quad (2.25)$$

- Hata Karelerinin Toplamı

$$ISE = \int_0^\infty e(t)^2 dt \quad (2.26)$$

- Zaman Ağırlıklı Hata Karelerinin Toplamı

$$ITAE=\int_0^\infty te(t)^2dt \tag{2.27}$$

- Zaman Ağırlıklı Mutlak Hatanın Toplamı

$$ITSE=\int_0^\infty t|e(t)|\,dt \tag{2.28}$$

Bu tez çalışmada ayrıca (2.25-2.28) denkleminde tanımlanan uygunluk fonksiyonuna ek olarak bir zaman alanına dayalı özel bir uygunluk fonksiyonu da kullanılmıştır. Bu uygunluk fonksiyonu, yükselme süresi, yerleşme süresi ve sabit durum hatası aşmasından oluşur (Lidbe ve ark., 2017). Uygunluk fonksiyonu Denklem 2.29'daki gibi tanımlanır:

$$min_{K(STABİL)}W(K) = \left(1 - e^{-\beta}\right).(M_P + E_{SS}) + e^{-\beta}.(t_S - t_R) \tag{2.29}$$

Burada M_P, maksimum aşımı; E_{SS}, kararlı hal hatasını; t_S, yerleşme süresini; t_R de yükselme süresini göstermektedir. Ağırlık faktörü β değiştirilerek $W(K)$'nin farklı sonuçları elde edilir. β 0.7'den büyük durumlarda, maksimum aşım ve kararlı hal hatası hataları azaltılırken β 0.7'den küçük durumlarda ise, yükselme süresi ve yerleşme süresi azaltılmaktadır. Tez çalışmasındaki simülasyonlarda, kriterleri eşit olarak ağırlıklandırmak için β değeri 0.7 olarak seçilmiştir (Gaing, 2004).

2.6. PID Denetleyicileri için Ayar Kuralları

Bir PID kontrol sistemi ayar noktası değişikliği ve yük bozulmaları ile uğraşmak zorundadır. Bu durumda iyi bir denge elde etmek için ayarlama yöntemine sahip olması arzu edilir. Denetleyici ayarlarında; parametreleri bulmak amacıyla, tasarımcıya rekabetçi hedeflerin analizinde yardımcı olacak şekilde bir optimizasyon tasarım prosedürü kullanılmaktadır.

PID denetleyicilerindeki araştırmalar her zaman özellikle ayar konusuna, yani belirli bir uygulama için en uygun PID parametrelerinin seçimine odaklanmıştır.

Özellikle, akort kurallarının geliştirilmesi, kullanıcının basit bir süreç modelinden başlayarak denetleyici kazanımlarını belirlemesi, Ziegler ve Nichols tarafından yapılan ilk öneriden beri büyük ilgi gören bir konudur.

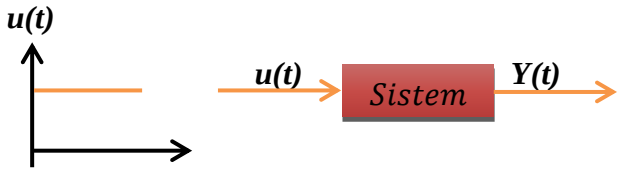
Geliştirilen farklı kontrol denetleyici yapılarında (örneğin, etkileşimli veya etkileşimsiz formda PID olabilir), farklı süreç modelleri (örneğin, birinci veya ikinci dereceden ölü zaman transfer fonksiyonu), yapmaları gereken farklı kontrol görevleri ile ilgili adres (örneğin, ayar noktası takibi veya yük bozulma reddi) ve bunların dayandığı farklı yaklaşımlardır (örneğin, deneysel, analitik veya optimal). Çıkış performansı açısından, yük bozulmalarının ve ayar noktası değişikliklerinin geri besleme kontrol sistemi üzerindeki etkilerini göz önünde bulundurarak bir tasarım girdi-çıktısı belirleyebiliriz (Basilio ve ark., 2002).

2.7. Ziegler-Nichols Metodu

Sistemi kararlı hale getirmek için açık çevrim cevabını kullanarak farklı yöntemler bulunmaktadır. Bu yöntemde PID katsayıları basamak veya frekans cevabına göre hesaplanabilmektedir (Ziegler & Nichols, 1942). Bu katsayıların hesaplanması Çizelge 2.2'de verilmektedir. Ziegler-Nichols Denklem 2.30'da verilmiştir.

$$G_P(s) = \frac{Ke^{-sL}}{Ts+1} \tag{2.30}$$

Açık çevrim transfer fonksiyonu $G_P(s)$ Şekil 2.9'da gösterilmiştir.



Şekil 2.9. Sistem birim basamak cevabı şeması

Denetleyici parametreleri $Y(t), L\ ve\ T$ yanıt eğrisi kullanılarak bulunur.

$$\alpha = T/KL\ ,\ \tau = \frac{L}{L+T} \tag{2.31}$$

Çizelge 2.2. Geçici tepki yöntemine göre $Kp, Ti, Td\ ve\ PID$ parametreleri

Denetleyici	K_p	T_i	T_d
P	$1/\alpha$	-	-
PI	$0.9/\alpha$	$3L$	-
PID	$1.21/\alpha$	$2L$	$0.5L$

2.8. Wang-Juang-Chan Metodu

Çizelge 2.3'te gösterilen K, L ve T parametre değerlerine göre hesaplaması yapılan Wang-Juang-Chan metodunda PID katsayıları daha önce ifade edilen metotlardaki gibi hesaplanmaktadır (Xue ve ark., 2007).

Çizelge 2.3. PID katsayılarının Wang-Juang-Chan yöntemiyle elde edilmesi

Denetleyici	K_p	T_i	T_d
PID	$\frac{(0.7303+\frac{0.5307T}{L})(0.5L+T)}{K(L+T)}$	$0.5L+T$	$\frac{0.5LT}{0.5L+T}$

2.9. Cohen-Coon Metodu

Sistemin basamak cevabından faydalanılarak PID denetleyici katsayıları Ziegler-Nichols metodu gibi Çizelge 2.4'teki eşitlikler kullanılarak hesaplanabilmektedir (Xue ve ark., 2007).

Çizelge 2.4. PID katsayılarının Cohen-Coon metodu ile elde edilmesi

Denetleyici	K_p	T_i	T_d
P	$\frac{1}{\alpha}(1+\frac{0.35\tau}{1-\tau})$	-	-
PI	$\frac{0.9}{\alpha}(1+\frac{0.92\tau}{1-\tau})$	$\frac{3.3-3\tau}{1+1.2\tau}L$	-
PID	$\frac{1.35}{\alpha}(1+\frac{0.18\tau}{1-\tau})$	$\frac{2.5-2\tau}{1-0.39\tau}L$	$\frac{0.37-0.37\tau}{1-0.81\tau}L$

2.10. Chien-Hrones-Reswick (CHR) Metodu

Ziegler-Nicholas metodundan geliştirilen Chien-Hrones-Reswick (CHR) metodunda PID katsayıları, sistemin açık çevrim birim basamak cevabından faydalanılarak elde edilen eşitlikler hesaplamalarda kullanılmak üzere, Çizelge 2.5'te gösterilmektedir (Xue ve ark., 2007, Arora ve ark., 2011).

Çizelge 2.5. PID katsayılarının CHR yöntemiyle elde edilmesi

	Ayar değeri düzenleme						Bozucu bastırma					
	% 0 *Aşım*			% 20 *Aşım*			% 0 *Aşım*			% 20 *Aşım*		
	K_p	T_i	T_d	K_p	T_i	T_d	K_p	T_i	T_d	K_p	T_i	T_d
P	$\frac{0.3}{\alpha}$	-	-	$\frac{0.7}{\alpha}$	-	-	$\frac{0.3}{\alpha}$	-	-	$\frac{0.7}{\alpha}$	-	-
PI	$\frac{0.35}{\alpha}$	$1.2T$	-	$\frac{0.6}{\alpha}$	T	-	$\frac{0.6}{\alpha}$	$4L$	-	$\frac{0.7}{\alpha}$	$2.3L$	-
PID	$\frac{0.6}{\alpha}$	T	$0.5L$	$\frac{0.95}{\alpha}$	$1.4T$	$0.47L$	$\frac{0.95}{\alpha}$	$2.4L$	$0.42L$	$\frac{1.2}{\alpha}$	$2L$	$0.42L$

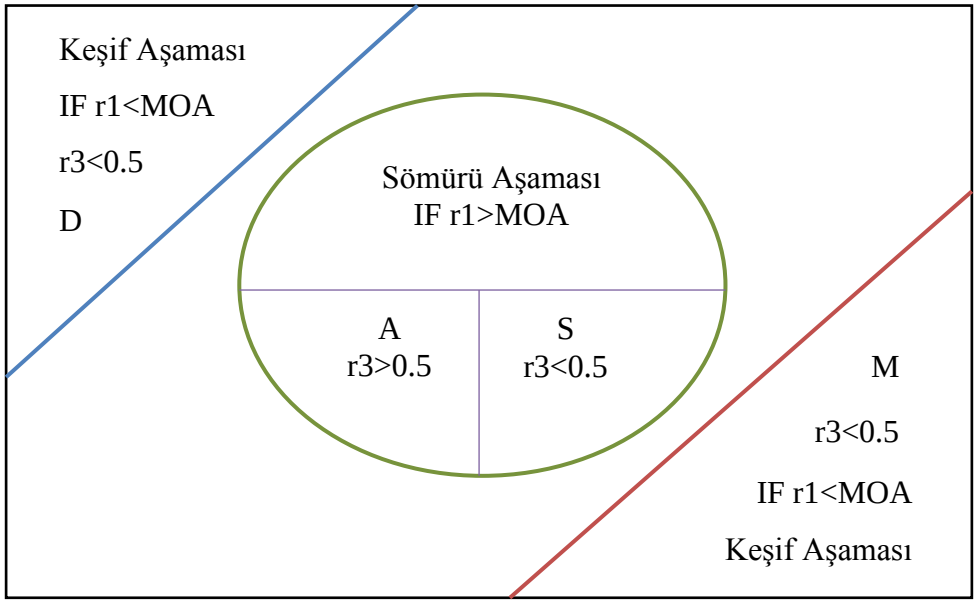
3. MATERYAL ve YÖNTEM

Bu bölümde tez çalışmasında kullanılan güncel metasezgisel algoritmalar ayrıntılı bir şekilde açıklanmıştır.

3.1. Aritmetik Optimizasyon Algoritması (AOA)

Popülasyon tabanlı bir optimizasyon algoritmasıdır. Popülasyon tabanlı algoritmalar, en iyileme süreçlerine rastgele oluşturulan bir diziyle başlarlar. Bu çözüm aşamalı olarak bir dizi optimizasyon yöntemiyle geliştirilir ve yinelemeli olarak değerlendirilir. Optimizasyon yöntemleri tek seferde çözüm bulmayı garanti etmez, çünkü optimal sonucu stokastik yöntemler ile bulmaya çalışırlar. Genel en iyi çözümü elde etme olasılığı, yeterli sayıda rastgele çözüm ve optimizasyon yinelemeleri ile arttırılmaktadır. Optimizasyon süreci iki ana bölümden oluşur. Bunlar keşif aşaması ve sömürü aşamalarıdır (Mirjalili, 2016).

Keşif aşamasında arama aracılarını kullanarak geniş çaplı bir arama alanı hedeflenirken, sömürü aşamasında ise keşif aşamasındaki bulunan çözümlerin iyileştirilmesi hedeflenir. Şekil 3.1 AOA'daki keşif ve sömürü aşamalarını göstermektedir. Popülasyon tabanlı olan bu algoritma, problemlerin türevlerini hesaplamadan çözebilir.



Şekil 3.1. AOA'nın arama aşamaları (Abualigah ve ark. 2020'den değiştirilerek alınmıştır)

3.1.1. Başlatma Aşaması

AOA'da optimizasyon süreci, Denklem 3.1'de gösterildiği gibi rastgele oluşturulan bir dizi aday çözüm (X) ile başlar ve her yinelemedeki en iyi aday çözüm, şimdiye kadar elde edilen en iyi çözüm olarak kabul edilir.

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{3.1}$$

AOA çalışmaya başlamadan önce, arama aşaması (yani keşif veya kullanım) seçmelidir. MOA (Math Optimizer Accelerated) fonksiyonu Denklem 3.2'de verilmektedir.

$$MOA(C_{Iter}) = Min + C_{Iter} \times \left(\frac{maks-min}{M_{Iter}}\right) \tag{3.2}$$

MOA; C_{Iter}, ile hesaplanan $t.$ iterasyondaki fonksiyon değerini gösterir. M_{Iter} ile maksimum yineleme sayısı C_{Iter} ise geçerli yineleme saysını ifade etmektedir. Hızlandırılmış fonksiyon değerleri $maks$ ve min ile gösterilmektedir.

3.1.2. Keşif Aşaması

Matematiksel hesaplamalardaki "M" çarpma ve "D" bölme operatörünü kullanan hesaplamalar, keşif arama mekanizmasına uyması için yüksek dağılıma sahip değerler sağlar. Yüksek dağılıma sahip olduklarından dolayı "M" çarpma ve "D" bölme operatörleri "S" Çıkarma ve "A" Toplama'ya göre hedefe daha zor yaklaşırlar. Şekil 3.2, matematiksel hesaplamalarda aritmetik operatörlerin etkisini ve davranışını göstermektedir (Abualigah ve ark., 2020).

Farklı operatörlerin dağılım değerlerinin etkisini görebilmek için dört işlemin tamamına dayalı bir fonksiyon kullanılır. Bu nedenle, keşif araştırması, birkaç denemeden (yinelemelerden) sonra çıkarılabilecek optimale yakın çözümü tespit eder. Optimizasyonun bu aşamasında sömürü aşamasını desteklemek için arama operatörleri çalıştırılır. AOA'nın keşif aşaması arama alanını birkaç bölge için rastgele olarak bulur

ve Bölme “D” ile Çarpma “M” arama stratejisine dayalı daha iyi bir çözüm bulmaya çalışır.

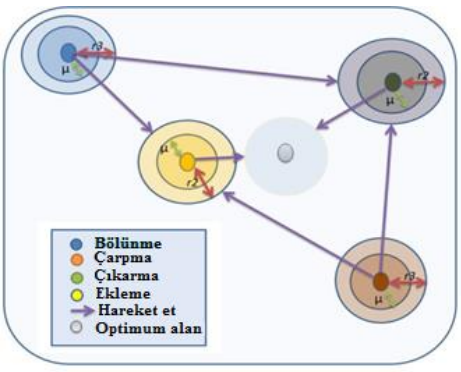
Bu aşamada $r2 < 0.5$ durumunda “D” operatörü kullanılır ve diğer operatör “M” bu operatör mevcut görevini bitirene kadar ihmal edilir. $r2 > 0.5$ durumunda ise “M” operatörü kullanılacaktır. Keşif bölümleri için Denklem 3.3’teki konum güncelleme denklemleri önerilmiştir:

$$x_{i,j}(C_{Iter}+1) = \begin{cases} eniyi(x_j) \div (MOP+\epsilon) \times \left((UB_j - LB_j\ ) \times \mu + LB_j \right), & r2 < 0.5 \\ eniyi(x_j) \times MOP \times \left((UB_j - LB_j\ ) \times \mu + LB_j \right), & diğer \end{cases} \quad (3.3)$$

$x_i(C_{Iter}+1)$ bir sonraki yinelemedeki i. çözümü gösterir, $x_{i,j}(C_{Iter})$ mevcut yinelemede i 'inci çözümün j'inci konumunu belirtir ve $eniyi(x_j)$ en iyi elde edilen çözümde j. konumdur, bu nedenle ıraksar. ϵ küçük bir tam sayıdır, UB_j ve LB_j sırasıyla j. pozisyonun üst ve alt sınır değerini belirtir.

$$MOP(C_{Iter}) = 1 - \frac{{C_{Iter}}^{1/a}}{{M_{Iter}}^{1/a}} \quad (3.4)$$

MOP (Math Optimizer Probability) bir katsayı olduğundan, MOP(C_{Iter}) yinelemedeki işlev değerini, C_{Iter} geçerli yinelemeyi ve M_{Iter} maksimum yineleme sayısını belirtir. α iterasyonlar üzerindeki kullanım doğruluğunu tanımlamakta ve 5 olarak seçilmesinin uygun olduğu ifade edilmektedir (Abualigah ve ark., 2020).



Şekil 3.2. Kullanılan operatörlerinin optimum alana yaklaşma modeli (Abualigah ve ark. 2020’den değiştirilerek alınmıştır)

3.1.3. Sömürü Aşaması

Aritmetik operatörlere göre, Çıkarma (S) veya Toplama'yı (A) kullanan matematiksel hesaplamalar, sömürü arama mekanizmasına karşılık gelen yüksek yoğunluklu değerler sağlar. Düşük dağılımlı olduklarından dolayı "S" veya "A" hedefe daha kolay yaklaşırlar. Bu nedenle sömürü aşaması optimum sonucu daha az denemeden sonra tespit eder.

Aritmetik operatörlerden (S ve A), düşük dağılımlardan dolayı hedefe kolay yaklaştıkları için optimizasyonun bu aşamasında çalıştırılır. Bu arama aşaması (S veya A'yı çalıştırarak yararlanma araması), $r1$'in mevcut MOA(C_{Iter}) değerinden büyük olmaması için MOA fonksiyon değeriyle koşullandırılır.

AOA' da, sömürü operatörleri (S ve A) arama alanını birkaç yoğun bölgede derinlemesine araştırır ve Denklem 3.5'te gösterildiği gibi ifade edilir.

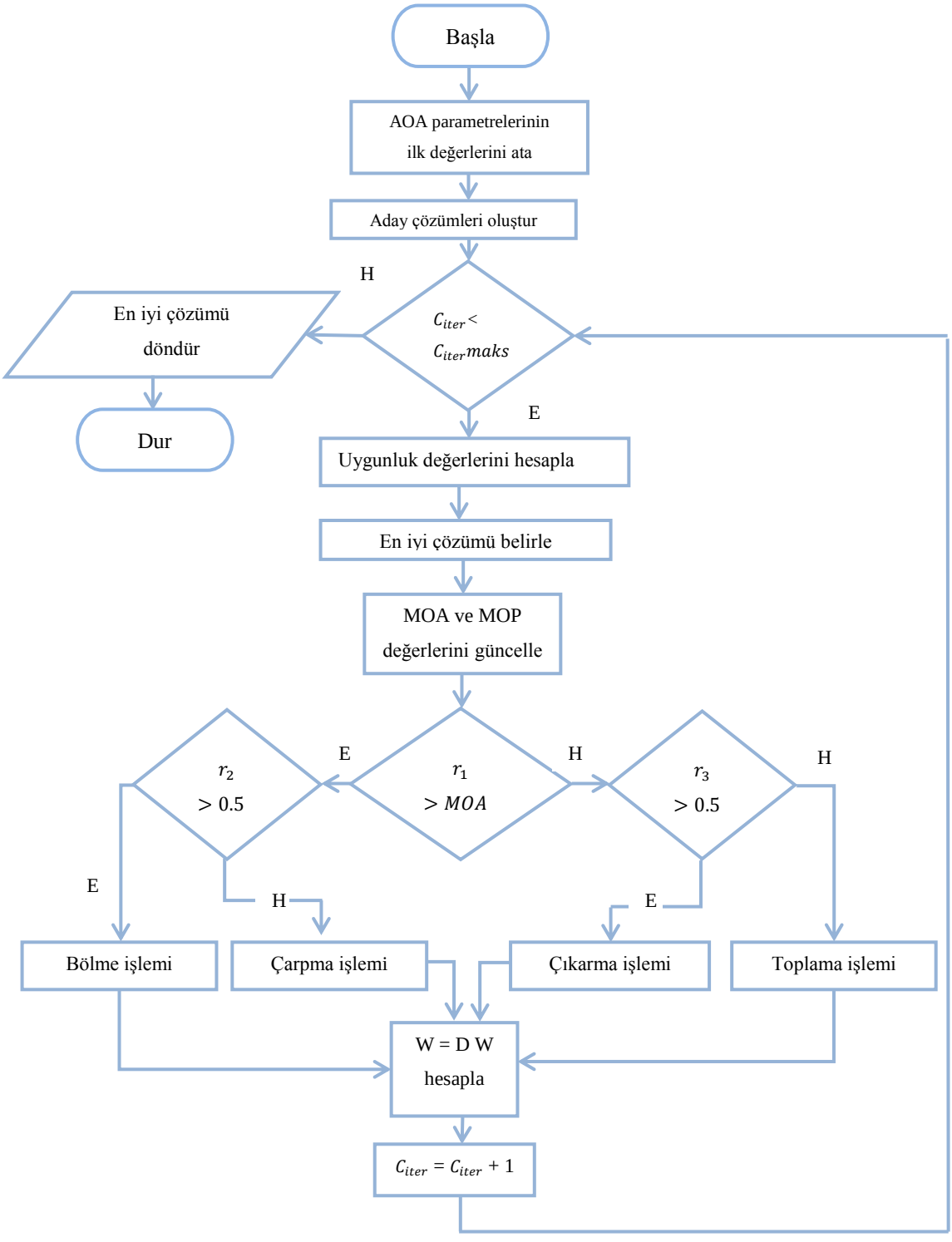
$$x_{i,j}(C_{Iter}+1) = \begin{cases} eniyi(x_j) - (MOP) \times \left((\mathrm{UB}_j - \mathrm{LB}_j) \times \mu + \mathrm{LB}_j \right), & r3 < 0.5 \\ eniyi(x_j) + MOP \times \left((\mathrm{UB}_j - \mathrm{LB}_j) \times \mu + \mathrm{LB}_j \right), & diğer \end{cases} \quad (3.5)$$

Bu aşamadaki çıkarma operatörü (S) $r3 < 0.5$ durumunda çalışır ve toplama (A) operatörü ihmal edilebilir. Aksi takdirde, çıkama yerine ikinci operatör toplama mevcut görevi yerine getirecektir.

Şekil 3.2, bir arama çözümünün değişkenlerini (konumlarını) iki boyutlu bir arama uzayında D, M, S ve A'ya göre nasıl güncellediğini açıklamaktadır. Diğer kavramlarda, D, M, S ve A optimuma yakın çözümün konumunu tahmin eder ve diğer çözümler, optimuma yakın çözüm alanı etrafındaki konumlarını stokastik olarak günceller. Son olarak sonlandırma kriteri karşılandığında, AOA'nın arama süreci sonlandırılır. AOA'nın sözde kodu Algoritma 1'de ve akış diyagramı da Şekil 3.3'te gösterilmektedir.

Algoritma 1. Aritmetik Optimizasyon Algoritması Sözde Kodu

M_{iter} ← maksimum iterasyon sayısı
N ← popülasyon boyutu
D ← problem boyutu
MOP_{Maks} ← MOP fonksiyonu maksimum değeri
MOP_{Min} ← MOP fonksiyonu minimum değeri
a ← hassasiyet değeri
μ ← arama sürecini düzenleyen kontrol parametresi
for i = 1: N
  $P_i \leftarrow i.$ çözümü rastgele üret
end for
for $g = 1 : M_{iter}$
  for $i = 1: N$
    if f(P_i) < $f(eniyi)$
      en iyi ← P_i
    end if
  end for
  MOA ← Denklem 3.2 ile MOA değerini güncelle
  MOP ← Denklem 3.4 ile MOA değerini güncelle
  for $i = 1: N$
    for $j = 1: D$
      r_1, r_2, r_3 ← [0,1] aralığında rastgele sayı üret
      if $r_1 > MOA$
        $x_{i,j}$ ← Denklem 3.3 ile çözümü güncelle
      else
        $x_{i,j}$ ← Denklem 3.5 ile çözümü güncelle
      end if
    end for
  end for
end for
en iyi sonucu göster



Şekil 3.3. Aritmetik optimizasyon algoritması akış diyagramı

3.2. Dinamik Diferansiyel Tavlı Optimizasyon Algoritması (DDAO)

İmalat sürecinde yaygın olarak kullanılan çeliğin mekanik özelliklerinin iyileştirilmesi, gerekli kütlesinin azaltılması ve maliyetinin düşürmesi için birçok çalışma yapılmıştır (Dutta ve ark., 2019). Bu çalışmalar, yüksek mukavemetli çeliklerin geliştirilmesi ile ilgilidir. Çift fazlı çelik, süper mekanik özelliklerini veren benzersiz bir mikro yapıya sahip bir tür gelişmiş yüksek mukavemetli çeliktir. Mekanik özelliklerini benzersiz bir mikro yapıya sahip olmasından alan çift fazlı çeliğin, yüksek mukavemetli olması ve deformasyon olmaması için hassas bir tavlama yapılması gerekir. Şekil 3.4 çift fazlı çeliğin oluşum aşamalarını göstermektedir. Genel olarak yüksek bir sıcaklıkta demir, sıvı halde erir ve sürekli soğutma ile oda sıcaklığında katı hale dönüşür.

Katı halde olan metallerin fazları vardır. Bunlar ferritik, martensit, bainit ve östenit gibi fazlardır. Bu fazlar çeliğin mekanik özelliklerine etki ederler (Ghafil ve ark., 2020). Çift fazlı çelik üretiminde; kristalleşme noktası üzerinde yüksek sıcaklığa kadar ısıtıldıktan sonra kalınlığı incelene kadar haddelenmesi gerekir. Bu sırada metal soğutma işlemi gerçekleşmiş olur.

Mekanik deformasyon işlemi yapıldıktan sonra metal diferansiyel soğutmaya tabi tutulur. Bu aşamalar; hızlandırılmış soğutma, hava soğutma ve son olarak da oda sıcaklığında soğutmadır. Bu süreç, adalar şeklinde sert bir martensitik faz içeren bir ferritik matris mikroyapısına sahip çeliğe yol açar. Yumuşak ferrit ve sert martensitin fazlarını bulunduran çift fazlı çelikler ferritten dolayı yüksek sünekliğe ve martensit nedeniyle yüksek mukavemete sahiptir.

Şekil 3.4. Çift fazlı çelik üretim sürecinin açıklaması (Ghafil ve ark. 2020'den değiştirilerek alınmıştır)

Yüksek kaliteli bir çelik türü (çift fazlı) üretmek için daha önce açıklanan sistematik prosedür, açık bir optimizasyon sürecidir. Bu süreçte metal kalitesi arttırılmış olup ilgili işleyiş DDAO algoritmasının oluşmasına katkı sağlamıştır. DDAO işleyişi aşağıdaki şekilde özetlenebilir:

1- Çelik kütlesi, ferrit ve martensit karışımı molekül gruplarından oluşur.
2- Çift fazlı çelik üretiminde farklı sıcaklıklarda kritik tavlama yapılarak farklı geçirgenlik birimi (mho)'na sahip çift fazlı çelikler üretilebilir. Bu durum global bir çözüm için matematiksel optimizasyondaki yineleme sürecini belirler.
3- Her fazın kendi iç enerjisi vardır ve optimizasyondaki amaç fonksiyonunun değerine eşdeğerdir.
4- Şekil 3.4, çift fazlı çeliğin soğutma hızlarını göstermektedir. Bu aşamalar; hızlandırılmış, hava ve son olarak oda sıcaklığında soğutmadır.

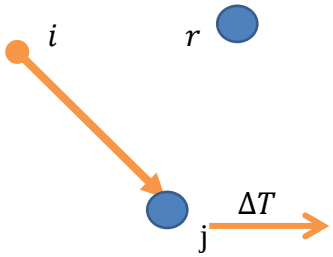
$$S^k = \left(Sc_i - Sc_j\right) + Sr \tag{3.6}$$

burada S^k, yineleme sayısı k için önerilen yeni bir çözümdür, $k = 1 \ldots n$ olup n yineleme sayısıdır. Sc_i ile Sc_j, popülasyondaki rastgele i ve j endeksleridir.

Sr, arama alanı içinde olup popülasyonun içinde olmayan rastgele oluşturulmuş bir çözümdür. Sıcaklığın ΔT kadar i noktasından j noktasına değiştirildiği Şekil 3.5 göz önüne alındığında objektif değere sahip enerji seviyeleri için bir çözüm vardır. İki enerji seviyesi arasındaki farkda bir çözüm belirtir.

Bu, önerilen algoritmanın temel denklemi Denklem 3.6'da verilmiştir ve optimizasyon problemlerinin ana yakınsamasından sorumludur. Denklem 3.6 ve onun tamamlayıcısı olan Denkleminin 3.7; diferansiyel evrimde belirtilen mutasyon sürecine benzerdir, ancak bu algoritma işleyişinde evrim yoktur, rastgele seçim vardır.

5- Isının diferansiyel indirgenmesi sırasında metal haddelenir ve bu mekanik işlemin matematiksel olarak modellenmesi gerekir. Bu yüzden metalin haddeleme yerine dövme işlemine tabi tutulduğu varsayılır.

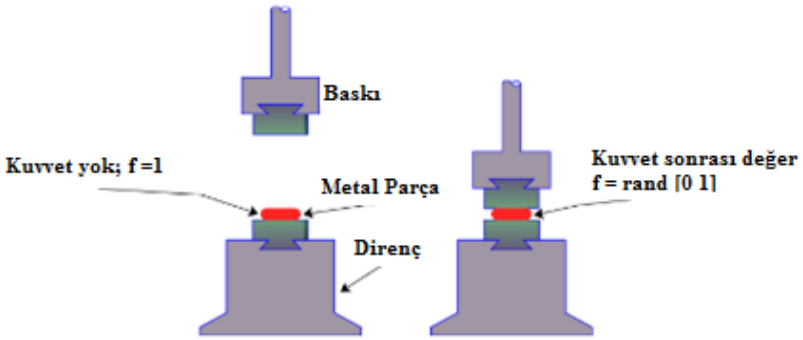


Şekil 3.5. Farklı enerji seviyeleri için arama alanındaki farklı aday çözümleri

$$f = \begin{cases} 1 & , \ eğer\ kalan(iterasyon, 2) = 1 \\ rastgele\ [0,1] & , \ eğer\ kalan(iterasyon, 2) = 0 \end{cases} \quad (3.7)$$

Çekicin dövme sırasında dinamik davranışı, 1 ile rastgele bir sayı arasında dalgalanan bir parametre olarak gösterilebilir; burada f dövme parametresidir ve 2'ye bölünmeden sonra kalan kısımdır. Yinelemeler sırasında denklem, geçerli yineleme sayısının tek olduğu, yani 1.3.5... vb. olduğu anlamına gelir.

Yinelemeler sırasında Denklem 3.5, yinelemenin geçerli sayısı tek olduğunda $f = 1$ olacak, mevcut iterasyon sayısı çift ise $f = (0,1)$ arasında rastgele bir sayı olacaktır. Dövme işlemi diferansiyel soğutma ile birlikte yapıldığından, Denklem 3.6'da gösterilmektedir.



Şekil 3.6. Diferansiyel tavlı dövme işlemi(Ghafil ve ark. 2020'den değiştirilerek alınmıştır)

6- Gerçek tavlama işleminde, düşük sıcaklıklara göre yüksek sıcaklıklarda yeni fazların oluşumunu kabul etme olasılığı daha yüksektir. Optimizasyon sürecinde, Benzetilmiş tavlama (SA) algoritması tarafından açıklanan olasılık formülüne bağlı olarak Denklem 3.8'deki gibi ifade edilir.

$$P = e^{\frac{-\Delta E}{T}} \tag{3.8}$$

$$\Delta E = \frac{Cost(S^k) - Cost(S_L)}{Cost(S_L)} \tag{3.9}$$

P yeni bir çözümü kabul etme olasılığı olduğunda, ΔE Denklem 3.9'dan önerilen çözümün nesnel değeri ile popülasyondaki L indeksinin bir çözümü olan S_L çözümünün nesnel değeri arasındaki farktır. $L = 1, .., N$ popülasyon büyüklüğü ve T güncellenmesi gereken sıcaklık değeridir. Bu parametre yüksek sıcaklıkla başlayıp yinelemeler ile sürekli olarak düşük değere güncellenmelidir. Önerilen çözüm, $P > rastgele\, sayı \in [0,1]$ ise kabul edilebilir. Aramanın başlangıcında, T yüksek değerle başlar; sonuç olarak, Denklem 3.8'e göre P 1'e yakın olacaktır. Bu, rasgele sayı aralığının birden az olabileceği ve çözümün seçileceği anlamına gelir. T 'nin düşük değerinde, P olasılığı sıfıra yakın olacaktır; Denklem 3.8'e göre bu, çok dar bir rasgele sayı aralığının P'den daha az olabileceği ve çözümün seçilme olasılığının daha düşük olduğu anlamına gelir. Örneğin $e^{-1/300} = 0.9967$ iken $e^{-1/0.3} = 0.0357$ dir. Dolayısıyla bu, sıcaklık düştükçe yeni bir çözümün seçilme olasılığını azaltan basit bir mekanizmadır.

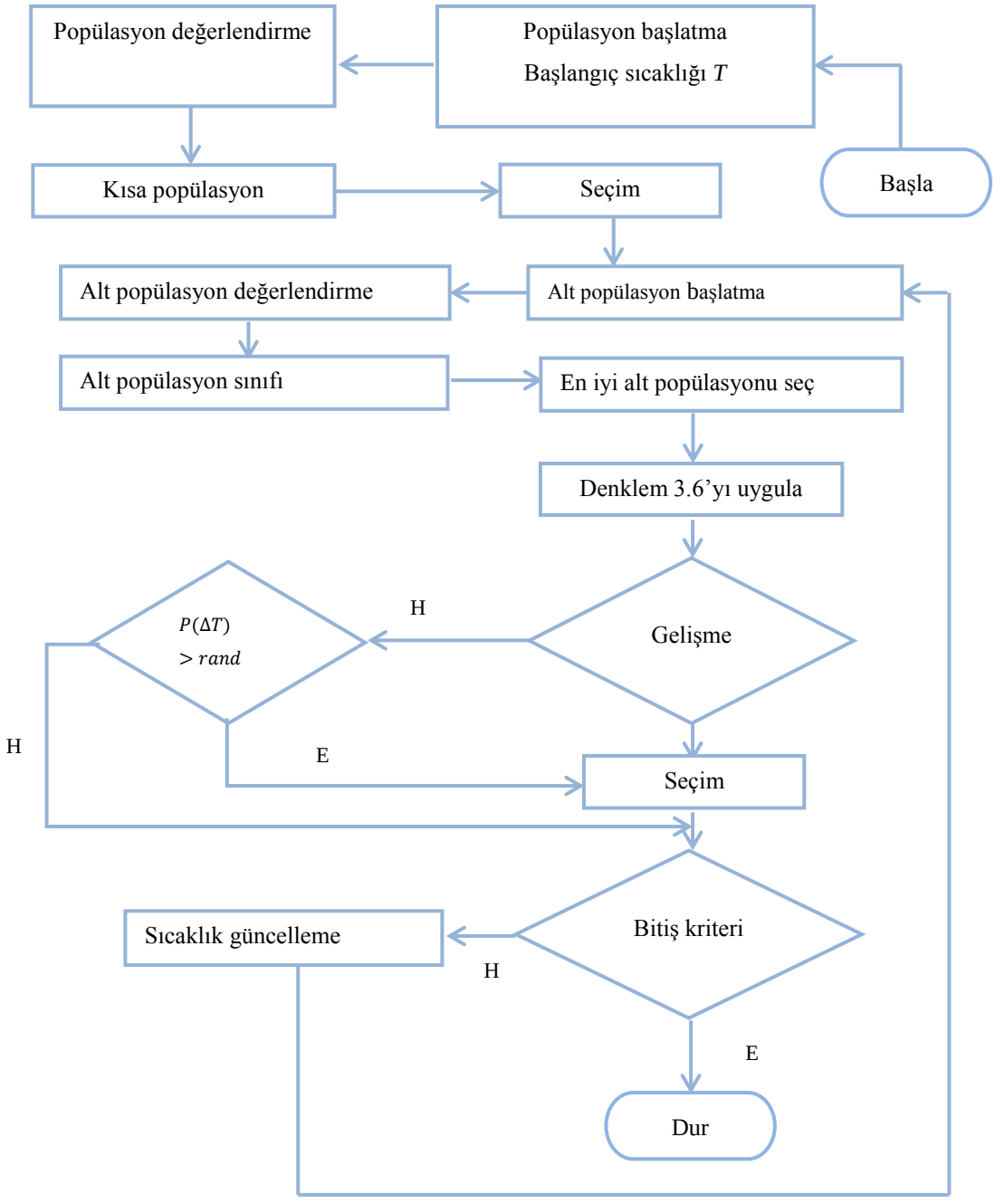
7- İşlem 4. adımdan itibaren tekrarlanır ve her yineleme için en iyi çözüm saklanır.

Dövme parametresi f, matematiksel problemlerin optimize edilmesi sırasında DDAO' nun genel performansı üzerinde gözle görülür bir etkiye sahiptir. Bazı problemler için Denklem 3.7'de $f = 1'e$ eşit olduğunda daha iyi bir sonuç bulunurken, f rastgele bir değere eşit olduğunda daha kötü bir sonuç bulunabilir ve bu başka bir problem seti için tam tersi olabilir. Bu nedenle Denklem 3.7'deki yinelemelerin yarısı $f = 1$ olarak kabul edilir ve yinelemenin kalan yarısı $f = rastgele\,[0\ 1]$ alınarak bu soruna bir çözüm sunulmaktadır.

DDAO, üç parametreli basit bir yapıya sahiptir. Bunlar maksimum yineleme sayısı, alt yineleme sayısı ve ayarlanabilen soğutma hızı olarak ifade edilebilir. Algoritma 2 DDAO algoritmasının sözde kodu ve Şekil 3.7'de akış diyagramı gösterilmektedir.

Algoritma 2. Dinamik Diferansiyel Tavlama Optimizasyon Algoritması Sözde Kodu

1. Başlangıç popülasyonunu oluştur $X_i = (i = 1,2,3, \dots n)$
2. soğutma hızı, başlangıç parametresi T
3. Çözüm için amaç fonksiyonunu hesapla
4. X_b en iyi sonuç
5. **while** ($t <$ maks. iterasyon)
6. S alt popülasyonu başlat
7. Alt popülasyonun amaç fonksiyonunu hesapla
8. Alt popülasyonu sırala
9. S_r alt popülasyondaki en iyi çözüm
10. Popülasyon $X_m$ve $X_n$denkleminden $S_k$hesapla Denklem 3.6
11. Popülasyon x'i sırala
12. **for** (Popülasyondaki her çözüm için)
13. **if** iyilişme varsa
14. $X_i = S_k$
15. X popülasyonundaki en kötü çözümü değiştir Denklem (3.8 , 3.9)
16. **end if**
17. **end for**
18. X_B'yi güncelle
19. $T = T * Soğutma\ hızı$
20. $t = t + 1$
21. **end while**
22. X_B yi döndür
23. en iyi sonucu göster



Şekil 3.7. Dinamik diferansiyel tavlama optimizasyonu akış diyagramı

3.3. Kel Kartal Arama Optimizasyon Algoritması (BES)

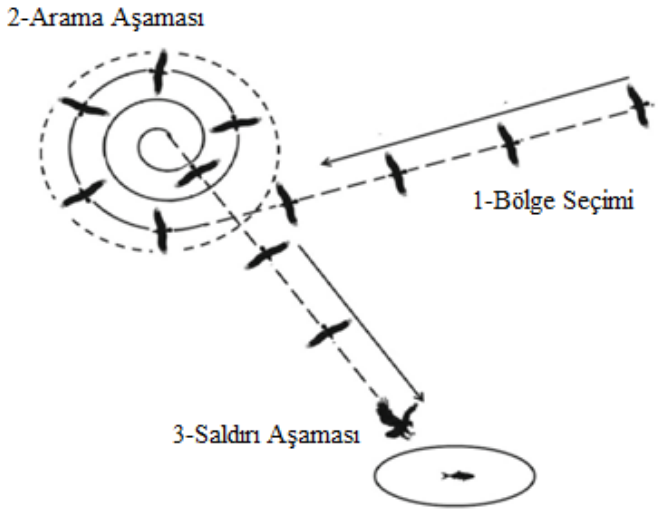
3.3.1. Kel Kartalın Avlanma Davranışı

Kel kartallar; esas olarak balıkları (canlı veya ölü), özellikle somonu birincil besin olarak seçen fırsatçı avcılardır. Kel kartallar uçarken avlanabildiği gibi gizlenerek de avlanabilirler. Sudan balık elde etmek zor olduğu için çok uzak mesafelerden balıkları tespit edebilirler. Böylece 20 saldırı girişiminden sadece 1'inde başarılı olabilirler (Stalmaster & Kaiser, 1997). Başarılı olduktan sonra avlanma sırasında harcadıkları enerjiden dolayı dinlenirler. Şekil 3.8'de kel kartalın avlanma davranışları gösterilmektedir. Kel kartallar belirli bir yönde arama yapmaya başlarlar ve belirli bir alan seçimi yaparlar.

Şekil 3.8. Kel kartalın avlanma sırasındaki davranışı(Alsattar 2020'den değiştirilerek alınmıştır)

Daha sonra belirledikleri alana giderler. Avlanma davranışının ilk aşaması bölge seçimidir. Kıyıdan 5 m mesafede (% 47), suyun derinliklerine kıyasla yiyecek arama başarısı yüksektir. Kara yüzeyi ile derin su arasındaki bölgeyi kendilerine arama alanı seçerler. Spesifik olarak, bir çift kartal her gün 250 hektarlık açık otlakta avlanır. Enerji; aramada kritik bir etken olduğu için kartalların aramaya başlayacakları alan yuvalarından 700 m'den daha uzak değildir çünkü enerji yönü aramada kritik bir faktördür (Lasserre, 2001). Kartalların bir seferde saatlerce kayan, zarif, hareketsiz uçuşlara sahip oldukları gözlemlenmiştir (Stalmaster ve Kaiser 1997; Hansen 1986; Hansen ve ark. 1984).

Keskin görüşe sahip olan kartallar çok uzun mesafeden sudaki canlıları, balıkları görebilirler. Kartallar havada binlerce fit yükseklikte uçarken, bir bükülme hareketiyle tarama yapmaları kolaylaşmakta ve kartal yaklaşık 3 m^2'lik alanda bir avı tespit edebilmektedirler. Arama aşaması, avlanma davranışının ikinci aşamasıdır. Kartallar avı gördüklerinde, ava yüksek hızda ulaşmak ve balığı sudan kapmak için kademeli bir hareket akışı ile alçalan avlanma davranışının son aşamasına başlayacaklardır (Şekil 3.9).



Şekil 3.9. BES avlanma aşamasında izlediği yol (Alsattar 2020'den değiştirilerek alınmıştır)

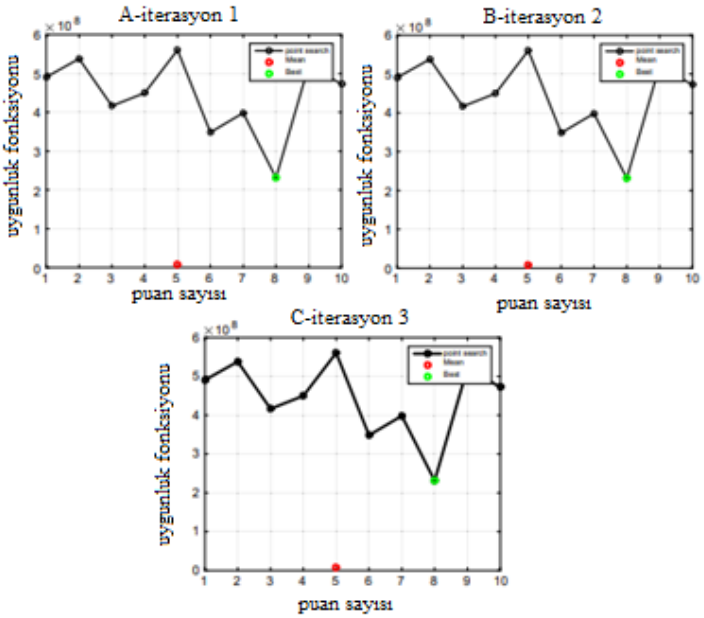
3.3.2. Bölge Seçimi

Seçme aşamasında; kel kartallar, seçilen arama alanında av için avlayabilecekleri en iyi alanı (yiyecek miktarı açısından) belirler ve seçerler. Denklem 3.10'da bu davranışın matematiksel ifadesi verilmektedir.

$$P_{yeni,i} = P_{eniyi} + a * r(P_{ort} - P_i) \tag{3.10}$$

Arama alanı değişikliği α, 1.5 ile 2 arasında ve r rastgele değer alan bir sayı olup 0 ile 1 arasında değişmektedir. P_{eniyi} önceki arama sırasında belirlenen en iyi konuma dayalı seçilen arama alanıdır ve P_{ort} da önceki noktalardan gelen tüm bilgilerin kullanıldığını göstermektedir. Dolayısıyla kel kartallar - önceki aşamadaki mevcut bilgilere dayanarak - seçilen arama alanının yakınındaki tüm noktaları rastgele aramaktadırlar.

Kel kartalların mevcut hareketi, pozisyon değişimlerini kontrol eden a ile çarpılmasıyla belirlenir.



Şekil 3.10. Doğaçlama seçim aşaması (Alsattar 2020'den değiştirilerek alınmıştır)

Şekil 3.10, seçim aşamasının ortalama ve en iyi noktalara dayalı olarak arama içindeki tüm çözümleri etkili bir şekilde iyileştirdiğini göstermektedir. Şekil 3.10.a, arama alanı içerisindeki en iyi ve ortalama çözümlerin konumunu göstermektedir. Ortalama nokta, arama alanında en iyi konuma sahipken seçilen alan, arama ve ortalama noktalar arasındaki farka bağlıdır. Şekil 3.10.b, arama noktası 10'un en iyi noktanın yakınında olduğunu gösterir. Şekil 3.10.c, arama noktası 3'teki yeni en iyi noktayı göstermektedir.

3.3.3. Arama Aşaması

Bu aşamada kel kartallar; seçilen arama alanı içinde av ararken aramalarını hızlandırmak için spiral bir alan içerisinde farklı yönlere hareket ederler. Bu davranışın matematiksel modeli Denklem 3.11'de verilmiştir.

$$P_{i,yeni} = P_i + y(i) * (P_i - P_{i+1}) + x(i) * (P_i - P_{ort}) \tag{3.11}$$

$$x(i) = \frac{xr(i)}{\max(|xr|)}, \quad y(i) = \frac{yr(i)}{\max(|yr|)} \tag{3.11a}$$

$$xr(i) = r(i) * \sin(\theta(i)),\ yr(i) = r(i) * \cos\ (\theta(i)) \quad (3.11b)$$

$$\theta(i) = a * \pi * rand \quad (3.11c)$$

$$r(i) = \theta(i) + R * rand \quad (3.11d)$$

burada a, merkezi noktanın nokta araması ile köşeye uzaklığını belirlemek için 5 ile 10 arasında değer alan bir parametredir. R arama döngülerinin sayısını belirlemek için 0.5 ile 2 arasında bir değer almaktadır.

3.3.4. Saldırı Aşaması

Hedef avına doğru salınan kel kartallar, arama alanı içerisindeki en iyi noktaya hareket ederler. Bu davranış matematiksel olarak Denklem 3.12'de gösterilmektedir.

$$P_{i,yeni} = rand * P_{eniyi} + x1(i) * (P_i - c1 * P_{ort}) + y1(i) * \left(P_i - c2 * P_{eniyi}\right) \quad (3.12)$$

$$x1(i) = \frac{xr(i)}{\max(|xr|)},\quad y1(i) = \frac{yr(i)}{\max(|yr|)} \quad (3.12a)$$

$$xr(i) = r(i) * \sinh[\theta(i)],\ yr(i) = r(i) * \cos h[\theta(i)] \quad (3.12b)$$

$$\theta(i) = a * \pi * rand \quad (3.12c)$$

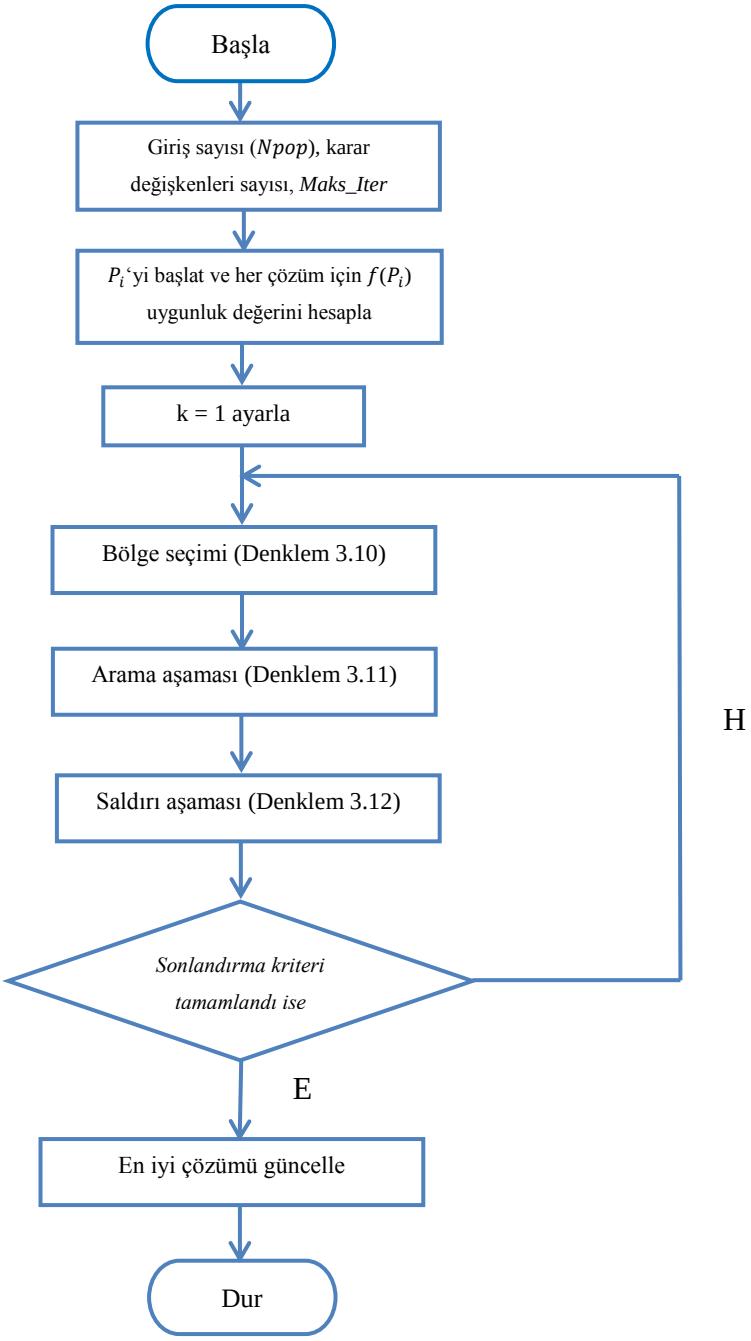
$$r(i) = \theta(i) \quad (3.12d)$$

Burada $c1, c2\ \in [1,2]$'dir. Kartalların hareketi, farklı şekiller alır. Uçuş hareketi için kutupsal bir denklem kullanılmaktadır. Algoritma 3'te Kel kartal optimizasyonu algoritmasının sözde kodu verilmiştir. P popülasyonundaki her bir çözüm için Algoritma 3'teki adımlar uygulanır. 4–12. satırlarda yeni alan için iki eksende rastgele sayının üretildiği spiral hareketi kullanarak arama ve seçim alanını belirler. Çözüm, bir sonraki noktaya ve merkezi noktaya doğru hareket eder. Yeni av pozisyonunu 13-21. satırlarda belirlenmektedir.

Yeni çözüm, 22-30. satırlar kullanılarak değerlendirilmektedir. Yineleme sayacı k, üç adım çalıştırıldıkça 31. satırda 2 artırılır ve başlangıçta ayarlanan yineleme sayısı kadar tekrarlanır. Son olarak, P'deki çözümler, nihai popülasyon ve problemi çözmek için popülasyonda elde edilen en iyi çözüm gösterilir. Algoritma 3 BES algoritmasının sözde kodu ve Şekil 3.11'de akış diyagramı verilmektedir.

Algoritma 3. Kel Kartal Arama Optimizasyon Algoritması Sözde Kodu

1. $n.$ nokta için P_i noktasını rastgele başlat
2. Başlangıç noktasının uygunluk değerlerini hesapla (P_i)
3. **while** (sonlandırma kriteri sağlanıncaya kadar)
Bölge seçimi
4. **for** (Popülasyondaki her i noktası)
5. $P_{yeni} = P_{eniyi} + a * \text{rand}(P_{ort} - P_i)$
6. **if** $f(P_{yeni}) < f(P_i)$
7. $P_i = P_{yeni}$
8. **if** $f(P_{yeni}) < f(P_{eniyi})$
9. $P_{eniyi} = P_{yeni}$
10. **end if**
11. **end if**
12. **end for**
Arama aşaması
13. **for** (Popülasyondaki her i noktası)
14. $P_{yeni} = P_i + y(i) * (P_i - P_{i+1}) + x(i) * (P_i - P_{ort})$
15. **if** $f(P_{yeni}) < f(P_i)$
16. $P_i = P_{yeni}$
17. **if** $f(P_{yeni}) < f(P_{eniyi})$
18. $P_{eniyi} = P_{yeni}$
19. **end if**
20. **end if**
21. **end for**
Saldırı aşaması
22. **for** (Popülasyondaki her i noktası)
23. $P_{yeni} = rand * P_{eniyi} + x_1(i) * (P_i - c1 * P_{ort}) + y1(i) * (P_i - c2 * P_{eniyi})$
24. **if** $f(P_{yeni}) < f(P_i)$
25. $P_i = P_{yeni}$
26. **if** $f(P_{yeni}) < f(P_{eniyi})$
27. $P_{eniyi} = P_{yeni}$
28. **end if**
29. **end if**
30. **end for**
31. Set $k = k + 1$
32. **end while**



Şekil 3.11. Kel kartal arama optimizasyonu akış diyagramı

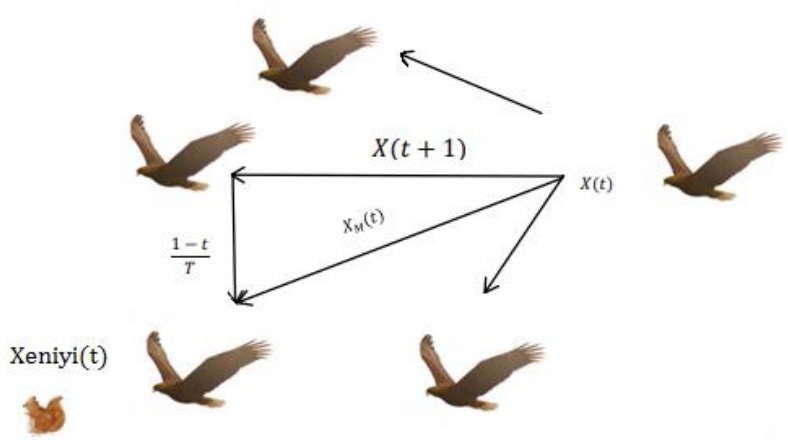
3.4. Aquila (Kartal) Optimizasyon Algoritması (AO)

3.4.1. AO'nun matematiksel modeli

Önerilen AO algoritması, kartalın avlanma sırasındaki davranışını simüle eder ve avın her adımında 160 eylemi gösterir. AO algoritmasının optimizasyon prosedürleri dört yöntemde temsil edilmektedir; dikey eğimle süzülerek arama alanı seçme, kısa süzülme saldırısı, sınır çizgisi ile keşif yapma ve yavaş inişli saldırıyla bir arama alanı içinde sömürü yaparak avını yakalamaya çalışmaktadır (Watson, 2010). Kartal davranışlarından esinlenerek matematiksel modellere dönüştürülmesi adım adım anlatılmıştır.

- Adım 1: Genişletilmiş keşif (X_1)

Birinci adımda X_1, kartal av bölgesini tanır ve dikey eğimle yüksekten süzülerek en iyi avlanma alanını seçer/belirler. Şekil 3.12 kartalın dikey eğimle yüksek süzülme davranışını göstermektedir ve bu davranış matematiksel olarak Denklem 3.13'te verilmektedir.



Şekil 3.12. Kartal dikey eğimli süzülme davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)

$$X_1(t+1) = X_{eniyi}(t) \times \left(1 - \frac{t}{T}\right) + (X_M(t) - X_{eniyi}(t) * rand) \quad (3.13)$$

Burada, $X_1(t+1)$, birinci arama yöntemi X_1 tarafından üretilen t'nin bir sonraki yinelemesinin çözümüdür. Avın yaklaşık yerini belirten en iyi çözüm $X_{eniyi}(t)$, t. iterasyonuna kadar elde edilen çözüm olan bu denklem $(1 - t\ /\ T)$, yineleme sayısı ile genişletilmiş arama kontrolünde kullanılmaktadır.

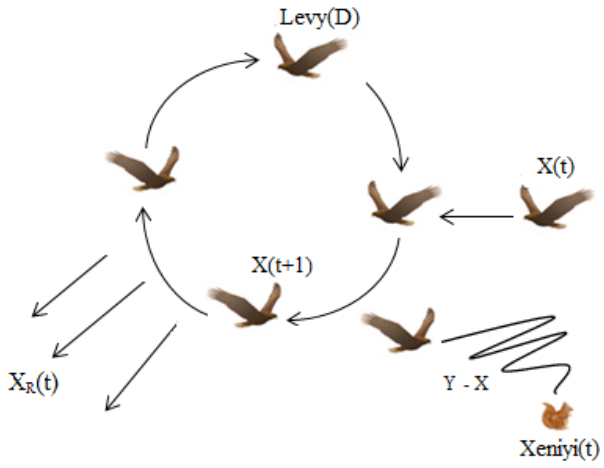
$X_{M(t)}$, Denklem 3.14 kullanılarak hesaplanan t'inci yinelemede çözümlerinin konumu için ortalama değerini belirtir.

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t), \forall j = 1,2,\ldots,Dim \qquad (3.14)$$

Burada *Dim*, problemin boyutunu, N ise aday çözüm sayısını gösterir (popülasyon boyutu).

- Adım 2: Daraltılmış keşif (X_2)

İkinci adımda (X_2), av alanı yüksek uçularak bulunur. Kartal hedef avın üzerinde daireler çizer, av bölgesini hazırlar ve ardından saldırır. Kontür uçuşu denilen bu yöntem kısa süzülme davranışıdır. Burada AO, saldırıya hazırlanırken hedef avın seçilen bölgesini dar bir şekilde araştırır (Şekil 3.13). Bu davranış matematiksel olarak Denklem 3.15'te verilmektedir.



Şekil 3.13. Kartal kısa süzülme kontür uçuşu davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)

$$X_2(t+1) = X_{eniyi}(t) \times Levy(D) + X_R(t) + (y - x) * rand \qquad (3.15)$$

Burada $X_2(t+1)$, ikinci arama yöntemi (X_2) tarafından üretilen bir sonraki $t+1$ iterasyonunun çözümüdür. D boyut uzayıdır ve *Levy(D)*, Denklem 3.16 kullanılarak hesaplanan dağılım fonksiyonudur. $X_{R(t)}$, $i.$ yinelemesinde $[1\ N]$ aralığında alınan rastgele bir çözümdür.

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{3.16}$$

Burada s, 0.01'e sabitlenmiş bir değer olup u ve v, 0 ile 1 arasındaki rastgele sayılardır. σ, Denklem 3.17 kullanılarak hesaplanır.

$$\sigma = \left(\frac{\Gamma(1+\beta) \times sine(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right) \tag{3.17}$$

Burada β değeri 1.5'tir. r_1, 1 ile 20 arasında bir değer alır ve U ise 0.00565'e eşittir. D_1, 1'den arama alanının uzunluğuna (Dim) kadar olan tam sayılardır ve $\omega, 0.005$ olan küçük bir sabittir. Şekil 3.14, AO'nun spiral şeklindeki davranışını göstermektedir.

- Adım 3: Genişletilmiş sömürü kullanımı (X_3)

Bu adımda av alanını belirleyen kartal, iniş-saldırı için hazır olduğunda, av reaksiyonunu keşfetmek için bir ön saldırı ile dikey olarak aşağı inmektedir. Buna alçak uçuş denir. Hedefin seçilen bölgesini, avına yaklaşmak ve saldırmak için sergilediği davranışı Şekil 3.14 gösterir. Bu davranışın matematiksel ifadesi Denklem 3.18'de verilmektedir.

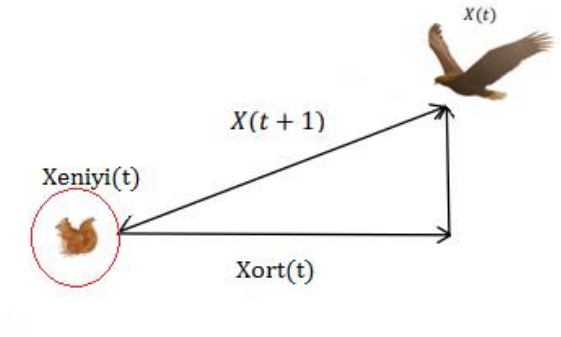
Şekil 3.14. AO'nın spiral şeklindeki davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)

$$X_3(t+1) = (X_{eniyi}(t) - X_M(t)) \times a - rand + ((\text{UB} - \text{LB}) \times \text{rand} + \text{LB}) \times \delta \quad (3.18)$$

Burada $X_3(t+1)$, üçüncü arama yöntemi (X_3) tarafından üretilen t'nin bir sonraki yinelemesinin çözümüdür. α ve δ, burada 0.1 gibi küçük bir değere sabitlenmiş sömürü ayarlama parametreleridir. LB, verilen problemin alt sınırını ve UB de üst sınırını belirtir.

- Adım 4: Daraltılmış kullanım (X_4)

Dördüncü adımda (X_4), kartal avına yaklaştığında, stokastik hareketlerine göre karada saldırır. Yürüyüş ve av yakalama adı verilen bu adım ile AO avına son konumda saldırır (Şekil 3.15). Bu davranışın matematiksel ifadesi Denklem 3.19'da görülmektedir.



Şekil 3.15. Kartal yavaş alçalma davranışı (Abualigah ve ark. 2021'den değiştirilerek alınmıştır)

$$X_4(t+1) = QF \times X_{eniyi}(t) - (G_1 \times X(t) \times rand) - G_2 \times Levy(D) + rand \times G_1 \quad (3.19)$$

Burada $X_4(t+1)$, dördüncü arama yöntemi (X_4) tarafından üretilen t'nin bir sonraki yinelemesinin çözümüdür. QF, Denklem 3.20 kullanılarak hesaplanmaktadır. Denklem 3.21 kullanılarak oluşturulan G_1, kaçırma sırasında avı izlemek için kullanılan AO'nun çeşitli hareketlerini gösterir. Denklem 3.21 kullanılarak oluşturulan G_2, ilk konumdan (1) son konuma (t) kaçış sırasında avı takip etmek için kullanılan AO'nun uçuş eğimini ifade eden, 2' den 0' a azalan bir değerdir. $X(t)$, t. yinelemesindeki mevcut çözümdür.

$$QF(t) = t^{\frac{2\times \text{rand}()-1}{(1-T)^2}} \quad (3.20)$$

$$G_1 = 2 \times rand() - 1 \quad (3.21)$$

$$G_2 = 2 \times (1 - \frac{t}{T}) \quad (3.21)$$

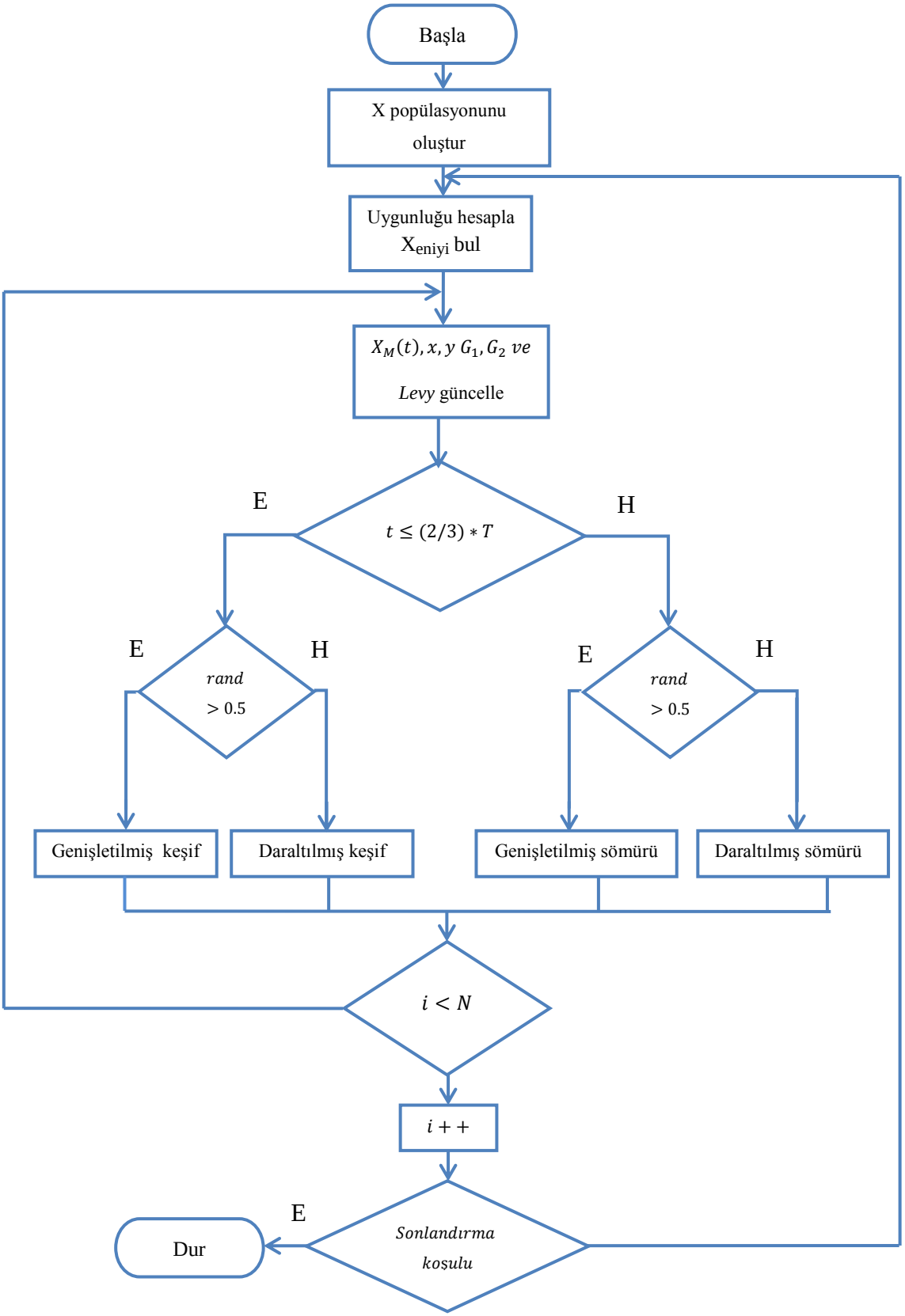
$QF(t)$, t. yinelemesindeki kalite fonksiyonu değeri ve $rand$ da 0 ile 1 arasında rastgele bir değerdir. AO popülasyon tabanlı bir algoritma olan; keşif aşaması, geniş çözüm uzayı, rastgele ve küresel olarak arama sürecini ifade eder.

AO'nun 4 farklı aşamada arama stratejisi tamamlanmış olur. Son olarak, sonlandırma kriteri karşılandığında AO'nun arama süreci tamamlanır. AO'nun sözde kodu, Algoritma 4 olarak aşağıda açıklanmıştır. Şekil 3.16'da AO algoritmasının akış diyagramı gösterilmiştir.

Algoritma 4. Aquila (Kartal)Optimizasyon Algoritması Sözde Kodu

Başlangıç Aşaması
X popülasyonunu başlat
AO parametreleri başlat
while$(iterasyon < maks\ iterasyon)$**do**
  Amaç fonksiyonu hesapla
  $X_{eniyi(t)}$ = Amaç fonksiyonuna göre elde edilen en iyi sonucu belirle
  for $(i = 1,2, \dots, n)$ **do**
    Şuan ki çözümün ortalama değerini $X_M(t)$ güncelle
    G_1, G_2 ve $Levy$ değerlerini güncelle
      if $t \leq \frac{2}{3} * T$ **then**
     if $rand \leq 0.5$ **then**
      Adım 1 : Genişletilmiş keşif $(\boldsymbol{X_1})$
      Denklem 3.13 kullanılarak mevcut çözüm güncellenir
      if $fitness(X_1(t+1)) < fitness(X(t))$ **then**
      $X(t) = (X_1(t+1))$
      if $fitness(X_1(t+1)) < fitness(X_{eniyi}(t))$ **then**
        $X_{eniyi}(t) = X_1(t+1)$
      end if
     end if
    else
      Adım 2 : Daraltılmış keşif $(\boldsymbol{X_2})$
      Denklem 3.15 kullanılarak mevcut çözüm güncellenir
      if $fitness(X_2(t+1)) < fitness(X(t))$ **then**
       $X(t) = (X_2(t+1))$
        if $fitness(X_2(t+1)) < fitness(X_{eniyi}(t))$ **then**
          $X_{eniyi}(t) = X_2(t+1)$
        end if

end if
end if
else
if $rand \leq 0.5$ **then**
Adım 3 : Genişletilmiş sömürü ($\boldsymbol{X_3}$)
Denklem 3.18 kullanılarak mevcut çözüm güncellenir
if $fitness(X_3(t+1)) < fitness(X(t))$ **then**
$X(t) = (X_3(t+1))$
if $fitness(X_3(t+1)) < fitness(X_{eniyi}(t))$ **then**
$X_{eniyi}(t) = X_3(t+1)$
end if
end if
else
Adım 4 : Daraltılmış sömürü ($\boldsymbol{X_4}$)
Denklem 3.19 kullanılarak mevcut çözüm güncellenir
if $fitness(X_4(t+1)) < fitness(X(t))$ **then**
$X(t) = (X_4(t+1))$
if $fitness(X_4(t+1)) < fitness(X_{eniyi}(t))$ **then**
$X_{eniyi}(t) = X_4(t+1)$
end if
end if
end if
end if
end for
end while
return en iyi çözüm X_{eniyi}



Şekil 3.16. Aquila (Kartal) optimizasyonu akış diyagramı

3.5. Martı Optimizasyon Algoritması (SOA)

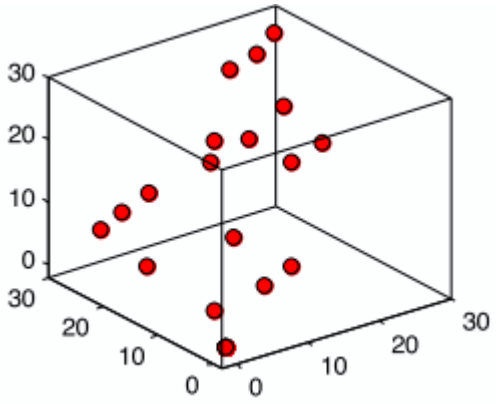
Martılar gezegenin her yerinde bulunabilen, farklı kütle ve uzunluklara sahip, çok çeşitli türleri olan; böcek, balık ve solucanlar ile beslenen gövdesi beyaz tüylerle kaplı olan zeki kuşlardır. Martılar avlanma sırasında değişik yöntemler kullanırlar. Örneğin; balık avlarken balıkları çekmek için ekmek kırıkları kullanırlar (Del Hoyo ve ark., 1992)

3.5.1. Matematiksel Model

- Göç (keşif)

Algoritma, göç sırasında martı grubunun bir konuma doğru nasıl hareket ettiğini simüle eder. Bu aşamada bir martı üç koşulu yerine getirmelidir:

• Çarpışmalardan kaçınma: Komşular (yani diğer martılar) arasındaki çarpışmayı önlemek için, yeni arama aracı konumunun hesaplanmasında ek bir değişken (A) kullanılır (Şekil 3.17).



Şekil 3.17. Arama aracıları arasında çarpışmadan kaçınma (Dhiman ve ark. 2018)

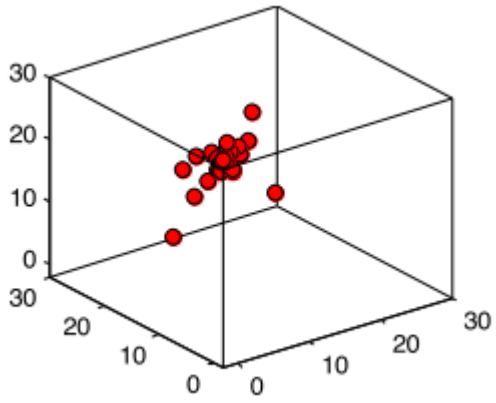
$$\vec{C}_s = A \times \vec{P}_s(x) \tag{3.22}$$

Burada $\vec{C}_s$, diğer arama aracı ile çarpışmayan arama aracısının konumunu; $\vec{P}_s$, arama aracısının mevcut konumunu; x, mevcut yinelemeyi ve A da belirli bir arama alanındaki arama aracısının hareketine ait davranışı temsil eder.

$$A = f_c - (x \times \left(\frac{f_c}{Maks_{iter}}\right)) \tag{3.23}$$

Burada : $x = 0, 1, 2, \ldots$, $Maks_{iter}$ olup f_c, doğrusal olarak f_c'den 0'a düşen A değişkenini kullanma sıklığını kontrol etmek için tanımlanmıştır. Bu çalışmada, f_c'nin değeri 2 olarak ayarlanmıştır.

• En iyi komşunun yönüne doğru hareket: Komşular arasındaki çarpışmadan kaçınıldıktan sonra, arama aracı en iyi komşu yönüne doğru hareket eder (Şekil 3.18).



Şekil 3.18. Arama aracıları en iyi komşuya doğru hareketi (Dhiman ve ark. 2018)

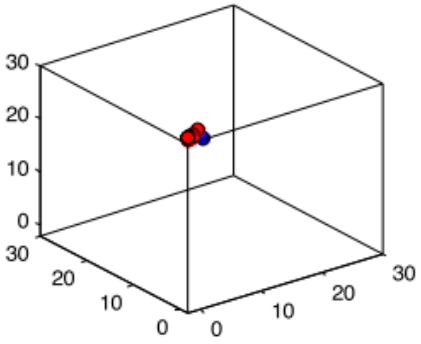
$$\vec{M}_s = B \times (\overrightarrow{P_{bs}}(x) - \vec{P}_s(x)) \tag{3.24}$$

Burada $\vec{M}_s$, en uygun arama aracı $\overrightarrow{P_{bs}}$'ye doğru arama yapan $\vec{P}_s$'lerin pozisyonlarını temsil eder. B keşif ve sömürü arasındaki dengeyi belirten bir kontol parametresidir. B Denklem 3.25 ile hesaplanır:

$$B = 2 \times A^2 \times rd \tag{3.25}$$

Burada rd rastgele bir sayıdır ve [0, 1] aralığında yer alır.

• En iyi arama aracına yakın kalma: Son olarak, arama aracısı, Şekil 3.19'da gösterilen en iyi arama aracısına göre konumunu güncelleyebilir.



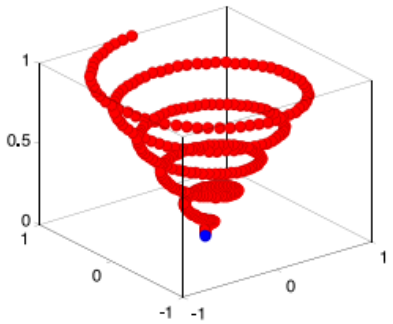
Şekil 3.19. En iyi arama aracısına yakınsama (Dhiman ve ark. 2018)

$$\vec{D}_s = \left|\vec{C}_s + \vec{M}_s\right| \tag{3.26}$$

Burada $\vec{D}_s$, arama aracısı ile en uygun arama aracısı arasındaki mesafeyi temsil eder (yani, uygunluk değeri daha az olan en iyi martı).

- Saldırı (sömürü) Aşaması

Geçmiş arama süreçlerinden yararlanarak saldırı sırasında hızını ve açısını sürekli değiştirebilir. Martıların belirledikleri bir ava saldırı sırasındaki spiral hareketi Şekil 3.20'de gösterilmiştir. Düzlem davranışları Denklem 3.27-3.30'daki denklemlerde açıklanmıştır.



Şekil 3.20. Martının ava spiral saldırı davranışı (Dhiman ve ark. 2018)

$$x' = r \times \cos(k) \tag{3.27}$$

$$y' = r \times \sin(k) \tag{3.28}$$

$$z' = r \times k \tag{3.29}$$

$$r = u \times e^{kv} \tag{3.30}$$

burada r, spiral şeklindeki her dönüşünün yarıçapıdır, k $[0 \leq k \leq 2\pi]$ aralığında rastgele bir sayıdır. u ve doğal logaritmik taban e üzerindeki v, spiral şeklini tanımlayan sabit değerlerdir. Arama aracının güncellenmiş konumu, Denklem 28 - 30 kullanılarak hesaplanır.

$$\overrightarrow{P_s}(x) = (\vec{D}_s \times x' \times \grave{y}' \times z') + \overrightarrow{P_{bs}}(x) \tag{3.31}$$

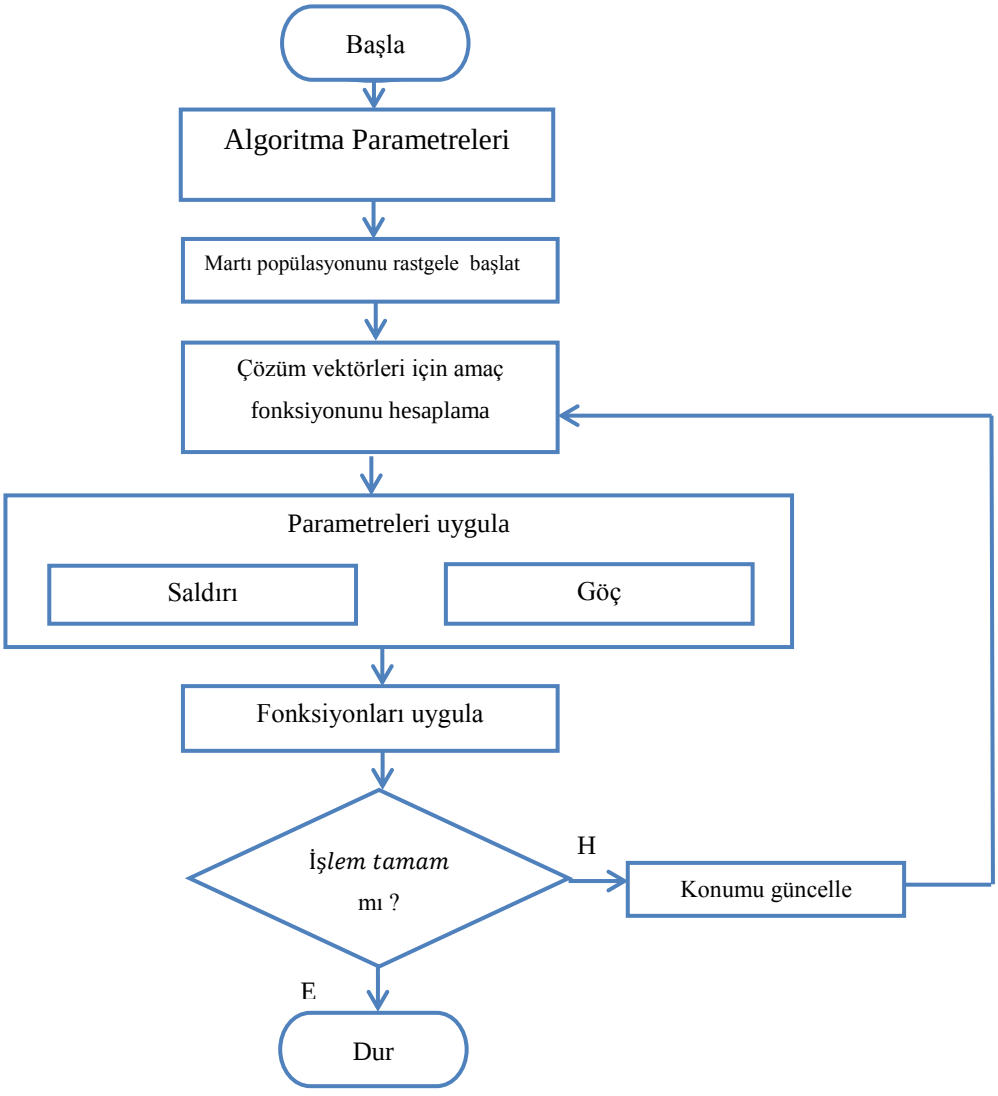
$\overrightarrow{P_s}(x)$, en iyi çözümü kayıt eder ve diğer arama aracılarının konumunu günceller. SOA çalışmaya rastgele oluşturulmuş bir popülasyon ile başlar ve en iyi arama aracısına göre konumunu günceller. A değişkenin değeri, her bir yinelemede, f_c'den 0'a doğru lineer olarak azalırken B değişkeni, göç (keşif) ve ava saldırı (sömürü) arasındaki geçiş dengesinin yumuşaklığından sorumludur.

Bu nedenle, SOA, daha iyi keşif ve kullanım kapasitesi nedeniyle küresel bir optimize edici olarak kabul edilir (Algoritma 5). Şekil 3.21'de SOA'nın Akış Diyagramı verilmektedir.

Algoritma 5. Martı Optimizasyon Algoritması Sözde Kodu

Girdi $\overrightarrow{P_s}$ martı popülasyonu
Çı**ktı** $\overrightarrow{P_{bs}}$ en iyi arama aracı
1. $A, B,$ ve Mak_{iter} parametrelerini ayarla
2. $f_c \leftarrow 2,$ u $\leftarrow 1, v \leftarrow 1$ ayarla
3. **while** $(x < Mak_{iter})$ **do**
4. $\overrightarrow{P_{bs}} \leftarrow \overrightarrow{P_s}$ uygunluk değeri hesapla
Göç − keşif davranışı
5. $rd \leftarrow$ rast(0,1)
6. $k \leftarrow$ rast(0,2π)
ava salıdırı − sömürü davranışı
7. $r \leftarrow u\ x\ e^{kv}$ spiral davranış
8. $\overrightarrow{D_s}$ mesafesini hesapla Denklem 3.26
9. $P \leftarrow x'\ xy\ \grave{}xz'$ Denklem(3.27-3.30) ile hesapla
10. $\overrightarrow{P_s}(x) \leftarrow \left(\vec{D}_s\ x\ P\right) + \overrightarrow{P_{bs}}(x)$
11. **end while**
12. return $\overrightarrow{P_{bs}}$
13. **basla** $\overrightarrow{P_s}$ prosedürü
14. **for** $i \leftarrow 1 : n$ **do**
15. $Fit_{s[i]} \leftarrow$ Her birey için uygunluk değeri hesapla
16. **end for**
17. $Fit_{s[eniyi]} \leftarrow$ En iyi uygunluk değeri hesapla

18. return $Fit_{s[eniyi]}$
19. **bitir**
20. **başla**
21. $eniyi \leftarrow Fit_{s[0]}$
22. **for** $i \leftarrow 1 : n$ **do**
23. **if** ($Fit_{s[i]} < eniyi$) **then**
24. $eniyi \leftarrow Fit_{s[i]}$
25. **end if**
26. **end for**
27. **return** *eniyi* sonucu döndür
28. **bitir**



Şekil 3.21. Martı optimizasyon algoritması akış diyagramı

3.6. Serçe Arama Algoritması (SSA)

3.6.1. Biyolojik Özellikler

Serçeler genellikle toplu halde yaşayan kuşlardır ve birçok türü vardır. İnsan hayatının olduğu yerlerde olmak üzere dünyanın her yerine dağılmışlardır. Dahası, omnivor kuşlardır ve çoğunlukla tahıl veya yabani ot tohumları ile beslenirler. Son derece zeki ve güçlü hafızalara sahip olan serçelerin türleri vardır (Barnard & Sibly, 1981) .

Üreticiler aktif olarak besin kaynağını ararlarken, tüketiciler üreticiler tarafından bulunan yiyeceklerden alırlar. Kuşlar davranış stratejilerini esnek bir şekilde kullandıkları için, üretici ve tüketici türleri arasında geçiş yapabilirler. Serçelerin yiyeceklerini bulmak için genellikle hem üreticinin hem de tüketicinin stratejisini kullandıkları söylenebilir. Grup olarak hareket ettikleri için birbirlerinden etkilenirler ve grup içerisindeki saldırganlar besin kaynaklarını avlanma için kullanırlar. (Lendvai ve ark., 2004).

Yırtıcı hayvanlar tarafından saldırıya uğrama ihtimalleri olduğu için sürekli bir pozisyon almaya ihtiyaç duyan serçeler tehlike alanlarını azaltmak için toplu halde bulunmaya özen gösterirler. Serçelerde doğal bir merak içgüdüsü olduğu ve bir tehlike algıladığı anda ise cıvıltı sesi çıkararak grubu uyarmaktalar ve tüm grup uçup gitmektedir.

3.6.2. Matematiksel Model ve Algoritma

Serçe arama algoritması matematiksel modeli, aşağıdaki maddelere göre oluşturulmaktadır. Serçelerin aşağıdaki davranışları idealize edilmiş olup ve bunlara karşılık gelen kuralları gösterilmiştir (Xue ve ark., 2020).

(1) Üreticiler yüksek düzeyde enerjiye sahiptir. Gıda kaynaklarının bulunabileceği alanları belirlerler. Enerji rezerv seviyesi, bireylerin uygunluk değerine bağlıdır.
(2) Serçelerin tehlikeyi fark ettikleri zaman cıvıldamaya başlayarak alarm seviyesini güvenlik eşiğinin üzerine çıkarıp üreticileri güvenli alana götürmeleri gerekir.
(3) Serçeler üretici veya tüketici olabilirler. Üretici ve tüketicilerin oranı popülasyonda değişmez.

(4) Enerjisi yüksek olan serçeler üretici olarak hareket ederler. Enerjisi düşük olan tüketiciler ise daha fazla yiyecek bulabilmek için uzak bölgelere uçma olasılıkları yüksektir.

(5) Tüketiciler, en iyi yiyeceği aramak için üreticiyi takip ederler. Bazıları ise avlanmasını arttırmak için üreticileri izleyip onların avlarından gözlerine kestirdiklerini avlamaya çalışırlar.

(6) Grup halinde gezerken tehlikeyi kenardaki serçeler fark ederler ve buna göre yeni konum belirlemeye çalışırlar. Ortadaki serçeler ise kenardakilere göre rastgele bir konum belirlerler.

Serçelerin konumu Denklem 3.32'deki matriste gösterilebilir:

$$X=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d} \end{bmatrix} \tag{3.32}$$

Burada n ile serçe sayısını ve d ile optimize edilecek değişkenlerin boyutu gösterilir. Ardından, tüm serçelerin uygunluk değeri Denklem 3.33'teki vektörle ifade edilebilir:

$$F_X=\begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d}]) \end{bmatrix} \tag{3.33}$$

Burada n, serçe sayısını gösterir ve F_X'teki her satırın değeri, bireyin uygunluk değerini temsil eder. SSA'da daha iyi uygunluk değerine sahip olan üreticiler, arama sürecinde yiyecek elde etme önceliğine sahiptir. Bu nedenle, üreticiler yiyecekleri tüketicilerden çok daha geniş bir yerde arayabilirler. Denklem 3.32 ve 3.33'e göre, her bir yineleme sırasında, üreticinin konumu Denklem 3.34'teki gibi güncellenir:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \times exp\left(\frac{-i}{\alpha \times iter_{maks}}\right) & eğer \;\; R_2 < ST \\ X_{i,j}^{t} + Q \times L & eğer \;\; R_2 \geq ST \end{cases} \tag{3.34}$$

t, geçerli yinelemeyi belirtir, $j = 1, 2, \ldots, d$. $X_{i,j}^{t}$ t. yinelemesinde i. serçenin j. boyutunun değerini temsil eder. $iter_{maks}$, en fazla sayıda yinelemeye sahip bir sabittir. $\alpha \in (0, 1]$ rastgele bir sayıdır. R_2 ($R_2 \in [0, 1]$) ve ST ($ST \in [0.5,\ 1.0]$) sırasıyla alarm değeri ve güvenlik eşiği, Q ise normal dağılımın rastgele bir sayısıdır. L, içindeki her bir elemanın bir olduğu $1 \times d'lik$ bir matrisi gösterir. Üretici yırtıcı olmadığı zaman $R_2 < ST$ geniş arama moduna girer. $R_2 \geq ST$ ise serçelerin yırtıcıyı keşfettiği anlamına gelir ve serçelerin güvenli bölgeye uçmaları gerekir.

Tüketiciler için Denklem 3.34-3.35'in uygulaması gerekir. Üreticilerin bir yemek bulduğunu gözlemleyen tüketiciler mevcut konumlarını bırakıp üreticilerle mücadele ederler. Kazanırlarsa yemeği alırlar yoksa Denklem 3.35'i uygulamaya devam ederler. Tüketiciler için konum güncelleme, formülü Denklem 3.35 ile verilmektedir.

$$X_{i,j}^{t+1} = \begin{cases} Q \times exp\left(\frac{X_{enkötü}^{t} - X_{i,j}^{t}}{i^2}\right) & eğer\ i > n/2 \\ X_P^{t+1} + \left|X_{i,j}^{t} - X_P^{t+1}\right| \times A^{+} \times L & değilse \end{cases} \tag{3.35}$$

Burada X_P, üreticinin işgal ettiği en uygun konumdur. En kötü konum $X_{enkötü}$ ile gösterilir. A, içindeki her bir öğenin rastgele 1 veya −1 olarak atandığı ve $A = A^T (A\,A^T)^{-1}$ olduğu $1 \times d$'lik bir matrisi temsil eder. $i > \frac{n}{2}$ olduğunda, daha kötü uygunluk değerine sahip i. tüketici'nin açlıktan ölme olasılığı yüksektir. İlk konumlar popülasyonda rastgele oluşur.

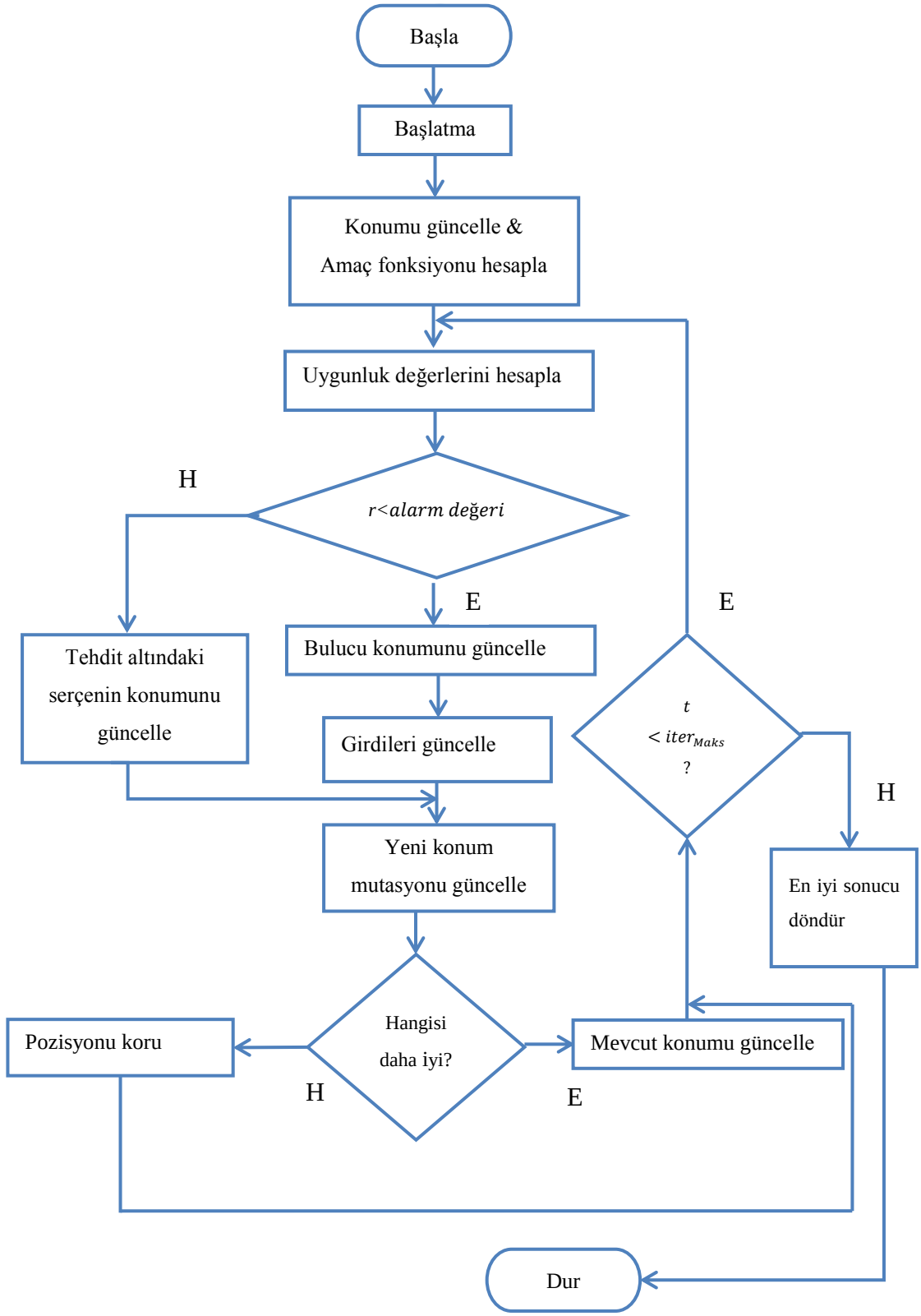
$$X_{i,j}^{t+1} = \begin{cases} X_{eniyi}^{t} + \beta \times \left|X_{i,j}^{t} - X_{eniyi}^{t}\right| & eğer\ f_i > f_g \\ X_{i,j}^{t} + K \times \left(\frac{\left|X_{i,j}^{t} - X_{enkötü}^{t}\right|}{(f_i - f_w) + \varepsilon}\right) & eğer\ f_i = f_g \end{cases} \tag{3.36}$$

Denklem 3.36'ya göre, matematiksel model X_{eniyi} mevcut küresel optimal konum olduğu durumda olabilir. β, adım boyutu kontrol parametresi olarak, ortalama değeri 0 ve varyansı 1 olan rastgele sayıların normal dağılımıdır. $K \in [-1, 1]$ rastgele bir sayıdır. Burada f_i mevcut serçenin uygunluk değeridir. f_g ve f_w sırasıyla serçenin en iyi ve en kötü uygunluk değerleridir. ε, sıfır bölme hatasını önlemek için küçük bir sabittir.

X_{eniyi}, popülasyon merkezinin konumunu temsil ederken çevresinin de güvenli olduğunu gösterir. $f_i = f_g$, grubun ortasındakilerin saldırının farkında olduğunu ve diğerlerinin gruba yaklaşmaları gerektiğini gösterir. K, serçenin hareket ettiği yönü belirtir ve aynı zamanda adım büyüklüğü kontrol katsayısıdır. Yukarıdaki modelin uygulanabilirliği ve SSA'nın sözde kodu Algoritma 6'da gösterilmiştir. Şekil 3.22'de SSA algoritması akış diyagramı verilmektedir.

Algoritma 6. Serçe Arama Algoritması Sözde Kodu

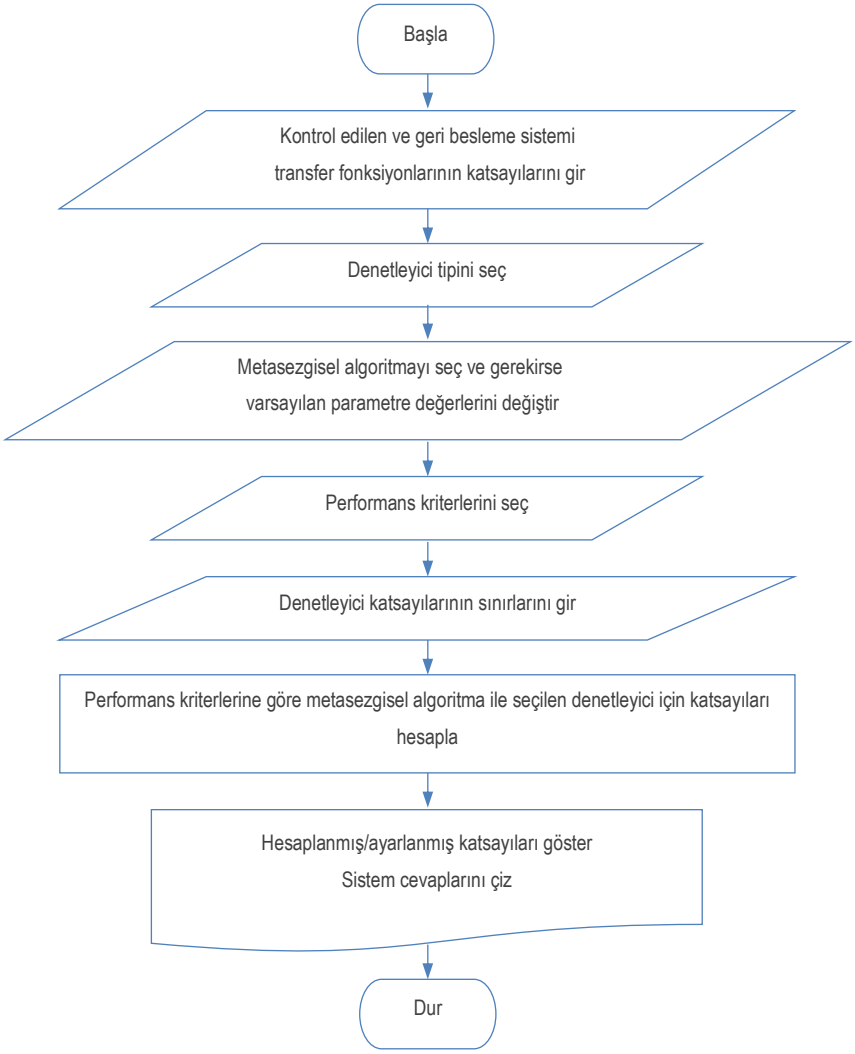
1. Maksimum yineleme sayısı $iter_{maks}$
2. P_d: üretici sayısı
3. Tehlike altındaki serçe sayısı S_d
4. Alarm değerini ayarla R_2
5. n: serçe sayısı
6. N başlatma durumunu ayarla
Çıktı: X_{eniyi}, f_g
7. **while** $(t < iter_{maks}\ )$
8. Uygunluk değerine göre en iyi en kötü değeri belirle
9. $R_2 = rand(1)$
10. **for** $i = 1 : P_d$
11. Serçe konumu güncelle(Denklem 3.34)
12. **end for**
13. **for** $i = (1 + P_d) : n$
14. Serçe konumu güncelle(Denklem 3.35)
15. **end for**
16. **for** $i = 1 : S_d$
17. Serçe konumu güncelle(Denklem 3.36)
18. **end for**
19. Mevcut yeni konumu al
20. Yeni konum öncekinden daha iyi ise güncelle
21. $t = t + 1$
22. **end while**
23. **return** X_{eniyi}, f_g



Şekil 3.22. Serçe arama algoritması akış diyagramı

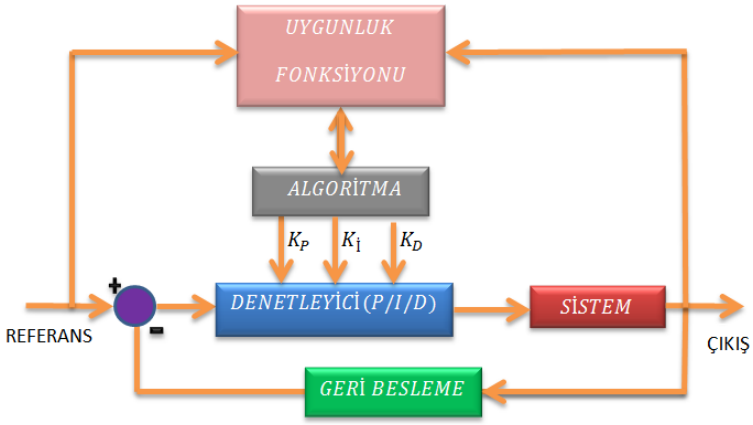
4. BULGULAR ve TARTIŞMA

Bu bölümde, seçilen algoritmaların etkinliğini göstermek için İntel® Core™ İ5 CPU M480 2.6GHz işlemcili bilgisayar kullanılarak farklı sistemler için MATLAB ortamında simülasyon uygulamaları yapılmıştır. Kullanılan algoritmaların çalışmasındaki süreç Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Tasarlanan yazılım aracındaki süreç için temel akış şeması

PID parametreleri, Şekil 4.2'de gösterildiği gibi farklı metasezgisel algoritma türleri ve farklı performas kriterlerine göre optimize edilebilir.



Şekil 4.2. PID parametrelerinin bulunmasındaki akış şeması

Birinci uygulama olarak Denklem 4.1'deki (Dorf & Bishop, 2011) transfer fonksiyonuna sahip sistem için PID denetleyici tasarımı gerçekleştirilmektedir. Kullanılan metasezgisel algoritmaların parametreleri Çizelge 4.1'deki gibi olup farklı performans kriterlerine göre elde edilen karşılaştırmalı sonuçlar sırasıyla Çizelge 4.2-4.5'te verilmektedir. Ayrıca karşılaştırmalı birim basamak cevapları da sırasıyla Şekil 4.3 - 4.6'da gösterilmektedir. Bütün algoritmalarda maksimum iterasyon sayısı 20, alt sınır $K_p = 0.1$ ve üst sınır $K_p = 50$ olarak seçilmiştir.

$$G(s) = \frac{1}{s^2+2s+12}, H(s) = 1 \qquad (4.1)$$

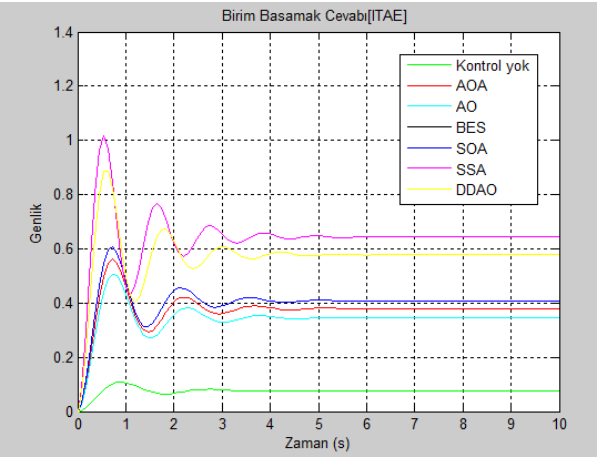
Çizelge 4.1. Sistem-1 için algoritma parametre ayarlaması

AOA	DDAO	BES	AO	SOA	SSA
Maksimum iterasyon sayısı: 20					
Çözüm no: 20	Maksimum alt yineleme: 20 $T_0 = 2000$ $\alpha = 0.995$ $N_{pop} = 3$	$N_{pop} = 10$	Çözüm no: 20	Arama aracı: 30	Arama aracı: 30

Çizelge 4.2. Sistem-1 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar

ITAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	7.3327	16.457	21.7012	6.328	6.2614	22.147
Maksimum aşım (%)	48.007	54.9116	57.7209	**46.8758**	48.6637	58.412
Yükselme süresi (s)	0.2826	0.2246	**0.2051**	0.2919	0.2744	0.2124
Yerleşme süresi (s)	3.8289	**3.7441**	3.9068	3.8971	3.7606	3.9474
Kararlı hal hatası	0.6207	0.4215	**0.3561**	0.6547	0.5923	0.3671
Uygunluk fonksiyonu	0.15181	0.1687	**0.13562**	0.15181	0.15191	0.1475
Hesaplama süresi (s)	**73.543**	114.421	100.947	137.182	104.051	178.593

Çizelge 4.2'de görüldüğü gibi Sistem-1 için ITAE performans kriterine göre en az maksimum aşım AO, en kısa yerleşme süresi DDAO, en kısa yükselme süresi ve en düşük kararlı hal hatası BES ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine BES algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

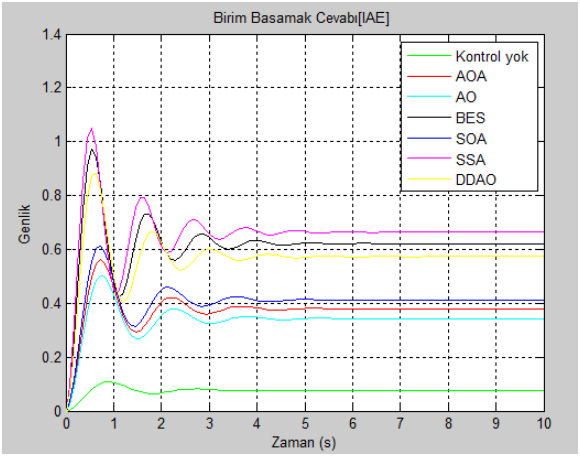


Şekil 4.3. Sistem-1 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.3. Sistem-1 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar

IAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	7.3303	16.12	19.7012	6.2579	8.3828	23.7012
Maksimum aşım (%)	48.0048	54.6863	56.536	**46.7797**	48.7255	58.2809
Yükselme süresi (s)	0.2826	0.2261	0.2121	0.2925	0.2734	**0.1986**
Yerleşme süresi (s)	3.8291	3.7627	**3.576**	3.9017	3.7516	3.8323
Kararlı hal hatası	0.6208	0.4267	0.3785	0.6572	0.5888	**0.3361**
Uygunluk fonksiyonu	15.1812	14.47	13.562	**11.784**	15.1916	13.745
Hesaplama süresi (s)	**73.142**	115.253	101.347	138.081	103.270	176.548

Çizelge 4.3'te görüldüğü gibi Sistem-1 için IAE performans kriterine göre en az maksimum aşım AO, en kısa yerleşme süresi BES, en kısa yükselme süresi ve en düşük kararlı hal hatası SSA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine AO algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

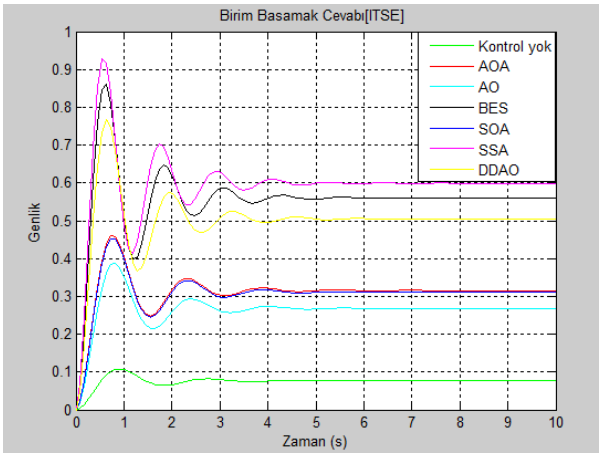


Şekil 4.4. Sistem-1 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.4. Sistem-1 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar

ITSE	AOA	DDAO	BES	AO	SOA	SSA
K_p	5.5331	12.23	15.27	4.3975	5.4223	17.94
Maksimum aşım (%)	46.1374	52.0779	54.0236	**44.788**	46.0363	55.589
Yükselme süresi (s)	0.2997	0.2459	0.2296	0.3124	0.3008	**0.2187**
Yerleşme süresi (s)	3.9359	3.916	3.8081	**3.4906**	3.9375	3.6659
Kararlı hal hatası	0.6844	0.4952	0.4401	0.7318	**0.4008**	0.6888
Uygunluk fonksiyonu	0.00292	0.003214	0.0035863	0.002914	**0.002901**	0.0039014
Hesaplama süresi (s)	**73.571**	114.450	100.971	137.302	145.245	174.29

Çizelge 4.4'te görüldüğü gibi Sistem-1 için ITSE performans kriterine göre, en düşük kararlı hal hatası SOA, en kısa yükselme süresi SSA, en kısa yerleşme süresi ve en az maksimum aşım AO ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SOA algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

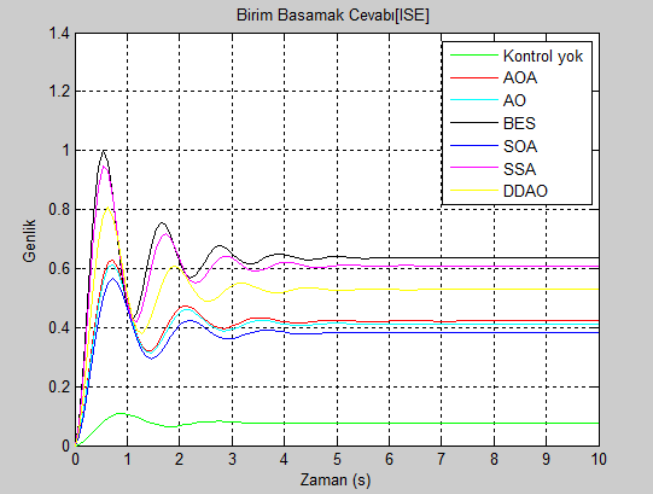


Şekil 4.5. Sistem-1 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.5. Sistem-1 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar

ISE	AOA	DDAO	BES	AO	SOA	SSA
K_p	8.7552	13.45	21.0372	8.4044	7.411	18.703
Maksimum aşım (%)	49.1879	52.9163	57.3995	48.7537	**48.0764**	55.77
Yükselme süresi (s)	0.2703	0.2384	**0.2074**	0.2732	0.2818	0.2158
Yerleşme süresi (s)	3.7248	3.8996	3.9223	3.7501	3.8236	**3.6266**
Kararlı hal hatası	0.5782	0.4715	**0.3633**	0.5881	0.6182	0.3908
Uygunluk fonksiyonu	3.0654	3.457	3.58637	2.9112	**2.901**	3.5864
Hesaplama süresi (s)	**77.129**	113.272	102.118	137.896	143.574	176.128

Çizelge 4.5'te görüldüğü gibi Sistem-1 için ISE performans kriterine göre en az maksimum aşım SOA, en kısa yerleşme süresi SSA, en kısa yükselme süresi ve en düşük kararlı hal hatası BES ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SOA algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.



Şekil 4.6. Sistem-1 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları

İkinci uygulama olarak Denklem 4.2'deki (Dorf & Bishop, 2011) transfer fonksiyonuna sahip sistem için PID denetleyici tasarımı gerçekleştirilmektedir.

Kullanılan metasezgisel algoritmaların parametreleri Çizelge 4.6'daki gibi olup farklı performans kriterlerine göre elde edilen karşılaştırmalı sonuçlar sırasıyla Çizelge 4.7-4.10'da verilmektedir. Ayrıca karşılaştırmalı birim basamak cevapları da sırasıyla Şekil 4.7 - 4.10'da gösterilmektedir.

$$G(s) = \frac{0.5}{0.00077s^2 + 0.009907s + 0.2501}, H(s) = 1 \qquad (4.2)$$

Çizelge 4.6. Sistem-2 için algoritma parametre ayarlaması

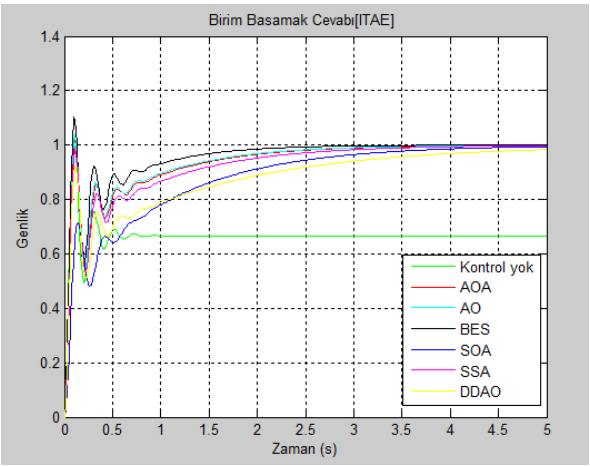
AOA	DDAO	BES	AO	SOA	SSA
Maksimum iterasyon sayısı: 20					
Çözüm no: 20	Maksimum alt yineleme: 20 $T_0 = 2000$ $\alpha = 0.995$ $N_{pop} = 3$	$N_{pop} = 10$	Çözüm no: 20	Arama aracı: 30	Arama aracı: 30

Bütün algoritmalarda maksimum iterasyon sayısı 20, alt sınır K_p=0.1, K_i=0.1 ve üst sınır K_p=1.5, K_i=2.5 olarak seçilmiştir.

Çizelge 4.7. Sistem-2 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar

ITAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.87863	0.741	1.0363	0.91472	0.41394	0.88169
K_i	1.5972	0.785	2.3268	1.6738	0.82647	1.3144
Maksimum aşım (%)	2.715	**0**	10.3102	4.3489	**0**	**0**
Yükselme süresi (s)	0.0634	0.0818	**0.0542**	0.0603	1.8219	0.0687
Yerleşme süresi (s)	2.4518	4.6917	**1.8011**	2.3787	3.6118	2.8465
Kararlı hal hatası	9.9790e-04	0.0164	**1.463e-04**	8.579e-04	0.0056	0.0022
Uygunluk fonksiyonu	0.3254	0.4325	**0.2874**	0.4278	0.7481	0.6547
Hesaplama süresi (s)	**69.106**	105.724	96.813	147.972	97.881	167.684

Çizelge 4.7'de görüldüğü gibi Sistem-2 için ITAE performans kriterine göre en az maksimum aşım DDAO, SOA ve SSA ile elde edilmiştir. En kısa yükselme süresi, en kısa yerleşme süresi ve en düşük kararlı hal hatası BES algoritmasında görülmektedir. Ayrıca en düşük uygunluk fonksiyonu değerine BES algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

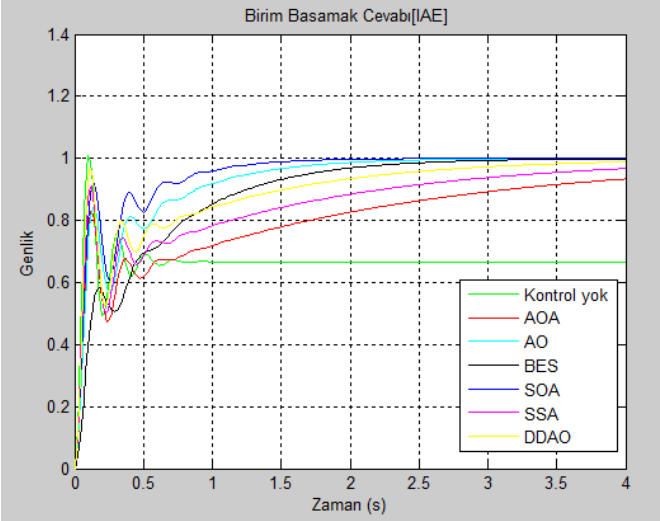


Şekil 4.7. Sistem-2 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.8. Sistem-2 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar

IAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.59882	0.796	1.0813	0.48619	0.52714	0.72508
K_i	0.52142	1.132	0.22472	1.7334	2.4643	0.76914
Maksimum aşım (%)	0	0	0	0	0	0
Yükselme süresi (s)	6.4918	**0.0712**	1.2163	0.8544	0.0973	0.0854
Yerleşme süresi (s)	3.1236	3.3428	2.2874	1.7753	**1.283**	4.7943
Kararlı hal hatası	0.0663	0.0112	0.0014	3.466e-04	**2.097e-05**	0.0330
Uygunluk fonksiyonu	3.7666	3.1524	2.7485	**1.1587**	1.347	3.2784
Hesaplama süresi (s)	**68.073**	103.258	94.243	134.443	98.217	167.471

Çizelge 4.8'de görüldüğü gibi Sistem-2 için IAE performans kriterine göre en kısa yükselme süresi DDAO, en kısa yerleşme süresi ve en düşük kararlı hal hatasına SOA algoritması ile ulaşılmıştır. Ayrıca en düşük uygunluk fonksiyonu değerine AO algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

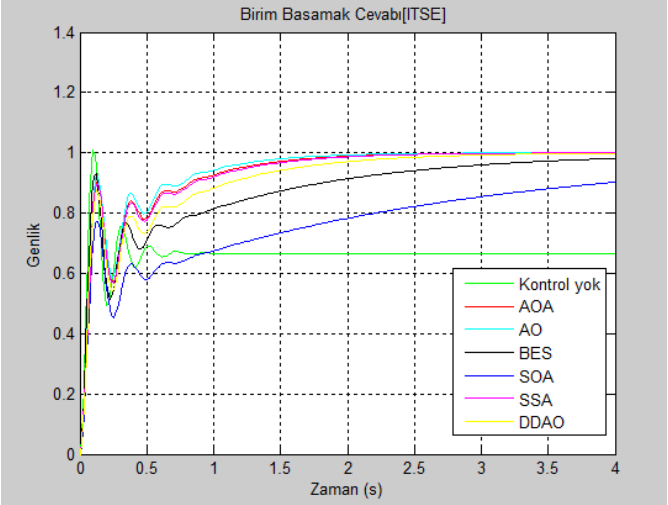


Şekil 4.8. Sistem-2 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.9. Sistem-2 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar

ITSE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.56859	0.579	0.74275	0.58913	0.53249	0.57304
K_i	1.9394	1.456	0.94822	2.1829	0.4001	1.8492
Maksimum aşım (%)	0	0	0	0	0	0
Yükselme süresi (s)	0.1043	1.0802	0.0911	**0.0805**	3.9137	0.1089
Yerleşme süresi (s)	1.6966	2.2802	**1.5043**	3.9094	7.9526	1.7738
Kararlı hal hatası	2.7986e-04	0.0019	0.0187	**1.0883e-04**	0.0976	3.8643e-04
Uygunluk fonksiyonu	0.002512	0.1240	0.1147	0.003547	0.541	**0.002374**
Hesaplama süresi (s)	**69.583**	107.542	96.077	139.672	98.193	168.975

Çizelge 4.9'da görüldüğü gibi Sistem-2 için ITSE performans kriterine göre en kısa yerleşme süresi BES, en kısa yükselme süresi ve en düşük kararlı hal hatası AO ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SSA algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

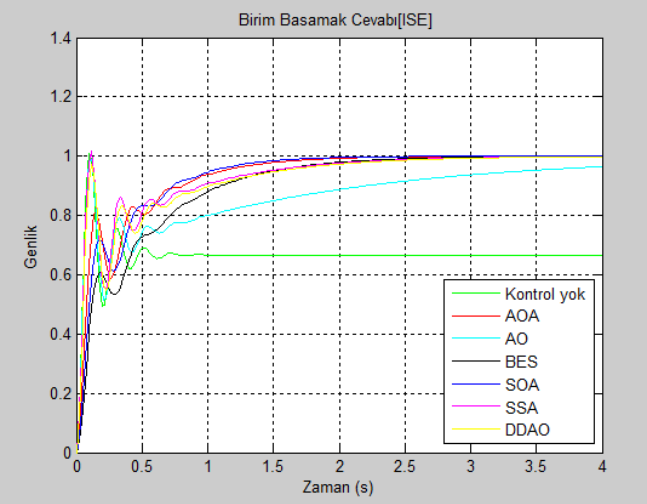


Şekil 4.9. Sistem-2 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.10. Sistem-2 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar

ISE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.40258	0.745	0.21514	0.89378	0.26687	0.8838
K_i	1.8747	1.642	2.2266	0.79343	1.8105	1.7299
Maksimum aşım (%)	0	0	0	0	0	0
Yükselme süresi (s)	0.7901	0.0733	1.0623	**0.0657**	0.6961	0.0661
Yerleşme süresi (s)	1.5338	2.2252	2.0183	4.9905	**1.3905**	2.1297
Kararlı hal hatası	9.722e-05	0.0018	6.071e-04	0.0354	**2.869e-05**	0.0015
Uygunluk fonksiyonu	**4.257**	7.8541	6.3274	8.1246	4.9342	7.2571
Hesaplama süresi (s)	**69.357**	108.323	95.081	130.891	97.203	167.572

Çizelge 4.10'da görüldüğü gibi Sistem-2 için ITSE performans kriterine göre en kısa yükselme süresi AO, en kısa yerleşme süresi ve en düşük kararlı hal hatası SOA ile elde edilmiştir. Ayrıca AOA algoritması en düşük uygunluk fonksiyonu değerine ve en kısa hesaplama süresine sahiptir.



Şekil 4.10. Sistem-2 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları

Üçüncü uygulama olarak Denklem 4.3'teki (Dorf & Bishop, 2011) transfer fonksiyonuna sahip sistem için PID denetleyici tasarımı gerçekleştirilmektedir. Kullanılan metasezgisel algoritmaların parametreleri Çizelge 4.11'deki gibi olup farklı performans kriterlerine göre elde edilen karşılaştırmalı sonuçlar sırasıyla Çizelge 4.12-4.15'te verilmektedir. Ayrıca karşılaştırmalı birim basamak cevapları da sırasıyla Şekil 4.11-4.14'te gösterilmektedir.

$$G(s) = \frac{2}{s^3+5s^2+4s},\ H(s) = 1 \tag{4.3}$$

Çizelge 4.11. Sistem-3 için algoritma parametre ayarlaması

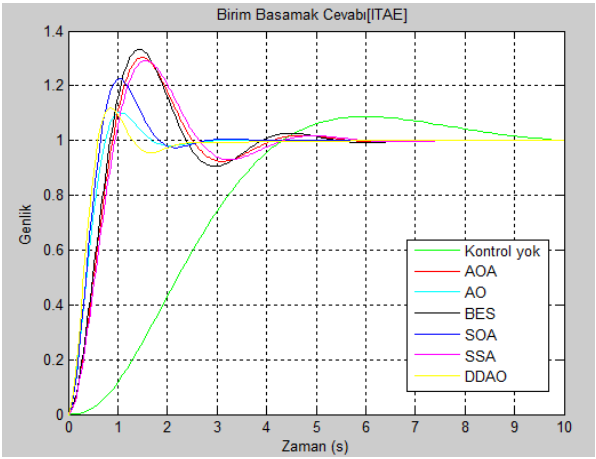
AOA	DDAO	BES	AO	SOA	SSA
Maksimum iterasyon sayısı: 20					
Çözüm no: 20	Maksimum alt yineleme: 20 $T_0 = 2000$ $\alpha = 0.995$ $N_{pop} = 3$	$N_{pop} = 10$	Çözüm no: 20	Arama aracı: 30	Arama aracı: 30

Bütün algoritmalarda maksimum iterasyon sayısı 20, alt sınır K_p=0.1, K_d=0.1 ve üst sınır K_p=16.7, K_d=8.7 olarak seçilmiştir.

Çizelge 4.12. Sistem-3 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar

ITAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	7.4287	6.4551	8.5809	5.9529	9.246	6.598
K_d	3.102	8.742	3.22614	6.5108	6.4046	2.87863
Maksimum aşım (%)	30.113	11.8102	33.3196	**10.1613**	22.4125	29.1373
Yükselme süresi (s)	0.6101	**0.4057**	0.5781	0.49103	0.4423	0.644
Yerleşme süresi (s)	3.8678	2.1772	4.8515	**1.5533**	2.3786	4.0447
Kararlı hal hatası	1.9169e-04	5.7981e-05	5.446e-06	4.383e-06	**7.4678e-08**	2.4402e-04
Uygunluk fonksiyonu	0.8147	1.2691	0.6547	0.3257	0.7895	**0.2987**
Hesaplama süresi (s)	**73.403**	121.393	103.577	146.253	104.529	178.771

Çizelge 4.12'de görüldüğü gibi Sistem-3 için ITAE performans kriterine göre en düşük kararlı hal hatası SOA, en kısa yükselme süresi DDAO, en kısa yerleşme süresi ve en az maksimum aşım AO ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SSA algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

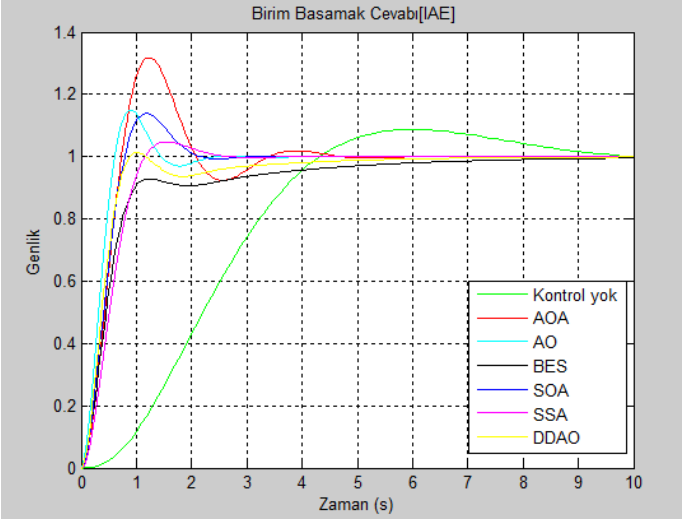


Şekil 4.11. Sistem-3 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.13. Sistem-3 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar

IAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	9.703	3.741	2.4456	7.5638	6.4816	4.1032
K_d	3.102	8.742	3.22614	6.5108	6.4046	2.87863
Maksimum aşım (%)	31.804	8.7419	**0**	14.927	13.8184	4.6851
Yükselme süresi (s)	0.4979	2.7588	0.8154	**0.4084**	0.5320	0.7419
Yerleşme süresi (s)	3.1977	3.9298	6.0168	2.029	**1.9254**	2.0962
Kararlı hal hatası	2.1635e-06	0.0010	0.0043	2.5179e-06	3.133e-07	**7.683e-08**
Uygunluk fonksiyonu	7.3065	8.1794	9.3241	7.1654	6.3287	**5.9624**
Hesaplama süresi (s)	**63.023**	120.547	100.576	137.812	103.600	178.171

Çizelge 4.13'te görüldüğü gibi Sistem-3 için IAE performans kriterine göre en az maksimum aşım BES, en kısa yükselme süresi AO, en kısa yerleşme süresi SOA ve en düşük kararlı hal hatası SSA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SSA algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

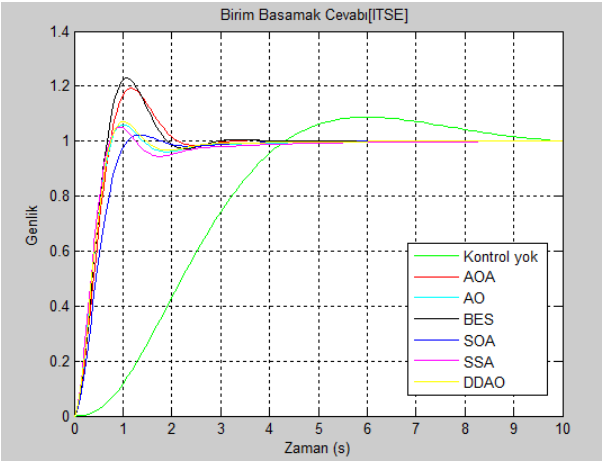


Şekil 4.12. Sistem-3 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.14. Sistem-3 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar

ITSE	AOA	DDAO	BES	AO	SOA	SSA
K_p	7.6648	5.173	9.0802	4.8962	3.9919	4.5444
K_d	5.3782	6.87	5.825	6.9265	4.0256	7.6222
Maksimum aşım (%)	19.1973	7.0138	23.0035	5.9582	**2.3161**	5.2233
Yükselme süresi (s)	0.5074	0.4992	**0.4585**	0.5032	0.6829	0.4769
Yerleşme süresi (s)	1.9717	2.3909	2.5302	2.5057	**1.4623**	2.948
Kararlı hal hatası	6.442e-08	7.165e-05	**6.007e-08**	1.357e-04	4.821e-05	4.610e-04
Uygunluk fonksiyonu	0.00573	0.01351	**0.00529**	0.02487	0.01478	0.01581
Hesaplama süresi (s)	**75.122**	121.520	140.917	138.173	128.425	179.451

Çizelge 4.14'te görüldüğü gibi Sistem-3 için ITSE performans kriterine göre en az maksimum aşım ve en kısa yerleşme süresi SOA algoritması ile elde edilmiştir. Ayrıca en kısa yükselme süresi, en düşük kararlı hal hatası ve en düşük uygunluk fonksiyonu değerine BES algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

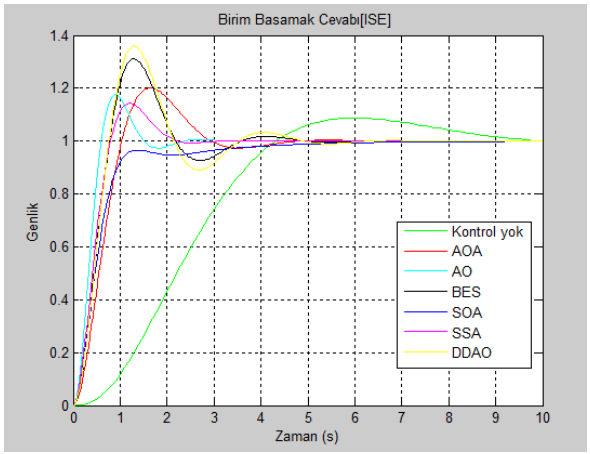


Şekil 4.13. Sistem-3 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.15. Sistem-3 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar

ISE	AOA	DDAO	BES	AO	SOA	SSA
K_p	5.6345	9.754	13.5283	8.4402	3.0521	6.4676
K_d	3.06656	3.78	2.05921	8.0774	4.8249	5.31953
Maksimum aşım (%)	20.0624	35.9302	31.0488	17.5374	**0**	14.2566
Yükselme süresi (s)	0.7056	0.5183	0.5256	**0.4029**	0.7784	0.5397
Yerleşme süresi (s)	3.9407	4.4925	3.3589	2.0125	3.9528	**1.9813**
Kararlı hal hatası	2.0704e-05	1.7842e-04	4.6641e-05	3.3537e-07	7.5006e-04	**2.0525e-07**
Uygunluk fonksiyonu	2.3252	1.9547	1.6463	1.8524	**1.3147**	2.1471
Hesaplama süresi (s)	**67.394**	119.572	101.145	140.429	103.757	197.521

Çizelge 4.15'te görüldüğü gibi Sistem-3 için ISE performans kriterine göre en az maksimum aşım SOA, en kısa yükselme süresi AO, en kısa yerleşme süresi ve en düşük kararlı hal hatası SSA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SOA algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.



Şekil 4.14. Sistem-3 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları

Dördüncü uygulama olarak Denklem 4.4'teki (Jayachandran & Ashok, 2013) transfer fonksiyonuna sahip sistem için PID denetleyici tasarımı gerçekleştirilmektedir. Kullanılan metasezgisel algoritmaların parametreleri Çizelge 4.16'daki gibi olup farklı performans kriterlerine göre elde edilen karşılaştırmalı sonuçlar sırasıyla Çizelge 4.17-4.21'de verilmektedir. Ayrıca karşılaştırmalı birim basamak cevapları da sırasıyla Şekil 4.15-4.19'da gösterilmektedir.

$$G(s) = \frac{2.4767}{(s+0.0476)(s+1)(s+5)}, H(s) = 1 \quad (4.4)$$

Çizelge 4.16. Sistem-4 için algoritma parametre ayarlaması

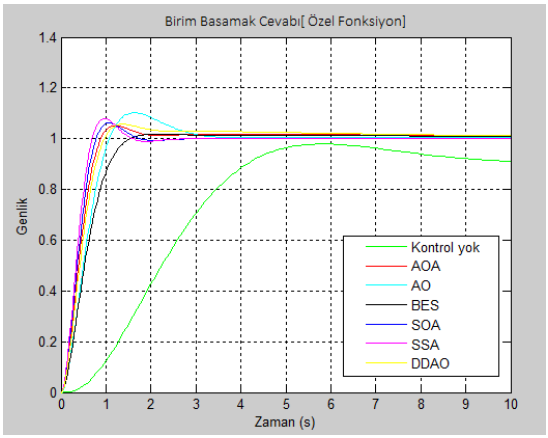
AOA	DDAO	BES	AO	SOA	SSA
Maksimum iterasyon sayısı: 20					
Çözüm no: 20	Maksimum alt yineleme: 20 $T_0 = 2000$ $\alpha = 0.995$ $N_{pop} = 3$	$N_{pop} = 10$	Çözüm no: 20	Arama aracı: 30	Arama aracı: 30

Bütün algoritmalarda maksimum iterasyon sayısı 20, alt sınır K_p=0.1, K_i=0.1, K_d=0.1 ve üst sınır K_p=7, K_i=1, K_d=7 olarak seçilmiştir.

Çizelge 4.17. Sistem-4 için özel fonksiyon tasarım kriterine göre karşılaştırmalı sonuçlar

Özel	AOA	DDAO	BES	AO	SOA	SSA
K_p	5.4031	5.0991	3.6997	5.1314	6.1245	6.8141
K_i	0.5726	0.64885	0.26871	0.30707	0.28874	0.31519
K_d	5.4031	4.7851	3.4858	3.4656	6.0552	6.8152
Maksimum aşım (%)	4.1325	5.7153	1.911	10.2229	6.2582	**1.4044**
Yükselme süresi (s)	0.5791	0.6292	0.8917	0.7259	0.5179	**0.4631**
Yerleşme süresi (s)	1.539	6.4705	**1.3645**	2.9161	1.5214	7.9004
Kararlı hal hatası	0.0102	0.0117	0.0090	0.0033	4.2693e-04	**1.9827e-04**
Uygunluk fonksiyonu	0.57261	0.96623	**0.2536**	0.75713	0.487	0.26379
Hesaplama süresi (s)	17.156	21.266	27.625	23.544	**16.441**	32.548

Çizelge 4.17'de görüldüğü gibi Sistem-4 için Özel performans kriterine göre en az maksimum aşım, en kısa yükselme süresi ve en düşük kararlı hal hatası SSA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine ve en kısa yerleşme süresine BES algoritmasıyla ulaşılırken SOA en kısa hesaplama süresine sahiptir.

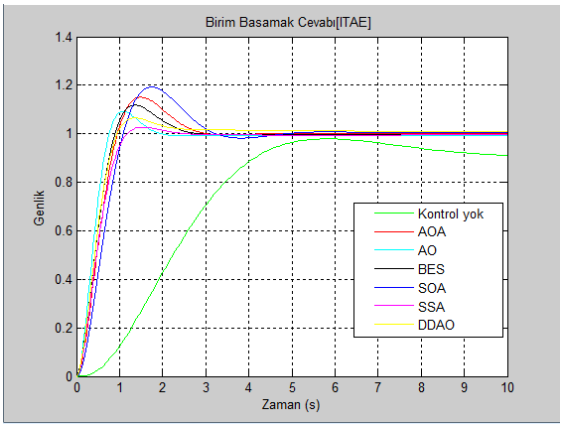


Şekil 4.15. Sistem-4 Özel fonksiyon kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.18. Sistem-4 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar

ITAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	6.1507	5.3573	6.304	6.8821	5.5293	4.5032
K_i	0.3337	0.4621	0.248	0.11855	0.35557	0.14487
K_d	3.5645	4.691	4.2578	5.7624	2.58	4.1105
Maksimum aşım (%)	15.0727	6.4898	11.7646	9.1928	19.2656	**2.7025**
Yükselme süresi (s)	0.6521	0.6242	0.6065	**0.5091**	0.7469	0.7363
Yerleşme süresi (s)	2.6441	2.4065	2.3382	**1.6887**	3.015	1.8361
Kararlı hal hatası	0.00169	0.0079	**0.00161**	0.0075	0.0037	0.0049
Uygunluk fonksiyonu	1.346	**1.081**	2.154	1.7038	1.476	2.142
Hesaplama süresi (s)	77.590	**75.123**	104.516	163.675	107.547	140.983

Çizelge 4.18'de görüldüğü gibi Sistem-4 için ITAE performans kriterine göre en az maksimum aşım SSA, en düşük kararlı hal hatası BES, en kısa yükselme süresi ve en kısa yerleşme süresi AO ile elde edilmiştir. Ayrıca DDAO en düşük uygunluk fonksiyonu değerine ve en kısa hesaplama süresinede sahiptir.

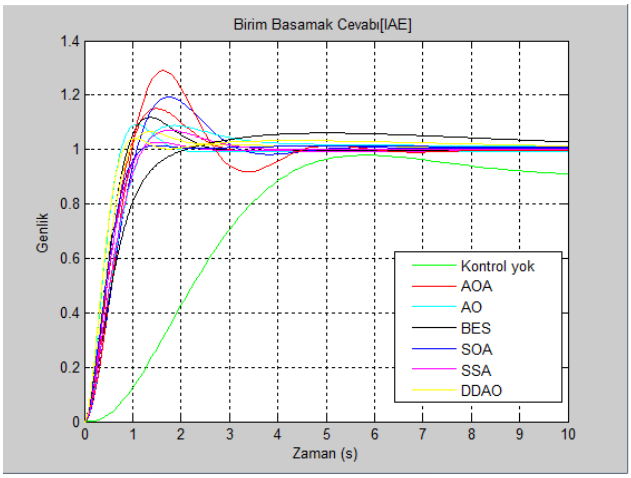


Şekil 4.16. Sistem-4 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.19. Sistem-4 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar

IAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	6.9618	5.304	2.871	4.4091	4.3596	4.5165
K_i	0.19814	0.91918	0.487	0.43621	0.33776	0.28026
K_d	2.26758	6.404	3.2921	3.2065	4.5388	3.2923
Maksimum aşım (%)	28.9906	4.0183	6.0745	8.8998	**1.4958**	7.085
Yükselme süresi (s)	0.6735	**0.5188**	1.0914	0.8081	0.7069	0.7995
Yerleşme süresi (s)	4.2613	8.0719	11.3775	4.5465	7.0738	**2.8941**
Kararlı hal hatası	**0.0044**	0.0130	0.0277	0.0107	0.0089	0.0074
Uygunluk fonksiyonu	7.1127	8.3823	9.517	**6.789**	8.47495	7.241
Hesaplama süresi (s)	75.723	**74.841**	103.801	130.492	123.157	160.876

Çizelge 4.19'da görüldüğü gibi Sistem-4 için IAE performans kriterine göre en az maksimum aşım SOA, en kısa yükselme süresi DDAO, en kısa yerleşme süresi SSA ve en düşük kararlı hal hatası AOA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine AO algoritmasıyla ulaşılırken DDAO en kısa hesaplama süresine sahiptir.

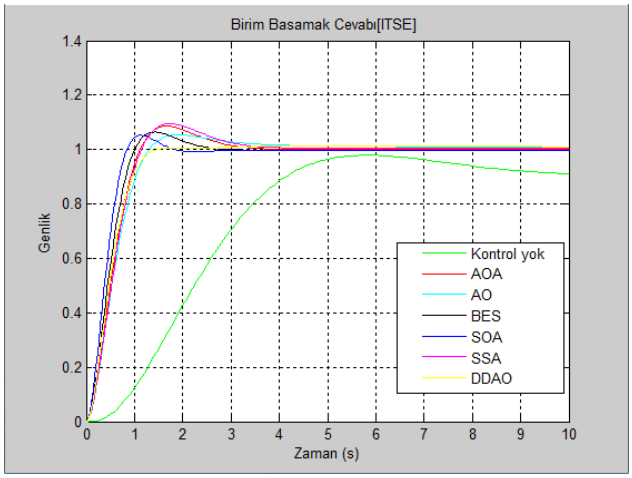


Şekil 4.17. Sistem-4 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.20. Sistem-4 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar

ITSE	AOA	DDAO	BES	AO	SOA	SSA
K_p	4.8975	3.8956	5.2899	4.09	5.7801	4.785
K_i	0.24822	0.31104	0.18772	0.30726	0.20847	0.3145
K_d	3.45	4.0596	4.2487	3.3062	5.6273	3.2724
Maksimum aşım (%)	8.6351	**0.6253**	6.4476	5.3855	5.2552	9.5484
Yükselme süresi (s)	0.7483	0.8026	0.661	0.8459	**0.5507**	0.7694
Yerleşme süresi (s)	2.8505	**1.2591**	2.1901	3.5228	1.5495	3.145
Kararlı hal hatası	**0.0013**	0.0106	0.0030	0.0082	0.0026	0.0049
Uygunluk fonksiyonu	0.752	0.987	0.647	0.75469	**0.5717**	1.241
Hesaplama süresi (s)	88.038	**74.508**	121.544	123.304	133.565	164.529

Çizelge 4.20'de görüldüğü gibi Sistem-4 için ITSE performans kriterine göre en düşük kararlı hal hatası AOA, en kısa yükselme süresi SOA, en kısa yerleşme süresi ve en az maksimum aşım DDAO ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine SOA algoritmasıyla ulaşılırken DDAO en kısa hesaplama süresine sahiptir.

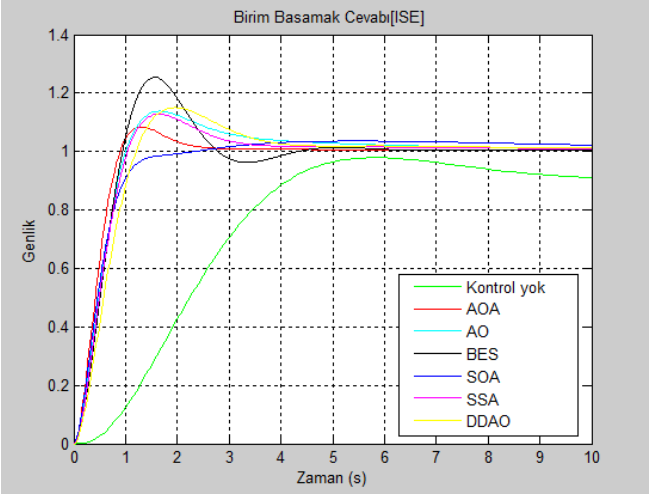


Şekil 4.18. Sistem-4 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.21. Sistem-4 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar

ISE	AOA	DDAO	BES	AO	SOA	SSA
K_p	5.857	4.528	6.984	5.2487	3.524	5.267
K_i	0.394	0.5784	0.524	0.964	0.485	0.582
K_d	4.735	2.672	2.824	3.656	4.352	3.457
Maksimum aşım (%)	8.3905	15.0357	25.4005	13.8101	**3.5925**	3.7044
Yükselme süresi (s)	**0.5979**	0.8264	0.647	0.6887	0.8168	0.7089
Yerleşme süresi (s)	**2.2697**	4.5131	3.8874	6.4851	10.5112	12.7893
Kararlı hal hatası	0.0044	0.0113	**0.0040**	0.0088	0.0218	0.0093
Uygunluk fonksiyonu	3.5458	5.789	3.53518	**2.4339**	3.5865	4.214
Hesaplama süresi (s)	76.182	**74.359**	104.814	121.237	122.878	160.981

Çizelge 4.21'de görüldüğü gibi Sistem-4 için ISE performans kriterine göre en az maksimum aşım SOA, en düşük kararlı hal hatası BES, en kısa yükselme süresi ve en kısa yerleşme süresi AOA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine AO algoritmasıyla ulaşılırken DDAO en kısa hesaplama süresine sahiptir.



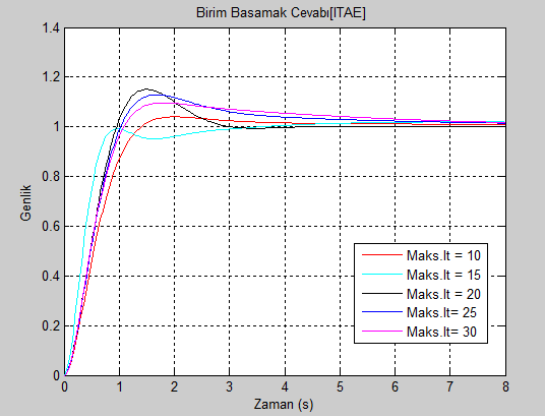
Şekil 4.19. Sistem-4 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları

Ayrıca bu tez çalışmasında kullanılan metasezgisel algoritma parametrelerinin tasarım/ayarlama sonuçlarına etkileri de incelenmiştir. Ele alınan örnek Sistem 4'ün ITAE kriterine göre tasarımında AOA algoritma parametrelerinin etkisi karşılaştırmalı olarak Çizelge 4.22 - 4.23'te ve Şekil 4.20 - 4.21'de verilmektedir.

Çizelge 4.22. AOA algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum iterasyon sayısı	**10**	**15**	**20**	**25**	**30**
K_p	3.86458	4.2551	6.1507	5.1212	4.6015
K_i	0.279	0.48293	0.33372	0.9199	0.94197
K_d	3.2825	6.8823	3.5645	3.6559	3.82854
Maksimum aşım (%)	4.014	**1.997**	15.0727	12.959	9.6981
Yükselme süresi (s)	0.8829	**0.5304**	0.6521	0.6985	0.7258
Yerleşme süresi (s)	3.3619	**2.5098**	2.6441	6.6683	7.5199
Kararlı hal hatası	0.0096	0.0198	**0.0018**	0.0147	0.0173
Uygunluk fonksiyonu	1.9997	1.784	**1.34629**	1.6759	1.779
Hesaplama süresi (s)	**43.465**	58.33	77.590	90.025	105.92

Çizelge 4.22'den de görüldüğü gibi en az maksimum aşım değeri, en kısa yükselme ve yerleşme süresi 15 iterasyon değerinde elde edilmiştir. 20 iterasyon değerinde en düşük kararlı hal hatasına ve uygunluk fonksiyonunun en düşük değerine ulaşılmıştır. En düşük hesaplama süresi de 10 iterasyonda görülmektedir.

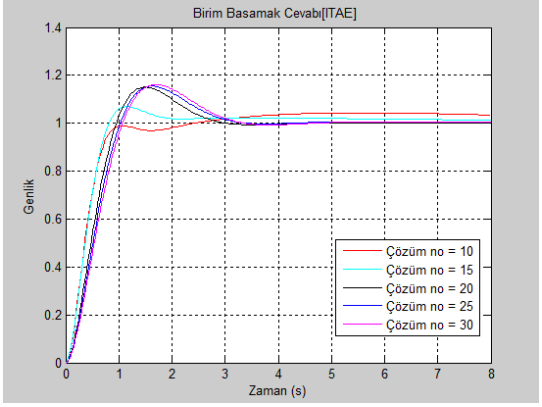


Şekil 4.20. Sistem-4 PID denetleyici tasarımında AOA algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.23. AOA algoritmasındaki çözüm no sayısının Sistem-4 PID denetleyici tasarımına etkisi

Çözüm no	10	15	20	25	30
K_p	4.0438	5.886	6.1507	5.6576	4.4453
K_i	0.8342	0.72107	0.33372	0.39298	0.34948
K_d	6.3697	5.6976	3.5645	3.1727	2.0932
Maksimum aşım (%)	**4.373**	6.8828	15.072	15.5116	16.168
Yükselme süresi (s)	0.5705	**0.5384**	0.6521	0.7035	0.7364
Yerleşme süresi (s)	10.319	4.9737	**2.6441**	2.9247	3.0023
Kararlı hal hatası	0.0343	0.0133	**0.0018**	0.0052	0.0092
Uygunluk fonksiyonu	2.01	1.897	**1.34629**	1.478	1.647
Hesaplama süresi (s)	**36.48**	53.76	77.59	89.51	104.07

Çizelge 4.23'ten de görüldüğü gibi en düşük yerleşme süresi, en düşük kararlı hal hatası ve uygunluk fonksiyonunun en düşük değeri 20 iterasyon değerinde elde edilirken 15 iterasyon değerinde en düşük yükselme süresine ulaşılmıştır. Benzer şekilde en az maksimum aşım değeri ve hesaplama süresi yine 10 iterasyonda görülmektedir.



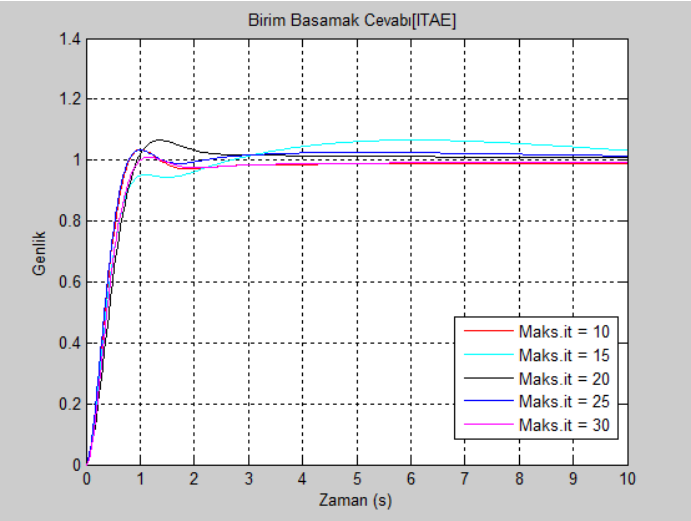
Şekil 4.21. Sistem-4 PID denetleyici tasarımında AOA algoritmasındaki çözüm no sayısının sistem cevaplarına etkisi

Ele alınan örnek Sistem 4'ün ITAE kriterine göre tasarımında DDAO algoritma parametrelerinin etkisi, karşılaştırmalı olarak Çizelge 4.24-4.28'de ve Şekil 4.22-4.26'da verilmektedir.

Çizelge 4.24. DDAO algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum iterasyon sayısı	**10**	**15**	**20**	**25**	**30**
K_p	5.5792	3.1295	5.3573	5.2431	4.8258
K_i	0.07442	0.89423	0.4621	0.75673	0.11782
K_d	6.4377	6.1853	4.691	6.6646	5.5491
Maksimum aşım (%)	3.359	6.601	6.4898	3.3004	**0.9801**
Yükselme süresi (s)	0.5133	0.6443	0.6242	**0.5077**	0.5991
Yerleşme süresi (s)	2.5261	11.0719	**2.4065**	7.8034	2.5926
Kararlı hal hatası	0.0113	0.0544	0.0095	0.0194	**0.0081**
Uygunluk fonksiyonu	2.0109	2.0837	**0.8791**	2.014	2.0077
Hesaplama süresi (s)	**35.24**	52.287	75.1271	85.12	103.24

Çizelge 4.24'ten de görüldüğü gibi en düşük yerleşme süresi ve uygunluk fonksiyonunun en düşük değeri 20 iterasyonda elde edilirken, en düşük yükselme süresi 25 iterasyonda elde edilmiştir. 30 iterasyon değerinde en az maksimum aşım ve en düşük kararlı hal hatasına ulaşılmıştır. Benzer şekilde en düşük hesaplama süresi yine 10 iterasyonda görülmektedir.

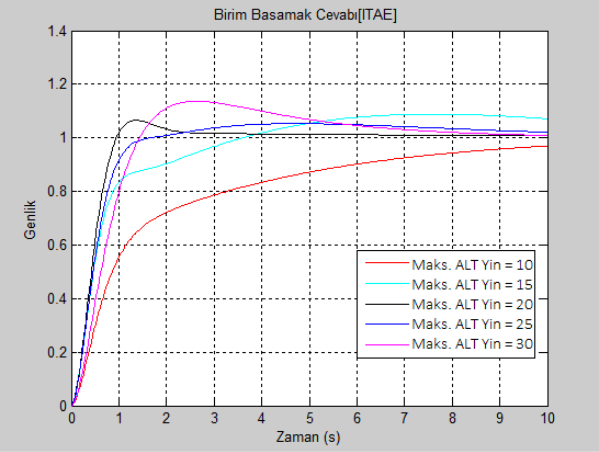


Şekil 4.22. Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.25. DDAO algoritmasındaki maksimum alt yineleme iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum altyineleme sayısı	10	15	20	25	30
K_p	1.0291	1.9613	5.3573	3.5729	3.4016
K_i	0.050643	0.547	0.4621	0.67747	0.789
K_d	2.3296	4.6453	4.691	4.3781	2.591
Maksimum aşım (%)	**0.5742**	8.8564	6.4898	5.3115	13.641
Yükselme süresi (s)	5.7062	1.7881	**0.6242**	0.7959	0.978
Yerleşme süresi (s)	11.2099	13.7177	**2.4065**	10.1308	8.0555
Kararlı hal hatası	0.0563	0.0880	**0.0095**	0.0342	0.0205
Uygunluk fonksiyonu	3.0304	3.071	**0.879**	2.0207	2.0092
Hesaplama süresi (s)	**36.52**	52.31	75.127	75.88	102.54

Çizelge 4.25'ten de görüldüğü gibi en düşük yükselme süresi, en düşük yerleşme süresi, en düşük kararlı hal hatası ve uygunluk fonksiyonunun en düşük değeri 20 iterasyonda elde edilmiştir. En az maksimum aşım ve en düşük hesaplama süresi ise 10 iterasyonda görülmektedir.

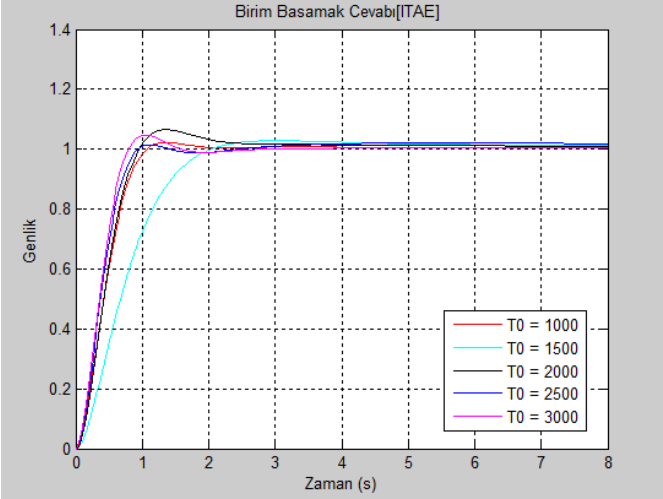


Şekil 4.23. Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki maksimum alt yineleme iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.26. DDAO algoritmasındaki $T0$ paramatresinin Sistem-4 PID denetleyici tasarımına etkisi

$T0$	1000	1500	2000	2500	3000
K_p	4.6479	2.8598	5.3573	4.74463	5.752
K_i	0.36541	0.19934	0.4621	0.55368	0.35166
K_d	4.8271	2.4337	4.691	5.91603	6.24
Maksimum aşım (%)	2.2592	2.8617	6.4898	**2.1196**	4.6944
Yükselme süresi (s)	0.6598	1.2211	0.6242	0.5727	**0.5165**
Yerleşme süresi (s)	1.5087	4.5463	2.4065	6.8789	**1.3664**
Kararlı hal hatası	0.0100	0.0124	0.0095	0.0180	**0.0042**
Uygunluk fonksiyonu	1.01	1.017	**0.879**	1.414	1.354
Hesaplama süresi (s)	**69.63**	69.77	75.127	79.53	85.57

Çizelge 4.26'dan da görüldüğü gibi en az maksimum aşım 2500 değerinde, en düşük yükselme süresi, en düşük yerleşme süresi ve en düşük kararlı hal hatası 3000 değerinde elde edilmiştir. En düşük uygunluk fonksiyonu 2000 değerinde iken en düşük hesaplama süresi 1000 değerinde görülmektedir.

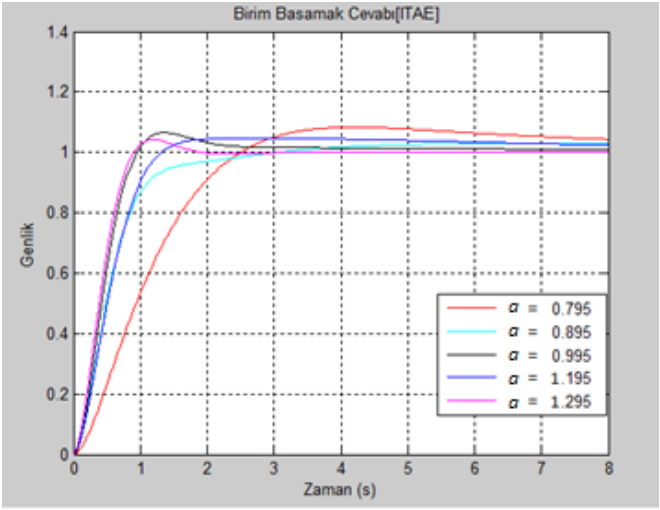


Şekil 4.24. Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki T0 paramatresinin sistem cevaplarına etkisi

Çizelge 4.27. DDAO algoritmasındaki a paramatresinin Sistem-4 PID denetleyici tasarımına etkisi

a	**0.795**	**0.895**	**0.995**	**1.195**	**1.295**
K_p	2.025	3.1068	5.3573	3.8733	5.4133
K_i	0.23966	0.35426	0.4621	0.56433	0.25547
K_d	1.4884	4.0597	4.691	3.695	5.3909
Maksimum aşım (%)	8.3255	**2.998**	6.4898	4.5496	4.2226
Yükselme süresi (s)	1.6747	0.948	0.6242	0.8272	**0.5793**
Yerleşme süresi (s)	13.1031	11.1854	2.4065	9.0122	**1.5539**
Kararlı hal hatası	0.0435	0.0282	0.0095	0.0241	**5.4278e-04**
Uygunluk fonksiyonu	1.0435	2.0282	**0.879**	1.041	1.0095
Hesaplama süresi (s)	70.60	**69.31**	75.127	69.78	69.59

Çizelge 4.27'den de görüldüğü gibi en az maksimum aşım 0.895 değerinde, en düşük yükselme süresi, en düşük yerleşme süresi ve en düşük kararlı hal hatası 1.295 değerinde elde edilmiştir. Uygunluk fonksiyonunun en düşük değeri 0.995'te görülmektedir. Benzer şekilde 0.895 değerinde en düşük hesaplama süresine sahiptir.

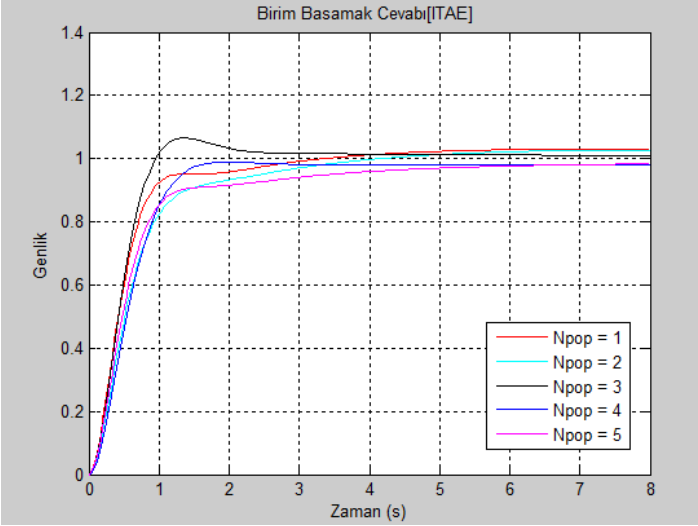


Şekil 4.25. Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki $\boldsymbol{a}$ paramatresinin sistem cevaplarına etkisi

Çizelge 4.28. DDAO algoritmasındaki $Npop$ paramatresinin Sistem-4 PID denetleyici tasarımına etkisi

$\boldsymbol{Npop}$	**1**	**2**	**3**	**4**	**5**
K_p	3.3296	2.5902	5.3573	3.5716	1.4342
K_i	0.41946	0.26177	0.4621	0.035196	0.072317
K_d	5.088	3.8993	4.691	3.4731	1.4342
Maksimum aşım (%)	2.9956	2.7492	6.4898	**0**	**0**
Yükselme süresi (s)	0.7647	1.2311	**0.6242**	0.9389	1.1525
Yerleşme süresi (s)	11.6191	12.8553	**2.4065**	5.6906	6.7388
Kararlı hal hatası	0.0292	0.0274	**0.0095**	0.0196	0.0164
Uygunluk fonksiyonu	1.547	1.357	**0.879**	1.2478	1.874
Hesaplama süresi (s)	69.27	**69.10**	75.12	69.661	69.90

Çizelge 4.28'den de görüldüğü gibi en az maksimum aşım $Npop$ = 4 ve 5 iken görülmüştür. En düşük yükselme süresi, en düşük yerleşme süresi, en düşük kararlı hal hatası ve uygunluk fonksiyonunun en düşük değeri $Npop$ =3 değerinde elde edilmiştir. Benzer şekilde $Npop$ =2 değerinde ise en düşük hesaplama süresine sahiptir.



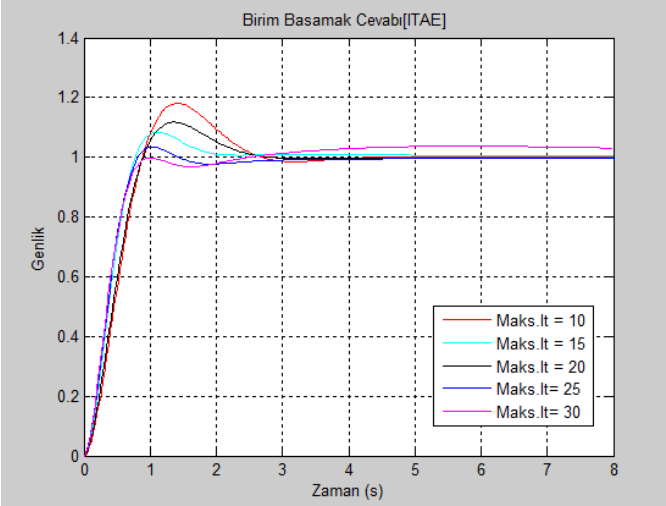
Şekil 4.26. Sistem-4 PID denetleyici tasarımında DDAO algoritmasındaki *Npop* paramatresinin sistem cevaplarına etkisi

Ele alınan örnek Sistem 4'ün ITAE kriterine göre tasarımında BES algoritma parametrelerinin etkisi, karşılaştırmalı olarak Çizelge 4.29 - 4.30'da ve Şekil 4.27 - 4.28' de verilmektedir.

Çizelge 4.29. BES algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum iterasyon sayısı	**10**	**15**	**20**	**25**	**30**
K_p	6.932	6.4367	6.304	5.4955	4.2737
K_i	0.33851	0.45739	0.248	0.19589	0.79263
K_d	3.7014	5.7168	4.2578	6.2911	6.661
Maksimum aşım (%)	18.0236	8.3601	11.7646	**3.3535**	3.725
Yükselme süresi (s)	0.6068	**0.5218**	0.6065	0.5232	0.5408
Yerleşme süresi (s)	2.4601	**1.7997**	2.3382	2.1561	10.1954
Kararlı hal hatası	**5.1886e-04**	0.0054	0.0017	0.0031	0.0306
Uygunluk fonksiyonu	**1.1092**	1.7997	2.154	2.261	1.3631
Hesaplama süresi (s)	**48.66**	71.76	104.51	119.88	141.97

Çizelge 4.29'dan da görüldüğü gibi en az maksimum aşım 25 iterasyon değerinde elde edilmiştir. En kısa yükselme ve en kısa yerleşme süresi 15 iterasyon değerinde görülmektedir. Benzer şekilde uygunluk fonksiyonunun en düşük değeri, en düşük kararlı hal hatası ve en kısa hesaplama süresine 10 iterasyonda ulaşılmıştır.

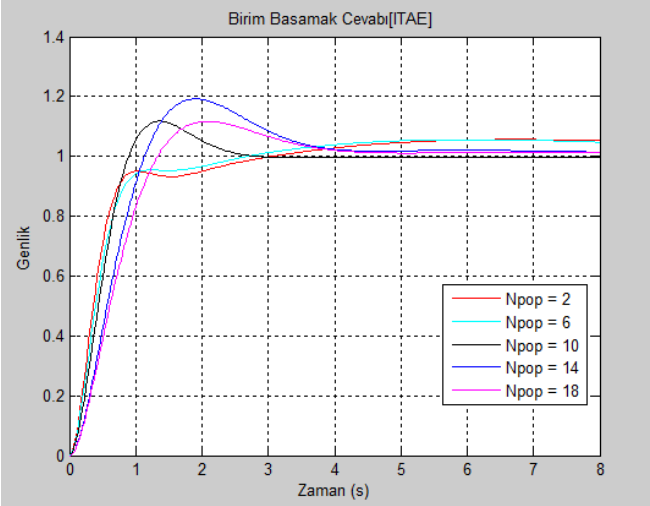


Şekil 4.27. Sistem-4 PID denetleyici tasarımında BES algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.30. BES algoritmasındaki $Npop$ paramatresinin Sistem-4 PID denetleyici tasarımına etkisi

$Npop$	**2**	**6**	**10**	**14**	**18**
K_p	3.1146	3.2587	6.304	4.8828	4.0812
K_i	0.77142	0.74246	0.248	0.79641	0.34869
K_d	6.52516	5.6326	4.2578	2.642	2.4853
Maksimum aşım (%)	5.6417	**5.6117**	11.7646	11.6803	11.6803
Yükselme süresi (s)	0.6144	0.6956	**0.6065**	0.7843	0.9086
Yerleşme süresi (s)	12.0316	11.3316	**2.3382**	4.0085	4.0085
Kararlı hal hatası	0.0525	0.0472	**0.0017**	0.0145	0.0115
Uygunluk fonksiyonu	1.4721	1.6574	2.1540	2.097	**1.4570**
Hesaplama süresi (s)	**16.70**	56.34	104.510	134.38	176.16

Çizelge 4.30'dan da görüldüğü gibi en az maksimum aşım $Npop$ = 6 değerinde elde edilirken, en kısa yükselme, en kısa yerleşme süresi ve en düşük kararlı hal hatasına $Npop$ = 10 değerinde ulaşılmıştır. Benzer şekilde uygunluk fonksiyonunun en düşük değeri $Npop$ = 18 iken en kısa hesaplama süresi $Npop$ = 2 değerinde görülmektedir.



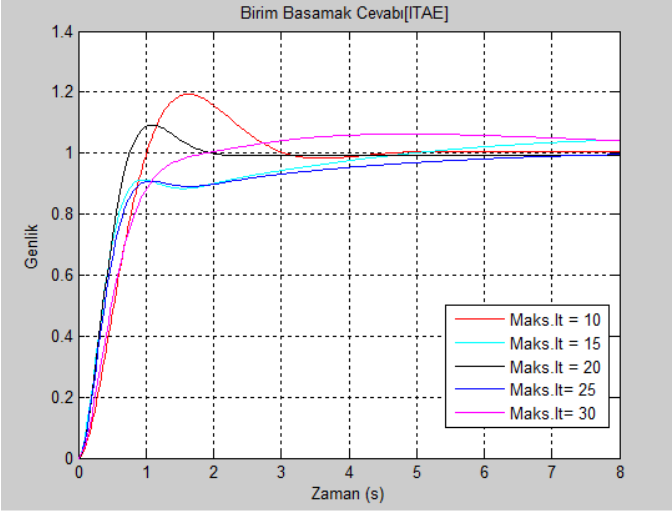
Şekil 4.28. Sistem-4 PID denetleyici tasarımında BES algoritmasındaki *Npop* paramatresinin sistem cevaplarına etkisi

Ele alınan örnek Sistem 4'ün ITAE kriterine göre tasarımında AO algoritma parametrelerinin etkisi, karşılaştırmalı olarak Çizelge 4.31-4.32'de ve Şekil 4.29-4.30'da verilmektedir.

Çizelge 4.31. AO algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum iterasyon sayısı	**10**	**15**	**20**	**25**	**30**
K_p	6.0213	2.2854	6.8821	2.6207	3.2186
K_i	0.38097	0.43808	0.11855	0.15738	0.65536
K_d	2.9058	6.7105	5.7624	5.9497	4.0289
Maksimum aşım (%)	2.8272	4.9728	9.1928	**0.6669**	6.2566
Yükselme süresi (s)	0.6945	0.6956	**0.5091**	0.798	0.893
Yerleşme süresi (s)	19.3852	16.7058	**1.6887**	5.8892	10.4837
Kararlı hal hatası	**0.0034**	0.0436	0.0078	0.0043	0.0404
Uygunluk fonksiyonu	2.146	1.6272	1.7038	1.974	**1.0497**
Hesaplama süresi (s)	**66.9**	98.46	163.67	163.27	195.13

Çizelge 4.31'den de görüldüğü gibi en az maksimum aşım 25 iterasyon değerinde elde edilmiştir. En kısa yükselme süresi ve en kısa yerleşme süresi 20 iterasyon değerinde elde edilirken uygunluk fonksiyonunun en düşük değerine 30 iterasyonda ulaşılmıştır. Benzer şekilde en kısa hesaplama süresi ve en düşük kararlı hal hatası yine 10 iterasyonda görülmektedir.

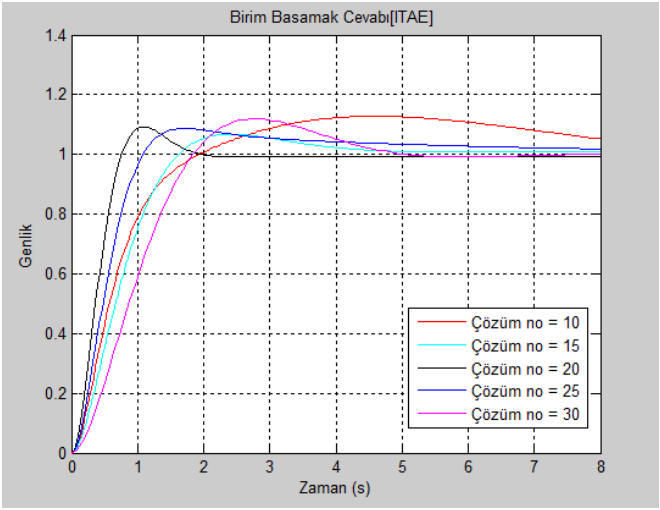


Şekil 4.29. Sistem-4 PID denetleyici tasarımında AO algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.32. AO algoritmasındaki çözüm no sayısının Sistem-4 PID denetleyici tasarımına etkisi

Çözüm no	10	15	20	25	30
K_p	1.4912	3.3486	6.8821	4.6947	2.903
K_i	0.8435	0.21268	0.11855	0.76693	0.14438
K_d	3.2769	2.3119	5.7624	3.8667	1.3367
Maksimum aşım (%)	12.7623	**6.7462**	9.1928	8.68	11.8566
Yükselme süresi (s)	1.1427	1.0747	**0.5091**	0.7196	1.2693
Yerleşme süresi (s)	9.2828	4.1408	**1.6887**	7.4897	4.5199
Kararlı hal hatası	0.1350	0.0074	0.0078	0.0179	**0.0012**
Uygunluk fonksiyonu	4.5907	2.4039	**1.7038**	1.709	2.77
Hesaplama süresi (s)	**66.24**	99.74	163.67	167.58	195.37

Çizelge 4.32'den de görüldüğü gibi en düşük yerleşme süresi, en düşük yükselme süresi ve uygunluk fonksiyonunun en düşük değeri 20 iterasyonda elde edilmiştir. 30 iterasyon değerinde en düşük kararlı hal hatasına ulaşılmıştır. Benzer şekilde en az maksimum aşım 15 iterasyon değerinde iken en kısa hesaplama süresi yine 10 iterasyon değerinde görülmektedir.



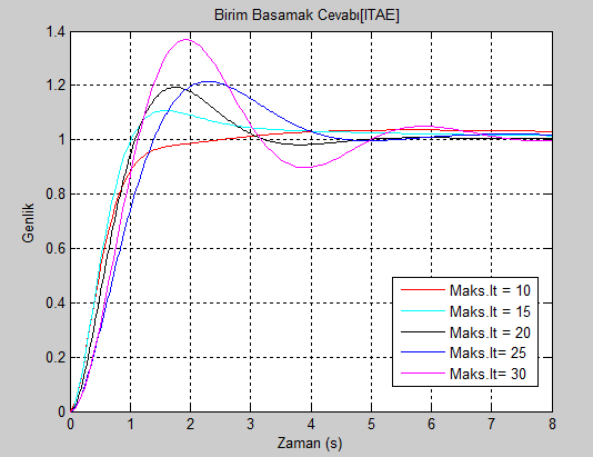
Şekil 4.30. Sistem-4 PID denetleyici tasarımında AO algoritmasındaki çözüm no sayısının sistem cevaplarına etkisi

Ele alınan örnek Sistem 4'ün ITAE kriterine göre tasarımında SOA algoritma parametrelerinin etkisi karşılaştırmalı olarak Çizelge 4.33 - 4.34'te ve Şekil 4.31 - 4.32' de verilmektedir.

Çizelge 4.33. SOA algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum iterasyon sayısı	**10**	**15**	**20**	**25**	**30**
K_p	3.301	5.1727	5.5293	3.9562	5.7665
K_i	0.43504	0.74767	0.35557	0.45963	0.5922
K_d	4.089	3.8865	2.58	1.6552	1.4321
Maksimum aşım (%)	**3.5964**	10.6858	19.2656	21.4547	36.7743
Yükselme süresi (s)	0.8899	**0.6837**	0.7469	0.7472	0.769
Yerleşme süresi (s)	10.9405	6.3442	**3.015**	7.5896	6.8203
Kararlı hal hatası	0.0306	0.0149	**0.0038**	0.0185	0.0040
Uygunluk fonksiyonu	2.57	1.578	**1.476**	1.741	2.34
Hesaplama süresi (s)	**50.23**	80.24	107.546	122.45	174.34

Çizelge 4.33'ten de görüldüğü gibi en az maksimum aşım ve en kısa hesaplama süresi 10 iterasyon değerinde elde edilirken, en kısa yükselme süresi 15 iterasyon değerinde görülmüştür. En kısa yerleşme süresi, en düşük kararlı hatası ve uygunluk fonksiyonunun en düşük değerine 20 iterasyonda ulaşılmıştır.

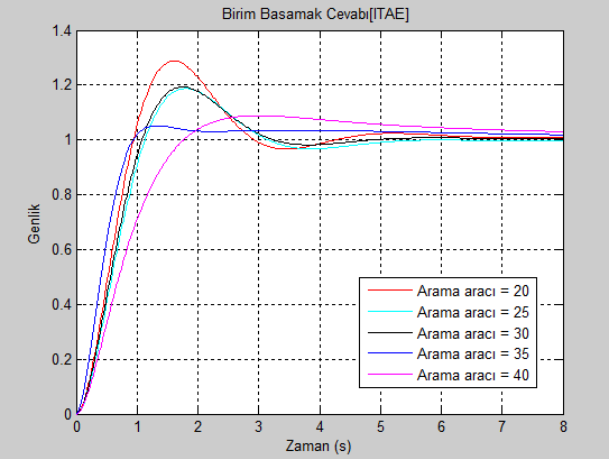


Şekil 4.31. Sistem-4 PID denetleyici tasarımında SOA algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.34. SOA algoritmasındaki arama aracı sayısının Sistem-4 PID denetleyici tasarımına etkisi

Arama aracı	**20**	**25**	**30**	**35**	**40**
K_p	6.8235	5.353	5.5293	5.0579	2.8936
K_i	0.94693	0.1752	0.35557	0.78718	0.40187
K_d	2.7102	2.3605	2.58	5.0579	2.212
Maksimum aşım (%)	28.8249	18.7555	19.2656	**5.1778**	8.8362
Yükselme süresi (s)	0.6488	0.7796	0.7469	**0.611**	1.1777
Yerleşme süresi (s)	5.7294	4.5484	**3.015**	7.5779	10.3095
Kararlı hal hatası	0.0087	0.0048	**0.0038**	0.0184	0.0300
Uygunluk fonksiyonu	1.6139	1.802	1.476	1.3316	**1.0884**
Hesaplama süresi (s)	**67.19**	83.88	107.5466	117.19	130.74

Çizelge 4.34'ten de görüldüğü gibi en az maksimum aşım değeri ve en kısa yükselme süresi 35 iterasyon değerinde elde edilmiştir. En kısa yerleşme süresi ve en düşük kararlı hal hatasına 30 iterasyon değerinde ulaşılmıştır. Uygunluk fonksiyonunun en düşük değeri 40 iterasyonda iken, en kısa hesaplama süresi yine 20 iterasyonda görülmektedir.



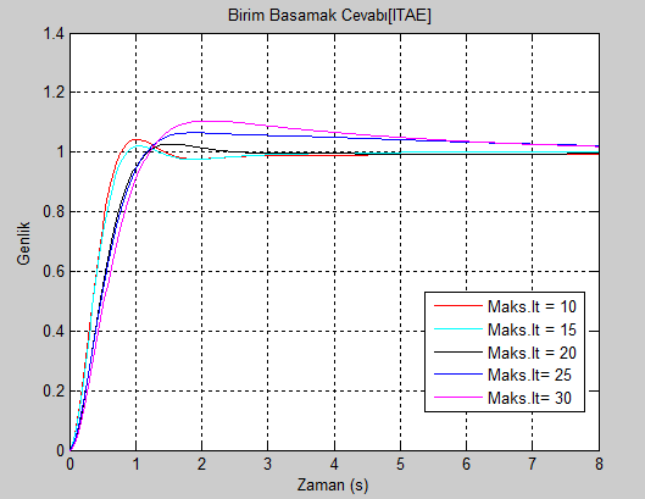
Şekil 4.32. Sistem-4 PID denetleyici tasarımında SOA algoritmasındaki arama aracı sayısının sistem cevaplarına etkisi

Ele alınan örnek Sistem 4'ün ITAE kriterine göre tasarımında SSA algoritma parametrelerinin etkisi karşılaştırmalı olarak Çizelge 4.35 - 4.36' da ve Şekil 4.33 - 4.34' te verilmektedir.

Çizelge 4.35. SSA algoritmasındaki maksimum iterasyon sayısının Sistem-4 PID denetleyici tasarımına etkisi

Maksimum iterasyon sayısı	**10**	**15**	**20**	**25**	**30**
K_p	3.301	5.1727	5.293	3.9562	5.7665
K_i	0.43504	0.74767	0.5557	0.45963	0.5922
K_d	4.089	3.8865	2.81	1.6552	1.4321
Maksimum aşım (%)	**3.5964**	10.6858	19.2656	21.4547	36.7743
Yükselme süresi (s)	0.8899	**0.6837**	0.7469	0.7472	0.769
Yerleşme süresi (s)	10.9405	6.3442	**3.015**	7.5896	6.8203
Kararlı hal hatası	0.0306	0.0149	0.0058	0.0185	**0.0040**
Uygunluk fonksiyonu	2.57	1.578	**1.476**	1.741	2.34
Hesaplama süresi (s)	**50.23**	80.24	107.546	122.45	74.34

Çizelge 4.35'ten de görüldüğü gibi en az maksimum aşım ve en kısa hesaplama süresi 10 iterasyon değerinde elde edilmiştir. En kısa yerleşme süresi ve uygunluk fonksiyonunun en düşük değerine 20 iterasyonda ulaşılmıştır. En düşük kararlı hal hatası 30 iterasyonda iken en kısa yükselme süresi 15 iterasyon değerinde görülmektedir.

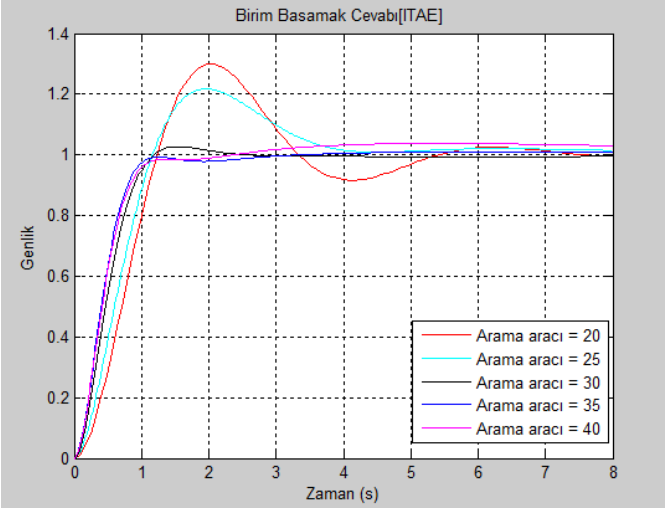


Şekil 4.33. Sistem-4 PID denetleyici tasarımında SSA algoritmasındaki maksimum iterasyon sayısının sistem cevaplarına etkisi

Çizelge 4.36. SSA algoritmasındaki arama aracı sayısının Sistem-4 PID denetleyici tasarımına etkisi

Arama aracı	**20**	**25**	**30**	**35**	**40**
K_p	5.1186	4.827	4.5032	4.247	5.167
K_i	0.2584	0.7756	0.14487	0.32704	0.9169
K_d	1.3636	2.357	4.1105	5.278	3.864
Maksimum aşım (%)	29.8658	21.7057	**2.7025**	3.247	3.781
Yükselme süresi (s)	0.8427	0.8063	0.7363	0.7145	**0.6873**
Yerleşme süresi (s)	6.6682	6.6954	**1.8361**	5.478	10.2746
Kararlı hal hatası	0.0401	0.0504	0.0871	**0.0357**	0.0874
Uygunluk fonksiyonu	1.997	1.958	2.142	**1.5389**	2.1253
Hesaplama süresi (s)	**134.24**	137.85	140.96	164.24	187.25

Çizelge 4.36'dan da görüldüğü gibi en az maksimum aşım değeri ve en kısa yerleşme süresi 30 iterasyon değerinde elde edilirken, en kısa yükselme süresine 40 iterasyon değerinde ulaşılmıştır. Uygunluk fonksiyonunun en düşük değeri ve en düşük kararlı hal hatası 35 iterasyonda iken en kısa hesaplama süresi 20 iterasyonda görülmektedir.



Şekil 4.34. Sistem-4 PID denetleyici tasarımında SSA algoritmasındaki arama aracı sayısının sistem cevaplarına etkisi

Beşinci uygulama olarak Denklem 4.5'teki (Chatterjee & Mukherjee, 2016) transfer fonksiyonuna sahip sistem için PID denetleyici tasarımı gerçekleştirilmektedir. Kullanılan metasezgisel algoritmaların parametreleri Çizelge 4.37'deki gibi olup farklı performans kriterlerine göre elde edilen karşılaştırmalı sonuçlar sırasıyla Çizelge 4.38-4.42'de verilmektedir. Ayrıca karşılaştırmalı birim basamak cevapları da sırasıyla Şekil 4.35 - 4.39'da gösterilmektedir.

$$G(s) = \frac{10}{0.04s^3+0.54s^2+1.5s+1}, H(s) = \frac{1}{0.01s+1} \quad (4.5)$$

Çizelge 4.37. Sistem-5 için algoritma parametre ayarlaması

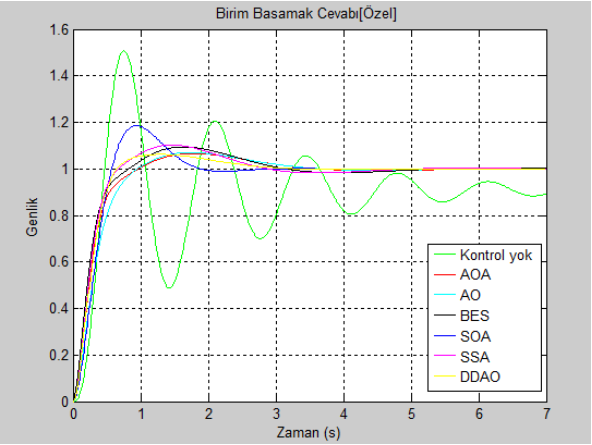
AOA	DDAO	BES	AO	SOA	SSA
Maksimum iterasyon sayısı: 20					
Çözüm no: 20	Maksimum alt yineleme: 20 $T_0 = 2000$ $\alpha = 0.995$ $N_{pop} = 3$	$N_{pop} = 10$	Çözüm no: 20	Arama aracı: 30	Arama aracı: 30

Bütün algoritmalarda maksimum iterasyon 20, alt sınır K_p=0.1, K_i=0.1, K_d=0.1 ve üst sınır K_p=1.5, K_i=1.5, K_d=1.5 olarak seçilmiştir.

Çizelge 4.38. Sistem-5 için özel fonksiyon tasarım kriterine göre karşılaştırmalı sonuçlar

Özel	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.4058	0.47354	0.41142	0.35763	0.51643	0.4936
K_i	0.50157	0.52452	0.63736	0.41618	0.56241	0.68518
K_d	0.17634	0.15422	0.19433	0.13336	0.10094	0.1851
Maksimum aşım (%)	6.566	**6.232**	9.4402	7.0422	18.57	10.2901
Yükselme süresi (s)	0.4639	0.4058	0.4164	0.5571	0.3949	**0.3909**
Yerleşme süresi (s)	2.993	2.454	2.8072	2.9896	**1.7079**	2.5806
Kararlı hal hatası	6.7622e-04	8.5151e-05	0.0016	3.9346e-04	**1.5322e-08**	3.1514e-04
Uygunluk fonksiyonu	1.2931	1.512	**1.2478**	1.3593	1.349	1.3083
Hesaplama süresi (s)	**16.521**	18.535	17.524	29.235	19.56	21.015

Çizelge 4.38'de görüldüğü gibi Sistem-5 için Özel performans kriterine göre en az maksimum aşım DDAO ve en kısa yükselme süresi SSA algoritmasında görülmektedir. En kısa yerleşme süresi ve en düşük kararlı hal hatası SOA ile elde edilmiştir. Ayrıca en düşük uygunluk fonksiyonu değerine BES algoritmasıyla ulaşılırken AOA en kısa hesaplama süresine sahiptir.

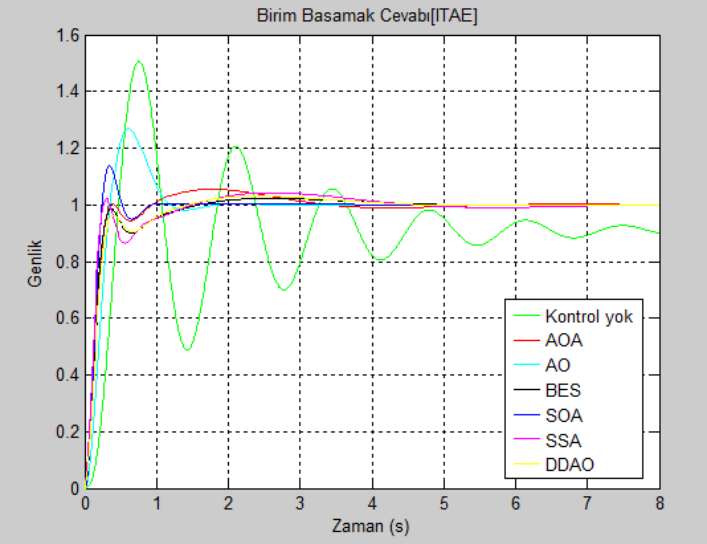


Şekil 4.35. Sistem-5 Özel Fonksiyon kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.39. Sistem-5 için ITAE tasarım kriterine göre karşılaştırmalı sonuçlar

ITAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.59977	0.5434	0.5979	0.9285	1.1209	0.54088
K_i	0.91008	0.57519	0.57519	1.1367	0.88342	0.74683
K_d	0.36146	0.33452	0.37073	0.1805	0.42534	0.49218
Maksimum aşım (%)	5.5368	3.086	**2.303**	26.912	13.753	4.2692
Yükselme süresi (s)	0.2107	0.2386	0.2098	0.2453	**0.1596**	0.1623
Yerleşme süresi (s)	2.8301	3.3125	3.1121	1.1156	**0.8185**	3.7682
Kararlı hal hatası	3.5943e-04	1.5476e-05	3.6285e-04	**1.9972e-09**	5.6809e-06	0.0010
Uygunluk fonksiyonu	**2.136**	2.854	2.652	2.619	2.3541	2.424
Hesaplama süresi (s)	**71.842**	102.563	97.852	129.243	69.401	152.575

Çizelge 4.39'da görüldüğü gibi Sistem-5 için ITAE performans kriterine göre en az maksimum aşım BES, en düşük kararlı hal hatası AO, en kısa yükselme süresi ve en kısa yerleşme süresi SOA ile elde edilmiştir. Ayrıca AOA en düşük uygunluk fonksiyonu değerine ve en kısa hesaplama süresine sahiptir.

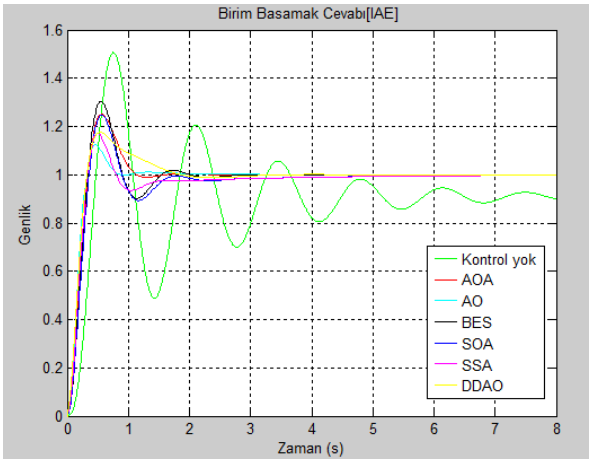


Şekil 4.36. Sistem-5 ITAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.40. Sistem-5 için IAE tasarım kriterine göre karşılaştırmalı sonuçlar

IAE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.98582	0.8542	1.1636	0.94939	1.0804	1.067
K_i	1.1143	1.435	0.7425	0.78461	0.42174	0.41298
K_d	0.199199	0.254	0.17018	0.27386	0.15856	0.2185
Maksimum aşım (%)	25.172	17.63	30.5039	**12.401**	24.8981	17.061
Yükselme süresi (s)	0.2331	0.2247	0.2278	**0.2141**	0.2438	0.2260
Yerleşme süresi (s)	1.0451	65.235	1.462	**0.7513**	2.6139	2.4511
Kararlı hal hatası	**2.2506e-08**	5.378e-06	8.517e-05	1.028e-06	0.0023	0.0021
Uygunluk fonksiyonu	3.214	3.84	3.874	3.548	3.125	**2.659**
Hesaplama süresi (s)	**61.621**	70.746	98.3027	70.3043	100.214	161.103

Çizelge 4.40'da görüldüğü gibi Sistem-5 için IAE performans kriterine göre en az maksimum aşım, en kısa yükselme süresi ve en kısa yerleşme süresine AO algoritmasıyla ulaşılmıştır. En düşük uygunluk fonksiyonu değeri SSA ile elde edilmiştir. AOA en düşük kararlı hal hatasına ve en kısa hesaplama süresine sahiptir.

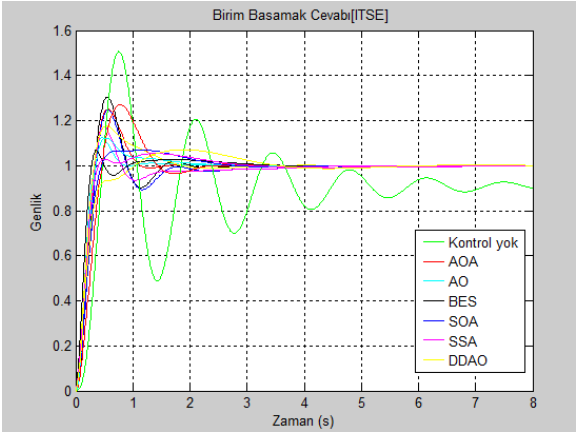


Şekil 4.37. Sistem-5 IAE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.41. Sistem-5 için ITSE tasarım kriterine göre karşılaştırmalı sonuçlar

ITSE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.68817	0.4241	0.83137	0.83143	0.61137	0.6493
K_i	0.83609	0.6561	0.92885	0.9043	0.79798	0.85433
K_d	0.12035	0.2546	0.39203	0.24123	0.212	0.21225
Maksimum aşım (%)	27.1621	7.1054	6.9841	12.0254	6.8512	**5.4066**
Yükselme süresi (s)	0.3138	0.3411	**0.181**	0.2385	0.2943	0.2504
Yerleşme süresi (s)	2.0303	3.08	2.2934	**1.6733**	2.1981	2.3952
Kararlı hal hatası	**2.4146e-08**	9.3318e-04	9.6445e-05	6.9346e-07	2.8240e-05	1.4923e-05
Uygunluk fonksiyonu	**1.925**	2.651	2.214	2.145	2.784	2.151
Hesaplama süresi (s)	**71.549**	136.641	97.0140	128.786	100.887	161.392

Çizelge 4.41'de görüldüğü gibi Sistem-5 için ITSE performans kriterine göre en az maksimum aşım SSA, en kısa yükselme süresi BES, en kısa yerleşme süresi AO ile elde edilmiştir. Ayrıca AOA en düşük uygunluk fonksiyonu değerine, en düşük kararlı hal hatasına ve en kısa hesaplama süresine sahiptir.

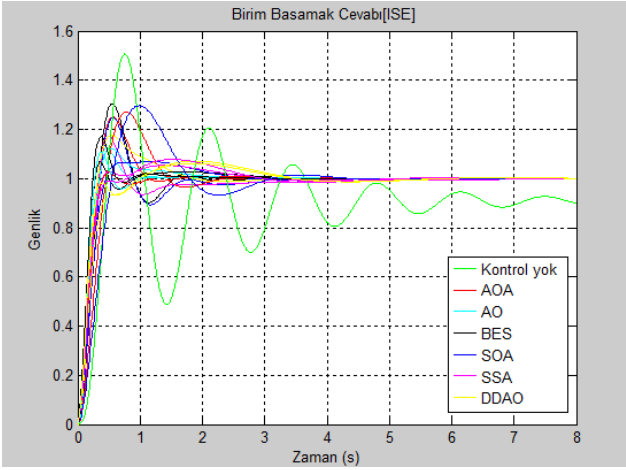


Şekil 4.38. Sistem-5 ITSE performans kriterine göre karşılaştırmalı birim basamak cevapları

Çizelge 4.42. Sistem-5 için ISE tasarım kriterine göre karşılaştırmalı sonuçlar

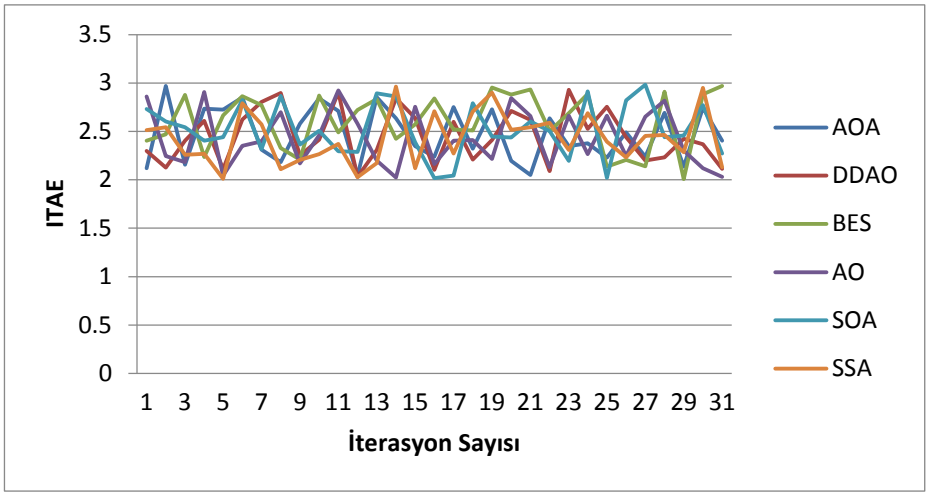
ISE	AOA	DDAO	BES	AO	SOA	SSA
K_p	0.70424	0.8651	1.1957	1.0023	1.12919	0.90564
K_i	0.56252	0.54243	1.1092	0.79701	0.49233	0.54344
K_d	0.26175	0.3524	0.37414	0.3799	0.10101	0.28454
Maksimum aşım (%)	**2.4682**	6.1241	17.085	10.8014	29.061	7.7972
Yükselme süresi (s)	0.2534	0.2229	**0.1684**	0.1765	0.3814	0.2637
Yerleşme süresi (s)	0.9197	2.9638	1.1025	**0.8588**	2.8412	2.5847
Kararlı hal hatası	1.2028e-05	7.184e-04	**4.235e-07**	7.2214e-06	5.993e-05	5.852e-04
Uygunluk fonksiyonu	**2.2684**	2.589	2.984	2.2841	3.0144	2.8746
Hesaplama süresi (s)	**71.647**	153.308	157.694	134.936	100.304	171.601

Çizelge 4.42'de görüldüğü gibi Sistem-5 için ISE performans kriterine göre en az maksimum aşım AOA, en kısa yerleşme süresi AO, en kısa yükselme süresi ve en düşük kararlı hal hatası BES ile elde edilmiştir. Ayrıca AOA en düşük uygunluk fonksiyonu değerine ve en kısa hesaplama süresine sahiptir.



Şekil 4.39. Sistem-5 ISE performans kriterine göre karşılaştırmalı birim basamak cevapları

Sistem-5 transfer fonksiyonu AOA, DDAO, BES, AO, SOA ve SSA algoritmaları 30 kez bağımsız olarak çalıştırıldı. Sistem-5 için elde edilen ITAE kriteri değerleri Şekil 4.39'da gösterilmiştir. En kötü, en iyi, ortalama, medyan ve standart sapma değerleri gibi objektif fonksiyonun istatistiksel değerleri Çizelge 4.43'te verilmiştir.



Şekil 4.40. Sistem-5 için elde edilen ITAE Değerleri

Çizelge 4.43. ITAE Kriteri Uygunluk Fonksiyonu Değerleri

İstatistik indeksi	AOA	DDAO	BES	AO	SOA	SSA
En kötü	2.890062	**3.185243**	2.7426	2.9436	2.8654	2.8343
En iyi	**2.004461**	2.048055	2.01071	2.03894	2.01821	2.04431
Ortalama	2.51086	2.492947	2.52872	2.52774	**2.39735**	2.44325
Medyan	2.495328	2.491080	2.56921	2.55208	**2.39080**	2.42821
Standart Sapma	**0.2455**	0.261367	0.28228	0.26707	0.2987	0.25085

Sistem-5 için ITAE performans kriterine göre elde edilen Çizelge 4.43'teki sonuçlara göre en kötü değer DDAO, en iyi değer AOA, en düşük ortalama ve en düşük medyan değeri SOA ile elde edilmiştir. Ayrıca en düşük standart sapma değerine AOA algoritmasıyla ulaşılmıştır.

5. SONUÇ

Metasezgisel algoritmalar son yıllarda kullanılan popüler yöntemlerdendir. Kullanımının yaygın olması sebebiyle literatüre birçok yeni metasezgisel algoritma girmektedir. Hesaplama gücünün iyi ve dönüşümlerinin kolaylığı sayesinde tercih edilen algoritmalar, tanımlanan her sistemde en iyi sonucu bulamadığı için yeni metasezgisel algoritmalar önerilmeye ve geliştirilmeye devam edilmektedir. Metasezgisel algoritmalar, farklı şekillerde sonuç bulurlar. Dolayısıyla optimizasyon algoritmaları her problemde farklı sonuçlar vermektedirler, bu yüzden farklı türde problemlerde farklı sonuçlar üretirler. Metasezgisel algoritmalar içerisinden problem türüne göre en uygun algoritma seçilmelidir.

Bu tez çalışmasında güncel metasezgisel algoritmalar kullanılarak PID türü denetleyicilerin tasarımları gerçeklenmiştir. Çalışmada, tanımlanan örnek sistemler için denetleyici katsayıları - farklı peformans kriterlerine göre - AOA, DDAO, AO, BES, SOA ve SSA algoritmaları kullanılarak ayarlanmıştır. Performans kriterlerine göre yapılan karşılaştırmalı denetleyici tasarımlarında, elde edilen sonuçlara göre kullanılan algoritmaların iyi ve kötü yönleri belirtilmiştir. Sistemlerin birim basamak cevabı parametreleri (yerleşme süresi, yükselme süresi, maksimum aşım, kararlı hal hatası) ve uygunluk fonksiyonu değeriyle hesaplama süresi kullanılarak yapılan karşılaştırmalar; hem sayısal hem de grafiksel olarak sunulmuştur. Ayrıca metasezgisel algoritma parametrelerinin tasarımlara etkileri de ilgili parametrelerin, belirli aralıklardaki değişimleriyle gözlemlenmiştir. Bunun yanında metasezgisel algoritmaların, belirli sayıda bağımsız çalıştırmalarıyla elde edilen en iyi, en kötü, ortalama, medyan ve standat sapma değerleri de hesaplanmıştır. Tüm bu sonuçlar, ayrıntılı yorumlanmıştır.

Elde edilen bulgular doğrultusunda; metasezgisel algoritmaların – diğer alanlarda da olduğu gibi - denetleyici tasarımlarında da etkin ve verimli bir şekilde kullanılabileceği görülmüştür. Mevcut yüksek hesaplama kapasitesine sahip araçlardan faydalanılarak, çok yüksek dereceli ve karmaşık sistemler için PID türü denetleyici katsayıları, metasezgisel algoritmalar kullanılarak ayarlanabilecek durumdadır.

Gerçek zamanlı uygulamalarda, denetleyici katsayılarının ayarlanması için de gelecek çalışma olarak bulut tabanlı sistemler ve düşük maliyetli haberleşme altyapılarının kullanılması hedeflenmektedir.

KAYNAKLAR

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A. H. (2021). The Arithmetic optimization algorithm. Computer Methods in Applied Mechanics and Engineering, 376, 113609. https://doi.org/1.1016/j.cma.2020.113609

Abualigah, L., Yousri, D., Elaziz, M.A., Ewees, A.A.i A. Al-qaness, M.A., Gandomi, A.H. (2021). Aquila Optimizer: A novel meta-heuristic optimization algorithm. Computers & Industrial Engineering, 157, https://doi.org/10.1016/j.cie.2021.107250

Ahmed S, Özbay H. (2016). Design of a switched robust control scheme for drug delivery in blood pressure regulation. IFACPapersOnLine. 49, 252-257.

Alsattar, H.A., Zaidan, A.A., Zaidan, B.B. (2020). Novel metaheuristic bald eagle search optimization search algorithm. Artificial Intelligence Review 53,2237-2264. https://doi.org/10.1007/s10462-019-09732-5

Altinoz, O.T. (2019). Optimal controller parameter tuning from multi/many objective optimization algorithms, in: M.J. Blondin, P.M. Pardalos, J. Sanchis Sáez (Eds.), Computational Intelligence and Optimization Methods for Control Engineering, Springer International Publishing, Cham, pp. 51–74, http://doi.org/10.1007/978-3-030-25446-9_3.

Ansari, U., Alam, S., Jafri, S. M. U. N. (2011). Modeling and control of three phase BLDC motor using PID with genetic algorithm, in 2011 UkSim 13th International Conference on Computer Modelling and Simulation, Cambridge, pp. 189–194.

Arora, A., Hote, Y.V., Rastogi, M. (2011). Design of PID Controller for Unstable System, ICLICC'2011, 19-26.

Barnard, C. J., Sibly, R. M. (1981). Producers and scroungers: A general model and its application to captive flocks of house sparrows. Animal Behaviour, 29, 543–550.

Basilio, J. C., Matos, S. R.. (2002). Design of PI and PID controllers with transient performance specification, IEEE Trans. Educ., 45 (4), 364–370.

Behbehani, K., Cross, R.R. (1991). A controller for regulation of mean arterial blood pressure using optimum nitroprusside infusion rate. IEEE Trans Biomed Eng. 38 (6), 513-521

Campo, A.B. (2012). PID control design. Içinde MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications – vol. 1. InTech.

Chang, G. H., Li, Y. F., Gnag, Q. (2011). Intelligent controller design for PM DC motor position control using evolutionary programming. 2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering, pp. 37-40, https://doi.org/10.1109/CCIENG.2011.6007951

Chatterjee, S. Mukherjee, V. (2016). PID controller for automatic voltage regulator using teaching–learning based optimization technique, International Journal of Electrical Power & Energy Systems, vol. 77, pp. 418-429, 0142-0615.

Del Hoyo J, Elliott A, Sargatal J. Handbook of the Birds of the World. Barcelona: Lynx edicions 1992, vol.1, no.8

Del-Hoyo, J., Elliott, A., Sargatal, J. (1992). Handbook of the Birds of the World Barcelona: Lynx edicions 1 (8).

Dhiman, G., Kumar, V. (2018). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowledge-Based Systems, 165, 169-196. https://doi.org/10.1016/j.knosys.2018.11.024

Diethorn, E. J. (1994). The generalized exponential time-frequency distribution, in *IEEE Transactions on Signal Processing*, 42 (5), 1028-1037, https://doi.org/10.1109/78.295214.

Dineva, A. Mosavi, S. F., Ardabili, I. Vajda, S., Shamshirband, T., Rabczuk, and Chau, K.-W. (2019). Review of soft computing models in design and control of rotating electrical machines, Energies,vol. 12, no. 6, pp. 1–29.

Dorf, R. C., Bishop, R. H. (2011). Modern Control Systems, 11th Ed., Pearson Prentice-Hall. New Jersey.

Dutta, T., Dey, S., Datta ,S., Das, D. (2019). Designing dual-phase steels with improved performance using ANN and GA in tandem Comput. Mater. Sci., 157, pp. 6-16

Gaing, Z.-L.(2004). A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE Trans.Energy Convers., 19 (2), 384–391.

Gandomi, A.H., Yang, X.-S., Alavi, A.H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1), 17–35. http://doi.org/10.1007/s00366-012-0308-4.

Ghafil, H.N., Jármai, K. (2018). Research and application of industrial robot manipulators in vehicle and automotive engineering, a survey Vehicle and Automotive Engineering, Springer, pp. 611-623, https://doi.org /0.1007/978-3-319-75677-6_53

Ghosal, S., Darbar, R. Neogi, B., Das, A., Tibarewala, D. N. (2012). Application of swarm intelligence computation techniques in PID controller tuning: A Review, in International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012), Visakhapatnam, India, 2012, pp. 195–208.

Hansen, A.J., (1986). Fighting behavior in bald eagles: a test of game theory ecological society of America. Ecology vol. 67 no.3, pp.787–797

Hansen, A.J., Boeker, E.L., Hodges, J.I., Cline, D.R., (1984). Bald eagles of the Chilkat Valley, Alaska: ecology, behavior, and management. National Audubon Society and U.S. Fish & Wildlife, Service, Juneau

Hernandez-Barragan, J., Rios J.D., Alanis A.Y., Lopez-Franco, C., Gomez-Avila, J., Arana-Daniel, N. (2020). Adaptive single neuron anti-windup PID controller based on the extended kalman filter algorithm. *Electronics*. 9 (4), 636. https://doi.org/10.3390/electronics9040636

Ibrahim, H., Mahmoud, A. (2014). DC motor control using PID controller based on improved ant colony algorithm, International Review of Automatic Control (IREACO), 7 (1), pp.1-6.

Jafari, S., Nikolaidis, T. (2019). Meta-heuristic global optimization algorithms for aircraft engines modelling and controller design; A review, research challenges, and exploring the future, Progress in Aerospace Sciences, 104, 40-53, ISSN 0376-0421, https://doi.org/10.1016/j.paerosci.2018.11.003.

Jain, R. V., Aware, M., Junghare, V., A. S. (2017). Tuning of Fractional Order PID controller using particle swarm optimization technique for DC motor speed control, in 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, pp. 1–4.

Jayachandran, R., Ashok, D. S., Narayanan, S. (2013). Fuzzy logic based modelling and simulation approach for the estimation of tire forces, Procedia Engineering, vol. 64,pp 1109-1118, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.2013.09.189.

Kaufman, H., Roy, R., Xu, X. (1984). Model reference adaptive control of drug infusion rate. Automatica. 20 (2), 205-209.

Keel, L. H., Bhattacharyya, S. P. (2008). Controller synthesis free of Analytical models :three term controllers. IEEE Transaction on Automatic Control, 53 (6), 1353-1369. https://doi.org/10.1109/TAC.2008.925810

Kiam, A.H., Chong, G., Yun, L. (2005). PID control system analysis, design, and technology. IEEE Trans Control Syst Technol; 13 (4), 559–76.

Klempka, R., Filipowicz, B., (2018). Optimization of a DC motor drive using a firefly algorithm, in 2018 International Symposium on Electrical Machines, SME 2018, Andrychów, pp. 1–6.

Koivo, A.J., Larnard, D., Gray, R. (1981). Digital control of mean arterial blood pressure in dogs by injecting a vasodilator drug. Ann Biomed Eng. 9 (3), 185-197.

Kok, L. Y., Elamvazuthi, I., Ramani, K.. (2017). Development of a nature inspired algorithm based controller for DC servo motor, in 2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation, ROMA 2017, Kuala Lumpur, pp. 1–5.

Lasserre, J.B. (2001). Global optimization with polynomials and the problem of moments. SIAM J Optim vol.11, no. 3, pp. 796–817

Lendvai, A. Z., Barta, Z., Liker, A., & Bokony, V. (2004). The effect of energy reserves on social foraging: Hungry sparrows scrounge more. Proceedings of the Royal Society of London. Series B: Biological Sciences, 271, 2467–2472.

Lidbe, A. D. Hainen, A. M. and Jones, S. L. (2017). Comparative study of simulated annealing, tabu search, and the genetic algorithm for calibration of the microsimulation model, Simulation, vol. 93 (1), 21–33

Loucif, F., Kechida, S., Sebbagh, A. (2020). Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator. J Braz. Soc. Mech. Sci. Eng. 42, 1. https://doi.org/10.1007/s40430-019-2074-3

MATLAB, (2013). The MathWorks Inc. https://mathworks.com/

Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-Based Systems, 96, 120-133.

Mishra, A. K., Tiwari, V. K., Kumar, R., Verma, T. (2013). Speed control of dc motor using artificial bee colony optimization technique, in 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE), Jabalpur, pp. 1–6.

Mohanty, M.D., Mohanty, M.N. (2020). Design of fuzzy controller for patients in operation theater. Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing. Singapore: Springer. https://doi.org/10.1007/978-981-15-1451-7_57

Ogata, K. (2002). Modern control enginering, Prentice Hall.

Oladipo, S., Sun, Y., Wang, Z. (2020). Optimization of PID controller with metaheuristic algorithms for DC motor drives: review. International Review of Electrical Engineering (IREE), 15 (5), 352. https://doi.org/10.15866/IREE.V15I5.18688

Payakkawan, P., Klomkarn, K., Sooraksa, P.. (2009). Dual-line PID controller based on PSO for speed control of DC motors, in 2009 9th International Symposium on Communications and Information Technology, Icheon, pp. 134–139.

Prabu, M. J., Poongodi, P., Premkumar, K., (2017). Genetic Algorithm Tuned PID controller for rotor Position control of BLDC motor, World Appl. Sci. J., 35 (2), 199–207.

Prommee, P., Angkeaw, K. (2018). High performance electronically tunable log domain current-mode pid controller, Microelectron. J. 72 126-137, http://doi.org/10.1016/j.mejo.2017.09.008.

Quresh, K., Rahnamayan, S., He, Y., Liscano, R. (2019). Enhancing lqr controller using optimized real-time system by gde3 and nsga–ii algorithms and comparing with conventional method, in: 2019 IEEE Congress on Evolutionary Computation (CEC), pp. 2074–2081, http://doi.org/10.1109/CEC.2019.8789894.

Rahman, A. R. A., Rahim A. R. A., Memari A. (2017). Metaheuristic algorithms: guidelines for implementation, J. Soft Comput. Decis. Support Syst., 4, (6), 1–6.

Ren, T. J., Chen, T. C., Chen, C. J. (2008). Motion control for a two-wheeled vehicle using a self-tuning PID controller. Control Engineering Practice, 16 (3), 365-375. https://doi.org/10.1016/j.conengprac.2007.05.007

Rodriguez-Molina, A., Villarreal-Cervantes, M. G., Aldape-Perez, M. (2018). Optimal Adaptive Control of a DC Motor using Differential Evolution variants, in 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, CYBER 2017, Honolulu, HI, pp. 283–288.

Roni, M.H.K., Rana, M.S., Pota, H.R. (2022). Recent trends in bio-inspired meta-heuristic optimization techniques in control applications for electrical systems: a review. Int. J. Dynam. Control . https://doi.org/10.1007/s40435-021-00892-3

SAHU, R. K., Shaw, B. (2018). Design of Solar System by Implementing ALO optimized PID based MPPT controller. Trends in Renewable Energy, 4, 44-55. https://doi.org/10.17737/tre.2018.4.3.0049

Sharma, R., Deepak, K.K., Gaur, P., Joshi. D. (2020). An optimal interval type-2 fuzzy logic control based closed-loop drug administration to regulate the mean arterial blood pressure. Comput Methods Prog Biomed. 185:105167.

Shatnawi, M., Bayoumi, E. (2019). Brushless DC motor controller optimization using simulated annealing, in 2019 International Conference on Electrical Drives & Power Electronics (EDPE), Slovakia, pp. 292–297.

Shehab, M., Abualigah, L., Al Hamad, H. (2020). Moth–flame optimization algorithm: variants and applications. Neural Comput & Applic 32, 9859–9884 . https://doi.org/10.1007/s00521-019-04570-6

Sheta, A., Braik, M., Maddi, D.R., Mahdy, A., Aljahdali, S., Turabieh, H. (2021). Optimization of PID controller to stabilize quadcopter movements using meta-Heuristic search algorithms. *Appl. Sci. 11,* 6492. https://doi.org/10.3390/app11146492

Silva S.J., Scardovelli T.A., Silva, S.R.M., Rodrigues SCM.. (2019). Simple adaptive PI controller development and evaluation for mean arterial pressure regulation. Res Biomed Eng. 35 (2), 157-165.

Singh, V., V. Garg, K. (2014). Tuning of PID controller for speed control of DC motor using soft computing techniques-A Review, Int. J. Appl. Eng. Res., 9 (9), 1141–1148.

Sondhi, S., Hote, Y.V. (2015). Fractional-order PI controller with specific gain–phase margin for MABP control. IETE J Res. 61 (2): 142-153.

Souza, D.A., Batista, J.G., Reis, L.L.N. (2021). PID controller with novel PSO applied to a joint of a robotic manipulator. J Braz. Soc. Mech. Sci. Eng. 43, 377. https://doi.org/10.1007/s40430-021-03092-4

Srivastava, S., Kumar, J., Kumar, V., Rana, KPS. Hasmat, M., Sharma, R. (2018). Design of robuts fractional order fuzzy sliding mode PID controller for two link robotic manipülatör system. Journal of intelligent & fuzzy systems, 35 (5), 5301-5315.

Stalmaster, P.A., Kaiser, K.H. (1997).Winter ecology of bald eagles on the Nisqually River Drainage,Washington. Northwest Sci 71, 214–223.

Su, T.J., Wang, S.M., Vu, H.Q., Jou, J.J., Sun, C.K. (2019). Mean arterial pressure control system using model predictive control and particle swarm optimization. Microsyst Technol. 24 (1), 147-153.

Sundari, K., Maruthupandi, P. (2022). Optimal design of PID controller for the analysis of Two TANK system using metaheuristic optimization Algorithm. J. Electr. Eng. Technol. 17, 627–640 (2022). https://doi.org/10.1007/s42835-021-00891-6

Susperregui, A., Herrero, J.M., Martinez, J. M., Tapia-Otaegui, G., Blasco, X. (2019). Multi-objective optimisation-based tuning of two second-order Sliding-Mode controller variants for DFIGs connected to non-ideal grid voltage" *Energies* 12, no. 19: 3782. https://doi.org/10.3390/en12193782

Vatansever, F., Hacıiskenderoğlu, E. (2022). PID tuning with up-to-date metaheuristic algorithms. Uludağ University Journal of The Facutly of Engineering, 27 (2), 573-584. https://doi.org/10.17482/uumfd.1090766

Wagner, M., Neumann, F. (2013). A fast approximation-guided evolutionary multiobjective algorithm, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, pp.687–94, http://doi.org/10.1145/2463372.2463448.

Watson, J., The golden eagle. 2010: Bloomsbury Publishing.

Wu, Z., Li. D., Xue Y., T. He, and Zheng S.. (2018). Tuning for fractional order PID controller based on probabilistic robustness, IFACPapersOnLine, 51 (4), 675–680.

Xue, D., Chen, Y.Q., Atherton, D.P. (2007). Linear Feedback Control (Analysis and Design with MATLAB), SIAM, USA.

Xue, J., Shen, B. (2020). A novel swarm intelligence optimization approach: sparrow search algorithm, System Science & Control Engineering, 8 (1), 22-34, https://doi.org /10.1080/21642583.2019.1708830

Zhong, S., Chunpeng, K., Dawei, X. (2007). Research of PID parameter selftuning applied in temperature control system, The Eighth International Conference on Electronic Measurement and Instruments, 360-363.

Zhu, J., Liu, H., Ren, Z. (2012). Optimization design of permanent magnetism brushless DC motor governing system based on artificial fish swarm algorithm, in Advanced Materials Research, 546–547, 278–283.

Ziegler, J.G., Nichols, N.B. (1942). Optimum settings for automatic controllers, Transactions of the A.S.M.E., 64, 759-768.

Zumbahlen, H. (2006). Basic Linear Design, Analog Devices, ISBN: 0-915550-28-1. Chapter 1.

ÖZGEÇMİŞ

Adı Soyadı : Emre HACIİSKENDEROĞLU
Doğum Yeri ve Tarihi : Bursa - 21.04.1996
Yabancı Dil : İngilizce

Eğitim Durumu
Lise : Nuri Nihat Aslanoba Anadolu Lisesi
Lisans : Bursa Uludağ Üniversitesi
Yüksek Lisans : Bursa Uludağ Üniversitesi

Çalıştığı Kurum/Kurumlar : Emko Elektronik

İletişim (e-posta) : emre.iskenderoglu1@gmail.com

Yayınları :

Vatansever, F., Hacıiskenderoğlu, E. (2022). PID tuning with up-to-date metaheuristic algorithms. Uludağ University Journal of The Facutly of Engineering, 27 (2), 573-584. https://doi.org/10.17482/uumfd.1090766