



T.C.
BATMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**HENRY GAZ ÇÖZÜNÜRLÜK OPTİMİZASYONU VE FARKLI
SİSTEMLERDEKİ KONTROLÖR TASARIMINA UYGULANMASI**

Veysi KAÇTI

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ocak-2021
BATMAN
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Veysi KAÇTI tarafından hazırlanan “**Henry Gaz Çözünürlük Optimizasyonu ve Farklı Sistemlerdeki Kontrolör Tasarımına Uygulanması**” adlı tez çalışması 04/01/2021 tarihinde aşağıdaki jüri tarafından oy birliği ile Batman Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Dr. Öğr. Üyesi Davut SEVİM

Danışman

Doç. Dr. Serdar EKİNCİ

Üye

Dr. Öğr. Üyesi Cafer BUDAK

İmza

Yukarıdaki sonucu onaylım.

.....
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Veysi KAÇTI

Tarih: 04/01/2021

ÖZET

YÜKSEK LİSANS TEZİ

HENRY GAZ ÇÖZÜNÜRLÜK OPTİMİZASYONU VE FARKLI SİSTEMLERDEKİ KONTROLÖR TASARIMINA UYGULANMASI

Veysi KAÇTI

Batman Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç.Dr. Serdar EKİNCİ

2021, 90 Sayfa

Jüri

Doç. Dr. Serdar EKİNCİ
Dr. Öğr. Üyesi Davut SEVİM
Dr. Öğr. Üyesi Cafer BUDAK

PID kontrol sistemi günümüz endüstrisinde ve sanayilerde sıkça kullanılan bir kontrol yöntemi olarak tercih edilmektedir. Piyasada kolaylıkla tedarik edilip hemen hemen her kontrol sistemiğine uygulanabilen bir sistemdir. Bundan dolayı oransal-integral-türevsel (PID) kontrol sistemini daha da geliştirip farklı algoritmalar ve optimizasyon yöntemleri araştırılıp kullanılmaya başlanmıştır. Sezgisel-üstü algoritmaların en iyi yanları işlem yükünün az olması ve kolay uygulanabilir olmasıdır. Sezgisel üstü algoritmaların farklı bir kullanım alanı da PID kontrolörlerdir. PID kontrolörler ile birlikte sezgisel-üstü algoritmalar kullanıldığında çok daha iyi sonuçlar alınabilmektedir. Bu tezimizde kontrolör olarak PID ve FOPID kontrolörler kullanılmıştır. Üç farklı yapıdaki sistemler için bu kontrolörler ile birlikte sezgisel-üstü algoritmalar ayrı ayrı uygulanmıştır. Bu üç sistemimizden birincisi aircraft pitch sistemi, ikincisi de otomatik gerilim regülatörü (AVR) ve üçüncü sistem ise DC motor hız kontrol sistemidir. Sezgisel-üstü optimizasyon tekniği olan güncel ve etkili HGSO algoritması, bu üç sistemlerdeki kontrolör tasarımına başarılı bir şekilde uygulandı. Ayrıca HGSO algoritma sonuçları, literatürdeki diğer çalışmalar ile karşılaştırıldı ve daha üstün sonuçlar elde edildi.

Anahtar Kelimeler: HGSO / OBL-HGSO Algoritması, Sezgisel-üstü Algoritmalar, Aircraft Pitch Kontrol Sistemi, AVR Sistemi, DC Motor Sistemi, PID / FOPID Kontrolör

ABSTRACT

MS/Ph.D THESIS

HENRY GAS SOLUBILITY OPTIMIZATION AND IT IS APPLICATION TO CONTROLLER DESIGN IN DIFFERENT SYSTEMS

Veysi KAÇTI

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF BATMAN UNIVERSITY THE DEGREE OF MASTER OF SCIENCE OF ELECTRICAL AND ELEKTRONICS ENGINEERING

Advisor: Assoc.Prof.Dr. Serdar EKİNCİ

2021, 90 Pages

Jury

**Assoc. Prof. Dr. Serdar EKİNCİ
Asst. Prof. Dr. Davut SEVİM
Asst. Prof. Dr. Cafer BUDAK**

The PID control system is preferred as a control method that is frequently used in today's industry and industries. It is a system that can be easily supplied in the market and can be applied to almost any control systematic. Therefore, the proportional-integral-derivative (PID) control system has been further developed and different algorithms and optimization methods have been researched and used. The best things about meta-heuristic algorithms are low processing load and easy implementation. Another area of use of meta-heuristic algorithms is PID controllers. Much better results can be obtained when using meta-heuristic algorithms with PID controllers. In this thesis, PID and FOPID controllers are used as controllers. For systems of three different structures, these controllers and meta-heuristic algorithms have been applied separately. The first of these three systems is the aircraft pitch system, the second is the automatic voltage regulator (AVR) and the third is the DC motor speed control system. The current and effective HGSO algorithm, which is a meta-heuristic optimization technique, has been successfully applied to the controller design in these three systems. In addition, HGSO algorithm results were compared with other studies in the literature and superior results were obtained.

Keywords: HGSO / OBL-HGSO Algorithms, Meta-heuristic Algorithms, Aircraft Pitch Control System, AVR System, DC Motor System, PID / FOPID Controller

ÖNSÖZ

Bu tez çalışmasının şekillenmesinde her türlü bilgi ve tecrübesini esirgemeyen çok değerli hocama sayın Doç. Dr. Serdar Ekinci'ye teşekkürlerimi sunarım. Çalışma hayatımdaki yoğunluktan kaynaklı göstermiş olduğu anlayış ve desteklerinden dolayı da çok teşekkür ederim.

Ayrıca her yönüyle desteğini hiç esirgemeyen çok değerli aileme teşekkürlerimi sunarım.

Veysi KAÇTI
BATMAN-2021

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	iv
SİMGELER VE KISALTMALAR	vi
1. GİRİŞ	1
1.1. Literatür.....	3
2. PID KONTROLÖR TASARIMI	6
2.1. PID Kontrolörde Oransal Terim.....	7
2.2. PID Kontrolörde İntegral Terim.....	8
2.3. PID Kontrolörde Türevsel Terim	9
2.4. PID Genel Özet:	10
3. FOPID KONTROLÖR	11
3.1 FOPID Denetleyicisinin Temelleri	11
3.2 FOPID denetleyicisinin tamsayı sıra yaklaşımı	12
4. PID KONTROLÖRLÜ AIRCRAFT PİTCH SİSTEMİ	14
4.1. Aircraft Pitch Sistem Modellemesi	17
4.2. Aircraft Pitch Sistem Analizi	18
4.2.1. Kontrol edilebilirlik.....	19
4.2.2. Ayrık durum uzayı	20
5. FOPID KONTROLÖRLÜ AVR SİSTEMİ	22
6. PID KONTROLÖRLÜ DC MOTOR HIZ KONTROL SİSTEMİ	24
7. SEZGİSEL-ÜSTÜ ALGORİTMALAR	26
7.1. Henry Gaz Çözünürlük Optimizasyonu (HGSO) Algoritması	26
7.1.1 Önerilen OBL / HGSO algoritması.....	28
7.2. Sinüs Kosinüs Algoritması (SCA)	30
7.3. Çekirge Optimizasyon Algoritması (GOA)	32
7.4. Parçacık Sürü Optimizasyon Algoritması (SSA)	35
7.5. Stokastik Fraktal Arama Algoritması (SFS)	36
7.6. Geliştirilmiş Böbrek Esintili Algoritma (IKA)	36
7.6.1. Böbrek Esintili Algoritma (KA)	36

7.6.2.	Geliştirilmiş Böbrek Esintili Algoritma (IKA).....	39
7.7.	Atom Arama Optimizasyon Algoritması (ASO).....	40
7.8.	Gri Kurt Optimizasyon Algoritması (GWO)	44
8.	PİTCH KONTROL SİSTEMİ SİMÜLASYON SONUÇLARI.....	47
8.1.	Amaç Fonksiyonu ve Önerilen Tasarım.....	47
8.2.	İstatistiksel Test Analizi	48
8.3.	Geçici Hal Cevap Analizi.....	49
8.4.	Kutup-sıfır Haritası Analizi.....	50
8.5.	Bode Analizi.....	52
9.	AVR KONTROL SİSTEMİ SİMÜLASYON SONUÇLARI.....	53
9.1.	Amaç Fonksiyonu ve Tasarımı	53
9.2.	Kutu Grafiği ve Yakınsama Profil Analizleri	54
9.3.	Geçici Hal Analizi	55
9.4.	Frekans Tepki Analizi	57
10.	DC MOTOR HIZ KONTROL SİSTEMİ SİMÜLASYON SONUÇLARI..	59
10.1.	DC Motor Optimizasyonu	59
10.2.	İstatistiksel Test Analizi	62
10.3.	Yakınsama Profili	63
10.4.	Aşma, Yükselme Süresi ve Oturma Süresinin Karşılaştırılması	64
10.5.	Performans Endeksinin Karşılaştırılması	67
10.6.	Frekans Alanı Analizlerinin Karşılaştırılması	67
10.7.	Gürebüzlük Analizinin Karşılaştırılması	68
10.8.	Yük Bozulma Analizinin Karşılaştırılması.....	73
10.9.	Kontrolörlerde Enerji Ve Maksimum Kontrol Sinyalinin Karşılaştırılması.	73
11.	GENEL SONUÇLAR VE ÖNERİLER.....	75
	KAYNAKLAR.....	76
	ÖZGEÇMİŞ.....	80

SİMGELER VE KISALTMALAR

Simgeler

α = hücum açısı (Angle Of Attack)

q = Eğim Oranı (Pitch Angle)

θ = Eğim Açısı (Pitch Angle)

δ = Sapma Açısı (Elevator Deflection Angle)

$$\mu = \frac{\rho S t}{4m}$$

ρ = Hava Yoğunluğu (Density Of Air)

S = Kanadın Platform Alanı (Platform Area Of The Wing)

\bar{c} = Ortalama Akor Uzunluğu (Average Chord Length)

m = Uçak Kütlesi (Mass Of Aircraft)

$$\Omega = \frac{2U}{t}$$

U = Uçuş Denge Hızı (Equilibrium flight speed)

C_T = İtme Katsayısı (Coefficient Of Thrust)

C_D = Sürüklenme Katsayısı (Coefficient Of Drag)

C_L = Kaldırma Katsayısı (Coefficient Of Lift)

C_W = Ağırlık Katsayısı (Coefficient Of Weight)

C_T = Eğim Momenti Katsayısı (Coefficient Of Pitch Moment)

γ = Uçuş Yolu Açısı (Flight Path Angle)

$$\sigma = \frac{1}{1+\mu C_L} = \text{Sabit}$$

\dot{I}_{yy} = Normalleştirilmiş Eylemsizlik Momenti (Normalized Moment Of Inertia)

$\eta = \mu \sigma C_M = \text{Sabit}$

Kısaltmalar

PID: Doğrusal İntegral Türev (Proportional İntegral Derivative)

HGSO: Henry Gaz Çözünürlük Optimizasyonu (Henry Gas Solubility Optimization)

SCA: Sinüs Cosinüs Algoritması (Sine Cosine Algorithm)

Z-N: Ziegler- Nichols

GOA: Çekirge Optimizasyon Algoritması (Grasshopper Optimization Algorithm)

SSA: Parçacık Sürü Optimizasyon Algoritması (Salp Swarm Algorithm)

SFS: Stokastik Fraktal Arama Algoritması (Stochastic Fractal Search Algorithm)

IKA: Geliştirilmiş Böbrek Esintili Algoritma (Improved Kidney-Inspired Algorithm)

ASO: Atom Arama Optimizasyon Algoritması (Atom Search Optimization Algorithm)

GWO: Gri Kurt Optimizasyon Algoritması (Grey Wolf Optimization Algorithm)

Kısaltmalar, tez içindeki kullanımlarında orijinal adları üzerinden yapılmıştır.

1. GİRİŞ

Günümüzde hızla gelişen teknolojiye karşın kontrolörler büyük öneme sahip olmaya başlamıştır. Kontrolörlerin büyük öneme sahip olmasından sonra bu kontrolörlerin en uygun şekilde tasarlanması ve optimize edilmesi söz konusu olmuştur. Bu gelişmelerden sonra özellikle en çok kullanılan PID kontrolör sisteminin geliştirilmesi ön plana çıkmıştır. PID kontrolör sistemini en iyi şekilde optimize etmek için çeşitli sezgisel-üstü algoritma çalışmaları olmuştur. Henry gaz çözünürlüğü algoritması (HGSO)'da bu sezgisel-üstü algoritmalar arasında büyük öneme sahiptir. Bu tezimizde özellikle HGSO algoritmasının farklı üç sistem üzerindeki etkisini detaylı bir şekilde inceleyeceğiz.

Henry gaz çözünürlüğü optimizasyonu (HGSO), Henry'nin sıvı ortamdaki gaz çözünürlüğü davranışını ifade etmek için ortaya koyduğu kanunu taklit ederek meydana getirilmiş fizik tabanlı bir algoritma olup çeşitli test fonksiyonları ve üç adet de gerçek dünya optimizasyon problemi üzerinde başarıyla test edilmiştir [1]. Söz konusu bu algoritma aynı zamanda motif keşif problemi [2], otomobil fren bileşenlerinin optimum yapısal tasarımı [3] ve özellik seçimi [4] için de kullanılmıştır. HGSO algoritmasını PID ve FOPID kontrolör kullanarak farklı sistemler üzerindeki davranışını incelerken aynı zamanda önerilen bir yaklaşım olan karşıt tabanlı öğrenme olan OBL/HGSO algoritması da üçüncü sisteme uygulanacaktır. İlk sistemimiz HGSO sisteminin uçak eğim kontrol sistemine uygulanması, ikinci sistemimiz HGSO algoritması ile birlikte FOPID kontrolör kullanılarak otomatik gerilim regülatörü sistemine uygulanması ve son sistem olarak ise OBL/HGSO algoritması kullanılarak DC motor hız kontrol sistemine uygulanacaktır. Bu üç sistem aşağıda detaylı bir şekilde anlatılmıştır.

Birinci sistemimiz olan uçak eğim kontrol sistemi, uçağın en güvenli, ekonomik ve konforlu bir uçuş deneyimi sağlamasıdır. Uçuşun başladığı ilk senelerde en önemli etkenlerden biri güvenlikti. Halihazırda uçuş sayılarının artması ve seyahat için uçak kullanan kişilerin bir hayli artması bu güvenlik durumunu daha da önemli bir hal almasına sebep olmuştur. Genel olarak uçak sistemine baktığımızda kanatçık, dümen ve havalanma gibi kontrol yüzeylerini kontrol ederek üç boyutlu bir düzlemde uçmaktadır. Bu kontrol yüzeyleri ise uçağın yalpalanma, eğim ve sapma gibi eksenlerdeki hareketini kontrol eder. Kaldırıcılar ise genel olarak uçağın kuyruğuna yerleştirilir ve uçağın eğimini ve dalma açısını değiştirerek uçağın yönünü bulası için kontrol eder [5] [6]. Doğrusal olmayan ve sürekli değişebilen belirsiz uçuş dinamikleri uçuş kontrol sisteminin tasarımı oldukça

zordur. Bundan dolayı belli uçuş şartları baz alınarak uçak dinamiği doğrusal hale getirilip, lineer olmayan bu sistemler lineer kontrolörler ile kontrol edilmeye çalışılmaktadır. Eğim yani alçalma/yükselme açısı, bir uçağın kalkışında ve inişinde önemli rol oynar. Son yıllarda, parçacık sürü optimizasyonu [6], genetik algoritma [7], [8], arı algoritması [9], yarası algoritması [10] ve bakteriyel yemleme optimizasyonu [6] gibi birçok sezgisel-üstü optimizasyon tekniği bir uçağın eğim açısını yani alçalma/yükselme açısını kontrol etmek için kullanılmıştır. Bu noktada belirtilmesinde fayda görülen husus; bu alanda bu çalışmaya benzer bir biçimde çalışmanın daha önce literatürde önerilmemiş olmasıdır. Bu sebeple söz konusu önerilen HGSO tabanlı PID kontrolörün etkinliğini gösterebilmek için, güncel sezgisel-üstü algoritmalarından sinüs kosinüs algoritması (SCA) ve çekirge sürüsü optimizasyon algoritması (GOA) ayarlı PID kontrolörlerle karşılaştırılmıştır.

İkinci sistemimiz olan otomatik gerilim regülatörü (AVR) sistemidir. Bu sistemde temel anlamda terminal gerilimi ayarlamak için kullanılmaktadır. Bir jeneratörün terminal voltajını ayarlamak ve izin verilen sınırlar içinde tutmak için bir AVR sistemi kullanılır, böylece güç şebekesinin / santralin stabilizasyonuna yardımcı olur [11]. Bu nedenle, bir AVR'den daha iyi bir performans elde etmek için uygun kontrol teknikleri gereklidir. AVR kullanarak bir jeneratörün terminal voltaj seviyesini kontrol etmek için önerilen birkaç kontrolör yapısı bulunmaktadır. Bunlar arasında PID ve FOPID denetleyicileri en çok tercih edilenlerdir [12]. Bununla birlikte, bir PID denetleyicisi, bir sistemin dinamik performansını iyileştirme açısından bir FOPID denetleyicisi kadar iyi değildir, çünkü ikincisi esas olarak tamsayı olmayan türev ve integral terimlere sahip bir PID'dir. Bu nedenle, sistemde dinamik belirsizlikler olması durumunda daha iyi kontrol performansı ve gelişmiş sistem sağlamlığı sağlama avantajına sahiptir [13], [14]. Yine de avantajlarından yararlanabilmek için bir FOPID kontrolörünün optimal parametrelerini elde etmek önemlidir [15]. Sezgisel-üstü algoritmalar, PID'yi ayarlamak için halihazırda kullanılan optimizasyon tekniklerinden biridir. Algoritmaları kullanmanın amacı, AVR için kullanılan denetleyici performansını daha da iyileştirmektir. Her açıdan diğerlerinden daha iyi performans gösteren bir optimizasyon algoritması olmadığından, araştırmacı her zaman mümkün olduğunda yenilerini deneyecektir. AVR sistemi için optimal bir FOPID denetleyicisi tasarlamak için HGSO algoritması kullanılmıştır. Önerilen ayarlama yönteminin verimliliği, kutu grafiği ve yakınsama analizleri ile doğrulanmıştır. Ayrıca, performans üstünlüğü geçici ve frekans tepki analizleri ile gösterilmiştir.

Üçüncü sisteminiz ise DC motor hız kontrol sistemidir. Doğru akım (DC) motor hızı kontrolü, performans değerlendirmeleri ve karşılaştırmaları için gözlemlenebilir bir test yatağı sağladığından, son on yılda gerçek dünya mühendislik uygulaması olarak birçok sezgisel-üstü algoritma için bir uygulama alanı olmuştur. Böyle bir gerçek dünya mühendislik uygulamasını göstermek için temel gereksinimler, uygun bir kontrolör, objektif bir fonksiyon ve bir optimizasyon algoritmasıdır. Ancak, optimizasyon algoritmaları açısından aynı kontrolörler için bile çeşitli seçenekler mevcuttur. Optimizasyon algoritma çalışmalarının ortak amacı, maksimum aşım, yerleşme süresi, yükselme süresi ve kararlı durum hatası gibi geçici ve kararlı durum kriterlerini en aza indirmek için bir optimizasyon algoritması aracılığıyla en uygun şekilde tasarlanmış bir kontrolörden yararlanmak ve sonuç olarak daha iyi bir hız elde etmektir. DC motorun kontrol performansı. Belirtilen hedefe ulaşmak için bir denetleyicinin en iyi kazanç parametrelerini elde etmek önemlidir, ancak performansı iyileştirmek için önerilen algoritmalar, sundukları zayıflıklar nedeniyle optimum denetleyici kazanımlarını bulamayabilir. Bu sistemde kullanılan HGSO'nun karşıt tabanlı öğrenme Henry gaz çözünürlük optimizasyonu (OBL/HGSO) olarak adlandırılan değiştirilmiş bir versiyonudur. OBL'nin HGSO'nun optimizasyon özellikleriyle birlikte kullanılması, standart HGSO'nun doğruluğunu ve performansını önemli ölçüde artırır. Bu nedenle, iyi optimizasyon yetenekleri korunurken standart HGSO'nun dezavantajları bu tür iyileştirmelerle ortadan kaldırılır. Önerilen HGSO-PID ve OBL / HGSO-PID yaklaşımlarının performansı ASO-PID [16], SFS-PID [17], GWO-PID [18] ve SCA-PID [19] ile tezde ayrıntılı olarak karşılaştırıldı. Geçici ve frekans tepkilerinin karşılaştırmalı sonuçları, sağlamlık ve yük bozucu reddetme analizlerinin tümü, önerilen karşıt tabanlı tasarımın, yani OBL/HGSO-PID denetleyicisinin etkinliğini ve diğer algoritmalara üstünlüğünü açıkça doğruladı.

1.1. Literatür

HGSO optimizasyon yöntemi özellikle son yıllarda teknolojinin de gelişmesiyle karşımıza çıkan bir yaklaşımdır. Mühendislik, işletme ve ekonomi gibi dallarda son dönemlerde karmaşık optimizasyon problemleri ortaya çıkmaya başlamıştır. Doğa bu gibi optimizasyona ihtiyaç duyulan karmaşık problemlerde farklı alternatif çözümler ile kaynaklar sağlamaktadır [20]. Bundan dolayı son yıllarda araştırmacılar özellikle doğayı taklit eden doğadan esinlenmiş birçok sezgisel-üstü algoritmalar geliştirmeye

başlamışlardır. Sezgisel-üstü yöntemler bir fonksiyonu defalarca çağırabilirler. Çağırdıkları fonksiyonlarda en optimum çözümü bulmaya çalışırlar. Çağırdıkları fonksiyonu her defasında biraz daha daraltarak en optimum sonucu elde etmek için en etkili çözümü bulmaya çalışırlar [20]. Geliştirilen algoritmalar arasındaki temel fark, keşif (global arama yeteneği) ile sömürü arasındaki denge (optimal çözüm çevresinde yerel arama yeteneği) yaklaşımında yatmaktadır [20].

Doğadan ilham alan algoritmaların birçoğu biyoloji biliminde ilham almaktadırlar. Biyoloji tabanlı algoritmalar sürü davranışlarını doğrudan kullanmamaktadırlar. Genetik algoritmaları [21], biyolojideki doğal seçim sürecinden ilham alırlar. İstilacı Yabani Ot Algoritması [22], aynı şekilde istilacı yabani otlardan ilham almaktadır. Atmosfer Bulutları Modeli Optimizasyon Algoritması [23], da aynı şekilde doğadan aynı bulutların hareketinden ilham almaktadırlar. Biyolojiden ilham alan algoritmalar bazılarını ise sürü zekasından ilham alınmıştır. Parçacık Sürü Optimizasyonu [24], kuş ve balık sürülerinden ilham almışlardır. Karınca Koloni Algoritması [25], karıncaların yiyecek ararken en kısa yolu bulma davranışlarını taklit etmektedir. Yapay Arı Koloni Algoritması [26], bal arılarının davranışlarını taklit etmektedir. Gri Kurt Optimizasyon Algoritması [27], gri kurtların avlanma tekniğini taklit etmektedir. Balina Optimizasyon Algoritması [28], özellikle kambur balinaların kabarcık ağı avlanma stratejisine dayanmaktadır.

Aslında tüm bu sezgisel-üstü algoritmalar bir matematiksel model kullanmasına rağmen matematiksel kurallardan ilham alan optimizasyon algoritmalarının sayıları oldukça azdır. Bunlardan biri Baz Optimizasyon Algoritması [29], optimum çözüm bulmak için yer değiştirme parametresi ile birlikte temel aritmetik operatörlerin kombinasyonunu kullanmaktadır.

Birinci sistemimiz olan uçak eğim (alçalma/yükselme) açısı, bir uçağın kalkışında ve inişinde önemli bir rol oynamaktadır. Son yıllarda bu sistemle ilgili, genetik algoritma [7], parçacık sürü optimizasyonu [6], [8], arı algoritması [9], yarasa algoritması [10] ve bakteriyel yemleme optimizasyonu [6] gibi birçok sezgisel-üstü optimizasyon teknikleri de bir uçağın alçalma/yükselme açısını kontrol etmek için kullanılmıştır. Meta sezgisel algoritmalar, sadece PID'yi değil aynı zamanda FOPID'yi ayarlamak için halihazırda kullanılan optimizasyon tekniklerinden biridir. İkinci sistemimiz olan AVR sisteminde kullandığımız FOPID kontrolünde, kaotik karınca sürüsü [13], guguklu arama [30], salp sürüsü algoritması [31], benzetilmiş tavlama optimizasyonu [32] ve sinüs-kosinüs algoritması [33] gibi FOPID'yi ayarlamak için meta-sezgisel algoritmalar da

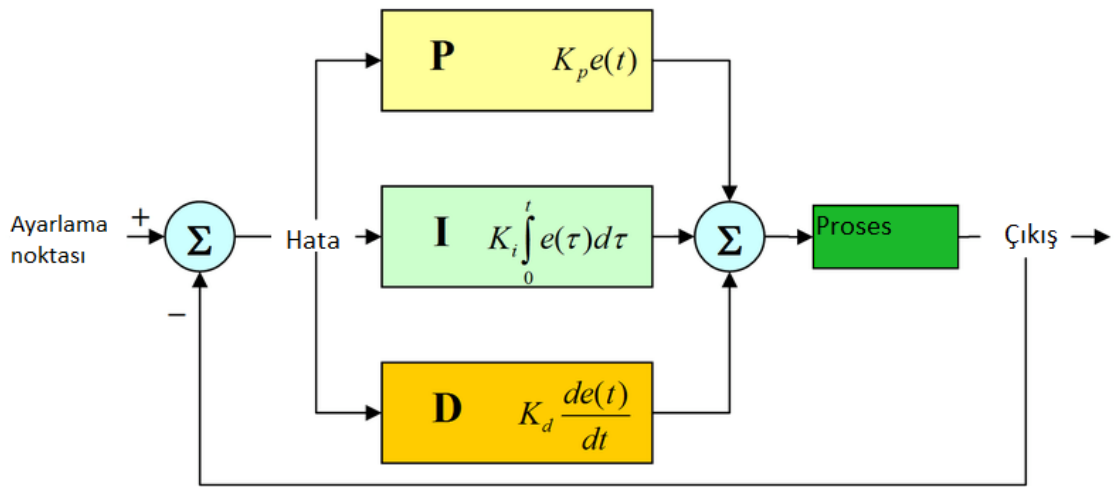
kullanılmıştır. Algoritmaları kullanmanın amacı, AVR için kullanılan denetleyici performansını iyileştirmektir. Üçüncü sistemimiz olan DC motor hız kontrol sisteminde daha önce de, yapay arı kolonisi (ABC) [34], atom araması (ASO) ve kaotik versiyonu (ChASO) [16], yarası (BA) [35], geliştirilmiş böcek anten araması (BAS) [36], diferansiyel evrim (DE) [35,37], geliştirilmiş DE [38], ateş böceği (FFA) [35], çiçek tozlaşması (FPA) [39], genetik (GA) [40], yerçekimsel arama (GSA) [41], gri kurt (GWO) [18,42,43], istilacı ot (IWO) [44], Jaya (JOA) [45], parçacık sürüsü (PSO) [35,46], kısıtlı PSO (CPSO) [47], dinamik PSO (dPSO) [48], salp sürüsü (SSA) [49], sinüs kosinüs (SCA) [19], geliştirilmiş SCA (ISCA) [50], stokastik fraktal arama (SFS) [51,17] ve kurt araması (WSA) [35], hız kontrolü için bir DC motorun kontrolör parametrelerini ayarlamak için önerilen optimizasyon algoritmalarıdır.

Bu çalışmada da kullanılan özellikle HGSO(henry gaz çözünürlük optimizasyonu) Henry gaz çözünürlük kanuna dayalı sezgisel-üstü bir algoritmadır [1].

2. PID KONTROLÖR TASARIMI

- Bir orantılı-integral-türev kontrolörü, endüstriyel kontrol sistemlerinde yaygın olarak kullanılan bir kontrol döngüsü olan geri besleme mekanizmasıdır.
- PID, en sık kullanılan geribildirim denetleyicisidir.
- Bir PID kontrol cihazı, ölçülen proses değişkeni ile istenen ayar noktası arasındaki fark olarak "hata" değerini hesaplar. Kontrolör, proses kontrol girişlerini ayarlayarak hatayı minimize etmeye çalışır.

Genel PID kontrolör şematiği şekil 2.1’de gösterilmiştir.



Şekil 2.1 Genel PID şematiği [52]

Genel PID Kontrol Şematiği

- PID kontrol cihazı hesaplaması üç ayrı parametre içerir ve buna göre bazen üç terim kontrolü olarak adlandırılır: orantılı, integral ve türev değerleri, P, I ve D olarak adlandırılır.
- P mevcut hataya, K geçmiş hataların birikimine bağlıdır ve D mevcut değişim oranına bağlı olarak gelecekteki hataların tahminidir.
- Bu üç işlemin ağırlıklı toplamı işlem değişkenini ayarlamak için kullanılır. Şekilde de görüldüğü üzere $\Sigma(t)$ aşağıdaki denklemde de ifade edilmiştir.

$$\Sigma(t) = P_{out} + I_{out} + D_{out}$$

- P_{out} , I_{out} ve D_{out} üç terimin her birinin PID denetleyicisinin çıktısına katkılarıdır[52].

2.1.PID Kontrolörde Oransal Terim

Oransal Terim:

$$P_{out} = K_p e(t) \quad (2.1)$$

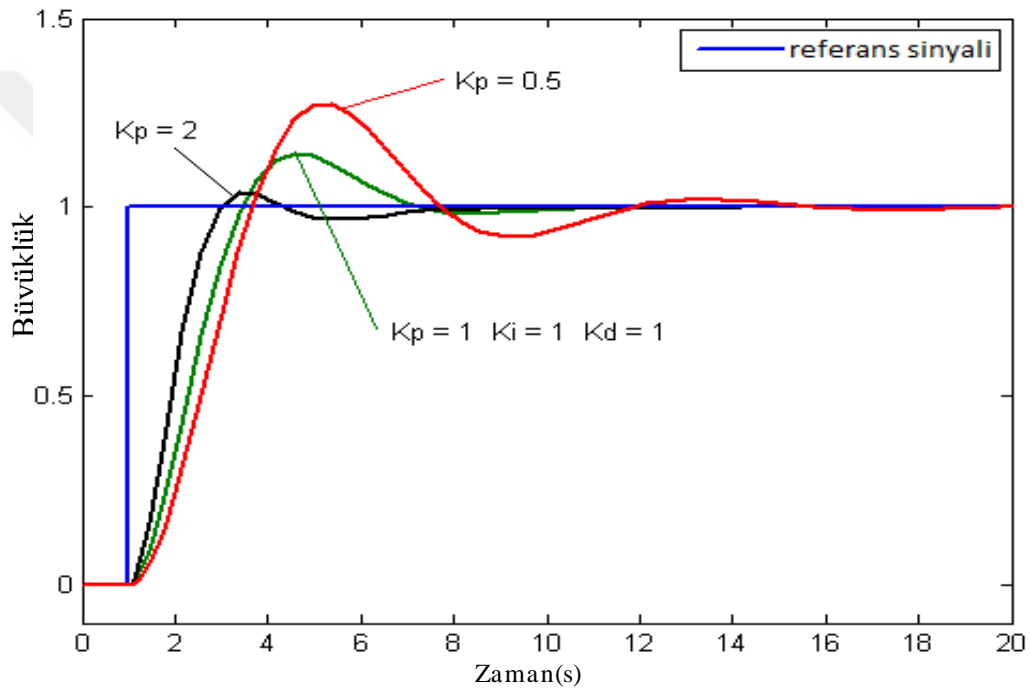
P_{out} : Oransal çıktı terimi.

K_p : Oransal kazanç, ayar parametresi.

PV: Proses değeri (veya proses değişkeni), ölçülen değer.

e: Hata = SP - PV.

t: Zaman veya anlık zaman (şimdiki zaman)



Şekil 2.2 Genliğin Zamana Karşı K_p Grafiği (K_i ve K_D sabit tutulmuş) [52]

Şekil 2.2'de K_i ve K_D sabit tutularak K_p parametresi de değiştirilerek sistem üzerindeki etkisi gözlemlenmiştir.

- Yüksek orantılı kazanç, hatadaki belirli bir değişiklik için çıktıda büyük bir değişiklikle sonuçlanır.
- Oransal kazanç çok yüksekse, sistem kararsız hale gelebilir.
- Eğer oransal kazanç çok düşükse, sistem bozukluklarına cevap verirken kontrol işlemi çok işlevsel olmayabilir[52].

2.2.PID Kontrolörde İntegral Terim

İntegral Terim:

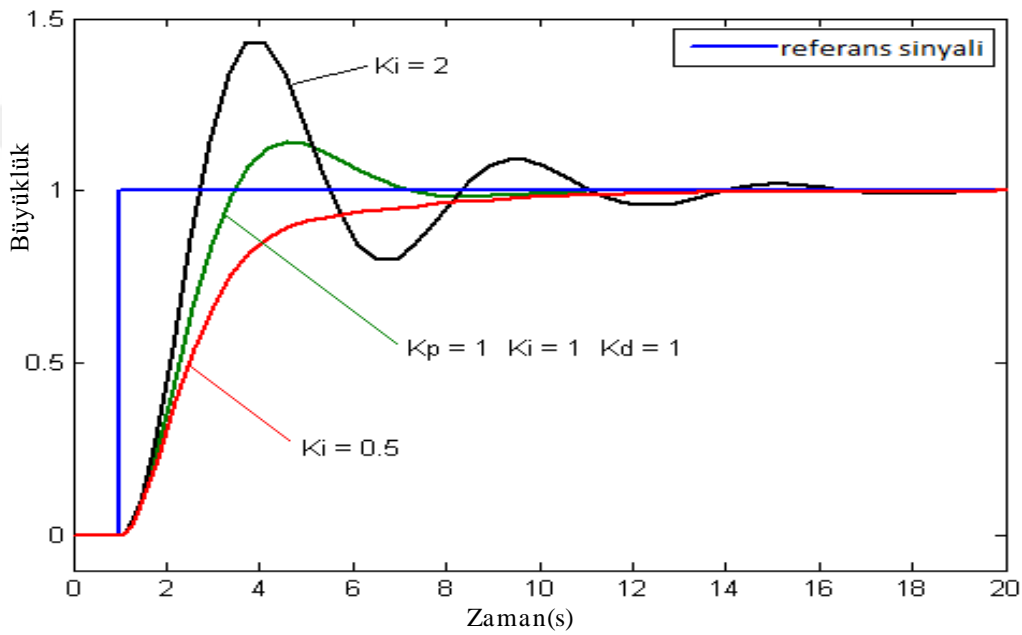
$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (2.2)$$

I_{out} : İntegral çıktı terimi

K_i : İntegral kazanç, bir ayarlama parametresi

e: Hata = SP - PV.

t: Zaman veya anlık zaman (şimdiki zaman)



Şekil 2.3 Genliğin Zamana Karşı K_i Grafiği (K_p ve K_D sabit tutulmuş)[52]

Şekil 2.3'de K_p ve K_D sabit tutularak K_i parametresi de değiştirilerek sistem üzerindeki etkisi gözlemlenmiştir.

- İntegral teriminin katkısı hem hatanın büyüklüğü hem de hatanın süresi ile orantılıdır.
- Zaman içindeki anlık hatanın toplanması (hatanın bütünleştirilmesi), daha önce düzeltilmesi gereken birikmiş ofseti verir.
- Biriken hata daha sonra integral kazanç ile çarpılır ve kontrolör çıkışına eklenir.
- İntegral teriminin genel kontrol eylemine katkısının büyüklüğü, integral kazancı K_i tarafından belirlenir.

- İntegral terimi (orantılı terime eklendiğinde) işlemin ayar noktasına doğru hareketini hızlandırır ve yalnızca orantılı bir denetleyici ile oluşan kalıcı sabit durum hatasını ortadan kaldırır.
- Bununla birlikte, integral terimi geçmişten gelen birikmiş hatalara cevap verdiği için, mevcut değerin ayar noktası değerini aşmasına neden olabilir[52].

2.3.PID Kontrolörde Türevsel Terim

Türevsel Terim:

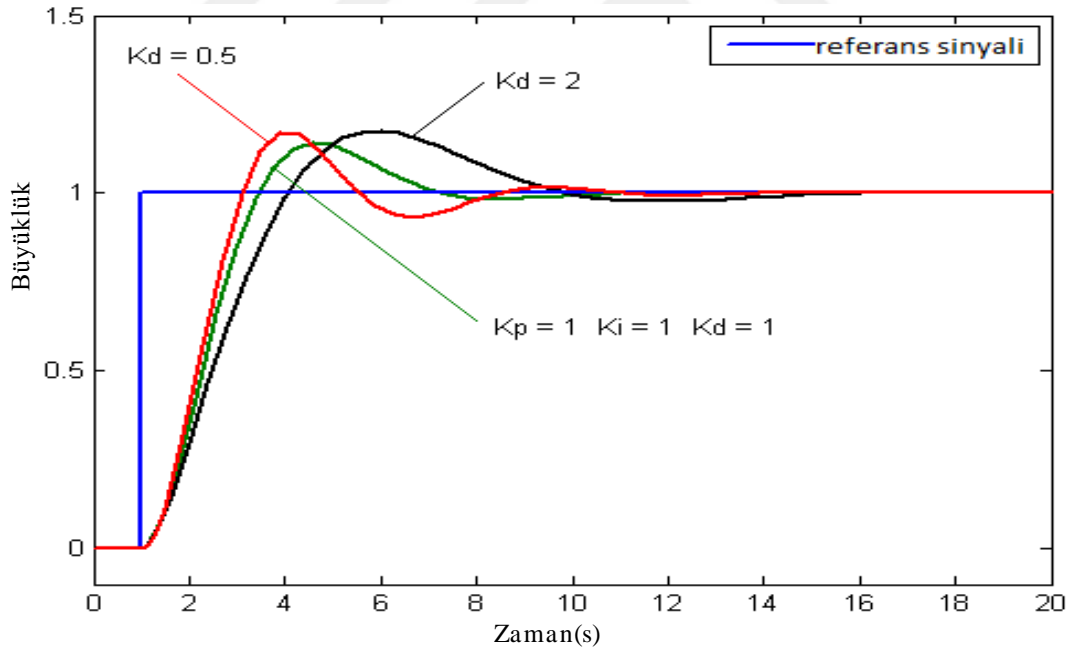
$$D_{out} = K_d \frac{d}{dt} e(t) \quad (2.3)$$

D_{out} : Türev çıktısı terimi

K_d : Türev kazancı,

e: Hata = SP - PV.

t: Zaman veya anlık zaman (şimdiki zaman)



Şekil 2.4 Genliğin Zamana Karşı K_D Grafiği (K_P ve K_I sabit tutulmuş) [52]

Şekil 2.4'de K_I ve K_P sabit tutularak K_D parametresi de değiştirilerek sistem üzerindeki etkisi gözlemlenmiştir.

- Türev terimi, denetleyici çıkışındaki değişim oranını yavaşlatır ve denetleyici çıkışını set değerine yakınlaştırmaya çalışır.

- Bu nedenle, türev kontrol, integral bileşen tarafından üretilen aşmanın büyüklüğünü azaltmak ve birleşik kontrolör işlem kararlılığını geliştirmek için kullanılır [52].

2.4.PID Genel Özet:

Oransal kazanç, K_p :

- Büyük değerler tipik olarak daha hızlı tepki anlamına gelir, çünkü hata büyüdükçe, orantılı terim telafisi artar.
- Aşırı büyük orantılı kazanç, işlem kararsızlığına ve salınımına yol açacaktır[52].

İntegral kazanç, K_i :

- Daha büyük değerler kalıcı durum hatalarını daha çabuk ortadan kaldırır.
- Deneme devri daha büyüktür: geçici müdahale sırasında bütünleşen negatif hatalar, sabit duruma ulaşmadan önce pozitif hatalarla ortadan kaldırılmalıdır[52].

Türev kazanç, K_d :

- Daha büyük değerler aşmayı azaltır, ancak geçici cevabı yavaşlatır ve hatanın farklılaşmasında sinyal gürültüsünün yükselmesi nedeniyle kararsızlığa neden olabilir[52].

3. FOPID KONTROLÖR

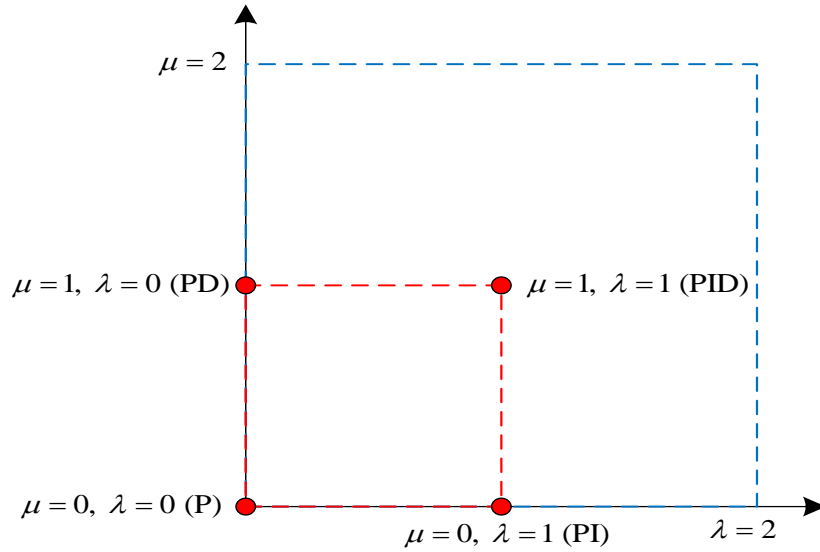
Burada kesir dereceli PID kontrolör tasarımı özetçe anlatılacaktır.

3.1 FOPID Denetleyicisinin Temelleri

Bir FOPID denetleyici, geleneksel PID denetleyicinin genişletilmiş bir sürümüdür. Fraksiyonel denetleyiciler, denetlenen sistem ve denetleyicinin parametrelerindeki değişikliklere daha az duyarlıdır [53] [54]. Fraksiyonel bir kontrolör, izo-sönümlenme özelliğini çok kolay bir şekilde elde edebilir, yani sistem yanıtı belirli bir frekans aralığında aynı faz marjına sahip olabilir. FOPID denetleyicisinin genelleştirilmiş transfer işlevi aşağıdaki gibi verilmiştir.

$$G_{FOPID}(s) = K_P + K_I s^{-\lambda} + K_D s^{\mu}, \quad (\lambda, \mu > 0) \quad (3.1)$$

K_P , K_I ve K_D sırasıyla orantılı, integral ve türev kazanç sabitleri olduğundan, λ integral terimin kesirli mertebesidir ve μ türev terimin hizipsel sırasındır. Genel olarak, kesirli siparişlerin aralığı 0 ile 2 arasındadır [14]. Eğer $\lambda = 1$ ve $\mu = 1$ ise, elde edilen kontrolör geleneksel bir PID kontrolördür; $\lambda = 0$ ve $\mu = 1$ ise, bu bir PD kontrolörüdür; $\lambda = 1$ ve $\mu = 0$ ise, bu bir PI kontrolörüdür; ve $\lambda = 0$ ve $\mu = 0$ ise, bu bir P denetleyicidir. Bu geleneksel tip kontrolörlerin tümü, FOPID kontrol cihazının özel durumlarıdır. FOPID kontrolörünün grafik temsili Şekil 3.1'de gösterilmektedir, burada x eksenini integral terimin kesirli sırasını temsil ederken y eksenini türev terimin kesirli sırasını gösterir. Bu şekil, FOPID denetleyicisi ile standart PID denetleyicisi arasındaki ilişkiyi açıklamaktadır. Şekilden görülebileceği gibi, FOPID kontrolörü tamsayı sıralı PID kontrol cihazını bir noktadan düzleme genişletir. Bu uzantı, kontrolör tasarımına daha fazla esneklik katarak gerçek yaşam süreçlerinin daha doğru kontrolüne izin verir [55].



Şekil 3.1 FOPID denetleyicisinin düzlemi.

3.2 FOPID denetleyicisinin tamsayı sıra yaklaşımı

Denklem (3.1) 'de s^r ($r \in -\lambda, \mu$) kesirli mertebeli Laplace dönüşüm değişkenidir. Bu, kesin analitik çözümü olmayan soyut bir kavram olduğundan, bu değişkenin çözümü ancak tamsayı sıralı bir yaklaşımla bulunabilir. Oustaloup'un özyinelemeli yaklaşım yöntemi CRONE [56], iyi bilinen tamsayı mertebeli yaklaşım yöntemlerinden biridir. CRONE, Fransızcadaki "Commande Robuste d'Ordre Non Entier" in kısaltmasıdır ve bu, fraksiyonel sağlam kontrol anlamına gelir. Kutuplar ve sıfırlar özyinelemeli olarak dağıtılmış CRONE modeli aşağıdaki gibi tanımlanır [56] [57].

$$s^r \cong C \prod_{i=1}^N \frac{1 + \frac{s}{\omega_{z,i}}}{1 + \frac{s}{\omega_{p,i}}} \quad , \quad r > 0 \quad (3.2)$$

$$\omega_{z,i} = \omega_l \left(\frac{\omega_h}{\omega_l} \right)^{\frac{(2i-1-r)}{2N}} \quad (3.3)$$

$$\omega_{p,i} = \omega_l \left(\frac{\omega_h}{\omega_l} \right)^{\frac{(2i-1+r)}{2N}} \quad (3.4)$$

$$C = \omega_l^r \quad (3.5)$$

N yaklaşıklık sayısı, C geçici kazanç, ω_l ve ω_h sırasıyla düşük ve yüksek geçiş frekanslarıdır. $r < 0$ olması durumunda s^r , ters çevrilerek elde edilir (3.2). Bununla birlikte,

$|r| > 1$ olması durumunda, yukarıdaki yaklaşım tatmin edici sonuçlar vermez, bu da kesirli sıranın aşağıdaki şekilde bölünmesini gerektirir.

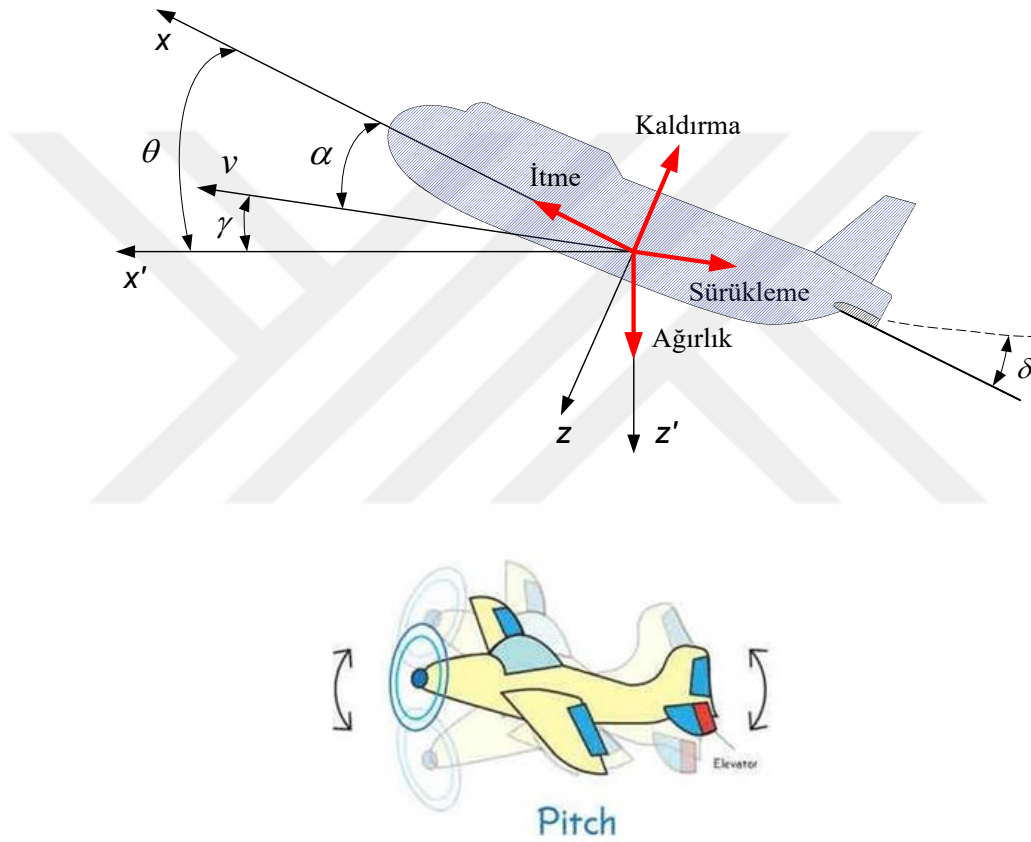
$$s^r = s^z s^x, \quad r = z + x, \quad z \in \mathbb{Z}, \quad x \in [0,1] \quad (3.6)$$

Bu nedenle, sadece (3.6)'deki s^x terimi yaklaştırılmalıdır.



4. PID KONTROLÖRLÜ AIRCRAFT PITCH SİSTEMİ

Bir uçağın hareketini düzenleyen denklemler, çok karmaşık yapıda altı adet doğrusal olmayan birleştirilmiş diferansiyel denklem setinden meydana gelmektedir. Bazı varsayımlar altında bu denklemler ayrıştırılabilir; uzunlamasına ve yanal denklemler olarak lineer hale de getirilebilir. Bir uçağın alçalma/yükselme açısı boyunca bir dinamiğe sahip olduğundan, söz konusu hareketler bu dinamikler ile kontrol edilmektedir. Bir uçağa etki eden temel kuvvetler ve koordinat eksenleri Şekil 4.1’de gösterilmiştir [58].



Şekil 4.1 Uçağa etkiyen temel kuvvetler

- i. **Kaldırma(Taşıma)**
- ii. **Ağırlık**
- iii. **İtme**
- iv. **Sürüklenme**

Sabit yükseklik ve hız ile seyir halinde olan bir uçağın itme, sürüklenme, ağırlık ve kaldırma kuvvetleri birbirlerini x ve y yönlerinde dengeler. Ayrıca, eğim açısındaki bir değişikliğin uçağın hızını değiştirmeyeceğini varsayarsak (gerçekte böyle bir durum söz konusu değildir ancak burada problemi biraz daha basitleştirmek adına böyle bir varsayımda bulunmaktadır), uçak için boyuna hareket denklemleri aşağıdaki gibi yazılabilir [58].

$$\frac{d\alpha}{dt} = \mu\Omega\sigma[-(C_L + C_D)\alpha + \frac{1}{\mu - C_L}q - (C_M \sin\gamma)\theta + C_L] \quad (4.1)$$

$$\frac{dq}{dt} = \frac{\mu\Omega}{2i_{yy}} [[C_m - \eta(C_L + C_D)]\alpha + [C_M + \sigma C_M(1 - \mu C_L)]q + (\eta C_W \sin\gamma)\delta] \quad (4.2)$$

$$\frac{d\theta}{dt} = \Omega q \quad (4.3)$$

Bu denklemlerde C_T , C_D , C_L , C_W ve C_M sırasıyla itme, sürüklenme, taşıma, ağırlık ve moment sabitleridir. μ havanın yoğunluğuna bağlı ve Ω ise uçuş denge hızına bağlı sabitlerdir [58]. Yukarıda bahsi geçen sistemde giriş, havalanma sapma açısı δ ve çıkış, uçağın eğim açısı θ olacaktır. Yukarıda verilen denklemleri basitleştirmek adına bazı sayısal değerler ekleyerek transfer fonksiyonunu bulabiliriz [58].

$$\frac{d\alpha}{dt} = -0.313\alpha + 56.7q + 0.232\delta \quad (4.4)$$

$$\frac{dq}{dt} = -0.0139\alpha - 0.426q + 0.0203\delta \quad (4.5)$$

$$\frac{d\theta}{dt} = 56.7q \quad (4.6)$$

Bu değerler Boeing'in ticari uçaklarından birinden elde edilen verilerden alınmıştır [59]. Yukarıdaki sistemin transfer fonksiyonunu bulmak için söz konusu denklemlerin Laplace

dönüşümü alınmalıdır. Yukarıdaki denklemlerin Laplace dönüşümü aşağıda gösterilmektedir.

$$sA(s) = -0.313A(s) + 56.7Q(s) + 0.232\Delta(s) \quad (4.7)$$

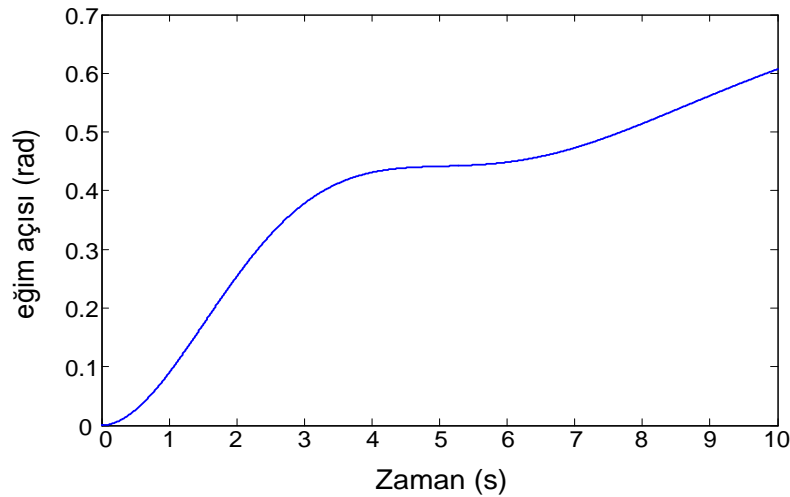
$$sQ(s) = -0.0139A(s) - 0.426Q(s) + 0.0203\Delta(s) \quad (4.8)$$

$$s\Theta(s) = 56.7Q(s) \quad (4.9)$$

Birkaç cebirsel işlemden sonra, aşağıdaki transfer fonksiyonu elde edilmektedir.

$$P(s) = \frac{\Theta(s)}{\Delta(s)} = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 0.9215s} \quad (4.10)$$

Kontrolsüz bir açık çevrim sisteminin nasıl performans gösterdiğini görmek için MATLAB yazılımından faydalanılmıştır (Şekil 4.2). Burada havalanma sapma açısı (δ) 0.2 radyan (yaklaşık 11.46 derece) olacak şekilde girişi temsilen alınmıştır. Açık çevrim cevabından anlaşılacağı üzere sistem kararsızdır ve sistemde etkili bir kontrolör kullanılmalıdır.



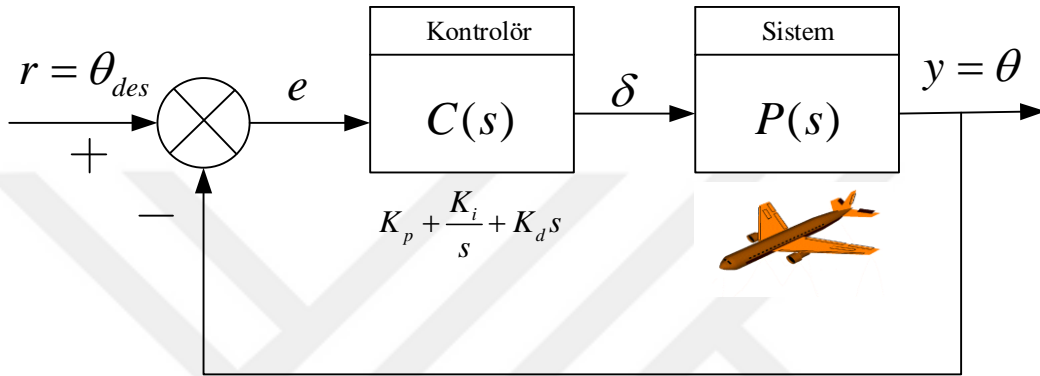
Şekil 4.2 Açık çevrim basamak cevabı

Uçağın dengede sağlıklı bir şekilde kalabilmesi için minimum istenilen gereksinimler aşağıda listelenmiştir.

Minimum Gereksinimler:

- Aşma(overshoot) %10 dan az olmalıdır.
- Yükselme zamanı(rise time) 2 saniyeden az olmalıdır.
- Oturma zamanı(settling time) 10 saniyeden az olmalıdır.
- Kararlı durum hatası(steady-state error) %2 den az olmalıdır.

Uçak alçalma/yükselme sistemini kararlı hale getirmek için geri-beslemeli bir PID kontrolör şekil 4.3'te gösterildiği gibi eklenebilir



Şekil 4.3 PID Kontrollü Bir Aircraft Pitch Sistemi

Şekil 4.3'te aircraft pitch sisteminin genel olarak kapalı çevrim kontrolör şematığı verilmiştir. Bu şematikteki P(s) denklem (4.10) ve C(s) değerleri yerine yazılarak aşağıdaki kapalı çevrim denklem elde edilir.

$$T(s) = \frac{C(s)P(s)}{1+C(s)P(s)} = \frac{(K_d s^2 + K_p s + K_i)(1.151s + 0.1774)}{s(s^3 + 0.739s^2 + 0.9215s) + (K_d s^2 + K_p s + K_i)(1.151s + 0.1774)} \quad (4.11)$$

PID kontrolörü ile aircraft pitch sisteminin kapalı çevrim transfer fonksiyonu $T(s)$ denklem (4.11) 'de verilmiştir.

4.1.Aircraft Pitch Sistem Modellemesi

Yukarıdaki modelleme denklemlerinin(4.10) halihazırda durum değişkeni formunda olduğu gerçeğini kabul ederek, bunları aşağıda gösterildiği gibi matris olarak yeniden yazabiliriz[58].

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0,313 & 56,7 & 0 \\ -0,039 & -0,46 & 0 \\ 0 & 56,7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0,232 \\ 0,0203 \\ 0 \end{bmatrix} \quad (4.12)$$

Çıktımız eğim açısı olduğundan, çıktı denklemi aşağıdaki gibidir.

$$y = [0 \quad 0 \quad 1] \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} \quad (4.13)$$

burada giriş asansör sapma açısı δ ve çıkış uçak eğim açısıdır θ . Yukarıdaki denklemler genel, doğrusal durum-uzay formuyla eşleşir.

$$\frac{dx}{dt} = Ax + Bu \quad (4.14)$$

$$y = Cx + Du \quad (4.15)$$

Bu sayfada bir durum-uzay denetleyicisi tasarım tekniği uygulayacağız. Özellikle, kontrolünü sistemin durumuna göre hesaplayan bir kontrolör tasarlayarak sistemin kapalı döngü kutuplarını yerleştirmeye çalışacağız.

Tasarım gereksinimleri

Bir sonraki adım bazı tasarım kriterleri seçmektir. Bu örnekte, bir eğim açısı adım komutuna yanıt olarak, gerçek eğim açısının% 10'dan daha az aştığı, 2 saniyeden daha az bir yükselme süresine, 10 saniyeden daha az bir çökme süresine sahip olacağı ve % 2'den az bir kararlı durum hatası. Örneğin, referans 0,2 radyan (11 derece) ise, adım açısı yaklaşık 0,22 radyonu geçmeyecek, 2 saniye içinde 0,02 rad'den 0,18 rad'e yükselecek, içindeki kararlı durum değerinin% 2'sine yerleşecektir. 10 saniye, ve kararlı durumda 0.196 ve 0.204 radyan arasında yerleşecektir [58].

Özet olarak, tasarım gereksinimleri aşağıdaki gibidir.

- Aşırı atış% 10'dan az
- Yükselme süresi 2 saniyeden az
- Ayar süresi 10 saniyeden az
- Kararlı durum hatası% 2'den az

4.2.Aircraft Pitch Sistem Analizi

Bu alanda uçak sisteminin kontrol edilebilirliği ve sistemin analizi yapılacaktır.

4.2.1. Kontrol edilebilirlik

Durum-uzay denetleyicisi tasarım tekniklerimizi uygulamak için önce önemli bir özelliği, kontrol edilebilirliği doğrulamamız gerekir. Kontrol edilebilirlik özelliği, sistemin durumunu istediğimiz yerde kullanma yetkisine sahip olduğumuzu garanti etmek için gereklidir. Bu, sistemin kapalı döngü kutuplarını karmaşık s düzleminde herhangi bir yere yerleştirme yeteneğine karşılık gelir.

Sistemin tamamen durum kontrol edilebilir olması için kontrol edilebilirlik matrisi

$$C = [B \ AB \ A^2 \ \dots \ A^{n-1}B] \quad (4.16)$$

rütbesi olmalı n . bir matrisin sırası, doğrusal olarak bağımsız satırların (veya sütunların) sayısıdır. Sayı n , sistemin durum değişkenlerinin sayısına karşılık gelir. Daha yüksek matris gücüne sahip kontrol edilebilirlik matrisine ilave terimler eklemek kontrol edilebilirlik matrisinin A sırasını arttırmayacaktır, çünkü bu ek terimler sadece önceki terimlerin doğrusal kombinasyonları olacaktır.

Kontrol edilebilirlik matrisimiz 3×3 olduğundan, matrisin sırası 3 olmalıdır. MATLAB komut sırası size bu matrisin sırasını verebilir. Yeni bir m dosyası oluşturun ve aşağıdaki komutları girin. Bu m dosyasını MATLAB komut penceresinde çalıştırmak aşağıdaki çıktıyı üretecektir [58].

```
A = [-0.313 56,7 0; -0.0139 -0.426 °; 0 56,7 0];
```

```
B = [0.232; 0.0203; 0];
```

```
C = [0 0 1];
```

```
D = [0];
```

```
co = ctrb (A, B);
```

```
Kontrol edilebilirlik = sıralama (ko)
```

```
Kontrol edilebilirlik = 3
```

Bu nedenle, kontrol edilebilirlik matrisi 3. sırada olduğundan sistemimiz tamamen devlet tarafından kontrol edilebilir.

4.2.2. Ayrık durum uzayı

Bir dijital kontrol sisteminin tasarımındaki ilk adım, tesisin örneklenmiş bir veri modelini oluşturmaktır. MATLAB, c2d komutu kullanılarak bu modeli sürekli zaman modelinden oluşturmak için kullanılabilir. C2D bir sistem modeli, örnekleme zaman (T_s) ve tutma devresi türü. Bu örnekte sıfır sıralı tutma (zoh) devresi olduğunu varsayacağız. Daha fazla bilgi için Giriş: Dijital Denetleyici Tasarımı sayfasına bakın.

Bir örnekleme süresi seçerken, sistemin örneklenmiş çıktısının sistemin tam davranışını yakalaması için örnekleme frekansının, sistemin dinamikleriyle karşılaştırıldığında hızlı olması arzu edilir; kaçırılmamalıdır. Bir sistemin "hızının" bir ölçüsü, kapalı döngü bant genişliğidir. Önemli bir kural, örnekleme frekansının, kapalı döngü Bode grafiğinden belirlenebilen kapalı döngü bant genişliği frekansından en az 30 kat daha büyük olmasıdır.

Kapalı döngü Bode grafiğinden bant genişliği frekansı yaklaşık 2 rad / sn ($0,32 \text{ Hz}$) olarak belirlenebilir. Bunu kendiniz doğrulayabilirsiniz. Bu nedenle, yeterince küçük bir örnekleme zamanımız olduğundan emin olmak için $1/100 \text{ sn}$ / örnek bir örnekleme zamanı kullanacağız. Şimdi c2d fonksiyonunu kullanmaya hazırız. Aşağıdaki komutlar bir m dosyasına girildi. Bu m dosyasını MATLAB komut penceresinde çalıştırıldı, örneklenmiş veri durum alanı modelini temsil eden aşağıdaki dört matrisi sonucu verildi [58].

```
A = [-0.313 56,7 0;
      -0.0139 -0.426 °;
      0 56,7 0];
B = [0.232;
      0.0203;
      0];
C = [0 0 1];
D = [0];

sys_ss = ss (A, B, C, D);

Ts = 1/100;
sys_d = c2d (sys_ss, Ts, ' zoh ' )
```

```
sys_d =
      A =
          x1 x2 x3
```

x1 0,9968 0,5649 0
 x2 -0.0001385 0.9957 0
 x3 -3.931e-05 0.5658 1

B =

u1
 x1 0.002374
 x2 0.0002024
 x3 5.744e-05

C =

x1 x2 x3
 y1 0 0 1

D =

u1
 y1 0

Örnekleme zamanı: 0.01 saniye

Ayrık zamanlı durum-uzay modeli.

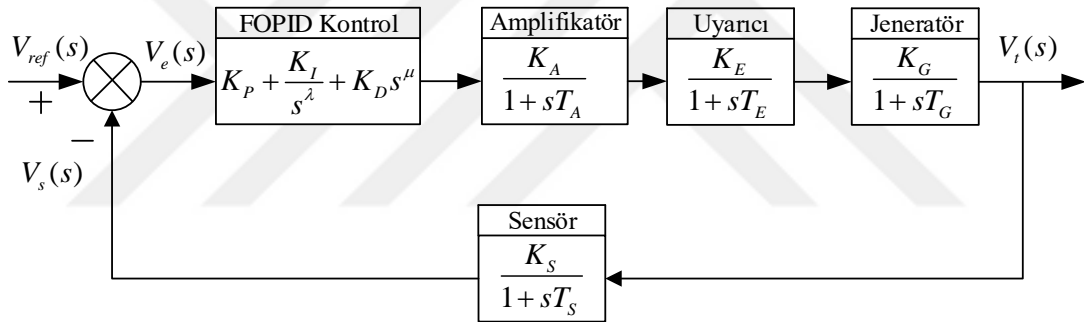
Şimdi aşağıda gösterilen ayrık zamanlı durum-uzay modelini elde edildi [18].

$$\begin{bmatrix} \alpha(k+1) \\ q(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} 0,9969 & 0,05649 & 0 \\ -0,0001 & 0,9957 & 0 \\ 0 & 0,5658 & 1 \end{bmatrix} \begin{bmatrix} \alpha(k) \\ q(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} 0,0024 \\ 0,0002 \\ 0,0001 \end{bmatrix}$$

$$y(k) = [0 \quad 0 \quad 1] \begin{bmatrix} \alpha(k) \\ q(k) \\ \theta(k) \end{bmatrix} + [0] [\delta(k)]$$

5. FOPID KONTROLÖRLÜ AVR SİSTEMİ

AVR sistemi, jeneratör terminal gerilimindeki dalgalanmalardan dolayı oluşabilecek sorunları azaltmak için bir elektrik güç şebekesinde etkili bir çözümdür [11]. Bu nedenle, güç ağını düzgün şekilde yönetmek için bir jeneratörün terminal geriliminin büyüklüğünü belirli bir seviyede tutmak önemlidir. Kararlı bir tepkiye sahip olmasına rağmen, AVR sisteminin bir denetleyicisiz adım tepkisi, uygun şekilde ayarlanmış bir denetleyici gerektiren zayıf kararlı durum ve geçici durum özellikleri ile oldukça salınımlıdır [60] [61]. Bu nedenle, bu çalışmada kontrol edilebilirlik açısından artırılmış esneklik ve serbestlik derecesi sağlamak için bir FOPID kontrolörü kullanılmıştır [13] [14]. Şekil 5.1'de V_{ref} , V_s , V_e ve V_t 'nin sırasıyla referans giriş gerilimi, sensor çıkış gerilimi, gerilim hatası ve jeneratör terminal gerilimi olduğu bir AVR sisteminin doğrusallaştırılmış matematiksel modelinin blok diyagramını göstermektedir.



Şekil 5.1 FOPID denetleyicili AVR sistemi

Şekil 5.1'deki FOPID denetleyici, amplifikatör, uyarıcı, jeneratör ve sensorun transfer fonksiyonları ile birlikte AVR'nin kapalı döngü transfer fonksiyonu (5.1), (5.2) ve (5.3) 'te verilmiştir, burada K_p , K_I , K_D , λ ve μ sırasıyla integral ve türev terimlerinin kesirli mertebeli orantılı, integral ve türev kazançlarıdır. K_x ve T_x ($x \in A, E, G, S$), AVR'nin ilgili elemanının kazanç ve zaman sabitleridir. Yukarıda belirtilen parametrelerin aralığı şu şekilde verilmiştir; FOPID denetleyicisi için $0.1 \leq K_p, K_I, K_D \leq 3.0$, $0.001 \leq \lambda, \mu \leq 2.0$, Amplifikatör için $10 \leq K_A \leq 40$; $0.02 \leq T_A \leq 0.1$, Uyarıcı için $1.0 \leq K_E \leq 10$; $0.4 \leq T_E \leq 1.0$; jeneratör için $0.7 \leq K_G \leq 1.0$; $1.0 \leq T_G \leq 2.0$ ve sensör için $1.0 \leq K_S \leq 2.0$; $0.001 \leq T_S \leq 0.06$. Seçilen parametre değerleri; Amplifikatör için $K_A = 10$, $T_A = 0.1$ s, uyarıcı için $K_E = 1.0$, $T_E = 0.4$ s, jeneratör için $K_G = 1.0$, $T_G = 1.0$ s ve sensör için $K_S = 1.0$, $T_S = 0.01$ s

$$G_{FOPID}(s) = K_P + K_I s^{-\lambda} + K_D s^\mu \quad (5.1)$$

$$G_{Amp}(s) = \frac{K_A}{1 + sT_A}$$

$$G_{Exc}(s) = \frac{K_E}{1 + sT_E}$$

$$G_{Gen}(s) = \frac{K_G}{1 + sT_G} \quad (5.2)$$

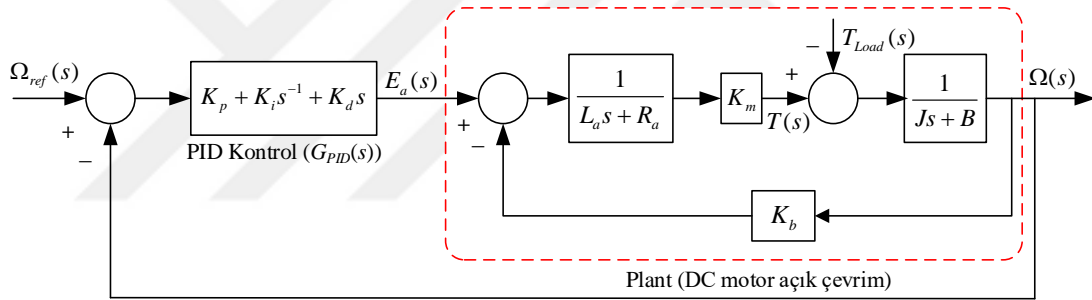
$$G_{Sen}(s) = \frac{K_S}{1 + sT_S}$$

$$T_{FOPID}(s) = \frac{V_t(s)}{V_{ref}(s)} = \frac{K_A K_E K_G (1+sT_S)(K_P + K_I s^{-\lambda} + K_D s^\mu)}{(1+sT_A)(1+sT_E)(1+sT_G)(1+sT_S) + K_A K_E K_G K_S (K_P + K_I s^{-\lambda} + K_D s^\mu)} \quad (5.3)$$

6. PID KONTROLÖRLÜ DC MOTOR HIZ KONTROL SİSTEMİ

Hız ayarı, armatür gerilimi ile kontrol edilir. Sistemin PID denetleyicisini kullanan kapalı döngüsü Şekil 6.1'de gösterilmektedir. PID transfer fonksiyonu denklemi (6.1)'de verilmiştir. Burada K_p , K_i , ve K_d PID'nin kazançlarıdır (sırasıyla orantılı, integral ve türev). ω ve ω_{ref} sırasıyla gerçek motor hızı ve referans hızıdır. R_a , 0.4Ω değerine eşit armatür direncidir, L_a , 2.7 H'ye eşit armatür endüktansdır, J , $0.0004 \text{ kg} \cdot \text{m}^2$ 'ye eşit motorun atalet torkudur, B , $0.0022 \text{ N} \cdot \text{m} \cdot \text{s/rad}$ 'e eşit motor sürtünme sabitidir, K_m $0,015 \text{ N m/A}$ 'ya eşit motor torku sabitidir ve K_b , $0,05 \text{ V}$ 'a eşit elektromotor kuvvet sabitidir [62].

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \quad (6.1)$$



Şekil 6.1 PID Kontrollü Bir DC Motor Hız Sistemi

DC motorun açık çevrim transfer fonksiyonu aşağıdaki şekilde elde edilir. Sabit bir akı için indüklenen gerilim $e_b(t)$, açısal hız $\omega(t) = d\theta/dt$ ile orantılıdır ve şu şekilde verilir:

$$e_b(t) = K_b \frac{d\theta(t)}{dt} = K_b \omega(t) \quad (6.2)$$

Armatür gerilimi $e_a(t)$, armatür kontrollü bir DC servo motorun hızını yönetmek için kullanılır. Armatür devresi diferansiyel denklemi aşağıdaki denklemde verilmiştir.

$$e_a(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + e_b(t) \quad (6.3)$$

Yük torkunun sıfır olduğunu varsayarsak, karşılık gelen bir tork (atalet ve sürtünme torklarının toplamı), denklem (6.4)'te sağlanan armatür akımı tarafından üretilir.

$$T(t) = J \frac{d\omega(t)}{dt} + B\omega(t) = K_m i_a(t) \quad (6.4)$$

Denklemlerin Laplace dönüşümleri. (6.2) - (6.4) sıfır başlangıç koşulları ile aşağıdaki şekilde verilmiştir:

$$E_b(s) = K_b \Omega(s) \quad (6.5)$$

$$E_a(s) = (L_a s + R_a) I_a(s) + E_b(s) \quad (6.6)$$

$$T(s) = (J s + B) \Omega(s) = K_m I_a(s) \quad (6.7)$$

Dolayısıyla, DC motorun giriş gerilimi ile çıkış hızı arasındaki ilişkiyi açıklayan açık döngü transfer fonksiyonu Denklem (6.8) 'deki gibi yazılabilir.

$$\frac{\Omega(s)}{E_a(s)} = \frac{K_m}{(L_a s + R_a)(J s + B) + K_b K_m} \quad (T_{Load} = 0 \text{ için}) \quad (6.8)$$

Ayrıca, giriş gerilimi (E_a) sıfır olduğunda motor hızı (ω) ve yük torku (T_{Load}) arasındaki ilişki aşağıdaki transfer fonksiyonu ile de verilebilir.

$$\frac{\Omega(s)}{T_{Load}(s)} = -\frac{(L_a s + R_a)}{(L_a s + R_a)(J s + B) + K_b K_m} \quad (E_a = 0 \text{ için}) \quad (6.9)$$

Denklem (6.8) ve (6.9) 'daki DC motor parametre değerlerini değiştirerek, aşağıdaki açık çevrim transfer fonksiyonu elde edilir:

$$G_{open-loop}(s) = \begin{cases} \frac{\Omega(s)}{E_a(s)} = \frac{15}{1.08s^2 + 6.1s + 1.63} , & (T_{Load} = 0 \text{ için}) \\ \frac{\Omega(s)}{T_{Load}(s)} = -\frac{2700s + 400}{1.08s^2 + 6.1s + 1.63} , & (E_a = 0 \text{ için}) \end{cases} \quad (6.10)$$

Son olarak, bir PID hız kontrolörü ile DC motor kapalı çevrim transfer fonksiyonu ($T_{closed-loop}(s)$) denklem (6.11) 'te verilmiştir.

$$= \begin{cases} \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{15(K_d s^2 + K_p s + K_i)}{1.08s^3 + 6.1s^2 + 1.63s + 15(K_d s^2 + K_p s + K_i)} , & (T_{Load} = 0 \text{ için}) \\ \frac{\Omega(s)}{T_{Load}(s)} = -\frac{(2700s + 400)s}{1.08s^3 + 6.1s^2 + 1.63s + 15(K_d s^2 + K_p s + K_i)} , & (\Omega_{ref} = 0 \text{ için}) \end{cases} \quad (6.11)$$

7. SEZGİSEL-ÜSTÜ ALGORİTMALAR

7.1. Henry Gaz Çözünürlük Optimizasyonu (HGSO) Algoritması

HGSO algoritması, 1800'lerde formüle edilen Henry kanununa dayanmaktadır. Söz konusu kanun ile gazların sıvı ortamdaki çözünürlüğü ifade edilmektedir. HGSO algoritması söz konusu bu fiziksel olayın matematiksel olarak taklit edilmesiyle oluşturulmuştur. Söz konusu matematiksel model 8 adımda ifade edilebilir [1].

İlk adımda, eşitlik (7.1) kullanılarak N adet gaz parçacığı üzerine rastgele bir arama başlatılır. Bu gaz parçacıkları aynı zamanda ilk popülasyonu da oluşturmaktadırlar [1].

$$X_i(t + 1) = X_{min} + r \times (X_{max} - X_{min}) \quad (7.1)$$

t ve r terimleri, sırasıyla yineleme süresini ve rastgele bir sayıyı (0 ile 1 arasında) temsil eder. Arama alanının alt ve üst sınırları X_{min} ve X_{max} ile gösterilir. X_i , popülasyondaki i . gaz parçacığının konumunu temsil etmektedir. Henry gaz sabiti $j(H_j(t))$, gaz parçacıklarının sayısı i , gaz parçacıklarının kısmi basıncı $P_{i,j}$ (j kümesindeki i parçacığı) ile $j(C_i)$ sabit değeri eşitlik (7.2) kullanılarak başlatılmaktadır. İlgili denklemde l_1, l_2, l_3 sırası ile 5×10^{-2} , 100 ve 10^{-2} değerlerindedir [11].

$$\begin{aligned} H_j(t) &= l_1 \times rand(0,1) \\ P_{i,j} &= l_2 \times rand(0,1) \\ C_j &= l_3 \times rand(0,1) \end{aligned} \quad (7.2)$$

HGSO algoritması, farklı tiplerde gazları barındırdığından, ikinci aşaması kümeleme ile ilgilidir. Bu sebeple her popülasyon farklı gaz sabitine (H_j) sahip kümelere ayrılmaktadır.

Üçüncü adım, j kümesindeki i gaz partikülünü değerlendirmek için bir amaç fonksiyonun kullanıldığı değerlendirme sürecinden oluşur. Bu aşamada, tüm popülasyondaki (X_{best}) ve her kümedeki ($X_{i,j}$) en iyi partikülü tanımlamak için bir sıralama aşaması da gerçekleştirilir.

Dördüncü adım, Henry katsayısının eşitlik (7.3) kullanılarak güncellendiği aşamadır. Burada T sıcaklığı, H_j Henry katsayısını, $iter$ ise j . kümedeki toplam yineleme sayısını ifade etmektedir. T^θ ise 298.15'e eşit olan bir sabittir [1].

$$\begin{aligned} H_j(t+1) &= H_j(t) \\ &\times e^{(-C_j \times (1/T(t) - 1/T^\theta))}, \end{aligned} \quad (7.3)$$

$$T(t) = e^{(-t/iter)}$$

Çözünürlüğü güncelleme eşitlik (7.4) aracılığıyla beşinci adımda gerçekleştirilir. Burada $P_{i,j}$ kısmi basıncı, $S_{i,j}$ kümedeki parçacığın (j kümesindeki i parçacığının) çözünürlüğünü ve K ise bir sabiti ifade etmektedir [1].

$$S_{i,j}(t) = K \times H_j(t+1) \times P_{i,j}(t) \quad (7.4)$$

Altıncı aşamada, eşitlik (7.5) kullanılarak pozisyon güncellemesi yapılmaktadır. Burada, her bir j kümesindeki i parçacığının $X_{i,j}$ pozisyonu $t+1$ iterasyonu için güncellenmektedir. Söz konusu denklemde $\varepsilon = 0.05$ değerine eşittir [1].

$$\begin{aligned} X_{i,j}(t+1) &= X_{i,j}(t) + F \times r_1 \times \gamma \times (X_{i,best}(t) - X_{i,j}(t)) \\ &+ F \times r_2 \times \alpha \times (S_{i,j}(t) \times X_{best}(t) - X_{i,j}(t)), \end{aligned} \quad (7.5)$$

$$\gamma = \beta \times e^{\left(\frac{F_{best}(t) + \varepsilon}{F_{i,j}(t) + \varepsilon} \right)}$$

Yukarıdaki denklemde, $X_{i,best}$ kümedeki en iyi gazı, r_1 ve r_2 iki farklı rastgele sayıyı, γ gaz partikülünün (j kümesindeki i gaz partikülü) kabiliyetini, α diğer gaz partiküllerinin etkisini (j kümesindeki i gaz partikülü) ve β ise kullanıcı tarafından tanımlanan bir sabiti ifade etmektedir. j kümesindeki i gaz parçacığının uygunluğunu $F_{i,j}$, tüm popülasyonlardaki en iyi gaz partikülünü ise F_{best} ifade etmektedir. Aramanın yönü F işaretçisi ile kontrol edilmektedir. Tüm popülasyondaki en iyi gaz X_{best} ile ifade

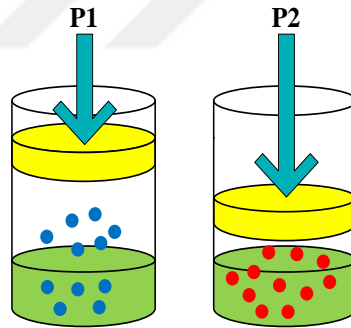
edilmektedir. Keşif ve sömürü kabiliyetleri $X_{i,best}$ ve X_{best} parametreleri ile sağlanmaktadır [1].

Yedinci basamak lokal minimumda takılı kalmamak için kullanılmaktadır. En kötü ajan sayısı N_w eşitlik (7.6)'e göre sıralanıp ve seçilir. Burada $c_1 = 0.1$ ve $c_2 = 0.2$ 'dir [1].

$$N_w = N \times (rand(c_2 - c_1) + c_1) \quad (7.6)$$

En son aşamada ise en kötü ajanların pozisyonu eşitlik (7.7) kullanılarak güncellenir. Burada, $G_{i,j}$ gaz parçacığının pozisyonunu, r rastgele bir sayıyı, G_{min} ve G_{max} ise problemin sınırlarını belirtmektedirler [1].

$$G_{i,j} = G_{\min(i,j)} + r \times (G_{\max(i,j)} - G_{\min(i,j)}) \quad (7.7)$$



Şekil 7.1 Henry gaz çözünürlüğü prensibi [1]

Şekil 7.1 Henry gaz çözünürlüğü prensibini canlandırmaktadır. Şekilden de görüleceği üzere, dengedeki gazın hacmi, basınç arttırıldıkça azalmaktadır.

7.1.1 Önerilen OBL / HGSO algoritması

Temel HGSO'nun yavaş yakınsama ve yerel bir en iyi çözüme takılıp kalma gibi birkaç dezavantajı vardır. Bu dezavantajlar, bazı çözümlerin yerel en iyi çözüme doğru güncellenmesinden kaynaklanmaktadır, ancak HGSO'nun bulamayacağı daha iyi ve

kullanılabilir ancak çok uzak çözümler vardır. Bu nedenle, bu dezavantajlardan kaçınmak için zıt yönlerdeki çözümlerin dikkate alınması gerekmektedir. İkinci gereksinim, HGSO'nun yanında OBL kullanan bu makalede önerilen algoritma ile elde edilir. Önerilen OBL / HGSO, orijinal HGSO'nun arama yeteneklerini OBL ile birleştirerek arama alanının keşif verimliliğini artırır. Benzer algoritmalarla karşılaştırıldığında, önerilen yaklaşım, ayarlamak için daha az parametre gerektirir. Ayrıca HGSO algoritmasının konfigürasyonu OBL'nin dahil edilmesinden etkilenmez, aksine en iyi çözümün doğruluğunun artmasına yardımcı olur. Böylece, optimum çözüme yakınsama açısından iyileştirme için gerekli olan başlangıç popülasyonunun boyutunu azaltmak mümkündür, çünkü OBL / HGSO tarafından daha geniş bir arama alanı keşfedilebilir.

Önerilen algoritma, HGSO'nun iki adımda geliştirilmesine yardımcı olur. İlk olarak, OBL, yerel bir en iyi çözümde durgunluktan kaçınmak için yakınsamayı hızlandırarak ve tüm çözümler için tüm arama alanını keşfederek popülasyonu başlatmak için kullanılır. Ek olarak, güncellenen çözümleri ters yönde kontrol etmek ve bu çözümleri mevcut olanla karşılaştırmak için başka bir en iyi çözüm olup olmadığını kontrol etmek için de kullanılır. Bunu yaparak, yerel en iyi minimumdan kaçınır. Aşağıdaki alt bölümler, önerilen algoritma için kullanılan aşamaları açıklamaktadır[63].

A. Başlatma aşaması

Önerilen yöntem, başlangıç çözümü için konum vektörünün $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ olarak tanımlanacağı şekilde N boyutuna sahip rastgele bir X popülasyonunu başlatmakla başlar; burada $i = 1, 2, \dots, N$. OBL daha sonra çözümleri her bir çözümün zıt yönlerinde hesaplamak için kullanılır ve karşıt bir \bar{X} popülasyonu oluşturur. Hem X hem de \bar{X} popülasyonlarını kullanarak, en iyi N çözüm sayısı seçilir. Bu aşamadaki adımlar aşağıdaki gibidir:

X popülasyonu için çözümlere rastgele başlayın.

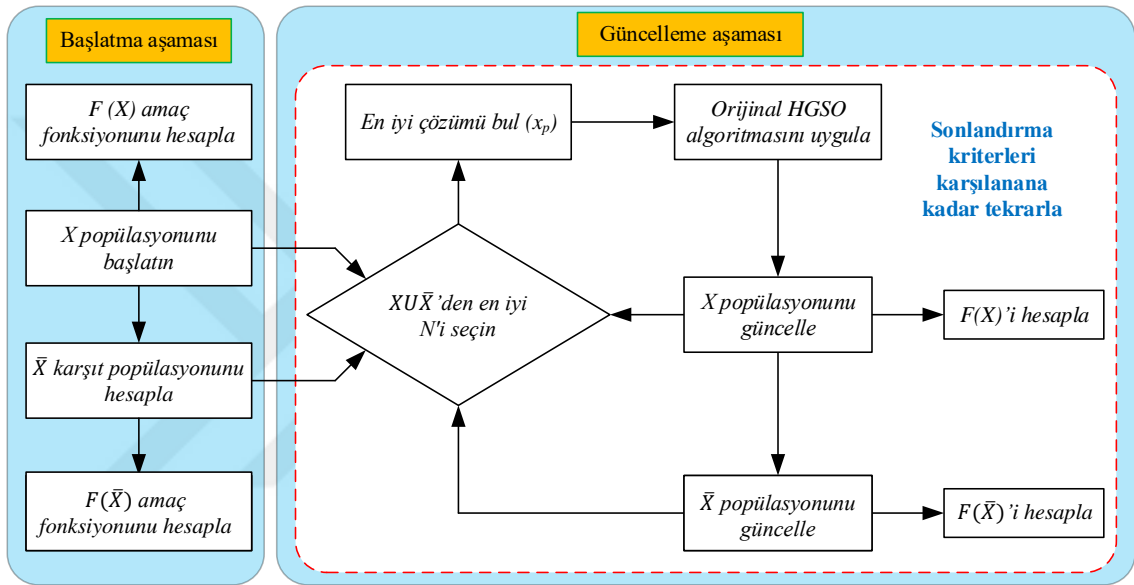
\bar{X} 'in zıt popülasyonunu şu şekilde hesaplayın:

$\bar{x}_{ij} = u_i + l_i - x_{ij}$ burada $i = 1, 2, \dots, N$ ve $j = 1, 2, \dots, n$. Burada, l ve u, sırasıyla arama alanı için alt ve üst sınırlardır. x_{ij} ve \bar{x}_{ij} , sırasıyla X popülasyonu için i pozisyonunun j'inci çözümünü ve \bar{X} 'e karşılık gelen karşıt popülasyonunu temsil eder.

Yeni bir popülasyon oluşturmak için $X \cup \bar{X}$ birliğinden N sayıda en iyi çözümü seçilir.

B. Aşama güncellemesi

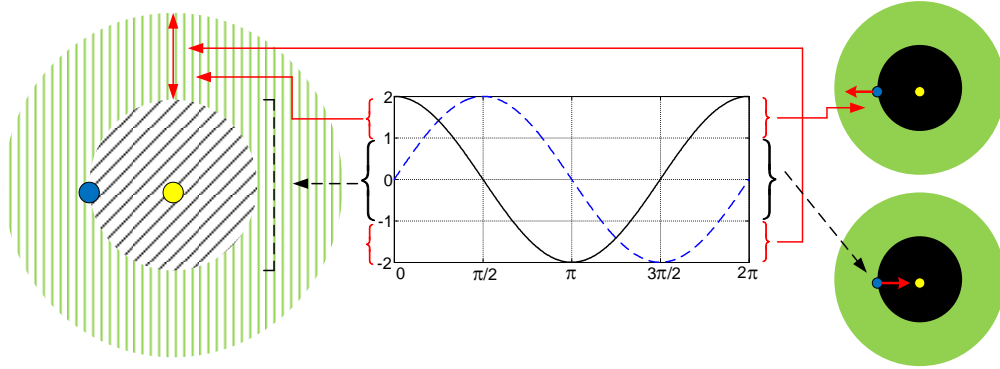
En iyi çözüm (x_p) en iyi N çözümü seçildikten sonra (ilk aşamadan itibaren) belirlenir. X popülasyonundaki ajanlar HGSO algoritması kullanılarak güncellenir ve uygunluk fonksiyonları hesaplanır. Ek olarak, \bar{X} 'in zıt popülasyonu OBL'ye göre hesaplanır ve her \bar{x} için uygunluk fonksiyonları belirlenir. OBL / HGSO'daki bir sonraki adım, her iki popülasyonun birleşiminden N sayıda en iyi çözümü seçmektir ($X \cup \bar{X}$). Durdurma durumuna ulaşılan kadar tüm adımlar tekrarlanır. Önerilen OBL / HGSO algoritmasının akış şeması Şekil 7.1.1.1'de gösterilmektedir[63].



Şekil 7.1.1.1. Önerilen OBL / HGSO algoritması

7.2. Sinüs Kosinüs Algoritması (SCA)

Sinüs kosinüs algoritması (SCA), yakın geçmişte önerilen popülasyon temelli sezgisel-üstü bir algoritmadır [64]. Bu algoritma, başlarken çoklu rastgele çözümler oluşturur. Oluşturulan bu çözümlerin daha sonra en iyi çözüme veya dışına doğru salınımlarını sağlar. Buna ek olarak, arama alanının keşfini ve sömürülmesini vurgulamak için algoritmaya rastgele ve uyarlanabilir çeşitli değişkenler entegre edilmiştir. Keşif ve sömürü, popülasyon temelli optimizasyon sürecinin ve SCA'nın yaygın olarak kullanılan iki aşamasıdır. Her iki faz da aşağıdaki pozisyon güncelleme denklemleri ile ifade edilmektedir [64] [65].



Şekil 7.3 Sinüs-kosinüs fonksiyonları $[-2, 2]$ aralığında bir çözümün hedefin ötesine (aralarındaki boşluğun dışına) veya çevresine (aralarındaki boşluğun içine) gitmesine izin verir [64], [65]

Algoritmada dengeli keşif ve sömürü aşamalarına ulaşmak için, eşitlik (7.8)'te sinüs-kosinüs fonksiyonlarının genliği aşağıdaki denklemle uyarlanabilir şekilde değiştirilir [64].

$$r_1 = a - t \frac{a}{T} \quad (7.9)$$

Burada t , mevcut yinelemedir, T ise toplam yineleme sayısıdır ve a , 2'ye eşit olan bir sabittir. Eşitlik (7.9)'te r_1 lineer bir biçimde a değerinden 0'a azalmaktadır.

7.3. Çekirge Optimizasyon Algoritması (GOA)

Çekirge optimizasyon algoritması (GOA), yakın zamanda önerilmiş olan tek hedefli, popülasyon tabanlı sezgisel-üstü bir algoritma olup doğada bulunan çekirge sürülerinin davranışını taklit ederek optimizasyon problemlerinin çözülmesi için oluşturulmuş matematiksel modeldir [66]. Algoritma çekirgeler arasındaki itme ve çekme kuvvetlerini simüle eder. İtme kuvvetleri çekirgelerin arama alanını keşfetmesine izin verirken, çekim kuvvetleri gelecek vaat eden bölgeleri sömürmeleri noktasında teşvikte bulunur. GOA, optimizasyon süresince keşif ve sömürü aşamalarını dengelemek için bir katsayı ile donatılmıştır. Bu sayede, yerel optimumda sıkışmanın önüne geçilip küresel optimumun kesin bir tahmininin bulunması kolaylaştırılmış olur. Mevcut duruma kadar elde edilmiş olan en iyi çözüm, sürü tarafından takip edilmesi gereken hedef olarak değerlendirildiğinden, çekirgelerin yinelemeler (iterasyonlar) boyunca hedefi

iyileştirerek küresel optimumu bulma şansı yükselmektedir. GOA'nın konum güncelleme denklemi eşitlik (7.10)'te olduğu gibi verilir [66] [67].

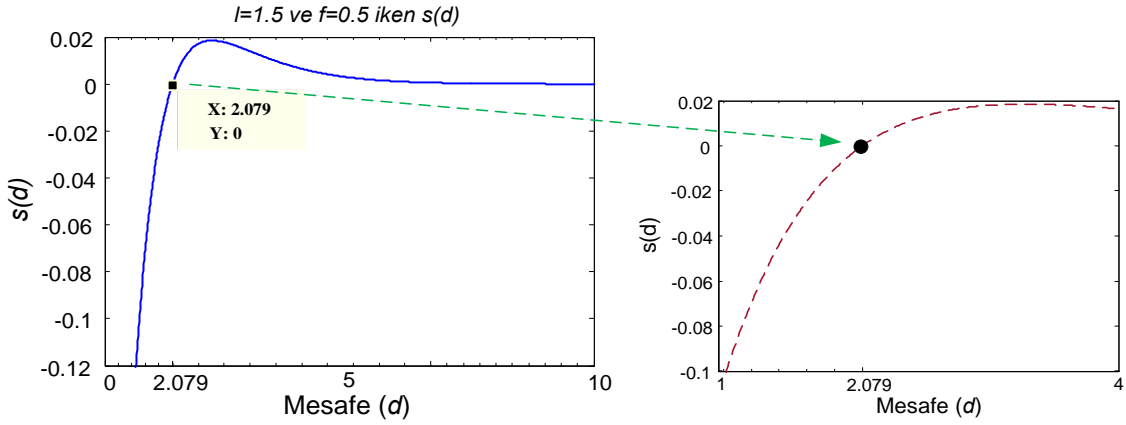
$$X_i^d = r \left(\sum_{\substack{j=1 \\ j \neq i}}^N r \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right) + T_d \quad (7.10)$$

Burada, X_i^d mevcut çözümün d boyutundaki pozisyonunu, r azalma katsayısını, ub_d ve lb_d ise d boyutunun üst ve alt sınırlarını, s çekirgeler arası sosyal kuvvetin fonksiyonunu, d_{ij} iki çekirge arası mesafenin mutlak değerini (j . (x_j) ve i . (x_i) çekirgeler arası mesafe) ve son olarak T_d ise d boyutunda şimdiye kadar ulaşılan en iyi çözüm olan hedef değeri ifade etmektedir. Eşitlik (15), bir çekirgenin bir sonraki konumunun mevcut konumuna, diğer tüm çekirgelerin konumuna ve hedefin konumuna bağlı olduğunu gösterir. Eşitlik (7.10)'daki sosyal kuvvet fonksiyonu aşağıdaki gibi verilmektedir [66] [67].

$$s = f e^{\frac{-d}{l}} - e^{-d} \quad (7.11)$$

Burada, f çekim gücünü ve l ilgili çekime ait uzunluk ölçeğinin ifadesidir. Çekirgeler arasındaki sosyal etkileşimi (çekim ve itme) göstermek için s fonksiyonu Şekil 7.4'te verilmiştir. Söz konusu şekilde 0 ile 10 arasındaki değerler dikkate alınmaktadır. İtme [0, 2.079] aralığında gerçekleşmektedir. Bir çekirge başka bir çekirgeden 2.079 birim uzakta olduğunda (konfor bölgesi veya rahat mesafe olarak adlandırılan mesafe), ne çekme ne de itme meydana gelmemektedir. Şekil 7.4 ayrıca çekim değerinin 2.079'dan yaklaşık 4'e kadar yükseldiğini ve sonra kademeli olarak azaldığını da göstermektedir. Eşitlik (7.11)'deki f ve l parametrelerini değiştirmenin yapay çekirgelerde farklı sosyal davranışlara neden olacağı açıktır ancak bu çalışmada söz konusu bu değerler sırası ile 0.5 ve 1.5 olarak alınmıştır [66] [67].

Bu fonksiyon 4'ten büyük mesafeler için 0'a yakın değerler verdiği için (Şekil 7'den de görüleceği üzere), çekirgeler arası mesafe [1, 4] aralığında belirtilmiştir. Söz konusu s fonksiyonu Şekil 7.4'te (sağ taraf) gösterilmiştir. Çekirgeler ile konfor bölgesi arasındaki sosyal etkileşimlerin s fonksiyonu kullanılarak elde edilmiş kavramsal modeli Şekil 7.5'te gösterilmektedir.

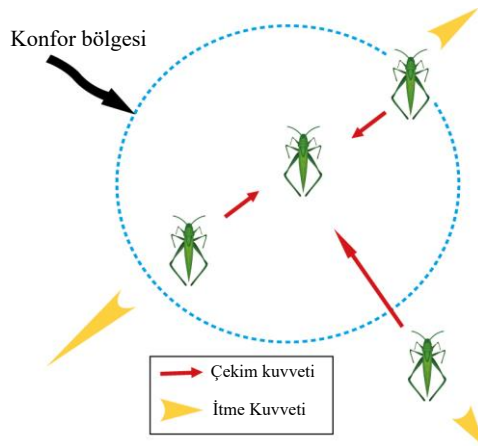


Şekil 7.4 Sosyal kuvvetler fonksiyonu $l = 1.5$ ve $f = 0.5$ değerlerinde olduğunda (solda) ve $d \in [1, 4]$ aralığında iken bu fonksiyonun sahip olduğu aralık (sağda) [66], [67]

Eşitlik (7.10)'teki r parametresinin, keşif ve sömürüyü dengelemek için yineleme sayısı ile orantılı olarak azaltılması gerekmektedir. Bu durum, yineleme sayısı arttıkça sömürüyü teşvik eder. Bu parametre ayrıca konfor bölgesini yineleme sayısı ile orantılı olarak azaltır ve aşağıdaki gibi hesaplanır [66] [67].

$$r = r_{max} - t \frac{r_{max} - r_{min}}{T} \quad (7.12)$$

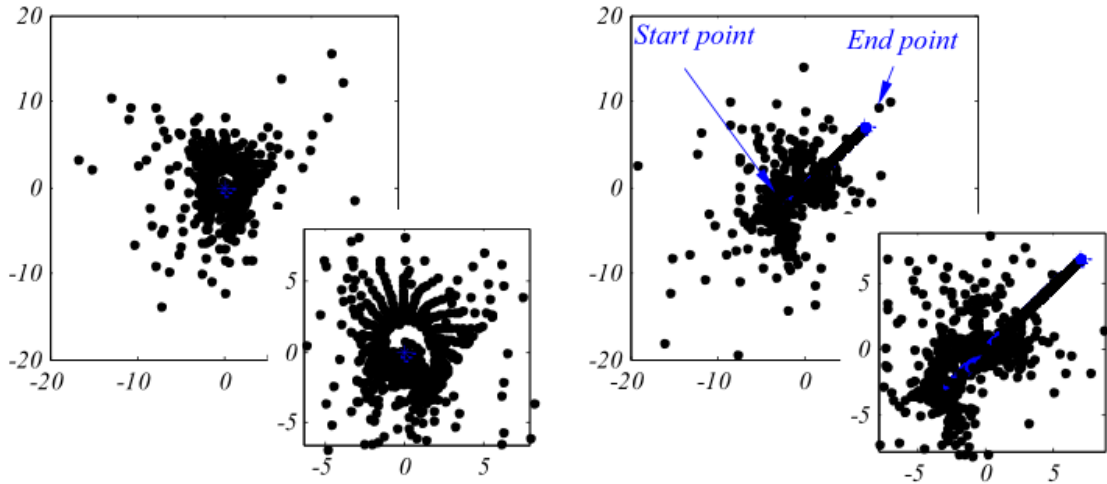
Yukarıdaki denklemde, r_{max} maximum değeri, r_{min} minimum değeri, t şu anki iterasyonu ve T ise maximum iterasyon sayısını ifade etmektedir. Bu çalışmada r_{max} ve r_{min} için sırası ile 1 ve 0.00001 kullanılmıştır.



Şekil 7.5 Çekirge ve konfor bölgesi arasındaki sosyal etkileşimler [66], [67]

7.4. Parçacık Sürü Optimizasyon Algoritması (SSA)

Bilindiği üzere salp zincirlerini simüle etmede matematiksel model, optimizasyon problemlerini çözmek için doğrudan kullanılamaz. Başka bir deyişle, optimizasyon problemlerine uygulanabilir hale getirmek için modeli biraz ince ayarlamaya ihtiyaç vardır. Tek bir hedef optimize edicinin nihai hedefi, küresel optimumun belirlenmesidir. SSA sürü modelinde, takipçi salpları önde gelen salp'ı takip eder. Önde gelen salp ayrıca besin kaynağına doğru hareket eder. Gıda kaynağı küresel optimumla değiştirilirse, bu nedenle salp zinciri otomatik olarak ona doğru hareket eder. Ancak sorun, optimizasyon problemlerinin genel optimumunun bilinmemesidir. Bu durumda, şimdiye kadar elde edilen en iyi çözümün küresel optimum olduğu ve salp zinciri tarafından takip edilecek besin kaynağı olduğu varsayılmaktadır [68].



Şekil 7.6 100 yinelemeden sonra 2B alanda sabit ve mobil gıda kaynakları etrafında arama geçmişi [68]

- SSA algoritması şimdiye kadar elde edilen en iyi çözümü kaydeder ve onu besin kaynağı değişkenine atar, böylece tüm popülasyon kötüleşse bile asla kaybolmaz.
- SSA algoritması, öncü salpın konumunu yalnızca gıda kaynağına göre günceller ve bu şimdiye kadar elde edilen en iyi çözümdür, böylece lider her zaman etrafındaki alanı araştırır ve kullanır.
- SSA algoritması, takipçi salplarının konumunu birbirlerine göre günceller, böylece kademeli olarak lider salp'a doğru ilerler.
- Takipçi tokatlarının kademeli hareketleri, SSA algoritmasının yerel optimada kolayca durmasını engeller.
- C1 parametresi, yinelemeler sırasında uyarlamalı olarak azaltılır, bu nedenle SSA algoritması önce arama alanını araştırır ve sonra onu kullanır.

- SSA algoritmasının yalnızca bir ana kontrol parametresi (c1) vardır.
- SSA algoritması basit ve uygulanması kolaydır.

7.5. Stokastik Fraktal Arama Algoritması (SFS)

Stokastik Fraktal Arama (SFS), fraktal kavramını kullanan büyüme süreci tarafından motive edilen, yakın zamanda önerilen bir optimizasyon algoritmasıdır. SFS, arama alanını keşfetmek için fraktallarda yaygın olarak görülen difüzyon özelliğini kullanır. Gauss yürüyüşleri gibi sto-chastic kuralları, rastgele fraktallar oluşturmak için yineleme sürecini değiştirmek için kullanılır. Başlangıçta konumlandırılan bir başlangıç parçacığı verildiğinde, yeni parçacıklar bu noktada rastgele rastgele bir şekilde yaratılır. Difüzyon süreci, algoritmanın global minimumları bulabilme yeteneğini artırır ve aynı zamanda yerel minimumda çarpılmayı önler. Bir sonraki güncelleme sürecinde, gruptaki diğer nokta konumuna bağlı olarak gruptaki noktaların konumları güncellenir. Bu işlemde, difüzyon işleminden en iyi birkaç parçacık alınır ve kalan parçacıklar reddedilir. Arama uzayının iyi organize edilmiş incelemesinin yanı sıra, algoritma güncellemek için Gaussian rasgele yaklaşımları kullanır ve böylece SFS algoritmasında çeşitlendirme özelliklerini sunar. Stokastik Fraktal Arama algoritması sonraki adımlarla açıklanabilir [69].

Başlatma: Parçacıkların (nokta) her konumu, daha düşük ve üst sınırlar aşağıdaki gibi belirtilerek sorun kısıtlamalarına bağlı olarak keyfi olarak başlatılır:

$$P = LB + \text{rand}(UB - LB) \quad (7.13)$$

Noktaların vektörü olduğunda, UB ve LB üst ve alt sınır vektörleri olduğunda, aralık [0,1] 'de eşit olarak dağıtılmış sayıyı oluşturur. Her parçacığın uygunluk işlevi, tüm parçacıklar arasında en iyi noktayı (BP) bulmak için değerlendirilir.

7.6. Geliştirilmiş Böbrek Esintili Algoritma (IKA)

Bu bölümde kısaca böbrek esintili algoritma (KA) ve geliştirilmiş böbrek esintili algoritmaları (IKA) özet olarak anlatılacaktır.

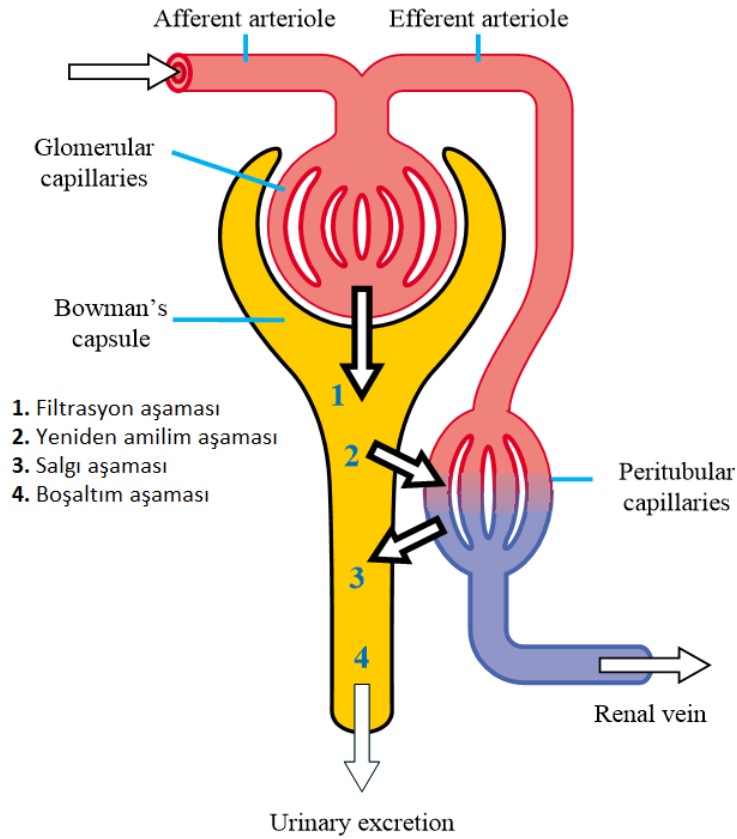
7.6.1. Böbrek Esintili Algoritma (KA)

Böbrekler, insan biyolojik yapısının üriner sisteminde idrar yoluyla fazla su ve atıkların giderilerek kanın süzülmesi ve kandaki iyon miktarlarının tutulması gibi hayati

fonksiyonları olan çok önemli organlardır. Bu nedenle, kan kimyası, sıvı dengesi ve dolayısıyla insan vücudunun genel sağlık durumundan sorumludurlar. Böbrek süreci, dört aşamada özetlenebilen tekrarlanan bir süreçtir: filtrasyon, yeniden emilim, salgılama ve boşaltım. Filtreleme, çözülmüş maddelerin hem Bowman kapsülü hem de kanın basıncı altında tübüllere aktarıldığı glomerüler kılcal damarlarda başlar. Daha sonra çözünen maddelerdeki faydalı maddeleri tübüllerden kan dolaşımına geri döndürmek için yeniden emilim gerçekleştirilir. Salgı, çözünen maddelerin böbrek tübülüne doğru hareketini takiben idrardaki fazla maddelerin (su, atıklar ve iyonlar) dışarı atılmasıdır. Şekil 5, böbreğin temel fizyolojik mekanizmasını göstermektedir.

İnsan vücudundaki böbreklerin yukarıda belirtilen işlevlerini simüle eden böbrekten esinlenen algoritma (KA), Jaddi ve ark. (2017) [70]. KA'nın ilk aşamasında, diğer popülasyon tabanlı algoritmalarda olduğu gibi, olası çözünen maddelerin rastgele bir popülasyonu oluşturulur ve bunların objektif işlevleri hesaplanır. Her bir çözünen, biyolojik böbrek sistemindeki plazmada bulunan su ve çözünen partiküller olarak düşünülebilir. Her yinelemede, şimdiye kadar elde edilen en iyi çözünen maddeye doğru ilerleyerek tüm çözünen maddeler için yeni bir çözünen oluşturulur. Bir çözünen maddenin bu algoritmadaki hareketi aşağıdaki gibi formüle edilir [70] [71]:

$$S_{i+1} = S_i + rand(S_{best} - S_i) \quad (7.14)$$



Şekil 7.7 Biyolojik böbrek sistemi. [72]

burada $S >$ popülasyonda bir çözünen madde, $S_i >$ i 'inci yinelemede bir çözünen, $\text{rand} >$ sıfır ve belirli bir sayı arasında rastgele bir sayı ve S_{best} ise önceki yinelemelerde şimdiye kadar elde edilen en iyi çözünen maddedir. Filtreleme operatörü uygulanarak, popülasyondaki daha yüksek kalitede çözünen maddeler filtrelenmiş kana (FB) süzülür ve kalan çözünenler atık (W) içine taşınır. Bu amaçla KA'da her yinelemede hesaplanan ve güncellenen bir filtreleme hızı kullanılır. Filtrasyon hızı (fr), biyolojik böbrek sistemindeki glomerüler filtrasyon hızına benzerdir ve şu şekilde hesaplanır [70] [71].

$$fr = \alpha \times \sum_{i=1}^p f(x_i) / p \quad (7.15)$$

burada α $[0,1]$ aralığında sabit bir değerdir, $f(x_i)$ i 'inci yinelemede çözünen x 'in nesnel fonksiyonudur, p popülasyon büyüklüğüdür. Bir çözünen maddenin FB veya W üyesi olarak kabul edilip edilmeyeceğini belirlemek için uyulması gereken kural şu şekildedir; Çözünen maddenin kalitesi fr 'den büyükse, FB'nin bir üyesi olarak kabul edilir, aksi takdirde W'nin bir üyesi olarak kabul edilir. Bir çözünen W'ye yerleştirilirse, yeniden emilim operatörü ona kendisini geliştirmek için bir şans daha sağlar. FB'nin bir parçası

olun. Bu, ancak (7.14) 'e hareket operatörünü tekrar uyguladıktan sonra filtreleme oranını karşılarsa olabilir. Bu, yararlı parçacıkların biyolojik böbrek sistemindeki kan dolaşımına yeniden emilmesine benzer. Bu şans karşılanmazsa, çözünen W'den atılır ve bunun yerine başka bir rastgele çözünen madde W'ye yerleştirilir. Ayrıca, filtreleme işleminden sonra FB'ye yerleştirilen bir çözünen, FB'deki en kötü çözünen maddeden daha iyiyse, en kötü çözünen FB'den salgılanır (çıkarılır). Ancak, bu çözünen FB'deki en kötü çözünen maddeden daha iyi değilse, o zaman salgılanır. Bu, biyolojik böbrek sistemindeki zararlı partiküllerin kan dolaşımından salgılanmasına benzer. Bundan sonra, şimdiye kadar elde edilen en iyi çözünen maddeyi güncellemek için FB'deki çözünenler sıralanır. Son olarak, FB ve W yeni bir popülasyon oluşturmak için birleştirilir ve filtrasyon hızı güncellenir. Fesih kriterine ulaşıldığında tekrarlanan bu süreç sona erer. Rasgele çözünen maddelerin eklenmesi, biyolojik böbrek sistemindeki glomerüler kılcal damarlara sürekli su ve çözelti ilavesi olarak düşünülebilir.

7.6.2. Geliştirilmiş Böbrek Esintili Algoritma (IKA)

Tüm optimizasyon algoritmalarının ana amacı, global olarak optimum çözümü bulmak olduğundan, algoritmanın keşif (global arama) ve kullanım (yerel arama) aşamaları arasında bir denge sağlamak çok önemlidir. Bu nedenle, bu amaca ulaşmak için, temel algoritma, her ikisi de Jaddi ve ark. Tarafından önerilen iki modifikasyon yoluyla geliştirilecektir. [73]. Bu yazıda, değiştirilmiş algoritmaya geliştirilmiş KA (IKA) denir.

1) Filtrasyon oranının kontrol edilmesi

Temel algoritmaya uygulanan ilk değişiklik, arama işlemi sırasında filtreleme oranını kontrol etmektir. Temel KA'nın yakınsama oranı, önceden sabit bir değere ayarlanan filtreleme hızındaki α değerine göre ayarlandığından, her yinelemede dinamik olarak ince ayarlanmış bir değere değiştirmek algoritmanın performansını önemli ölçüde artıracaktır. . Eğer α değeri büyükse, daha fazla çözünen W'ye hareket edecek ve bu, arama alanından daha fazla yararlanılmasına neden olacaktır. Bunun aksine, α 'nin değeri küçükse, daha fazla çözünen FB'ye taşınacak ve bu, arama alanının daha fazla araştırılmasına neden olacaktır. Yüksek sömürü olduğu zaman, FB'de yeterli keşif yapmak için yeterli çözünen madde olmayabilir. Bu, yerel optimal durgunluğa yol açabilir ve dolayısıyla algoritmanın performansını düşürebilir. Bu nedenle, algoritmanın yakınsama yeteneğini etkilememesi için uygun şekilde ince ayar yapmak çok önemlidir. Arama sürecinin başlangıcında, α değeri, arama alanının etkili bir şekilde

araştırılması için yeterince küçük olmalıdır. Sonraki yinelemelerde, algoritmanın yakınsamasını hızlandırmak için arama alanından yararlanmayı yavaşça artırmak için artırılması gerekir. Önerilen IKA'da, initi başlangıçta minimum değer olan sıfıra ayarlanmıştır ve ardından her yinelemede (7.16) 'da verilen oranla maksimum 1'e ulaşana kadar yavaşça artırılır [73].

$$IR_i^\alpha = 1 - \frac{iterasyon_i}{num_ite} \quad (7.16)$$

Burada IR_i^α , i th 'inci yinelemedeki artan oranı, $iteration_i$ geçerli yinelemenin sayısı ve num_ite toplam yineleme sayısını temsil eder. Filtreleme hızının dinamik değişimi, yalnızca algoritmadaki parametre sayısını azaltmakla kalmaz, aynı zamanda arama işlemi sırasında keşif ve kullanım aşamalarını da dengeler.

2) Kaotik lojistik haritanın kullanımı

Bir kaotik haritanın, yani lojistik haritanın kullanılması, keşif ve kullanım aşamaları arasındaki dengeyi korurken, algoritmanın yakınsama hızını daha da artırmak için gerçekleştirilen ikinci modifikasyondur. Lojistik harita, IKA'nın hareket formülünde rastgele bir sayı kullanmak yerine kaotik bir dizi oluşturmak için kullanılır. Bu tür bir harita, bir polinom haritalama biçimidir ve aşağıdaki gibi matematiksel olarak verilmiştir[73].

$$y_{k+1} = r \cdot y_k \cdot (1 - y_k) \quad (7.17)$$

Denklemden y bir değişken olduğunda $k=1,2,3,\dots, y_k$ ve y_{k+1} için sırasıyla y 'nin k th ve $(k+1)$ 'inci değerleridir.

KA ve IKA algoritmaları hakkında daha ayrıntılı bilgi ve işleme adımları için lütfen sırasıyla [70] ve [73] 'e bakın.

7.7. Atom Arama Optimizasyon Algoritması (ASO)

ASO, küresel optimizasyon problemleri için etkili bir arama mekanizması tasarlamak üzere etkileşim ve kısıtlama kuvvetleri tarafından kontrol edilen atomik hareketi taklit eden, yakın zamanda önerilen, fizikten esinlenen, popülasyon tabanlı bir sezgisel algoritmadır. [74]. Aslında itmenin vektör toplamı olan ve i 'inci atom üzerine dinamik olarak değişen komşu atomların uyguladığı çekimin vektör toplamı olan d 'inci boyutta

i 'inci Atoma etki eden toplam etkileşim kuvvetleri şu şekilde verilmiştir:

$$F_i^d(t) = \sum_{j \in K_{best}} rand_j F_{ij}^d(t) \quad (7.18)$$

$rand_j$, $[0,1]$ aralığında rastgele bir sayıdır ve K_{best} en iyi uygunluk fonksiyonu değerlerine sahip ilk K atomlarından oluşan atom popülasyonunun bir alt kümesidir. ASO'nun erken yinelemelerde daha fazla keşif yapmasını sağlamak için, her atomun K komşuları kadar daha iyi uygunluk değerlerine sahip atomlarla etkileşime girmesi ve algoritmanın son yinelemelerde daha fazla yararlanmasını sağlamak için her atomun birkaç atom olarak daha iyi etkileşime girmesi gerekir. K komşuları olarak uygunluk değerleri. Bu nedenle, aşağıdaki gibi iterasyonların seyri ile K değerinin kademeli olarak azaltılması gerekir.

$$K(t) = N - (N - 2) \times \sqrt{\frac{t}{T}} \quad (7.19)$$

Denklemden N atomik bir sistemdeki toplam atom sayısı, t mevcut yineleme ve T maksimum yineleme sayısıdır. (7.19)'teki F_{ij}^d , d 'inci boyutta j 'inci en iyi atomun i 'inci atoma uyguladığı etkileşim kuvvetidir. Etkileşim kuvveti, Lennard-Jones (L-J) potansiyelinin gradyanıdır ve optimizasyon problemlerinin üstesinden gelmek için bu modelin pozitif çekim ve negatif itme kuvvetleri ile revize edilmiş versiyonu şu şekilde verilmiştir. [74]

$$F_{ij}^d = -\eta(t) \left[2(h_{ij}(t))^{-13} - (h_{ij}(t))^{-7} \right] \frac{\vec{r}_{ij}}{r_{ij}} \quad (7.20)$$

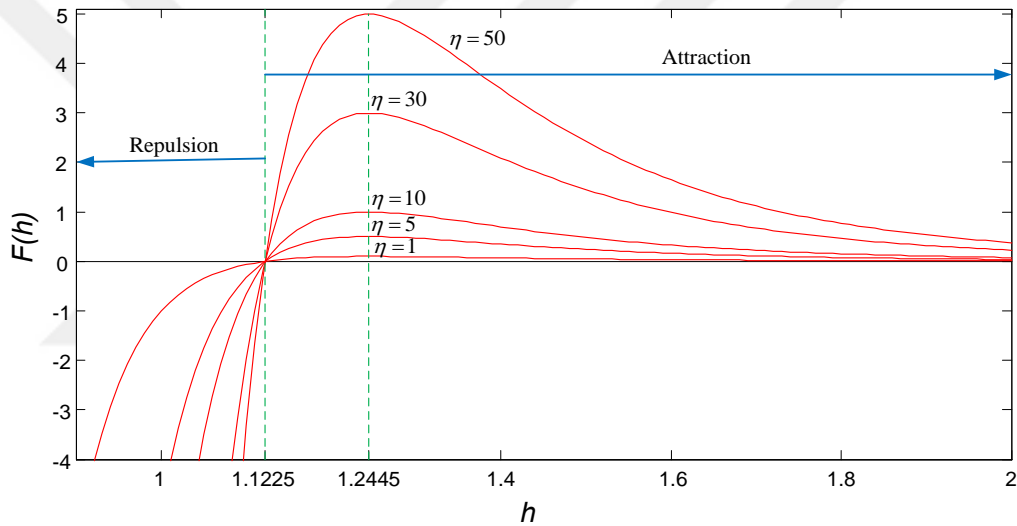
Denklemden $\eta(t)$, itme veya çekici bölgeleri ayarlamak için derinlik fonksiyonudur ve $h_{ij}(t) = r_{ij}/\sigma(t)$ iki atom arasındaki mesafenin ölçeklendirilmiş mesafe olarak adlandırılan uzunluk ölçeğine oranıdır. $\vec{r}_{ij} = \vec{x}_j - \vec{x}_i$ denklemi $\vec{x}_j = (x_{j1}, x_{j2}, x_{j3})$ denkleminin j 'inci atomun konum vektörü ve $\vec{x}_i = (x_{i1}, x_{i2}, x_{i3})$ denkleminin i 'inci atomun konum vektörü olduğu konum farkı vektörüdür. Dolayısıyla r_{ij} , i 'inci ve j 'inci atomlar arasındaki öklid mesafesidir ve şu şekilde verilir.

$$r_{ij} = \|\vec{x}_j - \vec{x}_i\| = \sqrt{(x_{j1} - x_{i1})^2 + (x_{j2} - x_{i2})^2 + (x_{j3} - x_{i3})^2} \quad (7.21)$$

Denklem (7.20) 'deki derinlik fonksiyonu şu şekilde tanımlanır:

$$\eta(t) = \alpha \left(1 - \frac{t-1}{T}\right)^3 e^{-\frac{20t}{T}} \quad (7.22)$$

Denklem (7.22)'da α derinlik ağırlığıdır ve burada 50'ye eşittir. Farklı derinlik değerlerine ($\eta = 1, 5, 10, 30$ ve 50) karşı $0,9$ ile 2 arasında değişen ölçeklenmiş mesafeye (h) sahip etkileşim kuvveti fonksiyonu Şekil 7.8'de gösterilmiştir.



Şekil 7.8 Farklı Derinlik Değerlerine ($\eta = 1, 5, 10, 30$ Ve 50) Sahip Ölçeklendirilmiş Mesafeye (h) Karşı Revize Edilmiş Etkileşim Kuvveti Fonksiyonu [74]

İki atom arasındaki ölçeklendirilmiş mesafe şu şekilde verilmiştir.

$$h_{ij}(t) = \begin{cases} h_{\min} & \frac{r_{ij}(t)}{\sigma(t)} < h_{\min} \\ \frac{r_{ij}(t)}{\sigma(t)} & h_{\min} \leq \frac{r_{ij}(t)}{\sigma(t)} \leq h_{\max} \\ h_{\max} & \frac{r_{ij}(t)}{\sigma(t)} > h_{\max} \end{cases} \quad (7.23)$$

h_{\min} and h_{\max} sırasıyla, ölçeklenen mesafenin (h) alt ve üst sınırlarını gösterir ve aşağıdaki

gibi tanımlanır.

$$\begin{cases} h_{\min} = g_0 + g(t) \\ h_{\max} = u \end{cases} \quad (7.24)$$

Denklemden g_0 , 1.1'e ayarlanan en düşük sınırdır ve u , 1.24'e ayarlanan üst sınırdır ve $g(t)$ algoritmayı keşiften yararlanmaya sürükleyebilen sürüklenme faktörüdür;

$$g(t) = 0.1 \times \sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) \quad (7.25)$$

Denklemin (7.23) 'daki uzunluk ölçüğü $\sigma(t)$ çarpışma çapını belirtir ve şu şekilde tanımlanır.

$$\sigma(t) = \left\| x_{ij}(t), \frac{\sum_{j \in K_{best}} x_{ij}(t)}{K(t)} \right\|_2 \quad (7.26)$$

ASO'daki her atomun en iyi atomla kovalent bir bağa sahip olduğunu varsayarsak, her atom ile en iyi atom arasındaki ağırlıklı konum farkı olan ortaya çıkan geometrik kısıtlama kuvveti şu şekilde verilebilir.

$$G_i^d(t) = \lambda(t)(x_{best}^d(t) - x_i^d(t)) \quad (7.27)$$

Denklemden $x_{best}^d(t)$ d'inci boyuttaki en iyi atomun konumudur ve $\lambda(t)$ Lagrangian çarpanı olarak ifade edilir.

$$\lambda(t) = \beta e^{-\frac{2\alpha t}{T}} \quad (7.28)$$

Denklemin (7.28)'de β bir çarpan ağırlığıdır ve bu yazıda 0.2'ye eşittir. Sırasıyla L-J potansiyeli ve bağ uzunluğu potansiyelinden kaynaklanan etkileşim kuvveti ve kısıtlama kuvveti tanımlandıktan sonra, iterasyon t'de d'inci boyuttaki i'inci atomun ivmesi artık aşağıdaki gibi hesaplanabilir.

$$\begin{aligned}
a_i^d(t) &= \frac{F_i^d(t)}{m_i^d(t)} + \frac{G_i^d(t)}{m_i^d(t)} \\
&= -\alpha \left(1 - \frac{t-1}{T}\right)^3 e^{-\frac{20t}{T}} \sum_{j \in K_{best}} \frac{rand_j [2(h_{ij}(t))^{-13} - (h_{ij}(t))^{-7}]}{m_i(t)} \cdot \frac{(x_j^d(t) - x_i^d(t))}{\|\vec{x}_i(t), \vec{x}_j(t)\|_2} \\
&\quad + \beta e^{-\frac{20t}{T}} \frac{(x_{best}^d(t) - x_i^d(t))}{m_i(t)} \tag{7.29}
\end{aligned}$$

Denklem (7.29)'de $m_i^d(t)$, t yinelemesinde n'inci boyuttaki i'inci atomun kütlesi olup, aşağıdaki gibi uygunluk fonksiyonu değeriyle hesaplanır.

$$M_i(t) = e^{\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \tag{7.30}$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \tag{7.31}$$

$Fit_i(t)$ iterasyon t'deki i'inci atomunun uygunluk fonksiyonu değeridir, $Fit_{best}(t)$ ve $Fit_{worst}(t)$ sırasıyla iterasyon t'deki en iyi ve en kötü atomların uygunluk değerleridir ve şu şekilde tanımlanır.

Son olarak i'inci atomunun iterasyondaki hız ve konum güncellemesi (t + 1) şu şekilde tanımlanır.

$$v_i^d(t+1) = rand_i^d \cdot v_i^d(t) + a_i^d(t) \tag{7.32}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{7.33}$$

ASO'nun daha ayrıntılı açıklaması için [74] 'ye başvurulabilir.

7.8. Gri Kurt Optimizasyon Algoritması (GWO)

Gri kurt algoritmasında ilk önce önerilen yöntemin ilhamı tartışılacaktır. Daha sonra matematiksel model verilir. 3.1. İlham Bozkurt (Canis lupus), Canidae ailesine aittir. Gri kurtlar en tepedeki yırtıcı hayvan olarak kabul edilir, yani besin zincirinin en üstünde yer

alırlar. Gri kurtlar çoğunlukla bir sürü içinde yaşamayı tercih ederler. Grup büyüklüğü ortalama 5–12'dir. Özellikle ilgi çekici olan, şekil 7.9'da gösterildiği gibi çok katı bir sosyal baskın hiyerarşiye sahip olmalarıdır. Liderler, alfa adı verilen bir erkek ve bir kadındır. Alfa çoğunlukla avlanma, uyuma yeri, uyanma zamanı vb. İle ilgili kararlar vermekten sorumludur. Alfa'nın kararları pakete dikte edilir. Bununla birlikte, bir alfa paketindeki diğer kurtları takip eden bir tür demokratik davranış da gözlemlenmiştir. Toplantılarda, tüm paket, kuyruklarını aşağıda tutarak alfayı kabul eder. Alfa kurdu, emirlerinin sürü tarafından takip edilmesi gerektiğinden baskın kurt olarak da adlandırılır [75]. Alfa kurtların sadece sürüde çiftleşmesine izin verilir. İlginç bir şekilde, alfa mutlaka paketin en güçlü üyesi değil, paketi yönetme açısından en iyisi. Bu, bir paketin organizasyonunun ve disiplininin gücünden çok daha önemli olduğunu gösterir.

Gri kurt hiyerarşisindeki ikinci seviye betadır. Birinciler, karar verme veya diğer sürü faaliyetlerinde alfaya yardımcı olan ikincil kurtlardır. Beta kurt erkek veya dişi olabilir ve alfa kurtlardan birinin ölmesi veya çok yaşlanması durumunda muhtemelen alfa olmaya en iyi adaydır. Beta kurt, alfaya saygı duymalı, ancak diğer alt düzey kurtlara da komuta ediyor. Alfa için bir danışman ve paket için disiplin rolünü oynar. Beta, paket boyunca alfa'nın komutlarını güçlendirir ve alfaya geri bildirim sağlar.

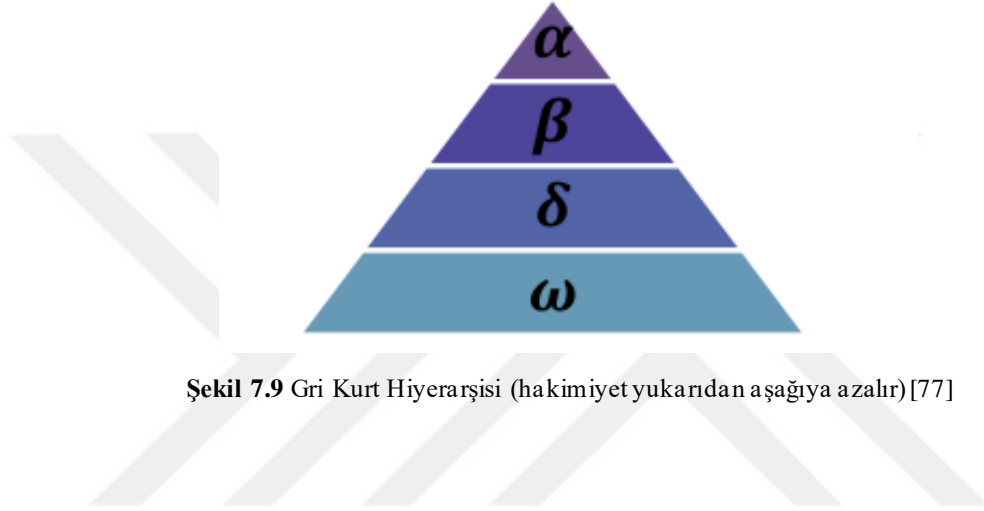
En alt sıradaki gri kurt omegadır. Omega günah keçisi rolünü oynar. Omega kurtları her zaman diğer baskın kurtların tümüne boyun eğmek zorundadır. Onlar yemesine izin verilen son kurtlardır. Omega, pakette önemli bir birey değil gibi görünebilir, ancak tüm paketin omegayı kaybetme durumunda iç çatışmalar ve sorunlarla karşılaştığı gözlemlenmiştir. Bunun nedeni, tüm kurtların omegalar tarafından şiddet ve hayal kırıklığı yaratmasıdır. Bu, tüm paketi tatmin etmeye ve hakimiyet yapısını korumaya yardımcı olur. Bazı durumlarda omega aynı zamanda paketindeki bebek bakıcılarıdır.

Bir kurt bir alfa, beta veya omega değilse, ona alt ordinat (veya bazı referanslarda delta) denir. Delta kurtları alfa ve betalara boyun eğmek zorundadır, ancak omega'ya hükmederler. İzciler, duyarlılar, yaşlılar, avcılar ve bakıcılar bu kategoriye girer. İzciler, bölgenin sınırlarını izlemek ve herhangi bir tehlike durumunda sürüyü uyarmakla sorumludur. Nöbetçiler, paketin güvenliğini korur ve garanti eder. Yaşlılar, alfa veya beta olan deneyimli kurtlardır. Avcılar, av avlarken ve sürü için yiyecek sağlarken alfa ve betalara yardım eder. Son olarak bakıcılar paketindeki zayıf, hasta ve yaralı kurtların bakımından sorumludur.

Kurtların sosyal hiyerarşisine ek olarak, grup avı gri kurtların bir başka ilginç sosyal davranışıdır. Muro ve arkadaşlarına [76] göre gri kurt avcılığının ana aşamaları şöyledir:

- Avı takip etmek, kovalamak ve ona yaklaşmak.
- Avı hareket etmeyi bırakana kadar takip etmek, çevrelemek ve taciz etmek.
- Avına saldırmak.

Bu çalışmada, bu avlanma tekniği ve gri kurtların sosyal hiyerarşisi, GWO'yu tasarlamak ve optimizasyonu gerçekleştirmek için matematiksel olarak modellenmiştir.



Şekil 7.9 Gri Kurt Hiyerarşisi (hakimiyet yukarıdan aşağıya azalır)[77]

8. PITCH KONTROL SİSTEMİ SİMÜLASYON SONUÇLARI

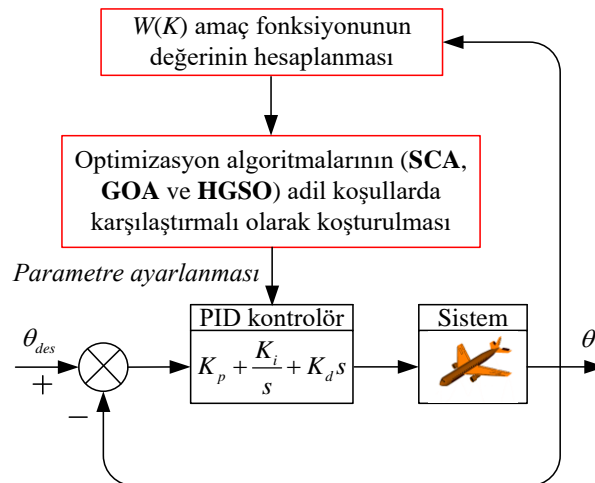
Son beş yıl içerisinde önerilen etkili sezgisel-üstü algoritmalar (SCA, GOA ve HGSO) çeşitli analizler vasıtasıyla detaylıca karşılaştırılmıştır.

8.1.Amaç Fonksiyonu ve Önerilen Tasarım

PID parametrelerini ayarlamak için genel olarak “integrated absolute error (IAE)”, “integral of squared-error (ISE)”, ve “integrated of time-weighted-squared-error (ITSE)” gibi çeşitli performans kriterleri amaç fonksiyonu olarak kullanılabilir [78]. Bununla birlikte, yukarıdaki performans kriterlerinin kendi dezavantajları da vardır [78]. Örneğin, IAE ve ISE’nin en aza indirilmesi uzun yerleşim süresine yol açabilir. ITSE performans kriteri IAE ve ISE kriterlerinin sahip olduğu dezavantajı elimine edebilir, ancak analitik formülün türetilme süreçleri karmaşık ve zaman alıcıdır [78]. Bu çalışmada kaynak [79]’te verilen kriter, amaç fonksiyonu olarak kullanılmıştır. Söz konusu bu kriter eşitlik (8.1)’de verilmiştir. Burada $K = [K_p, K_i, K_d]$, β ağırlık faktörü, M_p , E_{ss} , T_s ve T_r ise sırası ile maksimum aşım, sürekli hal hatası, yerleşme ve yükselme zamanlarıdır.

$$W(K) = (1 - e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(T_s - T_r) \quad (8.1)$$

SCA, GOA ve HGSO tabanlı PID kontrolör ile ayarlanmış uçak alçalma/yükselme açısı kontrol sistemine ait blok diyagramı Şekil 8.1’de gösterilmektedir. Bu çalışmada; $\beta = 1$, PID kontrolör parametrelerinin alt sınırlarının 0.001 ve üst sınırlarının ise 100 alınması uygun görülmüştür.



Şekil 8.1. Önerilen yaklaşımın blok diyagramı

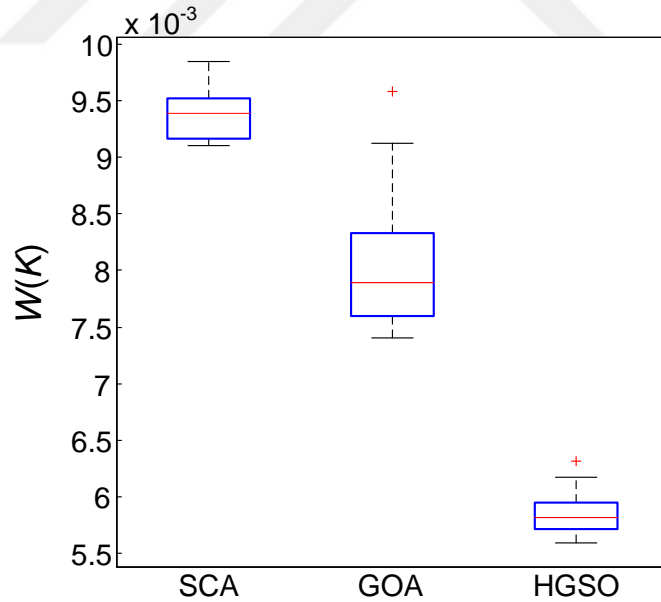
8.2. İstatistiksel Test Analizi

HGSO algoritma tabanlı tasarım yaklaşımı MATLAB / Simulink ortamında uygulanmış ve analizler Intel® i7 2.50 GHz işlemci ve 16.00 RAM'den oluşan Windows 7 işletim sistemine sahip bir bilgisayarda gerçekleştirilmiştir.

Uygun bir kıyaslama için her bir algoritmadaki maksimum iterasyon sayısı 100 ve sürü büyüklüğü ise 30 alınarak alçalma/yükselme kontrol sistemi optimize edilmiştir. Her bir algoritma 15 kere çalıştırdıktan sonra elde edilen $W(K)$ amaç fonksiyonunun istatistiksel sonuçları nümerik olarak Tablo 8.1'de listelenmiştir. Bahsi geçen $W(K)$ amaç fonksiyonunun istatistiksel kutu grafiği ise Şekil 8.2'de gösterilmiştir.

Tablo 8.1 $W(K)$ amaç fonksiyonunun istatistiksel değerleri

Algoritma	En iyi	En kötü	Ortalama	Standart sapma
SCA	0.0091	0.0098	0.0094	2.3515E-04
GOA	0.0074	0.0096	0.0081	6.3267E-04
HGSO	0.0056	0.0063	0.0059	2.0814E-04



Şekil 8.2 SCA, GOA ve HGSO algoritmalarına ait kutu grafikleri

Söz konusu tablo ve şekilden görüleceği üzere, $W(K)$ amaç fonksiyonunun minimize edilmesinde HGSO algoritması en iyi istatistiksel performansı sergilemiştir. Çoklu

çalıştırmalardan sonra her bir algoritmanın minimum $W(K)$ değerine karşılık gelen PID kontrolör parametreleri Tablo 8.2'de listelenmiştir.

Tablo 8.2 PID kontrolörün optimize edilen parametreleri

Kontrolör Türü	K_p	K_i	K_d
SCA ayarlı PID	70.8938	64.8932	72.4551
GOA ayarlı PID	63.8156	21.5434	77.6758
HGSO ayarlı PID	69.7726	3.6054	95.1465

Tablodaki parametreler kullanıldığında SCA, GOA ve HGSO ayarlı PID kontrolör içeren sistemlerin kapalı çevrim transfer fonksiyonları sırasıyla (8.2) – (8.4) eşitlikleri ile verilmiştir.

$$T_{SCA}(s) = \frac{83.4s^3 + 94.45s^2 + 87.27s + 11.51}{s^4 + 84.14s^3 + 95.38s^2 + 87.27s + 11.51} \quad (8.2)$$

$$T_{GOA}(s) = \frac{89.41s^3 + 87.23s^2 + 36.12s + 3.822}{s^4 + 90.14s^3 + 88.16s^2 + 36.12s + 3.822} \quad (8.3)$$

$$T_{HGSO}(s) = \frac{109.5s^3 + 97.19s^2 + 16.53s + 0.6397}{s^4 + 110.3s^3 + 98.11s^2 + 16.53s + 0.6397} \quad (8.4)$$

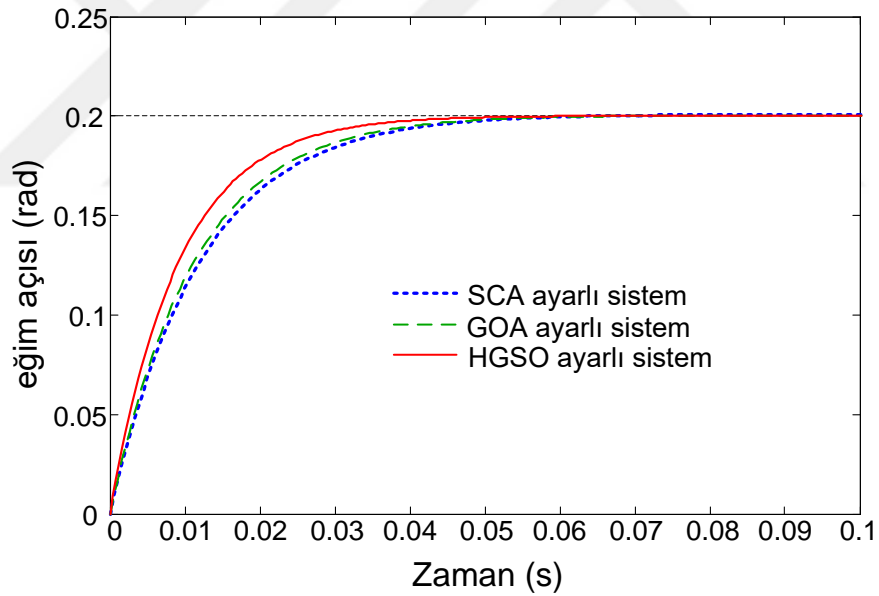
8.3. Geçici Hal Cevap Analizi

Farklı yaklaşımlar kullanılarak tasarlanan alçalma/yükselme açısı kontrol sistemine ait basamak cevapları ($\theta_{des} = 0.2 \text{ rad}$ girişi için) Şekil 8.3 ile verilmiştir. Şekilden görüleceği üzere HGSO ayarlı PID kontrolör ilaveli sistem daha iyi zaman cevabına sahiptir. Bu çalışmada basamak girişinin geçici hal cevabının karakterizasyonu için tepe değeri, maksimum yüzdeler aşım, yükselme zamanı, yerleşme zamanı ve tepe zamanı gibi parametreler performans ölçütü olarak kullanılmıştır. Performans ölçütleri açısından

karşılaştırmalı geçici hal cevabı sonuçları Tablo 8.3'te sunulmuştur. Tablodaki değerlere bakıldığında en iyi (minimum) sonuçlar HGSO ayarlı PID kontrolör ile tasarlanan sisteme aittir

Tablo 8.3 Geçici Hal Cevap Analizi Sonuçları

Kontrolör Türü	Tepe Değeri (rad)	Aşım (%)	Yükselme Zamanı (s)	Yerleşme Zamanı (s)	Tepe Zamanı (s)
SCA ayarlı PID	0.2007	0.3464	0.0260	0.0447	0.0827
GOA ayarlı PID	0.2002	0.1158	0.0244	0.0426	0.0770
HGSO ayarlı PID	0.2000	0.0000	0.0200	0.0352	0.0627

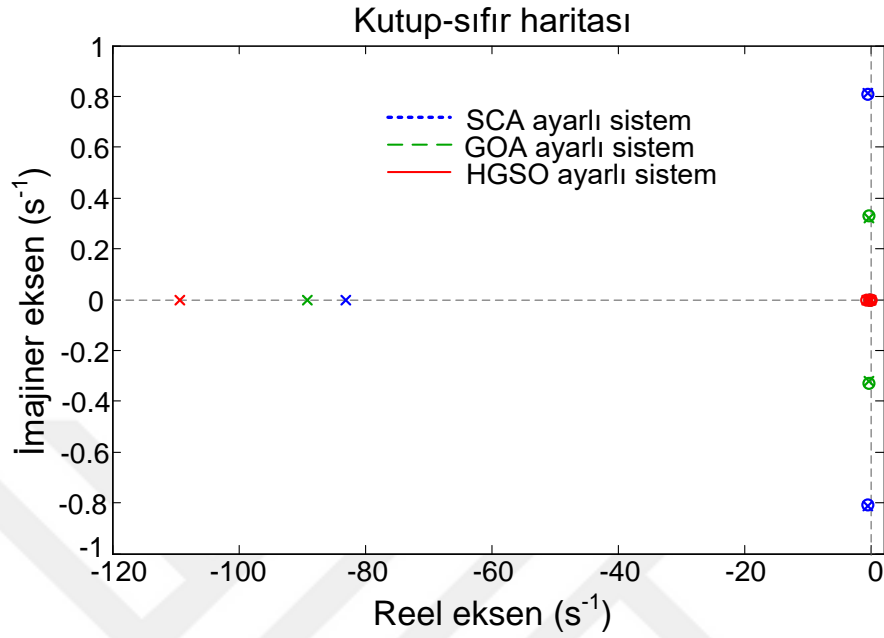


Şekil 8.3 Alçalma/yükselme (eğim) açısının değişim eğrileri

8.4. Kutup-sıfır Haritası Analizi

Kutup-sıfır haritası analizi, kapalı çevrim kutuplarının yerini ve karşılık gelen sönüm oranlarını verir. Aynı zamanda bir sistemin kararlılık özelliklerini de tanımlamaya yarar. Farklı algoritmalarla tasarlanan sistemlerin kutup-sıfır haritası çizimleri Şekil 8.4'te gösterilmiştir. Kapalı çevrim kutuplarının nümerik sonuçları ve ilgili sönüm oranları Tablo 8.4'te verilmiştir. Tablo 8.4 incelendiğinde, HGSO algoritması tarafından

ayarlanan sistemin hiçbir karmaşık eşlenik kutup çiftine sahip olmadığı ve SCA ile GOA algoritmaları tarafından ayarlanan sistemlere nazaran en büyük sönüm oranına sahip olduğu fark edilmektedir.



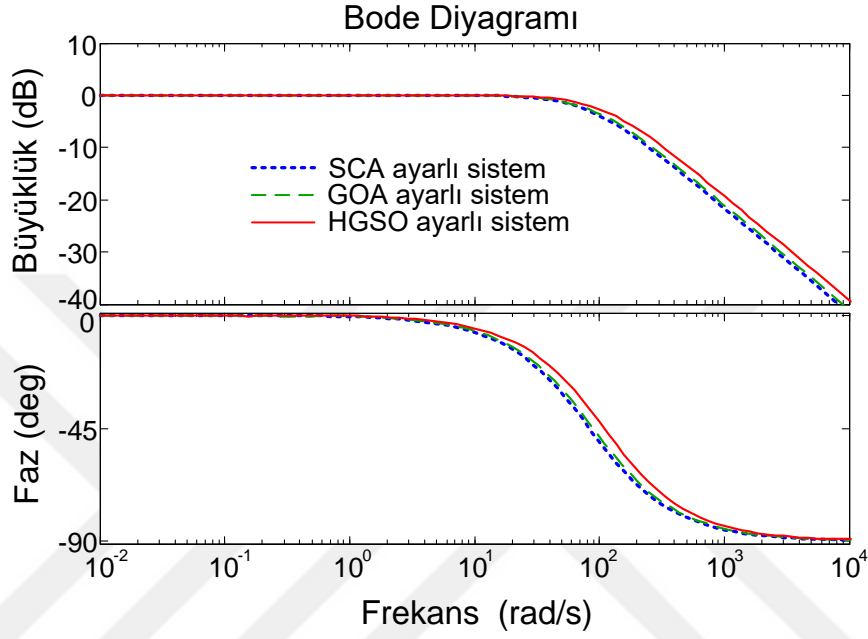
Şekil 8.4 Kutup-sıfır haritası çizimleri

Tablo 8.4 Kapalı çevrim kutupları ve sönüm oranları

Kontrolör Türü	Kapalı Çevrim Kutupları	Sönüm Oranları
SCA ayarlı PID	-0.1545	1
	$-0.4910 + 0.8106i$	0.5181
	$-0.4910 - 0.8106i$	0.5181
	-82.9991	1
GOA ayarlı PID	-0.1554	1
	$-0.4144 + 0.3226i$	0.7890
	$-0.4144 - 0.3226i$	0.7890
	-89.1604	1
HGSO ayarlı PID	-0.0564	1
	-0.1507	1
	-0.6887	1
	-109.3578	1

8.5. Bode Analizi

Bode analizi, kontrol sisteminin frekans cevabı hakkında bilgi verir [80]. Bu analizden elde edilen büyüklük (magnitude) ve faz grafikleri Şekil 8.5'te karşılaştırmalı olarak gösterilmiştir. Bu grafikten hesaplanan kazanç payları, faz payları ve bant genişlikleri de Tablo 8.5'te gösterilmektedir. Sonuç olarak, en iyi frekans cevabının HGSO algoritması tarafından ayarlanan sisteme ait olduğu söylenebilir.



Şekil 8.5 Bode diyagramları

Tablo 8.5 Frekans cevabı analizi sonuçları

Kontrolör Türü	Kazanç Payı	Faz Payı	Bant Genişliği
SCA ayarlı PID	∞ dB	174.4418°	83.5953 Hz
GOA ayarlı PID	∞ dB	175.9787°	89.4385 Hz
HGSO ayarlı PID	∞ dB	177.3049°	109.4116 Hz

9. AVR KONTROL SİSTEMİ SİMÜLASYON SONUÇLARI

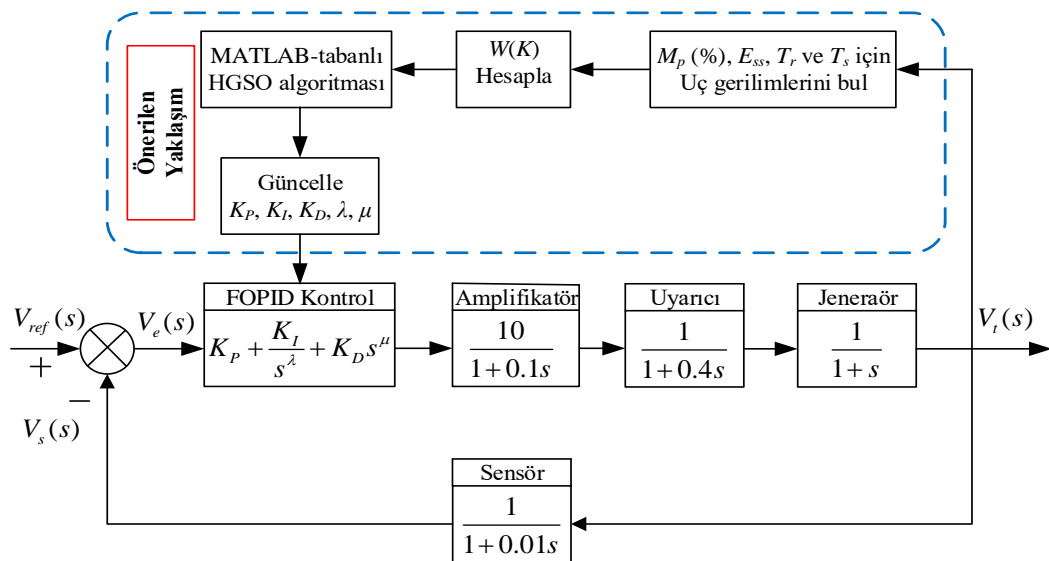
Önerilen HGSO-FOPID denetleyicisinin verimliliğini doğrulamak için kutu grafiği, yakınsama profili, geçici ve frekans yanıt analizleri gibi çeşitli analizler yapılmıştır. Ayrıca, önerilen yaklaşımın üstünlüğünü doğrulamak için, geçici yanıt ve frekans yanıt analizleri açısından diğer sezgisel-üstü algoritmalara dayanan çeşitli tasarım yaklaşımlarıyla performans karşılaştırmaları yapılmıştır. Karşılaştırmalar için seçilen kontrolörler SSA-FOPID [31], SFS-PID [81] ve IKA-PID [61] 'dir, çünkü bunlar literatürdeki en iyi kontrolör parametrelerini bulan en son çalışmalardır. Algoritma, amaç işlevi ve denetleyici yapısı açısından da birbirlerinden farklıdır.

9.1.Amaç Fonksiyonu ve Tasarımı

Bu sistemde, denklem (9.1)'de verilen etkili bir zaman alanı amaç fonksiyonu, beş denetleyici parametresini optimize etmek için kullanılmıştır [82], burada K , FOPID denetleyici parametre vektörünü temsil eder ve $K = [K_p, K_I, K_D, \lambda, \mu]$ olarak verilir.

$$W(K) = (1 - e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(T_s - T_r) \quad (9.1)$$

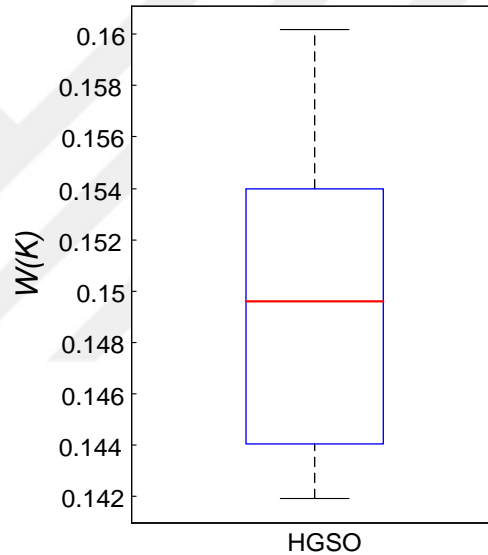
M_p maksimum aşmadır, $T_s \pm\% 2$ bant için yerleşme süresidir ve T_r yükselme süresidir. β genellikle 1'e ayarlanmış bir ağırlık faktörüdür [78]. Şekil 9.1'de gösterildiği gibi AVR sisteminin optimal kontrolör değerlerini belirlemek için HGSO algoritması kullanılmıştır.



Şekil 9.1 HGSO-FOPID denetleyicili AVR sistemi

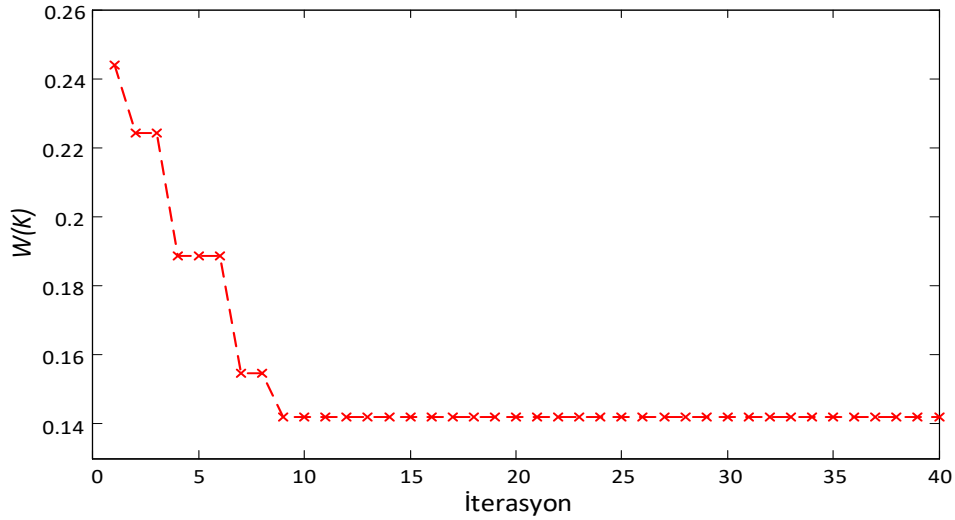
9.2. Kutu Grafiği ve Yakınsama Profil Analizleri

Önerilen HGSO algoritma tabanlı tasarım yaklaşımı MATLAB / Simulink ortamında uygulanmış ve analizler önceki sistemlerde kullanılan (8. bölümdeki pitch kontrol sistemi) bilgisayar üzerinden gerçekleştirilmiştir. Popülasyon boyutu ve maksimum yineleme sayısı sırasıyla 50 ve 40 olarak seçildi. HGSO algoritması 20 kez bağımsız olarak çalıştırıldı. Tüm çalışmalar için $W(K)$ amaç fonksiyonunun elde edilen değerleri şekil 9.2'de gösterilmektedir. Ortalama, en kötü, en iyi, standart sapma ve medyan değerleri sırasıyla 01494, 01602, 01419, 00058 ve 01496 gibi $W(K)$ objektif fonksiyonunun istatistiksel değerleri dikkate alındığında HGSO algoritmasının çok iyi bir istatistiksel performansa sahip olduğu sonucuna varılabilir.



Şekil 9.2 Tüm Çalışmalar İçin $W(K)$ 'nin Kutu Grafiği

Şekil 9.3'te HGSO algoritmasının en iyi çalışmasının yakınsamasını göstermektedir. Bu şekilden, HGSO algoritmasının en düşük $W(K)$ değerini yüksek yakınsama hızıyla sağlayabileceği ve en iyi çalışma için en uygun değerleri bulmak için yalnızca 9 yineleme aldığı sonucuna varılabilir. Optimizasyon işleminden sonra bulunan elde edilen HGSO-FOPID parametreleri Tablo 9.1'de gösterilmektedir. Bu kazançlar kullanılarak elde edilen transfer fonksiyonu (9.2) 'de verilmiştir.



Şekil 9.3 HGSO algoritmasının yakınsama özellikleri

$$T_{HGSO-FOPID}(s) = \frac{0.04559s^{3.6352} + 4.559s^{2.6352} + 0.26632s^{2.2689} + 26.632s^{1.2689} + 0.11314s + 11.314}{0.0004s^{5.2689} + 0.0454s^{4.2689} + 0.555s^{3.2689} + 4.559s^{2.6352} + 1.51s^{2.2689} + 27.632s^{1.2689} + 11.314} \quad (9.2)$$

9.3. Geçici Hal Analizi

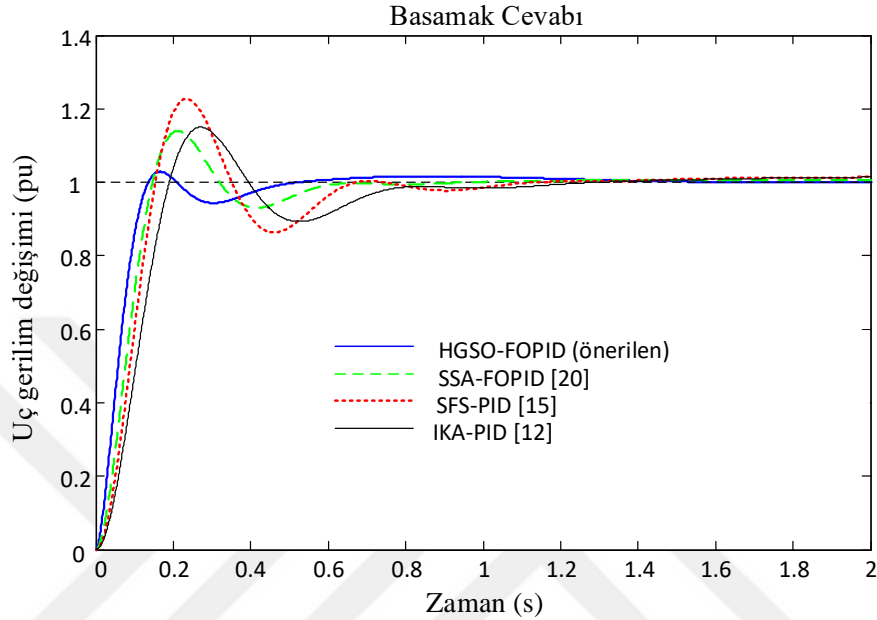
Karşılaştırma amacıyla seçilen diğer kontrolörler için kazanç parametreleri Tablo 9.1'de listelenmiştir. Bu kontrolörlerin transfer fonksiyonları sırasıyla (9.3), (9.4) ve (9.5) 'te verilmiştir. Farklı yaklaşımlarla tasarlanmış AVR sistemi adım yanıtları için karşılaştırmalı sonuçlar şekil 9.4'te sunulmuştur. Şekilden görüldüğü gibi, HGSO-FOPID daha iyi zaman yanıtına sahiptir.

Zaman alanı performans özellikleri açısından karşılaştırmalı geçici tepki analizi sonuçları Tablo 9.2'de verilmiştir. En düşük tepe değeri (pu cinsinden) aşma yüzdesi, yükselme süresi (% 10 \rightarrow % 90 için) ve çökme süresinin (\pm % 2 tolerans için) önerilen HGSO-FOPID denetleyicisi tarafından bulunduğu açıkça görülebilir.

Tablo 9.1 Çeşitli Algoritmalarla Ayarlanan Kontrolör Parametreleri

Ayarlama Yöntemi	PID / FOPID kontrolörlerinin parametreleri				
	K_p	K_I	K_D	λ	μ
HGSO-FOPID	2.6632	1.1314	0.4559	1.2689	1.3663
SSA-FOPID [20]	1.9982	1.1706	0.5750	1.1395	1.1656
SFS-PID [15]	1.2837	1.3392	0.7780	1	1
IKA-PID [12]	1.0426	1.0093	0.5999	1	1

Bu sayısal sonuçlar göz önüne alındığında, AVR sisteminin FOPID denetleyici tasarımı için önerilen HGSO algoritmasının, daha iyi geçici kararlılık, hızlı sönümlenme özellikleri ve minimum aşma ile SSA, SFS ve IKA gibi diğer algoritmalarla tasarlanan denetleyicilerden daha üstün olduğu sonucuna varılabilir.



Şekil 9.4 AVR'nin Adım Yanıtlarının Farklı Denetleyicilerle Karşılaştırılması

Tablo 9.2 Geçici Yanıtların Karşılaştırmalı Sonuçları

Ayarlama Yöntemi	Dinamik Yanıt Karşılaştırması			
	Tepe değeri (pu)	M_p %	T_r (s)	T_s (s)
HGSO-FOPID (önerilen)	1.0286	2.8626	0.0892	0.4257
SSA-FOPID [20]	1.1549	15.4900	0.0981	0.5510
SFS-PID [15]	1.2278	22.7814	0.1039	0.9532
IKA-PID [12]	1.1500	15.0009	0.1283	0.7531

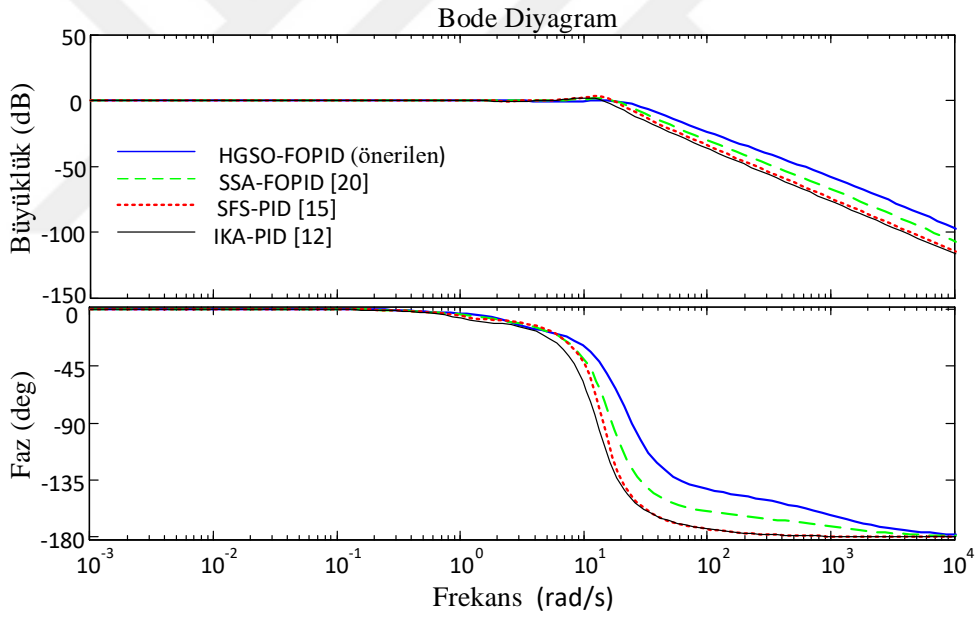
$$T_{SSA-FOPID}(s) = \frac{0.0575s^{3.3051} + 5.75s^{2.3051} + 0.19982s^{2.1395} + 19.982s^{1.1395} + 0.11706s + 11.706}{0.0004s^{5.1395} + 0.0454s^{4.1395} + 0.555s^{3.1395} + 5.75s^{2.3051} + 1.51s^{2.1395} + 20.982s^{1.1395} + 11.706} \quad (9.3)$$

$$T_{SFS-PID}(s) = \frac{0.0778s^3 + 7.908s^2 + 12.97s + 13.39}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 9.29s^2 + 13.84s + 13.39} \quad (9.4)$$

$$T_{IKA-PID}(s) = \frac{0.05999s^3 + 6.103s^2 + 10.53s + 10.09}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + 7.509s^2 + 11.43s + 10.09} \quad (9.5)$$

9.4.Frekans Tepki Analizi

AVR sisteminin farklı denetleyici tasarımlarına sahip karşılaştırmalı Bode diyagramları şekil 9.5'te gösterilmektedir. Ayrıca, tepe kazancı, faz ve gecikme marjları ve bant genişliği gibi karşılaştırmalı frekans yanıt performans kriterleri sonuçları Tablo 9.3'te sunulmaktadır. Bu tablodan, frekans yanıt kriterleri açısından en kararlı sistemin önerilen HGSO-FOPID kontrolörü olduğu sonucuna varılabilir.



Şekil 9.5 AVR'nin Bode Grafiklerinin Farklı Kontrolör Tasarımları İle Karşılaştırılması

Tablo 9.3 Farklı Kontrolörler İle AVR Vücut Analizi

Ayarlama Yöntemi	Frekans yanıtı karşılaştırması			
	Tepe kazanç	Faz marjı	Gecikme marjı	Bant genişliği
HGSO-FOPID (önerilen)	0.08 dB	168.8°	1.0920 s	25.3633 Hz
SSA-FOPID [20]	1.33 dB	88.8°	0.0906 s	21.1798 Hz
SFS-PID [15]	3.11 dB	62.4°	0.0634 s	19.8210 Hz
IKA-PID [12]	1.78 dB	76.7°	0.0954 s	16.7845 Hz

10. DC MOTOR HIZ KONTROL SİSTEMİ SİMÜLASYON SONUÇLARI

OBL / HGSO algoritmasının programlama kodları önceki sistemlerde kullanılan (8. bölümdeki pitch kontrol sistemi) bilgisayar üzerinden çalıştırılmıştır. Geçici tepki, frekans tepkisi, sağlamlık ve yük bozucu reddetme analizlerinin simülasyonları da belirtilen ortam kullanılarak gerçekleştirildi. Optimal kazanç seti kullanılarak elde edilen çeşitli analizlerin simülasyon sonuçları, aşağıdaki alt bölümlerde açıklandığı üzere tasarımın doğrulanması için diğer yeni yaklaşımlarla karşılaştırılmıştır.

10.1. DC Motor Optimizasyonu

ITAE işlevi, referanslar [16,17,18,19] ile adil bir karşılaştırma için bu çalışmada benimsenmiştir. Burada önceki sistemlerde kullanılan amaç fonksiyonlarından farklı olarak ITAE amaç fonksiyonu kullanılmıştır bu fonksiyon denklem (10.1) 'te gösterildiği gibi verilmiştir. Önerilen OBL / HGSO algoritmasının parametreleri Tablo 10.1'de listelenmiştir ve DC motor için OBL / HGSO tabanlı PID (OBL / HGSO-PID) denetleyicisinin blok şeması Şekil 10.1'de gösterilmektedir.

$$ITAE = \int_0^{tsim} t \cdot |e(t)| dt \quad (10.1)$$

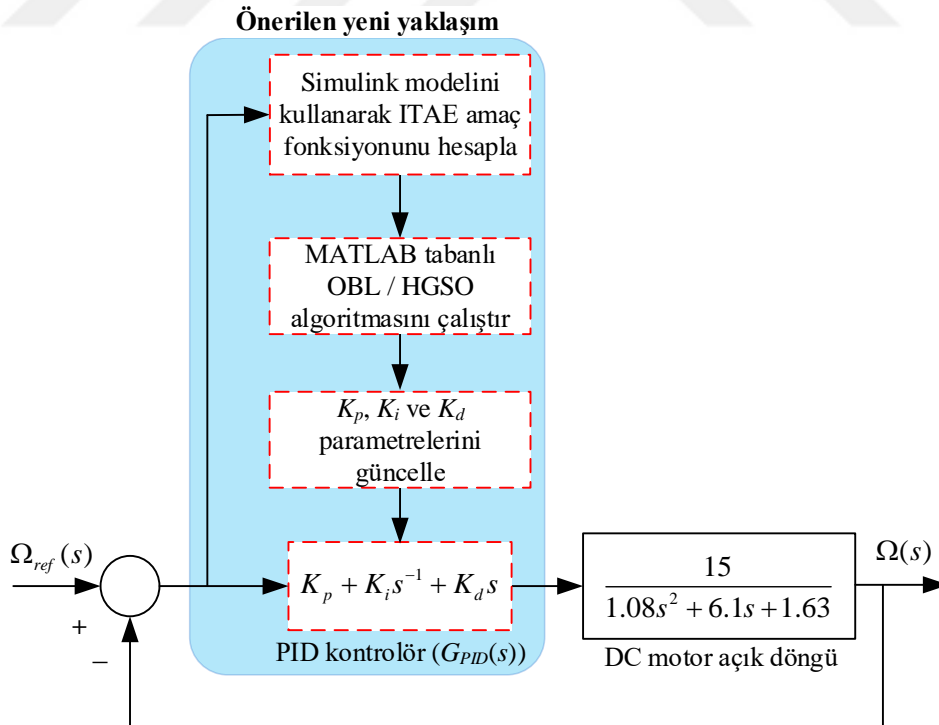
OBL / HGSO algoritması ile PID kontrolörün optimal kazanımlarını bulma prosedürü, DC motor hız kontrolü için geliştirilmiş bir MATLAB / Simulink modelinin OBL / HGSO ile entegre edildiği başlatma aşaması ile başlamıştır. Optimize edilmesi gereken PID kontrolör kazançları, popülasyondaki her bir gaz partikülünü temsil eden $P = (K_p; K_i; K_d)$ gibi bir gerçek sayı vektörüne atandı. Popülasyon, rastgele oluşturulan N sayıda gaz parçacığından ve bunlarla ilişkili karışıklıklardan oluşuyordu. Daha sonra, DC motorun hız kontrol sisteminin zaman etki alanı simülasyonu ile önerilen PID kontrolörü ve her bir gaz parçacığı için birlik geri beslemesi gerçekleştirilmiş ve ITAE değeri ile birlikte sistemin hız yanıt eğrileri elde edilmiştir.

Farklı parçacıklarla farklı hız çıktı eğrileri ve ITAE değerleri elde etmek tipiktir. Bu nedenle, HGSO algoritmasına dönmeden önce bir sonraki yineleme için güncellenmek üzere en iyi ITAE değerlerine sahip N sayıda çözüm (gaz parçacıkları) seçilmiştir. İkinci

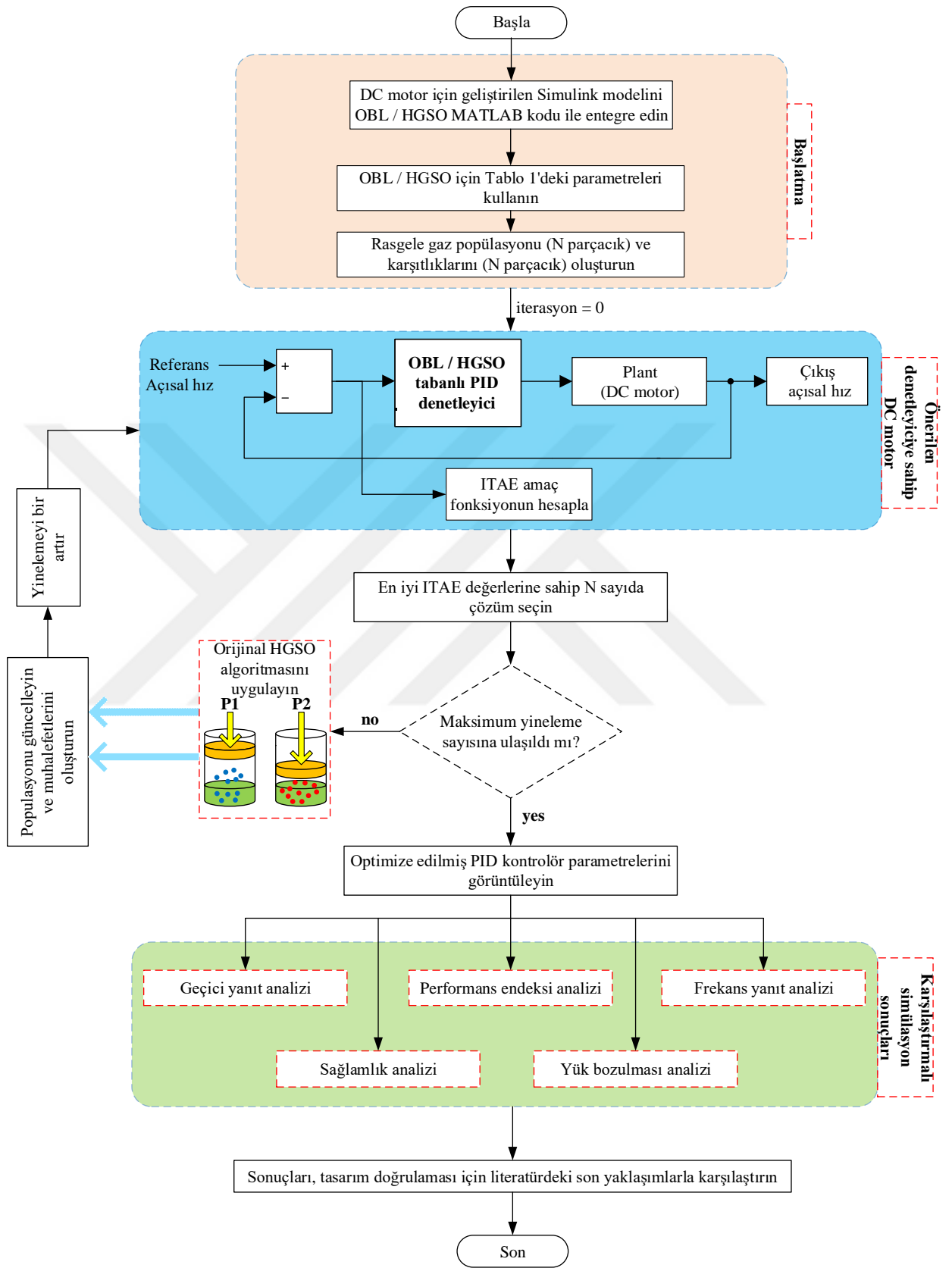
işlem sırasında, her bir gaz parçacığı güncellenirken karşıtlıklar üretildi. Bu süreç, kontrol sistemi ile OBL / HGSO algoritması arasında iki yönlü bir akıştı. Bu işlem, maksimum yinelemeye ulaşana kadar sürdürüldü. Son olarak, en düşük ITAE değerine sahip en iyi gaz parçacığı, optimizasyon sürecinin sonunda ayarlanan optimum PID denetleyici kazancı olarak gösterildi. Önerilen tasarım prosedürünün ayrıntılı akış şeması Şekil 10.2'de gösterilmektedir.

Tablo 10.1 Optimizasyon problemini çözmek için OBL / HGSO algoritmasının parametreleri

Parametre	Değer
Gaz parçacığı sayısı	40
Yineleme numarası	50
Bağımsız çalışma numarası	20
$[I_1; I_2; I_3]$	[0.05; 100; 0.01]
$[c_1; c_2; \beta; \alpha; K]$	[0.1; 0.2; 1; 1; 1]
$[K_p; K_i; K_d]$ için alt sınır	[0.001; 0.001; 0.001]
$[K_p; K_i; K_d]$ için üst sınır	[20; 20; 20]
Optimizasyon sorunu için boyut	3
Simülasyon süresi (t_{sim})	1 s



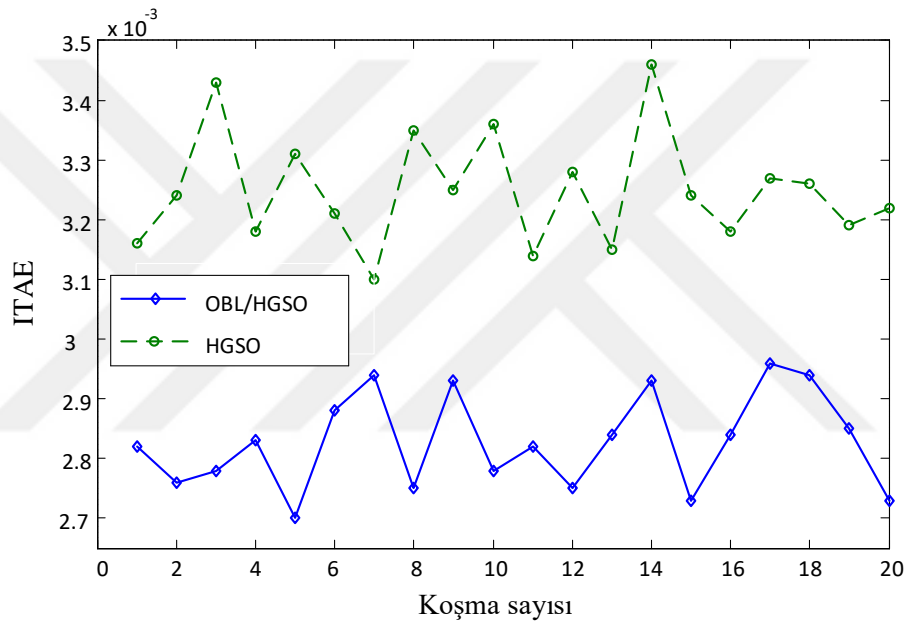
Şekil 10.1 DC Motor Hız Kontrolü İçin Önerilen OBL / HGSO-PID Yaklaşımı



Şekil 10.2 DC Motor Hız Kontrolü İçin Önerilen Tasarım Prosedürünün Akış Şeması

10.2. İstatistiksel Test Analizi

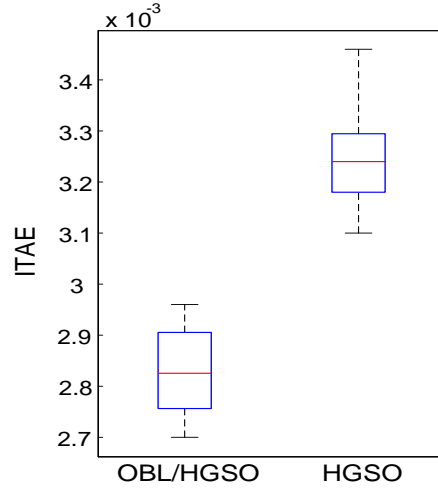
HGSO ve önerilen OBL /HGSO algoritmaları 20 kez bağımsız olarak çalıştırıldı. Tüm çalışmalar için elde edilen ITAE amaç fonksiyonunun değerleri şekil 10.3'te gösterilmiştir. En kötü, en iyi, ortalama, medyan ve standart sapma değerleri gibi objektif fonksiyonun istatistiksel değerleri Tablo 10.2'de ve kutu grafiği şekil 10.4'te verilmiştir. Hem şekillerden hem de tablodan, OBL / HGSO algoritmasının, OBL / HGSO algoritması ile elde edilen en kötü değer bile temel HGSO algoritması ile elde edilen en iyi değer çok altında olacağı şekilde oldukça iyi istatistiksel performansa sahip olduğu sonucuna varılabilir.



Şekil 10.3 Tüm Çalışmalardan Elde Edilen ITAE Değerleri

Tablo 10.2 ITAE Amaç Fonksiyonunun İstatistiksel Değerleri

İstatistik indeksi	HGSO	OBL/HGSO
En Kötü (Maksimum)	0.0035	0.0030
En İyi (Minimum)	0.0031	0.0027
Ortalama	0.0032	0.0028
Medyan	0.0032	0.0028
Standart sapma	9.5250E-05	8.0760E-05



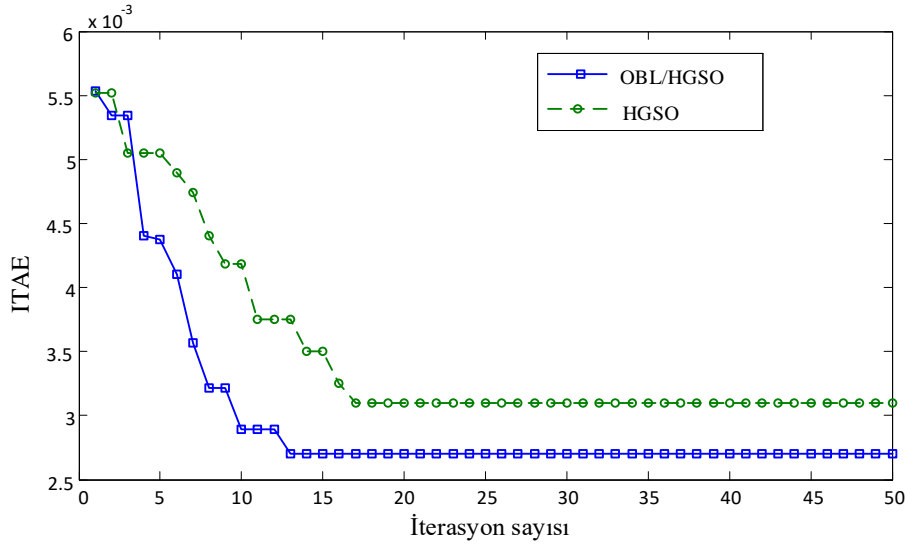
Şekil 10.4 HGSO ve OBL / HGSO İçin Kutu Grafiği

10.3. Yakınsama Profili

HGSO'nun en iyi çalışmasının ve önerilen OBL / HGSO algoritmalarının yakınsamaları Şekil 10.5'te gösterilmektedir. Bu şekilden, OBL / HGSO algoritmasının yüksek yakınsama hızıyla en düşük ITAE değerini sağlayabileceği sonucuna varılabilir. Bunun için en iyi çalışma için en uygun değerleri bulmak yalnızca 13 yineleme aldı. Optimizasyon işleminden sonra bulunan HGSO-PID ve OBL / HGSO-PID kontrolörlerinin parametreleri Tablo10.3'te verilmiştir. Bu kazançlar kullanılarak elde edilen transfer fonksiyonları Denklemlerde verilmiştir. (10.2) - (10.3).

$$T_{OBL/HGSO-PID}(s) = \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{42.77s^2 + 254s + 14.26}{1.08s^3 + 48.87s^2 + 255.6s + 14.26} \quad (10.2)$$

$$T_{HGSO-PID}(s) = \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{34.06s^2 + 201.6s + 18.09}{1.08s^3 + 40.16s^2 + 203.3s + 18.09} \quad (10.3)$$



Şekil 10.5 Karşılaştırmalı Yakınsama Eğrileri

10.4. Aşma, Yükselme Süresi ve Oturma Süresinin Karşılaştırılması

Karşılaştırma amacıyla seçilen diğer kontrolörler için kazanç parametreleri Tablo 10.3'te listelenmiştir. Bu kontrolörlerin transfer fonksiyonları Denklemlerde verilmiştir. (10.4) - (10.7). Farklı yaklaşımlarla tasarlanan hız kontrol sistemi için karşılaştırmalı adım yanıt sonuçları ve geçici yanıt analizi sonuçları, Şekil 10.6'nın adım yanıt karşılaştırmalarını sunduğu şekil 10.6 – 10.9'da sunulmaktadır; Şekil 10.7, 10.8 ve 10.9 maksimum aşma yüzdesinin çubuk grafiği karşılaştırmalarını göstermektedir sırasıyla; yükselme süresi (% 10 → % 90 için) ve oturma süresi ($\pm\% 2$ tolerans için).

$$T_{ASO-PID}(s) = \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{36.54s^2 + 179.2s + 30.78}{1.08s^3 + 42.64s^2 + 180.8s + 30.78} \quad (10.4)$$

$$T_{SFS-PID}(s) = \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{3.592s^2 + 24.47s + 4.197}{1.08s^3 + 9.693s^2 + 26.1s + 4.197} \quad (10.5)$$

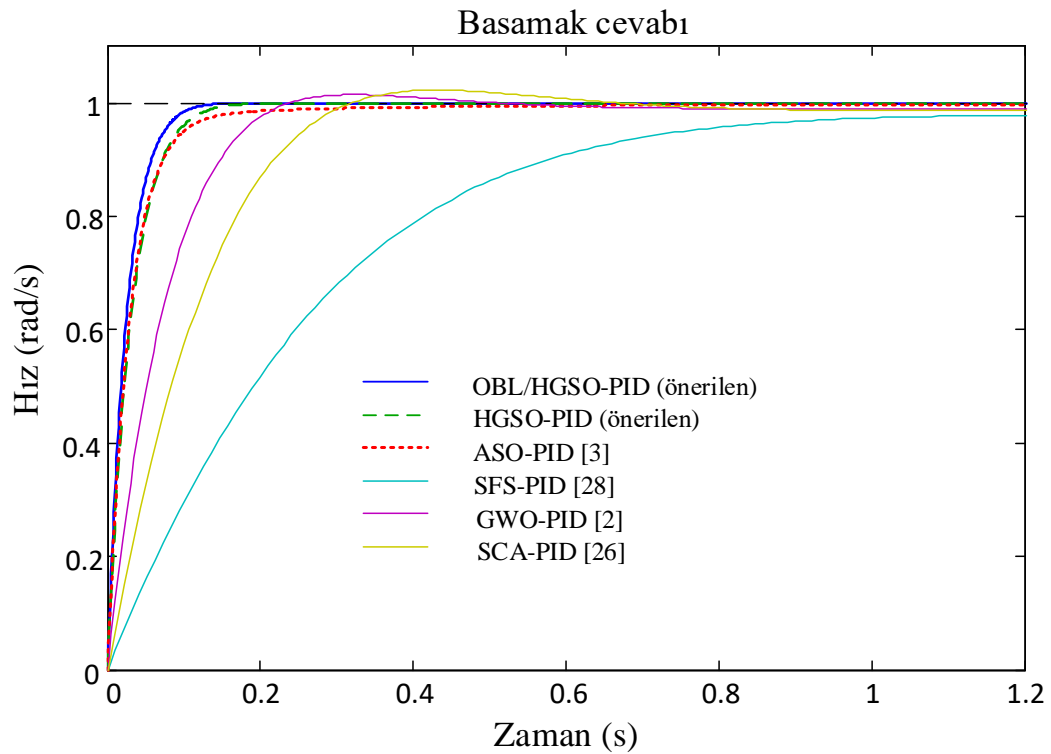
$$T_{GWO-PID}(s) = \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{13.94s^2 + 103.5s + 8.439}{1.08s^3 + 20.04s^2 + 105.1s + 8.439} \quad (10.6)$$

$$T_{SCA-PID}(s) = \frac{\Omega(s)}{\Omega_{ref}(s)} = \frac{7.953s^2 + 67.52s + 7.89}{1.08s^3 + 14.05s^2 + 69.15s + 7.89} \quad (10.7)$$

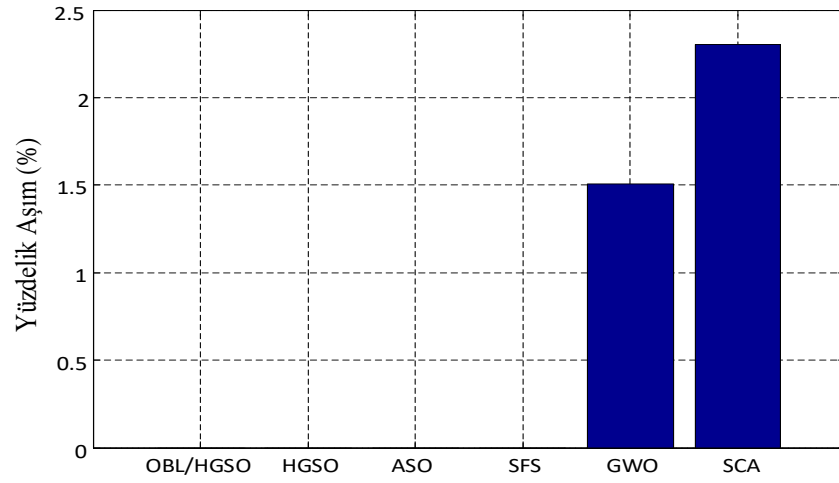
Bu şekillerden görüldüğü gibi, OBL / HGSO-PID kontrol cihazı, HGSO-PID kontrolörü dahil diğerlerinden açıkça daha iyi bir zaman cevabına sahiptir. Bu nedenle, önerilen kontrolör tasarım yaklaşımı sadece HGSO tabanlı tasarım yaklaşımından değil, aynı zamanda ASO [16], SFS [17], GWO [18] ve SCA [19] gibi daha iyi geçici kararlılık ile diğer kontrolör tasarım yaklaşımlarından daha üstündür.

Tablo 10.3 Önerilen Kontrolörlerin ve Karşılaştırılan Diğer Kontrolörlerin Kazanç Parametreleri

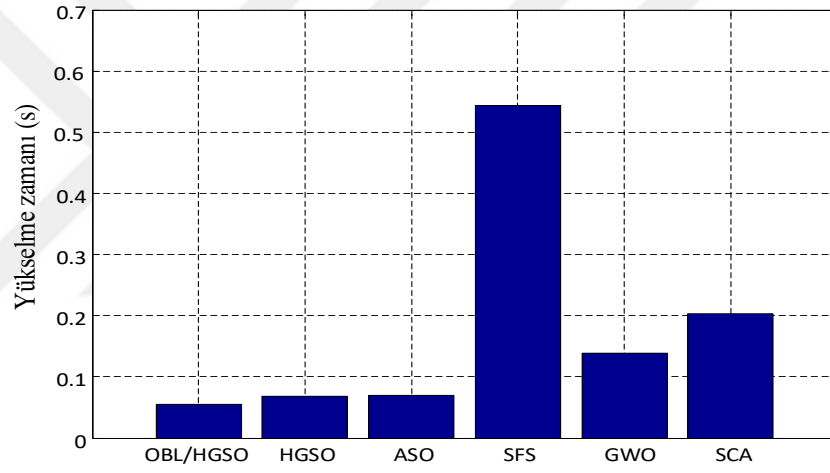
Kontrolör Tipi	K_p	K_i	K_d
OBL/HGSO-PID (önerilen)	16.9327	0.9508	2.8512
HGSO-PID (önerilen)	13.4430	1.2059	2.2707
ASO-PID [16]	11.9437	2.0521	2.4358
SFS-PID [17]	1.6315	0.2798	0.2395
GWO-PID [18]	6.8984	0.5626	0.9293
SCA-PID [19]	4.5012	0.5260	0.5302



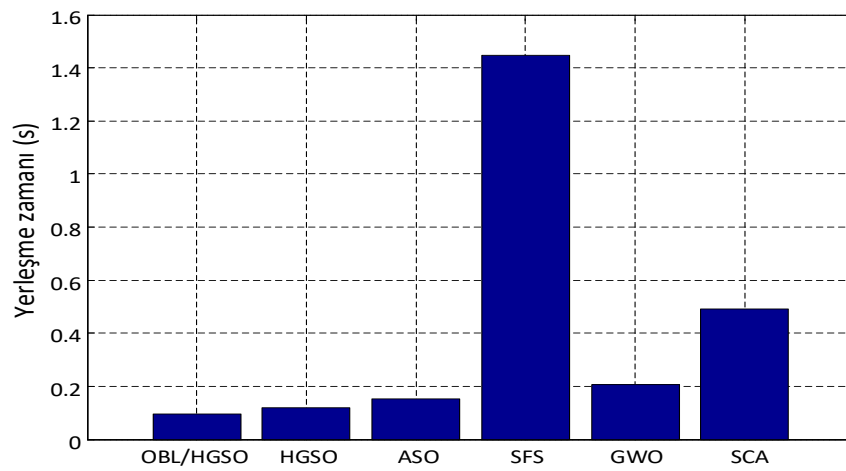
Şekil 10.6 Mevcut Yaklaşımlarla Hız Karşılaştırması



Şekil 10.7 Farklı Yaklaşımlar İçin Yüzde Aşımaları



Şekil 10.8 Farklı Yaklaşımlar İçin Yükselme Süreleri



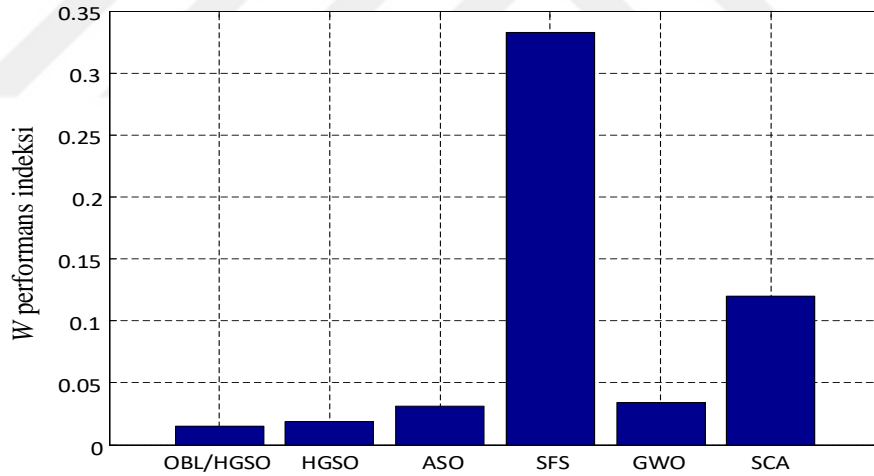
Şekil 10.9 Farklı Yaklaşımlar İçin Durma Süreleri

10.5. Performans Endeksinin Karşılaştırılması

Yaygın kullanımı nedeniyle, performans endeksi karşılaştırması için W amaç fonksiyonu da benimsenmiştir. W 'nin formülü E_{ss} , M_p , T_r ve T_s 'nin sırasıyla kararlı durum hatası, maksimum aşma, yükselme ve yerleşme sürelerini temsil ettiği denklem (10.8) 'de verilmiştir [83]. Ağırlıklandırma parametresi ρ genellikle 1.0 olarak seçilir [84]. Bir sistemin performansı, geçici yanıt açısından en düşük W değeri için maksimumdur.

$$W(K_p, K_i, K_d) = (1 - e^{-\rho})(M_p + E_{ss}) + e^{-\rho}(T_s - T_r) \quad (10.8)$$

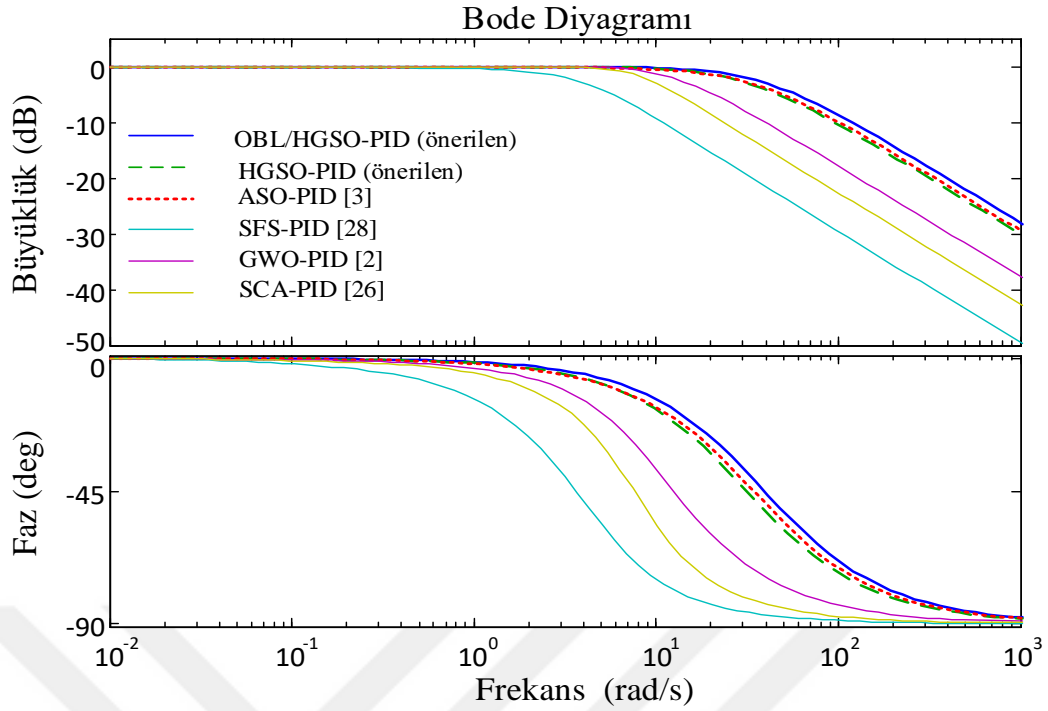
Farklı yaklaşımlardan elde edilen W performans indeksi değerlerinin karşılaştırmalı çubuk grafikleri şekil 10.10'da verilmiştir. Şekilde görüldüğü gibi önerilen OBL / HGSO-PID kontrolör en düşük W değerine sahiptir. Bu sonuç, OBL / HGSO'nun orijinal HGSO dahil diğer yaklaşımlara göre üstünlüğünü göstermektedir.



Şekil 10.10 Farklı Yaklaşımlar İçin W Performans İndeksi Değerleri

10.6. Frekans Alanı Analizlerinin Karşılaştırılması

Farklı kontrolör tasarımlarına sahip DC motor hız kontrol sisteminin karşılaştırmalı Bode grafikleri şekil 10.11'de gösterilmektedir ve kazanç marjı (desibel olarak), faz marjı (derece olarak) ve bant genişliği (Hertz cinsinden) gibi karşılaştırmalı frekans yanıtı performans analizi sonuçları Tablo 10.4'te sunulmuştur. Tablo, frekans yanıt kriterleri açısından en kararlı sistemin önerilen OBL / HGSO-PID denetleyicisi olduğunu açıkça göstermektedir.



Şekil 10.11 Farklı Kontrolör Tasarımları İçin Karşılaştırmalı Bode Grafikleri

Tablo 10.4 Karşılaştırmalı Frekans Tepkisi Performans Analizi Sonuçları

Kontrolör tipi	Kazanç marjı (dB)	Faz marjı (deg.)	Bant genişliği (Hz)
OBL/HGSO-PID (önerilen)	∞	180°	39.8561
HGSO-PID (önerilen)	∞	180°	31.7975
ASO-PID [16]	∞	180°	32.9113
SFS-PID [17]	∞	180°	4.1183
GWO-PID [18]	∞	180°	14.9018
SCA-PID [19]	∞	180°	10.1347

10.7. Gürbüzlük Analizinin Karşılaştırılması

Sistem yanıtını kabul edilebilir aralıklarda tutmak için sağlam bir kontrolör istenir. Bu nedenle, belirsizlik durumunda önerilen sistemin ne kadar kararlı olduğunu belirlemek için sağlamlık analizi yapılmıştır. Bunu yapmak için, DC motorun elektrik direnci (R_a) ve tork sabiti (K_m) sırasıyla $\pm\% 25$ ve $\pm\% 20$ ayrı ayrı değiştirilerek sistem davranışı gözlemlendi. Böylece, bu eylem dört olası işletim senaryosu yarattı. Bu senaryolar tablo 10.5'te sunulmuştur.

Tablo 10.5 Değişen DC Motor Parametreleriyle Dört Olası Çalışma Noktası Senaryosu

Motor parametresi	Senaryo I	Senaryo II	Senaryo III	Senaryo IV
R_a	0.30	0.30	0.50	0.50
K_m	0.012	0.018	0.012	0.018

Tüm senaryolar için geçici tepki analizinin karşılaştırmalı simülasyon sonuçları tablo 10.6 - tablo 10.9'da gösterilmektedir. Benzer şekilde, karşılaştırmalı hız adımı yanıt grafikleri şekil 10.12 – 10.15'te gösterilmektedir. DC motor parametrelerinde meydana gelen değişikliklere rağmen, önerilen OBL / HGSO -PID denetleyicisi, durum I ve II hariç, aşma olmaksızın hem çökme hem de yükselme süreleri için en düşük değerlere sahipti. Son iki durum, diğer kontrolörlerle karşılaştırıldığında sadece önemsiz bir aşma gösterdi. Önerilen OBL / HGSO-PID kontrol cihazının sağlamlığı, bir DC motorun hızını düzenleme açısından bu sonuçlarla doğrulanmıştır.

Tablo 10.6 Senaryo I İçin Geçici Yanıt Sonuçları

Kontrolör tipi	M_p (%)	T_r (s)	T_s (s)
OBL/HGSO-PID (önerilen)	0.0560	0.0678	0.1163
HGSO-PID (önerilen)	0.0127	0.0849	0.1455
ASO-PID [16]	0.0000	0.0872	0.1936
SFS-PID [17]	0.0000	0.6569	1.2031
GWO-PID [18]	1.5195	0.1683	0.2471
SCA-PID [19]	2.1514	0.2447	0.5618

Tablo 10.7 Senaryo II İçin Geçici Yanıt Sonuçları

Kontrolör tipi	M_p (%)	T_r (s)	T_s (s)
OBL/HGSO-PID (önerilen)	0.0959	0.0455	0.0785
HGSO-PID (önerilen)	0.0659	0.0570	0.0982
ASO-PID [16]	0.0000	0.0569	0.1198
SFS-PID [17]	0.0000	0.4452	0.7986

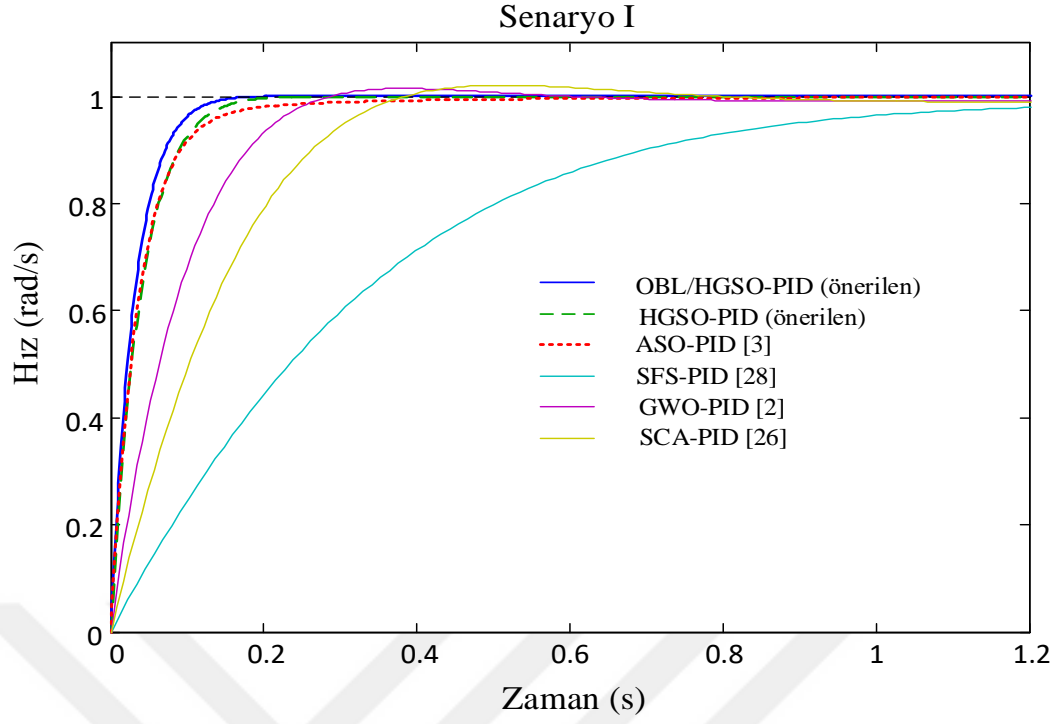
GWO-PID [18]	1.8415	0.1173	0.1731
SCA-PID [19]	2.9696	0.1733	0.4787

Tablo 10.8 Senaryo III İçin Geçici Yanıt Sonuçları

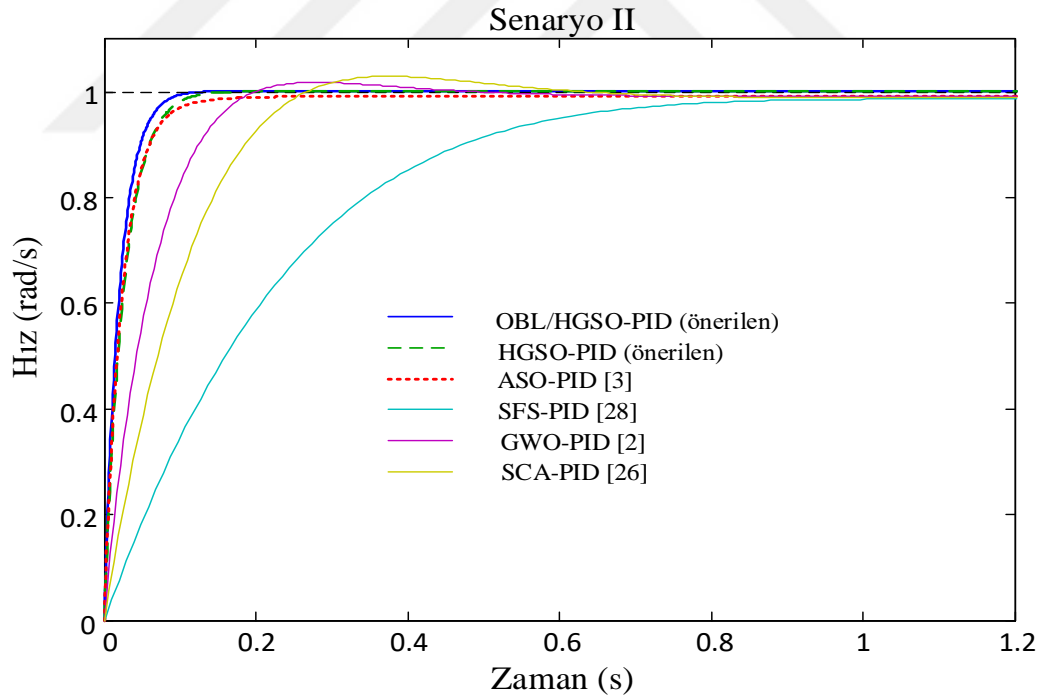
Kontrolör tipi	M_p (%)	T_r (s)	T_s (s)
OBL/HGSO-PID (önerilen)	0.0000	0.0682	0.1191
HGSO-PID (önerilen)	0.0000	0.0856	0.1499
ASO-PID [16]	0.0000	0.0881	0.2107
SFS-PID [17]	0.0000	0.7027	3.7214
GWO-PID [18]	0.9547	0.1706	0.2547
SCA-PID [19]	1.3036	0.2492	0.3573

Tablo 10.9 Senaryo IV İçin Geçici Yanıt Sonuçları

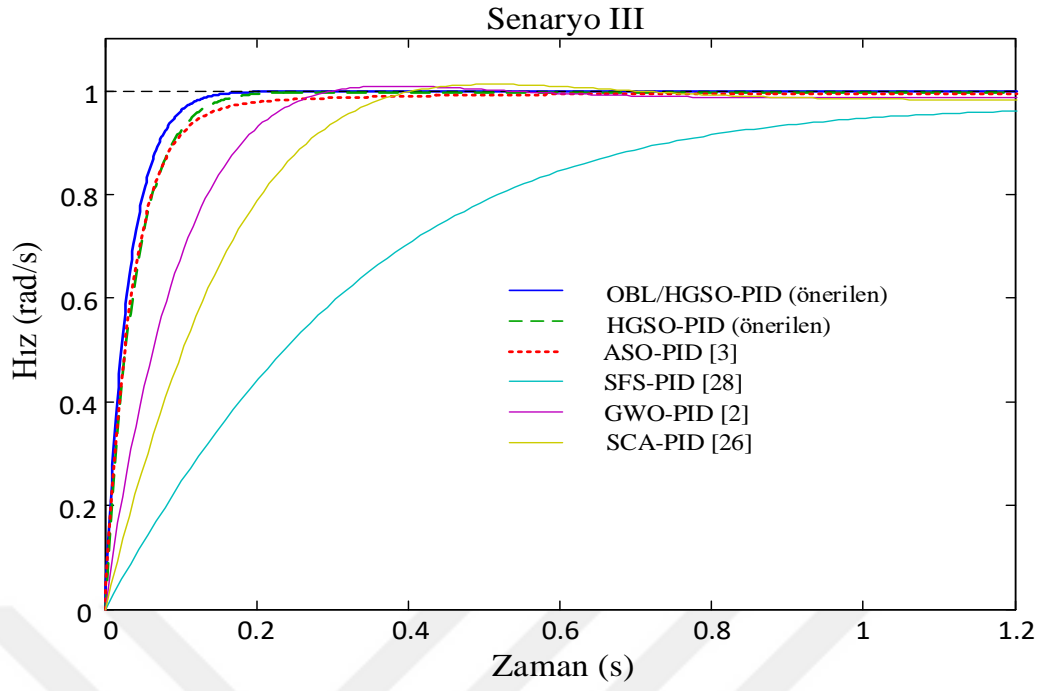
Kontrolör tipi	M_p (%)	T_r (s)	T_s (s)
OBL/HGSO-PID (önerilen)	0.0000	0.0457	0.0798
HGSO-PID (önerilen)	0.0000	0.0573	0.1002
ASO-PID [16]	0.0000	0.0573	0.1257
SFS-PID [17]	0.0000	0.4649	3.3544
GWO-PID [18]	1.4479	0.1185	0.1767
SCA-PID [19]	2.3664	0.1755	0.4326



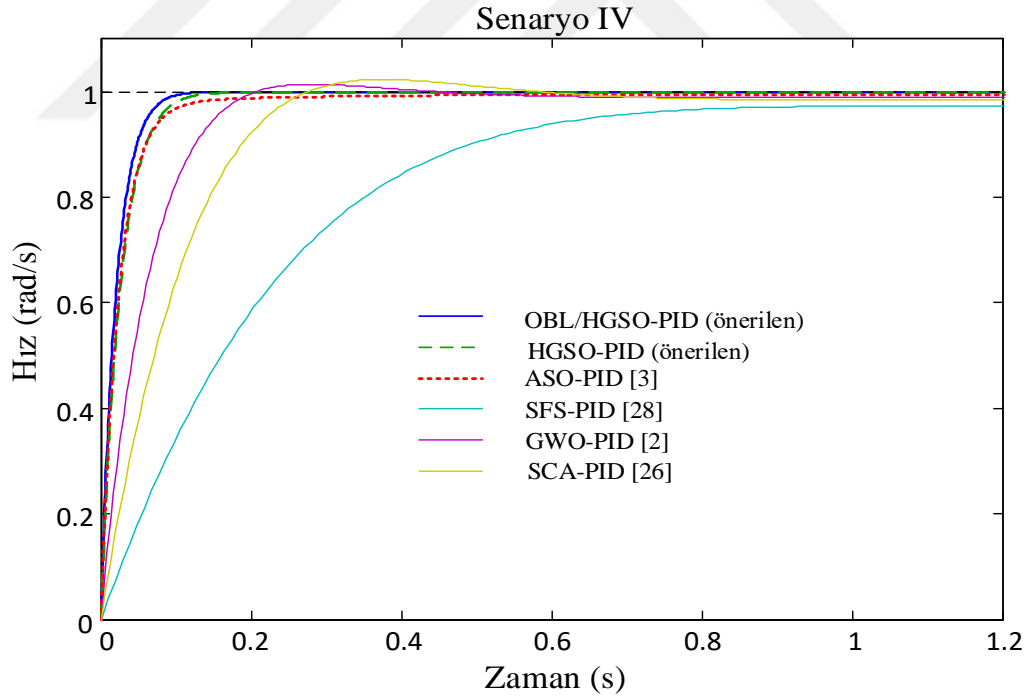
Şekil 10.12 Senaryo I için DC Motorun Hız Adımı Yanıtları



Şekil 10.13 Senaryo II için DC Motorun Hız Adımı Yanıtları



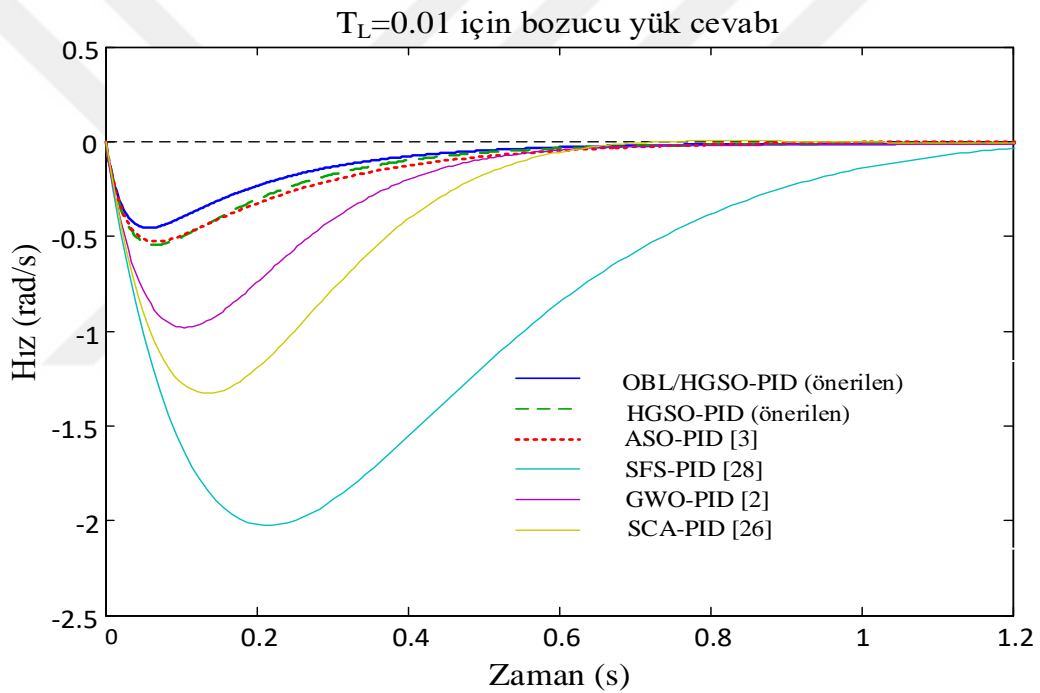
Şekil 10.14 Senaryo III İçin DC Motorun Hız Adımı Yanıtları



Şekil 10.15 Senaryo IV İçin DC Motorun Hız Adımı Yanıtları

10.8. Yük Bozulma Analizinin Karşılaştırılması

OBL/ HGSO-PID kontrolörünün hız kontrol sisteminde meydana gelen beklenmedik bozulma etkilerine yanıt verme yeteneği bu alt bölümde tartışılmaktadır. Bunu yapmak için, $t = 0$ s'de tork yükü ($T_L = 0.01$) için rahatsız edici bir sinyal oluşturuldu. Sistem performansı üzerindeki etkilerini en aza indirmek için, rahatsızlığın tasarlanmış kontrolör tarafından hızla bastırılması gerekir. Bozucu yük torku etkisine bağlı olarak DC motorun hız tepkisi şekil 10.16'da gösterilmektedir. Açıkça görüldüğü gibi, OBL / HGSO-PID kontrolör denetleyicisi HGSO-PID, ASO-PID [16], SFS-PID [17], GWO-PID [18], SCA-PID [19] kontrolörleri ile karşılaştırıldığında, minimum alt başlama ve çökme süresiyle rahatsız edici yük torku etkisine daha hızlı ve daha iyi bastırma sağlar.



Şekil 10.16 DC Motorun Yük Bozucu Yanıtları

10.9. Kontrolörlerde Enerji Ve Maksimum Kontrol Sinyalinin Karşılaştırılması

MATLAB / Simulink uygulamasında benimsenen kontrolör için ideal olmayan bir transfer fonksiyonu gerektirdiğinden, önerilen OBL / HGSO-PID kontrolörü birinci dereceden bir filtre ile uygulandı ve kontrolör çabalarını elde etmek için Simulink'te test edildi. Birinci dereceden bir filtre kullanan PID kontrolörünün transfer fonksiyonu aşağıdaki gibi verilmiştir.

$$G_{PID,filter}(s) = K_p + \frac{K_i}{s} + \frac{sK_d}{T_f s + 1} \quad (10.9)$$

Burada T_f birinci dereceden filtrenin zaman sabitidir ve 0,001'e ayarlanır. Kontrolör çabaları açısından çeşitli yaklaşımlarla ayarlanan kontrolör çıkışlarının karşılaştırmalı performansı Tablo 10.10'da verilmiştir. En yüksek kontrol eforuna sahip kontrolörün önerilen OBL / HGSO-PID kontrolör olduğu görülebilmektedir.

Tablo 10.10 Kontrolörlerin Enerji Ve Maksimum Kontrol Sinyali

Kontrolör tipi	Enerji	U_{max}
OBL/HGSO-PID (önerilen)	4277.5070	2868.1327
HGSO-PID	2714.9616	2284.1430
ASO-PID [16]	3113.8590	2447.7437
SFS-PID [17]	30.6680	241.1315
GWO-PID [18]	458.3019	936.1984
SCA-PID [19]	150.2166	534.7012

11. GENEL SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında sezgisel-üstü algoritma olan HGSO algoritması ve karşıt tabanlı öğrenme yani OBL/HGSO algoritmaları ile PID ve FOPID kontrolörler kullanılarak farklı sistemler üzerindeki etkileri incelenmiştir. Üç ayrı sistem olan uçak eğim kontrol sistemi, otomatik gerilim regülatörü ve DC motor hız kontrol sistemi gibi sistemler seçilerek bu sistemlerin matematiksel modelleri üzerinden HGSO algoritması uygulanmıştır. MATLAB programı kullanılarak farklı sistemlerdeki algoritmaların etkileri simüle edilmiş ve sonuçlar karşılaştırılmak üzere teker teker incelenmiştir. Simülasyon sonucunda özellikle sistemlerin basamak cevabı, Bode grafikleri, kutup-sıfır haritaları ve algoritmalara ait kutu grafikleri çıkarılmış ve gerekli yerlerde tablolara dökülmüştür.

HGSO ve OBL/HGSO algoritmaları her üç sistem için de daha önce denenmiş sezgisel-üstü algoritmalar ile karşılaştırılarak HGSO algoritmasının üstünlüğü; yerleşme zamanı, yükselme zamanı, maksimum aşım ve sürekli hal hatası gibi parametreler grafik üzerinde gösterilerek daha etkin olduğu gösterilmiştir.

KAYNAKLAR

- [1] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019, doi: 10.1016/j.future.2019.07.015.
- [2] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "A modified Henry gas solubility optimization for solving motif discovery problem," *Neural Comput. Appl.*, 2019, doi: 10.1007/s00521-019-04611-0.
- [3] B. S. Yıldız, A. R. Yıldız, N. Pholdee, S. Bureerat, S. M. Sait, and V. Patel, "The Henry gas solubility optimization algorithm for optimum structural design of automobile brake components," *Mater. Test.*, vol. 62, no. 3, pp. 261–264, Mar. 2020, doi: 10.3139/120.111479.
- [4] N. Neggaz, E. H. Houssein, and K. Hussain, "An efficient Henry gas solubility optimization for feature selection," *Expert Syst. Appl.*, vol. 152, p. 113364, 2020, doi: 10.1016/j.eswa.2020.113364.
- [5] Y. Işık and H. Korul, "Comparison of classical PD and fuzzy PD controller performances of an aircraft pitch angle control system," *Gazi University Journal of Science*, vol. 24, no. 4. Gazi University, pp. 781–789, 2011.
- [6] C. S. Mohanty, P. S. Khuntia, and D. Mitra, "Design of stable nonlinear pitch control system for a jet aircraft by using artificial intelligence," *Proc. Natl. Acad. Sci. India Sect. A - Phys. Sci.*, vol. 89, no. 1, pp. 57–66, 2019, doi: 10.1007/s40010-017-0396-z.
- [7] G. Altıntaş and Y. Aydin, "Comparison of fractional and integer order PID controllers on aircraft model using genetic algorithm," in *2016 National Conference on Electrical, Electronics and Biomedical Engineering (ELECO)*, 2016, pp. 242–246.
- [8] A. Chowdhury and V. G. Nair, "Optimization of PID controller gains of an aircraft pitch control system using particle swarm optimization algorithm," *Int. J. Mech. Prod. Eng. Res. Dev.*, vol. 7, no. 6, pp. 223–229, Dec. 2017, doi: 10.24247/ijmperdddec201724.
- [9] R. Zaeri, A. Ghanbarzadeh, B. Attaran, and Z. Zaeri, "Fuzzy logic controller based pitch control of aircraft tuned with bees algorithm," in *Proceedings - 2011 2nd International Conference on Control, Instrumentation and Automation, ICCIA 2011*, 2011, pp. 705–710, doi: 10.1109/ICCIAutom.2011.6356745.
- [10] P. Kumar and S. Narayan, "Multi-objective bat algorithm tuned optimal FOPID controller for robust aircraft pitch control," *Int. J. Syst. Control Commun.*, vol. 8, no. 4, pp. 348–362, Jan. 2017, doi: 10.1504/IJSCC.2017.087127.
- [11] H. Saadat, *Power system analysis*. New Delhi: Tata McGraw-Hill, 2002.
- [12] M. S. Ayas, "Design of an optimized fractional high-order differential feedback controller for an AVR system," *Electr. Eng.*, vol. 101, no. 4, pp. 1221–1233, Dec. 2019.
- [13] Y. Tang, M. Cui, C. Hua, L. Li, and Y. Yang, "Optimum design of fractional order $PI\lambda D\mu$ controller for AVR system using chaotic ant swarm," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6887–6896, 2012.
- [14] S. Ekinçi, B. Hekimoglu, A. Demiroren, and S. Kaya, "Harris hawks optimization approach for tuning of FOPID controller in DC-DC buck converter," in *Proc. IEEE IDAP*, 2019, pp. 1–9.
- [15] B. Hekimoglu, "Robust fractional order PID stabilizer design for multi-machine power system using grasshopper optimization algorithm," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 35, no. 1, pp. 165–180, 2020.
- [16] B. Hekimoğlu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, 7 (2019) 38100–38114. <https://doi.org/10.1109/ACCESS.2019.2905961>.
- [17] R. Bhatt, G. Parmar, R. Gupta, A. Sikander, "Application of stochastic fractal search in approximation and control of LTI systems," *Microsyst. Technol.* 25 (2019) 105–114. <https://doi.org/10.1007/s00542-018-3939-6>.
- [18] J. Agarwal, G. Parmar, R. Gupta, A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsyst. Technol.* 24 (2018) 4997–5006. <https://doi.org/10.1007/s00542-018-3920-4>.
- [19] J. Agarwal, G. Parmar, R. Gupta, "Application of sine cosine algorithm in optimal control of DC motor and robustness analysis," *Wulfenia J.* 24 (2017).
- [20] E. brahimi, A., Khamehchi, E. (2016). Sperm whale algorithm: an effective metaheuristic algorithm for production optimization problems. *Journal of Natural Gas Science and Engineering*, 29, 211–222.
- [21] Holland, J.H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, U Michigan Press.
- [22] Mehrabian, A.R., Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization.

- Ecological Informatics 1(4):355-366.
- [23] Yan, G.W. (2013) A novel optimization algorithm based on atmosphere clouds model. *International Journal of Computational Intelligence and Applications*, 12(1)
- [24] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization in neural networks, *IEEE International Conference*, 4, 1942–1948.
- [25] Dorigo, M. (1992). Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy.
- [26] Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3), 459-471.
- [27] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer, *Adv Eng Softw*, 69, 46-61.
- [28] Mirjalili, S., Mirjalili, S.M. (2016). The whale optimization algorithm, *Adv Eng Softw*, 95, 51-67.
- [29] Salem, S.A. (2012). BOA: A novel optimization algorithm. In *International Conference on Engineering and Technology (ICET)*, 1-5, Egypt.
- [30] K. Ogata, *Modern Control Engineering*, 4th Ed. Prentice-Hall, 2002.
- [31] I. A. Khan, A. S. Alghamdi, T. A. Jumani, A. Alamgir, A. B. Awan, and A. Khidrani, “Salp swarm optimization algorithm-based fractional order PID controller for dynamic response and stability enhancement of an automatic voltage regulator system,” *Electronics*, vol. 8, no. 12, 2019.
- [32] S. Ekinci, “Optimal design of power system stabilizer using sine cosine algorithm,” *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 34, no. 3. Gazi Üniversitesi, pp. 1329–1350, 2019, doi: 10.17341/gazimmfd.460529.
- [33] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimisation algorithm: Theory and application,” *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017, doi: 10.1016/j.advengsoft.2017.01.004.
- [34] A.K. Mishra, V.K. Tiwari, R. Kumar, T. Verma, Speed control of dc motor using artificial bee colony optimization technique, in: *CARE 2013 - 2013 IEEE Int. Conf. Control. Autom. Robot. Embed. Syst. Proc.*, 2013: pp. 1–6. <https://doi.org/10.1109/CARE.2013.6733772>.
- [35] A. Rodríguez-Molina, M.G. Villarreal-Cervantes, M. Aldape-Pérez, An adaptive control study for a DC motor using meta-heuristic algorithms, *IFAC-PapersOnLine*. 50 (2017) 13114–13120. <https://doi.org/10.1016/j.ifacol.2017.08.2164>.
- [36] X. Lin, Y. Liu, Y. Wang, Design and research of DC motor speed control system based on improved BAS, in: *Proc. 2018 Chinese Autom. Congr. CAC 2018*, 2018: pp. 3701–3705. <https://doi.org/10.1109/CAC.2018.8623171>.
- [37] L. Syafaah, Widiyanto, I. Pakaya, D. Suhardi, M. Irfan, PID designs using DE and PSO algorithms for damping oscillations in a DC motor speed, in: *2017 4th Int. Conf. Electr. Eng. Comput. Sci. Informatics*, 2017: pp. 1–5. <https://doi.org/10.1109/eecsi.2017.8239138>.
- [38] A. Rodríguez-Molina, M.G. Villarreal-Cervantes, J. Álvarez-Gallegos, M. Aldape-Pérez, Bio-inspired adaptive control strategy for the highly efficient speed regulation of the DC motor under parametric uncertainty, *Appl. Soft Comput. J.* 75 (2019) 29–45. <https://doi.org/10.1016/j.asoc.2018.11.002>.
- [39] D. Potnuru, K. Alice Mary, C. Sai Babu, Experimental implementation of Flower Pollination Algorithm for speed controller of a BLDC motor, *Ain Shams Eng. J.* 10 (2019) 287–295. <https://doi.org/10.1016/j.asej.2018.07.005>.
- [40] A.T. El-Deen, A.A. Hakim Mahmoud, A.R. El-Sawi, Optimal PID tuning for DC motor speed controller based on genetic algorithm, *Int. Rev. Autom. Control.* 8 (2015) 80–85. <https://doi.org/10.15866/ireaco.v8i1.4839>.
- [41] S. Duman, D. Maden, U. Güvenç, Determination of the PID controller parameters for speed and position control of DC motor using gravitational search algorithm, in: *ELECO 2011 - 7th Int. Conf. Electr. Electron. Eng.*, 2011: pp. I-225-I-229.
- [42] A. Madadi, M.M. Motlagh, Optimal Control of DC motor using Grey Wolf Optimizer Algorithm, *Tech. J. Eng. Appl. Sci.* 4 (2014) 373–379.
- [43] M. Muniraj, R. Arulmozhiyal, D. Kesavan, An Improved Self-tuning Control Mechanism for BLDC Motor Using Grey Wolf Optimization Algorithm, in: V. Bindhu, J. Chen, J.M.R.S. Tavares (Eds.), *Lect. Notes Electr. Eng.*, Springer Singapore, Singapore, 2020: pp. 315–323. https://doi.org/10.1007/978-981-15-2612-1_30.
- [44] M. Khalilpour, H.H. Razmjooy, N., P. Moallem, Optimal Control of DC motor using Invasive Weed Optimization (IWO) Algorithm, in: *Majlesi Conf. Electr. Eng.*, 2011.
- [45] R.K. Achanta, V.K. Pamula, DC motor speed control using PID controller tuned by jaya optimization algorithm, in: *2017 IEEE Int. Conf. Power, Control. Signals Instrum. Eng.*, 2017: pp. 983–987. <https://doi.org/10.1109/ICPCSI.2017.8391856>.

- [46] R. V Jain, M. V Aware, A.S. Junghare, Tuning of Fractional Order PID controller using particle swarm optimization technique for DC motor speed control, in: 2016 IEEE 1st Int. Conf. Power Electron. Intell. Control Energy Syst., 2016: pp. 1–4. <https://doi.org/10.1109/ICPEICES.2016.7853070>.
- [47] A. Roy, S. Srivastava, Design of optimal PI λ D δ controller for speed control of DC motor using constrained particle swarm optimization, in: Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2016, 2016: pp. 1–6. <https://doi.org/10.1109/ICCPCT.2016.7530150>.
- [48] S. Khubalkar, A. Junghare, M. Aware, S. Das, Modeling and control of a permanent-magnet brushless DC motor drive using a fractional order proportional-integral-derivative controller, Turkish J. Electr. Eng. Comput. Sci. 25 (2017) 4223–4241. <https://doi.org/10.3906/elk-1612-277>.
- [49] B. Hekimoğlu, S. Ekinici, V. Demiray, R. Doğurici, A. Yıldırım, Speed Control of DC Motor Using PID Controller Tuned By Salp Swarm Algorithm, in: 2018 Int. Eng. Nat. Sci. Conf. (IENSC 2018), 2018: pp. 1878–1889.
- [50] S. Ekinici, B. Hekimoğlu, A. Demirören, E. Eker, Speed Control of DC Motor Using Improved Sine Cosine Algorithm Based PID Controller, in: 2019 3rd Int. Symp. Multidiscip. Stud. Innov. Technol., 2019: pp. 1–7. <https://doi.org/10.1109/ISMSIT.2019.8932907>
- [51] I. Khanam, G. Parmar, Application of SFS algorithm in control of DC motor and comparative analysis, in: 2017 4th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Comput. Electron. UPCON 2017, 2017: pp. 256–261. <https://doi.org/10.1109/UPCON.2017.8251057>.
- [52] https://en.wikipedia.org/wiki/PID_controller 02.10.2020
- [53] I. Podlubny, “Fractional order systems and PID-controllers,” IEEE T Automat Contr, vol. 44, no. 1, pp. 208–214, 1999.
- [54] P. Shah and S. Agashe, “Review of fractional PID controller,” Mechatronics, vol. 38, pp. 29–41, 2016.
- [55] Y. Tang, M. Cui, C. Hua, L. Li and Y. Yang, “Optimum design of fractional order PI λ D μ controller for AVR system using chaotic ant swarm,” Expert Syst Appl, vol. 39, no. 8, pp. 6887–6896, 2012.
- [56] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot, “Frequency-band Complex Noninteger Differentiator: Characterization and Synthesis,” IEEE Trans Circuits Syst I, vol. 47, no. 1, pp. 25–39, 2000.
- [57] O. Atan, D. Chen, and M. Turk, “Fractional order PID and application of its circuit model,” J Chin Inst Eng, vol. 39, no. 6, pp. 695–703, 2016.
- [58] “Control Tutorials for MATLAB and Simulink - Aircraft Pitch: System Modeling,” Published with MATLAB® 7.14, 2012. [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=AircraftPitch§ion=SystemModeling> g. [Accessed: 23-Mar-2020].
- [59] A. Johari et al., “Improvement of pitch motion control of an aircraft systems,” Telkomnika (Telecommunication Comput. Electron. Control., vol. 16, no. 5, pp. 2263–2274, Oct. 2018, doi: 10.12928/TELKOMNIKA.v16i5.7434.
- [60] H. Gozde and M. C. Taplamacioglu, “Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system,” J. Franklin Inst., vol. 348, no. 8, pp. 1927–1946, 2011.
- [61] S. Ekinici and B. Hekimoglu, “Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system,” IEEE Access, vol. 7, pp. 39935–39947, 2019.
- [62] Serdar Ekinici a, Baran Hekimoğlu, Davut Izci, “Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor”, Engineering Science and Technology, an International Journal, 2020
- [63] Serdar Ekinici a, Baran Hekimoğlu, Davut Izci, “Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor”, Engineering Science and Technology, an International Journal, 2020
- [64] S. Mirjalili, “SCA: A Sine Cosine Algorithm for solving optimization problems,” Knowledge-Based Syst., vol. 96, pp. 120–133, 2016, doi: 10.1016/j.knosys.2015.12.022.
- [65] S. Ekinici, “Optimal design of power system stabilizer using sine cosine algorithm,” Journal of the Faculty of Engineering and Architecture of Gazi University, vol. 34, no. 3. Gazi Üniversitesi, pp. 1329–1350, 2019, doi: 10.17341/gazimmfd.460529.
- [66] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimisation algorithm: Theory and application,” Adv. Eng. Softw., vol. 105, pp. 30–47, 2017, doi: 10.1016/j.advengsoft.2017.01.004.
- [67] B. Hekimoğlu and S. Ekinici, “Grasshopper optimization algorithm for automatic voltage regulator system,” in 2018 5th International Conference on Electrical and Electronics Engineering, ICEEE 2018, 2018, pp. 152–156, doi: 10.1109/ICEEE2.2018.8391320.

- [68] S. Mirjalili et al. "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems", *Advances in Engineering Software* 000 (2017) 1–29
- [69] H. Salimi, Stochastic Fractal Search: a powerful metaheuristic algorithm, *Knowl. Based Syst.* 75 (2015) 1e18.
- [70] N. S. Jaddi, J. Alvankarian, and S. Abdullah, "Kidney-inspired algorithm for optimization problems," *Commun Nonlinear Sci Numer Simulat*, vol. 42, pp. 358-369, 2017.
- [71] S. Ekinci, A. Demiroren, and B. Hekimoglu, "Parameter optimization of power system stabilizers via kidney-inspired algorithm," *T I Meas Control*, vol. 41, no. 5, pp. 1405-1417, 2019.
- [72] J. E. Hall, Guyton and Hall Textbook of Medical Physiology, Philadelphia, PA, USA: Elsevier Health Sciences, 2015.
- [73] N. S. Jaddi and S. Abdullah, "Optimization of neural network using kidney-inspired algorithm with control of filtration rate and chaotic map for real-world rainfall forecasting," *Eng Appl Artif Intell*, vol. 62, pp. 246-259, 2018.
- [74] W. Zhao, L. Wang, and Z. Zhang, "A novel atom search optimization for dispersion coefficient estimation in groundwater," *Future Gener. Comput. Syst.*, vol. 91, pp. 601–610, Feb. 2019.
- [75] Mech LD. Alpha status, dominance, and division of labor in wolf packs. *Can J Zool* 1999;77:1196– 203.
- [76] Muro C, Escobedo R, Spector L, Coppinger R. Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behav Process* 2011;88:192–7.
- [77] S. Mirjalili et al., "Grey Wolf Optimizer", *Advances in Engineering Software* 69 (2014), pp. 46–61
- [78] B. Hekimoğlu, "Sine-cosine algorithm-based optimization for automatic voltage regulator system," *Trans. Inst. Meas. Control*, vol. 41, no. 6, pp. 1761–1771, 2019, doi: 10.1177/0142331218811453.
- [79] Z. L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, 2004, doi: 10.1109/TEC.2003.821821.
- [80] K. Ogata, *Modern Control Engineering*, 4th Ed. Prentice-Hall, 2002.
- [81] E. Celik, "Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system," *Neural Comput. Appl.*, vol. 30, no. 6, pp. 1991–2002, Sep. 2018.
- [82] Z. -L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, 2004.
- [83] Z.L. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Trans. Energy Convers.* 19 (2004) 384–391. <https://doi.org/10.1109/TEC.2003.821821>.
- [84] M.A. Sahib, A novel optimal PID plus second order derivative controller for AVR system, *Eng. Sci. Technol. an Int. J.* 18 (2015) 194–206. <https://doi.org/10.1016/j.jestch.2014.11.006>.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Veysi KAÇTI
Uyruğu : TC
Doğum Yeri ve Tarihi : Batman 01.04.1994
Telefon : 0544 630 78 63
Faks :
e-mail : veysikacti@gmail.com

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: 15 Temmuz A.L	Kozluk	Batman	2012
Üniversite	: Hacettepe Üniversitesi	Merkez	Ankara	2017
Yüksek Lisans	: Batman Üniversitesi	Merkez	Batman	2021
Doktora	: -			

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2017-2018	ASELSAN	Kontrol Sistemleri Tasarım Mühendisi
2018-	TÜPRAŞ	Enstrüman Proje Mühendisi

UZMANLIK ALANI

Kontrol Sistemleri, Güç Elektronikleri, Devreler ve Sistemler

YABANCI DİLLER

İngilizce

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

Yayınlanmış Makale:

V. Kaçtı , S. Ekinci ve D. İzci , "Henry Gaz Çözünürlük Optimizasyonu ile Uçak Eğitim Kontrol Sistemi için Etkin Kontrolör Tasarımı", Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, c. 11, sayı. 3, ss. 953-964, Eyl. 2020, doi:10.24012/dumf.709449

<https://dergipark.org.tr/tr/pub/dumf/issue/56929/709449>