



T.C.  
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**PARÇACIK SÜRÜ OPTİMİZASYONU (PSO)  
KULLANARAK ÖZ AYARLAMALI PID  
KONTROLÖR TASARIM**

**HUSSEİN ALRUİM ALHASAN**

**YÜKSEK LİSANS TEZİ  
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**KAHRAMANMARAŞ 2018**

**T.C.  
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**PARÇACIK SÜRÜ OPTİMİZASYONU (PSO)  
KULLANARAK ÖZ AYARLAMALI PID  
KONTROLÖR TASARIM**

**HUSSEİN ALRUİM ALHASAN**

**Bu tez,  
Elektrik-Elektronik Mühendisliği Anabilim Dalında  
YÜKSEK LİSANS  
derecesi için hazırlanmıştır.**

**KAHRAMANMARAŞ 2018**

Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü öğrencisi Hussein ALRUİM ALHASAN tarafından hazırlanan “PARÇACIK SÜRÜ OPTİMİZASYONU (PSO) KULLANARAK ÖZ AYARLAMALI PID KONTROLÖR TASARIM” adlı bu tez, jürimiz tarafından 11/1/2018 tarihinde oy birliği ile Elektrik-Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Mahit GÜNEŞ (DANIŞMAN) .....  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Kahramanmaraş Sütçü İmam Üniversitesi

Prof. Dr. Ahmet Serdar YILMAZ (ÜYE) .....  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Kahramanmaraş Sütçü İmam Üniversitesi

Yrd. Doç. Dr. M.Ali ÖZÇELİK (ÜYE) .....  
Elektrik Enerji Anabilim Dalı  
Gaziantep Üniversitesi

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylıyorum.

Doç. Dr. Mustafa ŞEKKELİ .....

Fen Bilimleri Enstitüsü Müdürü

## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada, alıntı yapılan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

.....  
Hussein ALRUİM ALHASAN

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

**PARÇACIK SÜRÜ OPTİMİZASYONU (PSO) KULLANARAK ÖZ AYARLAMALI  
PID KONTROLÖR TASARIM  
(YÜKSEK LİSANS TEZİ)**

**HUSSEİN ALRUİM ALHASAN**

**ÖZET**

Bu tezde, doğru akım (DA) motor sisteminde yeni adaptif parçacık sürüsü optimizasyonu (APSO) algoritması kullanarak öz ayarlamalı PID kontrolör tasarımı yapılmıştır. PID kontrolörün manuel ayarlanması iyi bir performans sağlamadığı gibi, parametrelerinin ayarlanması zor ve sıkıcıdır. PID kontrolörün ayar işlemi, parçacık sürüsü optimizasyon (PSO) algoritması ile yapılmıştır. Eylemsizlik ağırlığı PSO algoritmasında en önemli parametre olup global ve yerel arama süreci arasında denge sağlamaktadır. PSO algoritmasının başlangıcından bu yana, uygun eylemsizlik ağırlığı belirlemek için farklı PSO algoritması stratejileri önerilmiştir. Bu çalışmada, parçacıkların uygunluk değerlerinin karşılaştırılmasına dayanılarak eylemsizlik ağırlığını ayarlamak için yeni bir strateji önerilmektedir.

Adaptif parçacık sürüsü optimizasyonu algoritması (APSO), arama verimliliğini ve yakınsama hızını iyileştirmek için eylemsizlik ağırlığı, hızlanma katsayıları ve diğer algoritmik parametrelerin çalışma zamanında otomatik olarak kontrol edilmesini sağlamaktadır. Standart parçacık sürüsü optimizasyon algoritması (S-PSO) ve zamanla değişen eylemsizlik ağırlığı parçacık sürüsü optimizasyon algoritması (TVIW-PSO) ile karşılaştırıldığında, önerilen adaptif parçacık sürüsü optimizasyon (APSO) algoritması, hızlı yakınsama ve optimal çözüme doğru harekete devam açısından daha iyi performans vermektedir.

**Anahtar Kelimeler** : Parçacık sürüsü optimizasyon, PID kontrol, Optimal kontrol

Kahramanmaraş Sütçü İmam Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı, Ocak/ 2018

Danışman: Prof. Dr. Mahit GÜNEŞ

Sayfa sayısı: 59

**A SELF-TUNING PID CONTROLLER DESIGN USING A PARTICLE SWARM  
OPTIMIZATION ALGORITHM (PSO)  
(M.Sc. THESIS)**

**HUSSEİN ALRUİM ALHASAN**

**ABSTRACT**

This thesis presents a self-tuning PID controller design using a new adaptive particle swarm optimization algorithm (APSO) for DC motor. Manual tuning of PID controllers does not provide good performance, difficult and tedious. The tuning process of PID controller is done by particle swarm optimization (PSO) algorithm. Inertia weight is the most important parameter in PSO algorithm, which provides a balance between the global and local search process. Since the beginning of inertia weight in PSO algorithm, Different strategies of PSO algorithm have been proposed in order to determine the appropriate inertia weight. In this paper, we propose a completely new strategy to adapt the inertia weight based on the fitness value of the particles.

Adaptive particle swarm optimization algorithm (APSO) provides automatic control of inertia weight, acceleration coefficients and other algorithmic parameters over time to improve search efficiency and convergence speed. Comparing with standard particle swarm optimization (PSO) algorithm and time varying inertia weight particle swarm optimization (TVIW-PSO) algorithm, the proposed adaptive particle swarm optimization (APSO) algorithm gives better performance in terms of quick convergence capability and continues movement toward the optimal solution region.

**Key words:** Particle swarm optimization, PID control, Optimal control

University of Kahramanmaraş Sütçü İmam  
Graduate School of Natural and Applied Sciences  
Department of Electric-Electronic Engineering , January/ 2018

Supervisor: Prof. Dr. Mahit GÜNEŞ

Pages Number: 59

## TEŐEKKÜR

Tezimin hazırlanmasında maddi ve manevi hiçbir desteęini eksik etmeyen aileme, alıőmamın her aőamasında beni destekleyen deęerli danıőman hocam Prof. Dr. Mahit GÜNEŐ'a, bölüm başkanımız Prof. Dr. Hasan RIZA ÖZALIK'a teőekkür ve saygılarımı sunarım.

Ayrıca bu tezin hazırlanmasında yardımlarını esirgemeyen tüm hocalarımıza ve aileme bir teőekkürü bor bilirim.



# İÇİNDEKİLER

## Sayfa No

ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ .....	viii
SİMGELER VE KISALTMALAR DİZİNİ.....	x
1. GİRİŞ .....	1
2. ÖNCEKİ ÇALIŞMALAR.....	5
3. DOĞRU AKIM MOTORLARI.....	7
3.1. Doğru Akım Motorun Modeli .....	7
3.1.1. Transfer fonksiyonu .....	8
3.1.2. Durum uzay modeli.....	9
4. PARÇACIK SÜRÜSÜ OPTİMİZASYONU .....	10
4.1. Genel Bilgileri.....	10
4.2. Parçacık Sürüsü Optimizasyonu Algoritması .....	11
4.3. Parçacık Sürüsü Optimizasyonunun Ana Parametreleri.....	14
4.4. PSO Uygunluk Kriterleri .....	16
4.5. Parçacık Sürüsü Optimizasyonu İçin Farklı Eylemsizlik Ağırlık Stratejileri.....	17
4.5.1. Sabit eylemsizlik ağırlığının stratejisi.....	17
4.5.2. Rastgele eylemsizlik ağırlığının stratejisi .....	17
4.5.3. Eylemsizlik ağırlığının doğrusal azaltılmasının stratejisi .....	18
4.5.4. Sigmoid ile eylemsizlik ağırlığının azaltılmasının veya artırılmasının stratejileri .....	19
4.5.5. Kaotik eylemsizlik ağırlığı stratejisi .....	21
4.6. Adaptif Parçacık Sürüsü Optimizasyonu.....	23
4.6.1. Parçacıkların başarısına dayalı adaptif PSO.....	23
4.6.2. Global-yerel en iyi eylemsizlik ağırlığı adaptif PSO.....	24
4.7. Önerilen Adaptif Parçacık Sürü Optimizasyon Algoritması .....	24
5. PID DENETİMİ.....	27
6. SİMULASYON ÇALIŞMALARI .....	29
7. DENEYSEL ÇALIŞMALARI .....	47



**Sayfa No**

8. SONUÇLAR.....	51
9. ÖNERİLER .....	52
KAYNAKLAR.....	53
EKLER.....	55
ÖZGEÇMİŞ.....	59



## ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 3.1. DA motorunun elektrik ve mekanik eşdeğer diyagramı .....	7
Şekil 3.2. Doğru akım motor blok diyagramı .....	9
Şekil 4.1. PSO parametrelerinin vektör olarak gösterimi.....	12
Şekil 4.2. Parçacık sürüsü algoritmasının akış diyagramı .....	13
Şekil 4.3. Rastgele eylemsizlik ağırlığının stratejisine göre eylemsizlik ağırlığı değişimleri.....	18
Şekil 4.4. Eylemsizlik ağırlığının doğrusal azaltılmasının stratejisine göre eylemsizlik ağırlığı değişimleri.....	19
Şekil 4.5. Sigmoid ile eylemsizlik ağırlığının azaltılmasının stratejisine göre eylemsizlik ağırlığı değişimleri.....	20
Şekil 4.6. Sigmoid ile eylemsizlik ağırlığının artırılmasının stratejisine göre eylemsizlik ağırlığı değişimleri.....	21
Şekil 4.7. Kaotik eylemsizlik ağırlığı stratejisine göre eylemsizlik ağırlığı değişimleri ..	22
Şekil 4.8. Sürüdeki parçacıkların uygunluk değerleri ile pbest parçacıklarının uygunluk değerleri arasındaki farka göre $\lambda_1$ 'in değeri.....	25
Şekil 4.9. Pbest parçacıklarının uygunluk değerleri gbest parçacığının uygunluk değeri arasındaki farka göre $\lambda_2$ 'in değeri .....	26
Şekil 5.1. PID kontrolörün blok diyagramı .....	28
Şekil 6.1. PSO-PID kontrolör blok diyagramı .....	29
Şekil 6.2. DA motor için üç PSO stratejilerine göre adım cevabı .....	30
Şekil 6.3. '20' iterasyon için üç PSO algoritmasının en iyi uygunluk değeri.....	31
Şekil 6.4. 1'inci iterasyon sonunda parçacıkların konumları.....	44
Şekil 6.5. 2'inci iterasyon sonunda parçacıkların konumları.....	44
Şekil 6.6. 3'üncü iterasyon sonunda parçacıkların konumları.....	44
Şekil 6.7. 4'üncü iterasyon sonunda parçacıkların konumları.....	44
Şekil 6.8. 5'inci iterasyon sonunda parçacıkların konumları.....	45
Şekil 6.9. 6'ıncı iterasyon sonunda parçacıkların konumları.....	45

**Sayfa No**

Şekil 6.10. 7'inci iterasyon sonunda parçacıkların konumları.....	45
Şekil 6.11. 8'inci iterasyon sonunda parçacıkların konumları.....	45
Şekil 6.12. 9'inci iterasyon sonunda parçacıkların konumları.....	46
Şekil 6.13. 10'inci iterasyon sonunda parçacıkların konumları.....	46
Şekil 6.14. 20'inci iterasyon sonunda parçacıkların konumları.....	46
Şekil 6.15. DA motor için IAE kriterine göre adım cevabı.....	46
Şekil 7.1. Fırçalı 350 rpm (30:1) 37Dx52L mm metal dişli motor 64 CPR enkoderli .....	47
Şekil 7.2. L298N H köprülü çift motor sürücü kartı .....	49
Şekil 7.3. Kullanılan arduino uno r3 kartı .....	50

## ÇİZELGELER DİZİNİ

	<u>Sayfa No</u>
Çizelge 5.1. PID katsayılarının sistem üzerine etkisi .....	28
Çizelge 6.1. Doğru akım DA motoru ile ilgili parametreleri .....	30
Çizelge 6.2. Standart PSO algoritması ile DA motorun PID değerleri .....	31
Çizelge 6.3. Eylemsizlik ağırlığının doğrusal azaltmasının PSO algoritması ile DA motorun PID değerleri .....	31
Çizelge 6.4. Önerilen adaptif PSO algoritması ile DA motorun PID değerleri .....	31
Çizelge 6.5. Üç PSO stratejileri kullanarak öz ayarlamalı PID kontrolör ile DA motor sisteminin açılma hızının performans endeksleri .....	32
Çizelge 6.6. 1'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri .....	33
Çizelge 6.7. 1'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu .....	33
Çizelge 6.8. 2'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri .....	34
Çizelge 6.9. 2'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu .....	34
Çizelge 6.10. 3'üncü iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri .....	35
Çizelge 6.11. 3'üncü iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu .....	35
Çizelge 6.12. 4'üncü iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri .....	36
Çizelge 6.13. 4'üncü iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu .....	36
Çizelge 6.14. 5'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri .....	37
Çizelge 6.15. 5'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu .....	37
Çizelge 6.16. 6'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri .....	38

Çizelge 6.17. 6'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu.....	38
Çizelge 6.18. 7'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri.....	39
Çizelge 6.19. 7'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu.....	39
Çizelge 6.20. 8'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri.....	40
Çizelge 6.21. 8'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu.....	40
Çizelge 6.22. 9'uncu iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri.....	41
Çizelge 6.23. 9'uncu iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu.....	41
Çizelge 6.24. 10'uncu iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri.....	42
Çizelge 6.25. 10'uncu iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu.....	42
Çizelge 6.26. 20'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri.....	43
Çizelge 6.27. 20'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu.....	43
Çizelge 7.1. Kullanılan fırçalı dişli DA motor özellikleri .....	48
Çizelge 7.2. Motor/enkoder kablolarının fonksiyonları .....	48
Çizelge 7.3. Deneysel çalışmada, önerilen adaptif PSO algoritması ile DA motorun PID kazançları.....	50
Çizelge 7.4. Deneysel çalışmada, önerilen adaptif PSO kullanarak öz ayarlamalı PID kontrolör ile DA motor sisteminin açılma hızının performans değerleri.....	50

## SİMGELER VE KISALTMALAR DİZİNİ

<b>DA</b>	: Doğru Akım
<b>GA</b>	: Genetik Algoritma
<b>YAK</b>	: Yapay Arı Kolonisi
<b>KKA</b>	: Karınca Koloni Algoritması
<b>PSO</b>	: Parçacık Sürüsü Optimizasyonu
<b>S-PSO</b>	: Standard Particle Swarm Optimization, Standart Parçacık Sürüsü Optimizasyonu
<b>TVIW-PSO</b>	: Time Varying Inertia Weight Particle Swarm Optimization, Eylemsizlik Ağırlığının Doğrusal Azaltmasının Parçacık Sürüsü Optimizasyonu
<b>A-PSO</b>	: Adaptive Particle Swarm Optimization, Adaptif Parçacık Sürüsü Optimizasyonu
<b>V<sub>app</sub></b>	: Gerilim Kaynağı (V)
<b>R</b>	: Armatür Direnci ( $\Omega$ )
<b>L</b>	: Armatür İndüktansı (H)
<b>V<sub>emf</sub></b>	: Zıt Elektromotor Gerilimi (V)
<b>i</b>	: Armatür Akımı (A)
<b><math>\omega</math></b>	: Motor Milinin Açısal Hızı (rad/sec)
<b>K<sub>m</sub></b>	: Moment Katsayısı
<b>K<sub>b</sub></b>	: Zıt Elektromotor Gerilimin Sabiti
<b>K<sub>f</sub></b>	: Viskoz Sürtünme Katsayısı (N.m.s)
<b>J</b>	: Motor Eylemsizlik Momenti (Kg.m <sup>2</sup> )
<b>c<sub>1</sub>, c<sub>2</sub></b>	: Hızlanma Katsayıları
<b>r<sub>1</sub>, r<sub>2</sub></b>	: Rastgele Sayıları
<b><math>\omega</math></b>	: PSO Algoritmasında Eylemsizlik Ağırlığı
<b>pbest</b>	: Parçacığın En İyi Kendi Çözümü

<b>gbest</b>	: Tüm Parçacıkların En İyi Çözümü
<b>V<sub>i</sub></b>	: Parçacık Hızı
<b>χ<sub>i</sub></b>	: Parçacık Konumu
<b>χ<sub>i,best</sub></b>	: pbest Parçacığının Konumu
<b>χ<sub>gbest</sub></b>	: gbest Parçacığının Konumu
<b>IAE</b>	: İntegral Of Absolute Error- Mutlak Hataların Toplamı
<b>ITAE</b>	: İntegral Of Time Weighted Absolute Error – Zaman Ağırlıklı Mutlak Hataların Toplamı
<b>ISE</b>	: İntegral Of Squared Error - Hataların Karelerinin Toplamı
<b>ITSE</b>	: İntegral Of Time Weighted Squarred Error – Zaman Ağırlıklı Hataların Karelerinin Toplamı
<b>İter, k</b>	: İterasyon Sayısı
<b>iter<sub>max</sub>, gen</b>	: Maksimum İterasyon Sayısı
<b>rand(), z</b>	: Rastgele Sayıları
<b>u , α</b>	: Sigmoid Fonksiyonunun Netliğinin Ayarlama Sabiti
<b>n</b>	: Sigmoid Fonksiyonunun Ayırma Noktasının Ayarlama Sabiti
<b>P<sub>s</sub></b>	: Parçacıkların Başarısının Fonksiyonu
<b>n<sub>s</sub></b>	: Parçacıkların Sayısı
<b>PID</b>	: Proportional Integral Derivative - Oransal İntegral Türev Denetim
<b>K<sub>p</sub></b>	: PID Denetimin Orantı Değeri
<b>K<sub>i</sub></b>	: PID Denetimin İntegral Değeri
<b>K<sub>d</sub></b>	: PID Denetimin Türev Değeri
<b>RPM</b>	: DA Motorun Hızı (rpm)
<b>A, B</b>	: Hall-Effect Enkoder Kanalları
<b>PULSE</b>	: A, B Kanalların Hem Yükselen Hem De Düşen Kenarların Sayısı

## 1.GİRİŞ

Elektrik motorları, ürün işleme, şekillendirme, taşıma, hareket kontrolü, tork kontrolü, konum ve hız kontrolü gibi endüstriyel uygulamalarda yaygın olarak kullanılmaktadır. Her elektrik motoru istenen kontrole uygun bir davranış göstermeyebilir. Bunun için istenen işleme uygun motor seçilmelidir. Endüstriyel uygulamalarda genelde kullanılan elektrik motorları yapıya göre farklılıklar göstermektedir. Elektrik motorlarının ortak yanı, elektrik enerjisini mekanik enerjiye dönüştürmeleridir.

İstenilen işlemleri hızlı ve hassas yapmak için uygun kontrol yöntemi seçilmelidir. Bu nedenle sistemin analizinin yapılması gerekir. Analiz süreci, incelenen sistemin matematiksel modelinin çıkartılması, transfer fonksiyonunun elde edilmesi ve gerekli benzetim çalışmaları yapılmasını kapsamaktadır. Analiz işlemi sırasında ilgili problemin çözümüne yönelik bir optimizasyon algoritmasına ihtiyaç duyulabilir.

Doğru akım (DA) motoru bilinen en eski motor türüdür. Her motorun kendi avantajları ve dezavantajları vardır. DA motorlarının en büyük avantajı hız kontrolünün kolay bir şekilde yapılabilmesidir. Bu nedenle endüstride uzun bir süre kullanılmıştır. En büyük dezavantajı, DA makinelerde kollektör ve fırça teçhizatının bulunmasıdır. Buna ek olarak, bakım, yüksek üretim maliyetleri ve çalışma koşullarından dolayı kolay aşınmaları gibi olumsuz yönleri vardır.

Bir problem için, belirli şartlar altında tüm çözümler arasından en iyi çözümü elde etme işine optimizasyon denir. Problemlerin çözümüne yönelik birçok optimizasyon algoritması araştırmacılar tarafından geliştirilmiştir. Optimizasyon algoritmaları amaç fonksiyonunu en küçük ya da en büyük yapan değeri bulmayı hedefler. En çok bilinen algoritmalar, yapay arı koloni algoritması, genetik algoritma, karınca koloni algoritması ve parçacık sürüsü algoritmasıdır.

Genetik algoritmalar, doğal seçim ilkelerine dayanan bir optimizasyon yöntemidir. Genetik algoritmaların temel ilkeleri 1960 yılında Michigan Üniversitesindeki John Holland tarafından bulunmuştur (Tang et al., 1996). Seçim, çaprazlama ve mutasyon, genetik algoritmalarda kullanılan üç temel genetik operatörlerdir. Genetik algoritma yeni



bir popülasyon (yavru nesil) oluşturmak için bu operatörleri kullanır. Yeni nesil eski ebeveyn nesil ile yer değiştirir. Dizin (kromozomun) performansını belirlemek için bir amaç fonksiyon değeri vardır. Yeni kromozomlar bu amaç fonksiyon değerlerine göre seçilirler. Bir genetik algoritmanın evrim sürecinde, her yeni üretilen nesilde daha uyumlu iyi nesiller üretmeye çalışılır.

GA üç temel operatörden meydana gelmektedir. seçim, çaprazlama ve mutasyondur. Genetik algoritmadaki her çözüm bir kromozom olarak kodlanır. Genetik algoritmanın basit formunda başlangıç popülasyonu rastgele olarak oluşturulur. Popülasyondaki her birey ikili düzende veya tamsayı olarak kodlanmaktadır. Bundan sonra bu bireyler belli amaç fonksiyonuna göre değerlendirilmektedir. Yüksek uygunluk değerlerine sahip bireylerin seçilme olasılığı daha yüksektir. Seçim operatörü ile en iyi bireyler hayatta kalır. Seçim operatörü sonrasında hayatta kalan bireyleri çaprazlama topluluğundaki diğer bir birey ile rastgele eşleştirilmektedir. Eğer çaprazlama meydana gelmez ise ebeveynler aynen bir sonraki popülasyona aktarılmaktadır. Amaç, yüksek uygunluk değerlerine sahip bireylerin daha iyi bireyler üretebilmelerini sağlamaktır. Algoritmanın belirlenen bir durdurma koşulu sağlanıncaya kadar devam etmektedir.

Yapay arı kolonisi (YAK), bal arılarının akıllı davranışlarına dayanan en son tanımlanmış algoritmalarından biridir. YAK ilk olarak 2005 yılında optimizasyon problemlerini çözmek için Derviş Karaboğa tarafından geliştirmiştir (Karaboga, 2005). Son yıllarda, YAK algoritması oldukça popüler olmuş bir optimizasyon algoritmasıdır. Doğal yaşamda bal toplayan bir arı kolonisi içinde görev paylaşımı vardır. Kolonide görevli, gözcü ve kâşif arılar farklı görevleri yapmaktadırlar. Görevli arıların görevi, bal kaynaklarının çiçek özlerini toplamak ve topladığı balları kovana boşaltmaktır. Görevli arıların ikinci görevi, toplanan balın kaynağının kalitesini belirlemektir. Gözcü arıların görevi, görevli arıları yeni bal kaynaklarına yönlendirmektir. Diğer taraftan kâşif arılar rastgele olarak çevrede dolaşırken yeni bal kaynakları aramaktadırlar. Bu algorithmada her bir bal kaynağı yalnız bir görevli arı tarafından toplanır. Sonuç olarak, popülasyondaki çözüm sayısı, görevli arı sayısı ile aynıdır.

Az sayıda parametre gerektiren, kolay uygulanan ve basit bir çalışma prensibine sahip olan YAK algoritması arařtırmacılar tarafından farklı çeřitli optimizasyon problemlerinin çözümünde kullanılmıřtır.

Karınca koloni algoritması (KKA), Dorigo ve arkadaşları tarafından 1996 yılında önerilmiş en yeni sezgisel algoritmalarından biridir (Dorigo et al., 1991). Karınca kolonisi optimizasyon algoritması, gerçek karınca kolonilerinin akıllı davranıřlarına dayanan bir optimizasyon algoritmasıdır. Gerçek karıncalar, yiyecek kaynađı ile yuvaları arasındaki en kısa yolu bulma kabiliyetine sahiptirler. Karıncalar birbirleri arasındaki kimyasal haberleřmeden dolayı en kısa yolu bulma yeteneđine sahiptirler. Karıncalar birbirleriyle haberleřmede feromon denilen kimyasal bir madde kullanmaktadır. Karıncalar yürürken yolları üzerine bazı feromon maddesi bırakırlar ve her bir karınca yuva ya da yiyecek bulmak için bir yönü seçer. Bir yönün seçilme olasılıđı, bu yöndeki feromon maddesi miktarına bađlıdır. Bu nedenle, bütün karıncalar, feromon miktarına göre en kısa yolları seçecektir.

KKA parametrelerin dođru seçilmesi, etkili bir şekilde algoritmanın performansını etkilemektedir. Karıncaların sayısının arttırılması daha iyi bir çözüm getirir, ancak hesaplama süresini uzatır.

Bu çalışmada, ölçülen çıkıř ile referans arasındaki hatayı kaldırmak için pid kazançlarını ayarlamak için parçacık sürüsü optimizasyonu (PSO) yöntemi kullanılmıřtır. Parçacık sürüsü optimizasyonu 1995 yılında J.Kennedy ve R.C. Eberhart tarafından geliştirilmiş (Kennedy ve Eberhart, 1995). Parçacık sürü optimizasyonu aslında sürü zekâsına dayanan bir algoritmadır. PSO parçacıklar arasındaki sosyal bilgi paylaşımına dayanmaktadır.

PSO algoritmasındaki ilk popülasyon rastgele parçacıklardan oluřmaktadır. PSO algoritması, her iterasyonda parçacıklarını güncelleyerek optimum çözümü arar. Her bir iterasyonda, tüm parçacıklar sürüdeki en iyi parçacıđa dođru uçarlar. Her bir iterasyondaki sürede tüm parçacıkların uygunluk deđerleri hesaplanır. Her iterasyonda, tüm parçacıklar iki “en iyi” deđerlere göre güncellenirler. Parçacıđın en iyi kendi çözümü (pbest) ve tüm parçacıkların en iyi çözümü (gbest) güncellenen deđerlerdir. Maksimum iterasyon sayısına

ulaşıldığında veya uygunluk değeri istenilen seviyeye ulaştığında algoritma durur. Algoritmanın sonunda elde edilen gbest parçacığı aranan optimum çözümdür.

Eylemsizlik ağırlığı PSO algoritmasında büyük bir etkiye sahiptir. PSO algoritmasında eylemsizlik ağırlığı global ve yerel arama sürecini dengelemek için kullanılmaktadır. Eylemsizlik ağırlığı yüksek olduğunda, global arama yeteneği güçlüdür ancak yerel arama yeteneği düşüktür. Bunun yanında, eylemsizlik ağırlığı düşük olduğunda, yerel arama yeteneği güçlüdür ancak global arama yeteneği düşüktür. Eylemsizlik ağırlığı, adaptif PSO algoritmasında sürü durumuna göre değiştirilir. Bu çalışmada, eylemsizlik ağırlığını ayarlamak için yeni bir adaptif PSO algoritması önerilmiştir.

Bu yöntemin hızlı sonuç bulması, çok az algoritma parametresi kullanması, etkili ve uygulaması kolay olması, hızlı yakınsama özelliğine sahip olması gibi bir çok üstünlüğü bulunmaktadır. Parçacık sürüsü optimizasyonun ana avantajı optimize etmeye çalıştıkları problem hakkında herhangi bir bilgiye ihtiyaç duymamalarıdır. Sistemin matematik model bilgilerini çıkarmaya gerek yoktur. Algoritmada karmaşık matematik hesaplamaların yerine yalnızca giriş-çıkış bilgileri ile işlem yapılmaktadır. Bu özelliğinden dolayı araştırmacılar tarafından karmaşık problemlerin optimizasyonu için tercih edilebilir bir yöntemdir.

## 2. ÖNCEKİ ÇALIŞMALAR

Parçacık sürüsü optimizasyonu (PSO), Eberhart ve Kennedy tarafından 1995 yılında geliştirilmiş popülasyona dayalı bir stokastik optimizasyon algoritmasıdır (Kennedy ve Eberhart, 1995). PSO algoritması hem doğrusal hem de doğrusal olmayan problemleri çözmek için kullanılmaktadır. PSO algoritması, çok değişkenli ve çok parametrelili optimizasyon problemlerine çözüm bulmak için kullanılmaktadır. PSO algoritmasında, algoritmanın optimize etmeye çalıştığı problem hakkında matematik model bilgilerini çıkarmaya gerek yoktur. PID günümüzde basit yapısı nedeniyle endüstride yaygın olarak kullanılan bir kontrol tekniğidir. Bu kontrol tekniğinde bulunan parametrelerin (oransal, türev, integral) en uygun değerde ayarlanması çok önemlidir. Parametrelerin manuel ayarlanması iyi bir performans sağlamadığı gibi parametre ayarlama işlemi zaman alır ve sıkıcıdır. Bu çalışmada PID kontrolörün optimum kazanç parametrelerinin belirlenmesi için PSO algoritması ile seçilmiştir. Bu konularla ilgili yapılan çalışmalardan bir kısmı aşağıda verilmiştir.

1998'de ilk kez Shi ve Eberhart sabit eylemsizlik ağırlığı getirerek eylemsizlik ağırlığı kavramını sundular (Shi ve Eberhart, 1998). Eylemsizlik ağırlığı, global ve yerel arama süreci arasında denge kurmak için en önemli parametredir. Büyük eylemsizlik ağırlığı global arama sürecini iyileştirir. Ancak, küçük eylemsizlik ağırlığı yerel arama sürecini iyileştirir.

2002'de Eberhart ve Shi, PSO'nun verimliliğini arttırmak için rastgele bir eylemsizlik ağırlığı stratejisini önerdiler (Eberhart ve Shi, 2002).

Xin ve arkadaşları 2009 yılında PSO ile eylemsizlik ağırlığının doğrusal azaltılmasının stratejisini önerdiler (Xin et al., 2009).

Bazı araştırmacılar, uygun parametre seçiminin zorluğunu aşmak için adaptif parçacık sürüsü optimizasyonunu önermişlerdir (Arumugam ve Rao, 2006).

Junfeng Chen ve arkadaşları tarafından 2006 yılında yapılan çalışmada PID kontrolör parametrelerini ayarlamak için adaptif PSO algoritması kullanılmışlardır (Chen et al., 2006).

Seçkin Tamer ve Cihan Karakuzu tarafından 2006 yılında yapılan çalışmada parçacık sürüsü optimizasyonu ve benzetim örnekleri tartışılmıştır (Tamer ve Karakuzu, 2006).

Önder Belgin 2007 yılında yaptığı yüksek lisans tez çalışmasında PSO algoritması ve diğer sezgisel yaklaşımlar hakkında bilgiler vermiştir (Belgin, 2007).

Brahim Allaoua ve arkadaşları 2009 yılında yaptıkları çalışmada DA motor hız kontrolü için kullanılan PID parametrelerini bulmak için PSO algoritmasını kullanmışlardır (Allaoua et al., 2009).

Haluk Gözde ve arkadaşlarının 2010 yılında yaptıkları çalışmada iki bölgeyi ara ısıtmalı termal güç sisteminin yük-frekans kontrolü için kullanılan PI kontrolör parametrelerini bulmak için PSO algoritmasını kullanmışlardır (Gözde et al., 2010).

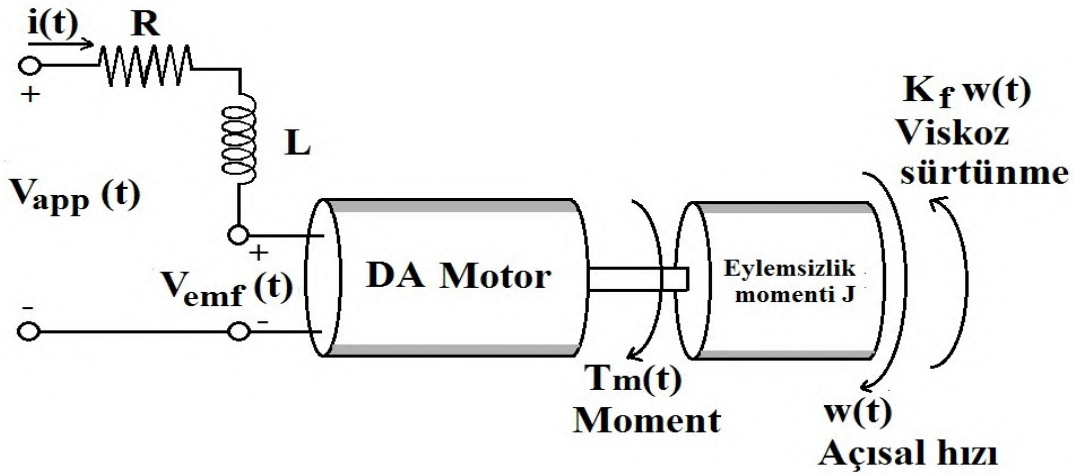
Serdar Özyön ve arkadaşları 2011 yılında PSO algoritmasını termik birimlerden oluşan çevresel ekonomik güç dağıtım problemlerine uygulamışlardır (Özyön et al., 2011).

### 3. DOĞRU AKIM MOTORLARI

Doğru akım motoru, doğru akım elektrik enerjisini mekanik enerjiye dönüştüren elektrik makinesidir. Doğru akım motoru istenen birçok özelliğe sahiptir. Bunlardan bazıları uzun ömürlü, pahalı değil, güvenli ve düşük voltajda çalışabilmesidir. Bu özelliklerinden dolayı doğru akım motorları bir çok uygulamada tercih edilmektedir.

#### 3.1. Doğru Akım Motorun Modeli

Bu çalışmada kullanılan kalıcı mıknatıslı DA motoru,  $V_{app}$  gerilim kaynağından sağlanan elektrik enerjisini oluşturduğu manyetik alan sayesinde dönen rotor vasıtasıyla mekanik enerjiye dönüştürür. DA motorunda şekil 3.1'de görüldüğü gibi uygulanan  $V_{app}$  giriş gerilim altında,  $R$  armatür direnci ve  $L$  indüktansı olan seri bir devre ile rotor döndüğünde  $V_{emf}$  zıt elektromotor gerilimi oluşur. Rotor iletkenlerinden akım geçtiğinde, iletkenler manyetik alan tarafından alanın dışına doğru itilirler. Böylece rotor çevresinde, rotoru döndüren bir moment meydana gelir.



Şekil 3.1. DA motorunun elektrik ve mekanik eşdeğer diyagramı.

Motor milinde oluşan ( $T_m$ ) moment, ( $K_m$ ) moment katsayısı ve ( $i$ ) armatür akımıyla doğru orantılıdır.  $T_m$  motor momenti

$$T_m(t) = K_m i(t) \quad (3.1)$$

ile ifade edilir. ( $V_{emf}$ ) zıt elektromotor gerilimi, ( $\omega$ ) motor milinin açısal hızı ile doğru orantılıdır. ( $V_{emf}$ ) zıt elektromotor gerilimi

$$V_{emf}(t) = K_b \omega(t) \quad (3.2)$$

ile ifade edilir.  $K_b$  zıt elektromotor gerilimin sabitidir.

DA motoru elektrik ve mekanik eşdeğer devresinde Kirchoff ve Newton kanunları uygulandığında, motorun matematik modeli elde edilir.

$$J \frac{d\omega}{dt} = -K_f \omega(t) + K_m i(t) \quad (3.3)$$

$$V_{app}(t) - V_{emf}(t) = L \frac{di}{dt} + Ri(t) \quad (3.4)$$

$K_f$ , viskoz sürtünme katsayısıdır. Yukarıdaki denklemleri basitleştirdikten sonra, motor dinamiklerini tanımlayan iki diferansiyel denklem elde edilecektir.

$$\frac{di}{dt} = -\frac{R}{L}i(t) - \frac{K_b}{L}\omega(t) + \frac{1}{L}V_{app}(t) \quad (3.5)$$

$$\frac{d\omega}{dt} = -\frac{K_f}{J}\omega(t) + \frac{K_m}{J}i(t) \quad (3.6)$$

Burada;  $\omega$  rotor açısız hızı ve J motor eylemsizlik momentini gösterir.

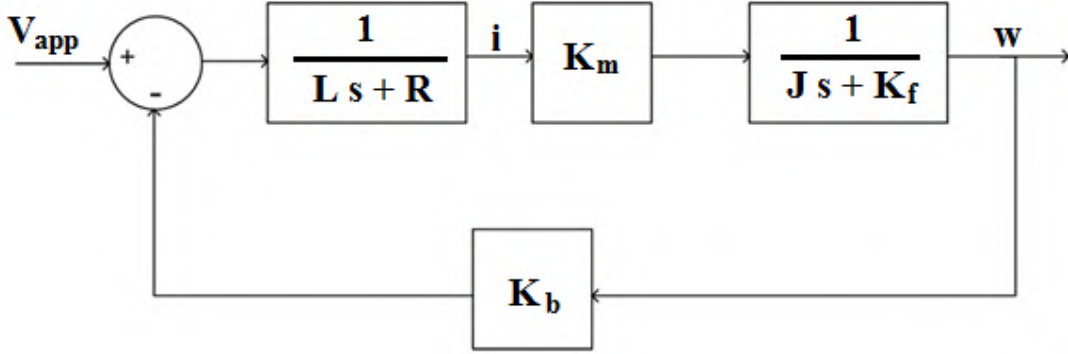
### 3.1.1. Transfer fonksiyonu

Denklem 3.5 ve 3.6'in  $i(0)=0$  ve  $\omega(0)=0$  başlangıç değerlerinde laplace dönüşümleri düzenlenirse, yüksüz DA motoru için armatür akımının ve motor milinin açısız konumunun frekans alanındaki ifadeleri

$$i(s) = \frac{1}{Ls + R} [V_{app}(s) - K_b \omega(s)] \quad (3.7)$$

$$\omega(s) = \frac{K_m}{Js + K_f} i(s) \quad (3.8)$$

elde edilir. Denklem 3.7 ve 3.8'den DA motorunun blok diyagramı şekil 3.2'de gösterildiği gibi elde edilmiştir.



Şekil 3.2. Doğru akım motor blok diyagramı

Denklem 3.7 ve 3.8'den yararlanarak DA motorun transfer fonksiyonu

$$\frac{w(s)}{V_{app}(s)} = \frac{K_m}{JLs^2 + (JR + K_f L)s + K_f R + K_m K_b} \quad (3.9)$$

ifade edilir.

### 3.1.2. Durum uzay modeli

Denklem 3.5 ve 3.6 vektör-matris biçiminde düzenlenirse, DA motoru durum uzay modeli

$$\frac{d}{dt} \begin{bmatrix} i \\ w \end{bmatrix} = \begin{bmatrix} -R/L & -K_b/L \\ K_m/J & -K_f/J \end{bmatrix} \begin{bmatrix} i \\ w \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} V_{app}(t) \quad (3.10)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ w \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} V_{app}(t)$$

ifade edilir. Armatür akımı (i) ve motor milinin açısal hızı (ω) durum değişkenleridir. Uygulanan gerilim (V<sub>app</sub>) sistem girişi veya kontrol sinyalidir ve açısal hızı (ω) sistem çıktısıdır.



## 4. PARÇACIK SÜRÜSÜ OPTİMİZASYONU

### 4.1. Genel Bilgileri

Parçacık sürü optimizasyonu kuş ve balık sürülerinden esinlenerek geliştirilmiş popülasyon tabanlı arama ve optimizasyon algoritmasıdır. Parçacık sürü optimizasyonu, ilk olarak Kennedy ve Eberhart tarafından 1995 yılında ortaya atılmıştır (Eberhart ve Kennedy, 1995). Diğer evrimsel algoritma ve matematiksel temelli algoritmalara göre fazla hafıza gerektirmeyen, hesapsal olarak etkili ve uygulanması kolay bir optimizasyon yöntemidir. Ayrıca hızlı yakınsama özelliğine sahiptir. PSO az parametre gerektirmesi, hızlı sonuç bulması ve çok etkili global arama algoritması nedeniyle diğer arama algoritmalarına göre daha iyi performans sağlar. PSO, PID parametrelerinin ayarlama, yapay sinir ağlarının eğitimi, fonksiyon optimizasyonu, bulanık mantık sistemleri, çizelgeleme vb. pek çok optimizasyon problemlerinde başarı ile kullanılmaktadır.

Parçacık sürü optimizasyonu aslında sürü zekâsına dayanan bir algoritmadır. PSO parçacıklar arasındaki sosyal bilgi paylaşımına dayanmaktadır. Her bireye parçacık denir ve parçacıklardan oluşan popülasyona da sürü denir. Her bir parçacık optimize edilmiş problemin birer aday çözümünü temsil etmektedir. Amaç sürüdeki en iyi parçacığın konumunu belirlemektir. Parçacıklar bir sonraki konumunu geçmiş tecrübelerine ve sürüdeki en iyi bireye dayanarak iyileştirmeye çalışırlar.

PSO algoritmasındaki ilk popülasyon rastgele parçacıklardan oluşmaktadır. PSO algoritması, her iterasyonda parçacıklarını güncelleyerek optimum çözümü arar. Sürü içerisindeki her bir parçacık bir cevabı temsil etmektedir. Parçacıkların sayısı PSO algoritmasında üretilen çözüm sayısına eşittir. Her parçacık, en iyi pozisyonu belirlemek için hafızasındaki o andaki ve önceki pozisyonlarını kullanır. Parçacığın en iyi pozisyonuna pbest adı verilir. Tüm parçacıkların en iyi pozisyonu gbest olarak adlandırılır. Durdurma kriteri sağlanıyor ise, PSO algoritması durdurularak çözüm olarak gbest parçacık seçilir.

## 4.2. Parçacık Sürüsü Optimizasyonu Algoritması

PSO algoritması rastgele üretilen parçacıkların popülasyonu ile başlar ve parçacıkları güncelleyerek en uygun değeri arar. PSO algoritmasındaki her bir iterasyonda parçacıklar farklı hızlarla en iyi çözüme doğru hareket ederler. Sürüdeki tüm parçacıklar, uygunluk fonksiyonuna göre değerlendirilir. Uygunluk fonksiyonu, çözümün ne kadar iyi olduğunu belirlemek için kullanılan bir değerlendirme fonksiyonudur. Arama alanındaki her bir parçacığın en iyi kendi çözümü (pbest) ve tüm parçacıkların en iyi çözümü (gbest) göre güncellenir. Bu değerler bulunduktan sonra; parçacığın hızı ve konumu sırasıyla aşağıdaki denklemlerine göre güncellenir.

$$V_i^{(k+1)} = w * V_i^{(k)} + c_1 * r_1 * (x_{i,best}^{(k)} - x_i^{(k)}) + c_2 * r_2 * (x_{gbest}^{(k)} - x_i^{(k)}) \quad (4.1)$$

$$x_i^{(k+1)} = x_i^{(k)} + V_i^{(k+1)} \quad (4.2)$$

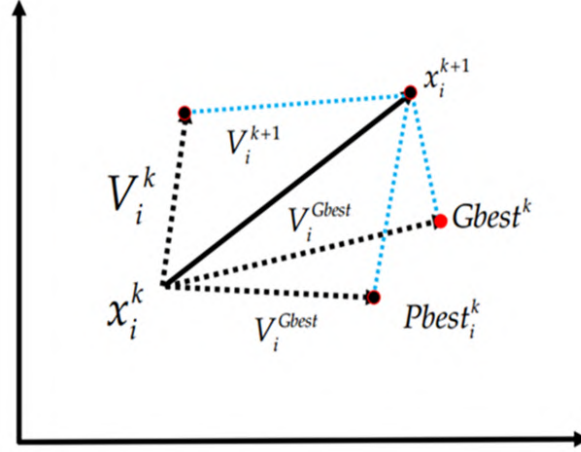
Denklem 4.1'deki  $c_1$  ve  $c_2$  sabit pozitif hızlanma katsayıları olup ve genellikle  $c_1=c_2=2$  olarak seçilir.  $r_1$  ve  $r_2$  katsayıları rastgele sayılardır ve her iterasyonda yenilenir.  $r_1$  ve  $r_2$  katsayıları 0 ile 1 aralığında değişmektedir.  $w$  ise eylemsizlik ağırlığıdır ve genellikle 0.1 ile 1 aralığında değişmektedir.

pbest ve gbest değerleri şu şekilde ifade edilebilir.

$$Pbest_i^{(k)} = x_{i,best}^{(k)} = \begin{cases} x_{i,best}^{(k-1)} & \text{if } f(x_i^{(k)}) \geq f(x_{i,best}^{(k-1)}) \\ x_i^{(k)} & \text{if } f(x_i^{(k)}) < f(x_{i,best}^{(k-1)}) \end{cases} \quad (4.3)$$

$$gbest^{(k)} = x_{gbest}^{(k)} = \min \{ f(x_{1,best}^{(k)}), f(x_{2,best}^{(k)}), \dots, f(x_{n,best}^{(k)}) \} \quad (4.4)$$

PSO'da parçacıklar (kuşlar), her bir iterasyonda çoklu arama alanındaki konumlarını değiştirirler. PSO'da arama alanındaki parçacıklar hareketi şekil 4.1'da gösterilmiştir (Allaoua et al., 2009).



Şekil 4.1. PSO parametrelerinin vektör olarak gösterimi

Burada;

$x_i^{(k)}$  : parçacığın anlık konumu.

$x_i^{(k+1)}$  : parçacığın bir sonraki konumu.

$V_i^{(k)}$  : parçacığın anlık hızı.

$V_i^{(k+1)}$  : parçacığın bir sonraki hızı.

$V_i^{(Pbest)}$  : Pbest parçacığın hızı.

$V_i^{(gbest)}$  : gbest parçacığın hızı.

Parçacık sürüsü optimizasyonu algoritması aşağıdaki adımlara göre çalışır.

**Adım 1:** Parçacıkların başlangıç konumları ( $x_i$ ) arama alanında rastgele olarak belirlenir.

**Adım 2:** Parçacıkları değerlendirmek için sürüdeki tüm parçacıkların uygunluk değerleri hesaplanır.

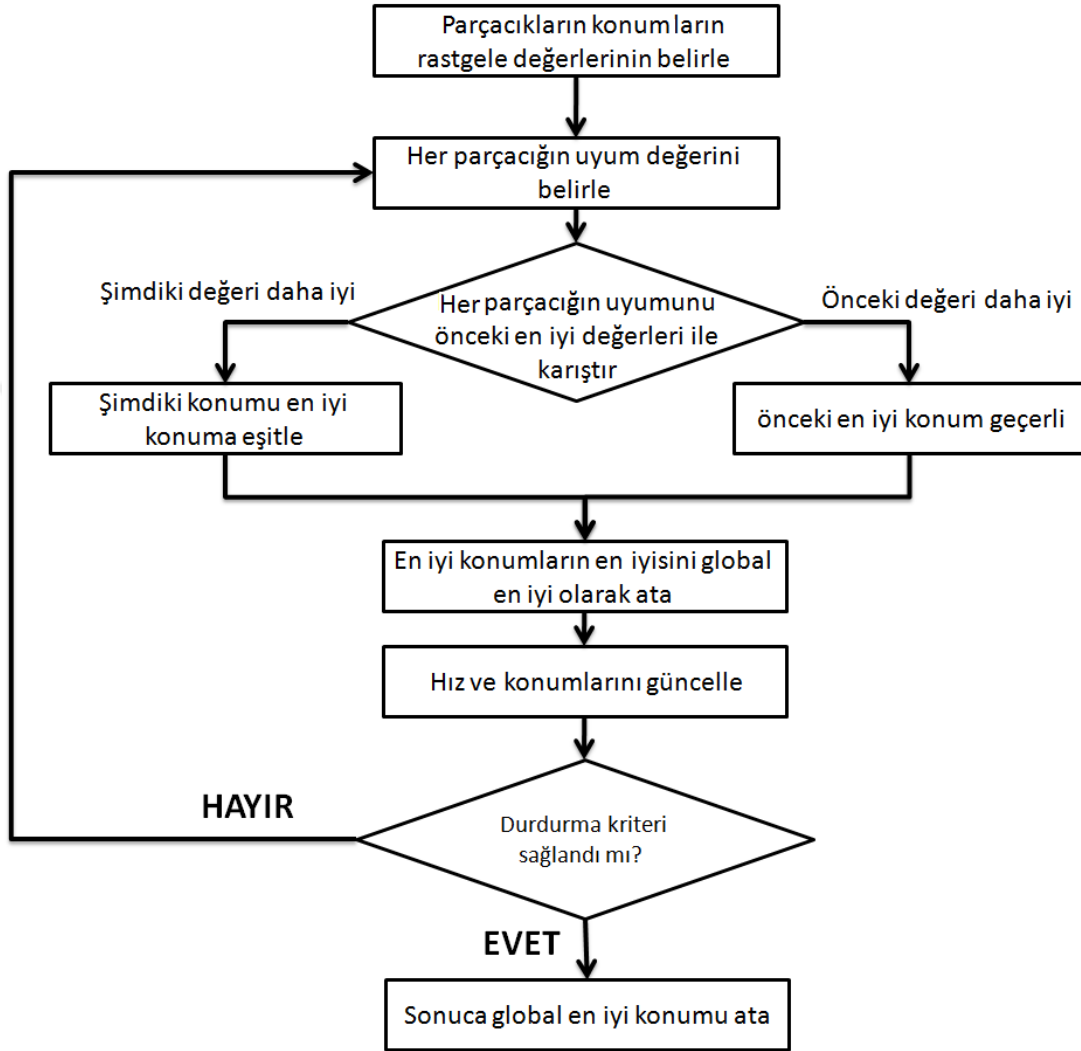
**Adım 3:** Eğer parçacık konumunun uygunluk değeri kendi yerel en iyi konumunun uygunluk değerinden daha iyi ise, parçacık konumu yerel en iyi konumu Pbest olarak güncellenir.

**Adım 4:** Yerel en iyi parçacık vektörünün en iyi uygunluk değerine sahip parçacık global en iyi parçacık gbest olarak güncellenir.

**Adım 5:** Parçacıkların hız ve konumları, denklem 4.1 ve 4.2'ye göre güncellenir.

**Adım 6:** Algoritma, durdurma kriterine ulaşıncaya kadar üçüncü adımdan itibaren tekrarlanır.

Bu ifadeler ışığında PSO'ya ait temel akış diyagramı şekil 4.2'de sunulmuştur.



Şekil 4.2. Parçacık sürüsü algoritmasının akış diyagramı

Uygunluk fonksiyonu, algoritmanın her iterasyonunda mevcut popülasyondaki parçacıkların kalitesini değerlendirmek için kullanılır. Uygunluk fonksiyonu, popülasyondaki bir parçacık ne kadar iyi olduğunu değerlendirir. Maksimum iterasyon sayısına ulaşıldığında veya uygunluk değeri istenilen seviyeye ulaştığında algoritma durdurulabilir. Evrim sonunda elde edilen global en iyi konum, problemin çözümü olarak alınır.

PSO algoritmasının Pseudo kodu aşağıda verilmiştir:

**For** her parçacık için

    rastgele bir konum değerini ata

**End**

**Do**

**For** her parçacık için uygunluk değerini hesapla

    Eğer parçacık uygunluk değeri, pbest parçacık uygunluk değerinden daha iyi ise; bu parçacığı yeni pbest parçacığı olarak ata

**End**

Pbest parçacıklarının en iyisini gbest parçacığı olarak ata

**For** her parçacık için

    (4.1) denklemine göre parçacık hızını hesapla

    (4.2) denklemine göre parçacık konumunu güncelle

**End**

**While** durdurma kriterine ulaşıncaya kadar devam et

### 4.3. Parçacık Sürüsü Optimizasyonunun Ana Parametreleri

Parçacık sürü optimizasyonu, problem boyutu, sürünün büyüklüğü, eylemsizlik ağırlığı, hızlanma katsayıları, bilişsel ve sosyal katkılarını belirleyen rastgele değerler gibi birçok kontrol parametresinden oluşmaktadır.

**Sürünün büyüklüğü:** Sürüdeki parçacıkların sayısıdır. Sürüde çok sayıda parçacığın olması her iterasyonda daha fazla sayıda çözüme ulaşılabilmesini sağlar. Ayrıca fazla sayıdaki parçacık her iterasyonda hesaplama zorluğunu artırır. Daha fazla sayıda parçacığın olması, daha az sayıdaki parçacığa göre iyi çözüme daha az iterasyonda ulaşılmasını sağlar. Sürüde en çok kullanılan parçacık sayısı genellikle 20 ve 40 parçacık arasında değişir. Aslında çoğu problem için sayıyı 10 almak kabul edilebilir çözümler elde etmek için yeterlidir. Bazı özel problemler için 40'tan daha fazla parçacık kullanılması gerekebilir.

**İterasyon sayısı:** Optimal çözümü elde etmek için gereken iterasyon sayısı probleme bağlı olarak değişmektedir. PSO algoritmasındaki düşük iterasyon sayısı iyi bir çözüm vermek için yeterli olmayabilir. Yüksek iterasyon sayısı da hesaplama zorluğunu artırır.

**Hızlanma katsayıları:** Hızlanma katsayıları  $c_1$  ve  $c_2$ , parçacıkların bilişsel ve sosyal bileşenlerini etkiler. Bilişsel bileşen, yerel arama yeteneğini kontrol eder ve sosyal bileşen, global arama yeteneğini kontrol eder.  $c_1$ , parçacıkların kendi yerel en iyi konumlarına göre hareket etmesini,  $c_2$  ise sürüdeki en iyi parçacığın konumuna göre hareket etmesini sağlar. Eğer hızlanma katsayıları sıfır olarak ayarlanırsa, parçacıklar arama alanındaki mevcut hızlarıyla hareketlerine devam ederler. Hızlanma katsayılarının düşük değerlerde seçilmesi parçacıkların yavaş yavaş hedef bölgeye doğru hareket etmesini sağlar. Ancak hızlanma katsayılarının yüksek değerlerde seçilmesi, hedefe ulaşmayı hızlandırırken, beklenmedik hareketlerin oluşmasına ve hedef bölgenin es geçilmesine neden olabilir. PSO algoritması üzerinde araştırmacıların yaptığı denemelerde " $c_1=c_2=2$ " olarak almanın iyi sonuçlar verdiği belirtilmiştir (Shi ve Eberhart, 1998).

**Eylemsizlik Ağırlığı:** PSO algoritmasında eylemsizlik ağırlığı global ve yerel arama süreci arasında denge kurmak için kullanılmaktadır. Büyük eylemsizlik ağırlığı global arama yeteneğini ve küçük eylemsizlik ağırlığı yerel arama yeteneğini artırır. Bu nedenle eylemsizlik ağırlığı, arama alanında yerel ve global araştırma arasında dengeyi sağlar ve en az sayıda iterasyonla optimum çözüme ulaşmayı amaçlar.

Eylemsizlik ağırlığı eski hız bilgisinin yeni hız bilgisine etkisini ayarlayan parametredir. Başka bir deyişle eski yöne ilişkin hafızanın yeni hızı nasıl etkileyeceğini belirler. Eylemsizlik ağırlığı  $\omega$ , hız güncelleme denkleminde önceki hız değerine çarpan olarak yazılır. Eğer eylemsizlik ağırlığı birden daha fazla ise parçacıkların hızları zamanla artar, sürü ıraksar ve parçacıklar daha iyi bölgelere doğru yön değiştirme kabiliyetini yitirirler. Eğer eylemsizlik ağırlığı sıfırdan daha az ise parçacıklar hızları sıfır oluncaya kadar yavaşlarlar.

#### 4.4. PSO Uygunluk Kriterleri

Kontrolörün performansını belirleyebilmek için bir çok değerlendirme kriteri bulunmaktadır. Bunlardan bazıları; integral of absolute error (IAE), integral of time weighted absolute error (ITAE), integral of squared error (ISE) ve integral of time weighted squared error (ITSE) gibi kriterlerdir. IAE mutlak hatayı zamanla bütünleştirir, ancak ISE hata karesini zamanla bütünleştirir. ITAE zaman ağırlıklı mutlak hatayı zamanla bütünleştirir, ancak ITSE zaman ağırlıklı hata karesini zamanla bütünleştirir.

Bu dört performans kriterinin çeşitli avantaj ve dezavantajları vardır. Örneğin IAE ve ISE kriterlerinin dezavantajı yerleşme zamanlarının uzun olmasıdır. Çünkü ISE performans kriterinde tüm hatalar zamandan bağımsızdır. ITSE performans kriteri ISE kriterinin dezavantajının üstesinden gelebilmesine rağmen analitik formülün türev işlemi zaman alıcıdır (Mitsukura, 1999; Lieslehto, 2001).

Buna göre; IAE, ISE, ITAE, ITSE performans kriterleri aşağıdaki gibi ifade edilebilir.

$$IAE = \int_0^{\infty} |r(t) - y(t)| dt = \int_0^{\infty} |e(t)| dt \quad (4.5)$$

$$ISE = \int_0^{\infty} e^2(t) dt \quad (4.6)$$

$$ITAE = \int_0^{\infty} t |e(t)| dt \quad (4.7)$$

$$ITSE = \int_0^{\infty} t \cdot e^2(t) dt \quad (4.8)$$

Verilen denklemlerde r referans, y sistem çıkışı ve e referans ve sistem çıkışı arasında hatadır.

Performans indeksleri, PSO-PID kontrolör sistemi için aşım, yerleşim zamanı, kararlı durum hatasını en aza azaltmak için kullanılır.

#### 4.5. Parçacık Sürüsü Optimizasyonu İçin Farklı Eylemsizlik Ağırlık Stratejileri

Eylemsizlik ağırlığı eski hız bilgisinin yeni hız bilgisine etkisini ayarlayan parametredir. Deneyler, eylemsizlik ağırlığı global ve yerel arama süreci arasında denge kurmak için en önemli parametre olduğunu göstermektedir. Eylemsizlik ağırlığı yüksek olduğunda, global arama yeteneği güçlüdür ancak yerel arama yeteneği düşüktür. Bunun yanında, eylemsizlik ağırlığı düşük olduğunda, yerel arama yeteneği güçlüdür ancak global arama yeteneği düşüktür.

##### 4.5.1. Sabit eylemsizlik ağırlığının stratejisi

Eberhart ve Kennedy tarafından 1995 yılında sunulan temel PSO, eylemsizlik ağırlığına sahip değildir (Eberhart ve Kennedy, 1995). 1998'de ilk kez Shi ve Eberhart sabit eylemsizlik ağırlığını getirerek eylemsizlik ağırlığı kavramını sundular (Shi ve Eberhart, 1998). Yaptıkları çalışmalarda, büyük eylemsizlik ağırlığı global arama, küçük eylemsizlik ağırlığı ise yerel arama yapmayı kolaylaştırdığını belirtmişlerdir.

$$\omega = c \quad (4.9)$$

$c$ , 0'dan 1'e değişen sabit bir sayıdır.

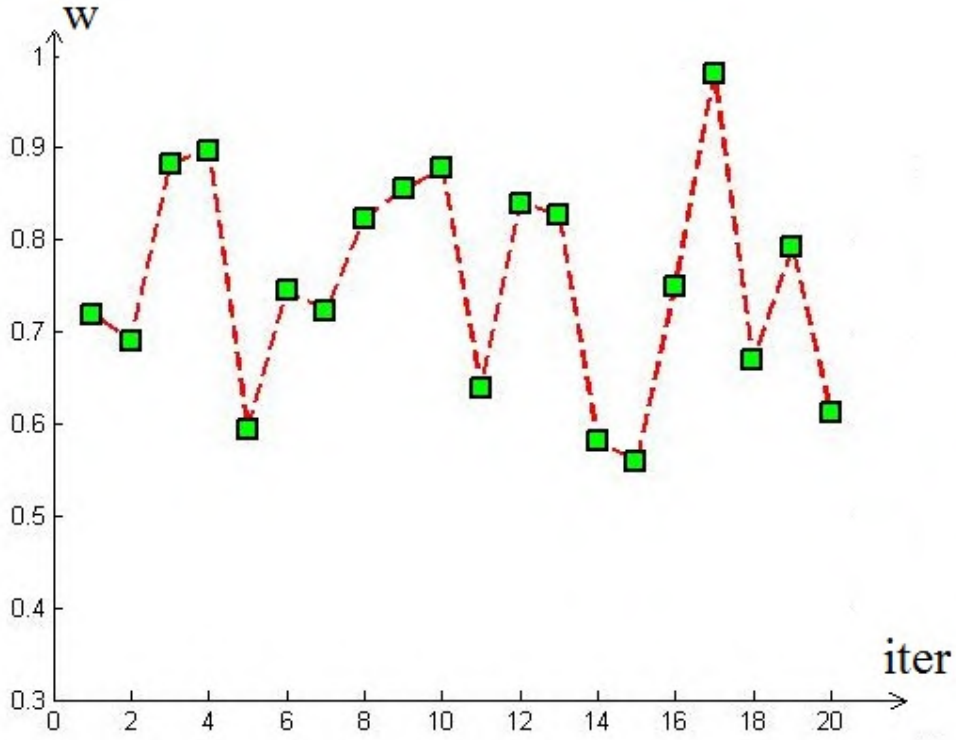
##### 4.5.2. Rastgele eylemsizlik ağırlığının stratejisi

Eberhart ve Shi 2001 yılında rastgele eylemsizlik ağırlığı stratejisini önerdiler (Eberhart ve Shi, 2001). Bu strateji erken iterasyonda PSO algoritmanın yakınsamasının hızlı bir şekilde sağlandığını buldular.

$$\omega_k = 0.5 + \frac{rand()}{2} \quad (4.10)$$

$rand()$ , 0'dan 1'e değişen rastgele bir sayıdır.  $k$  şimdiki iterasyon sayısıdır. Şekil 4.3'deki her iterasyonda eylemsizlik ağırlığı değişimlerinin rastgele eylemsizlik ağırlığının stratejisine göre değişimi göstermektedir.





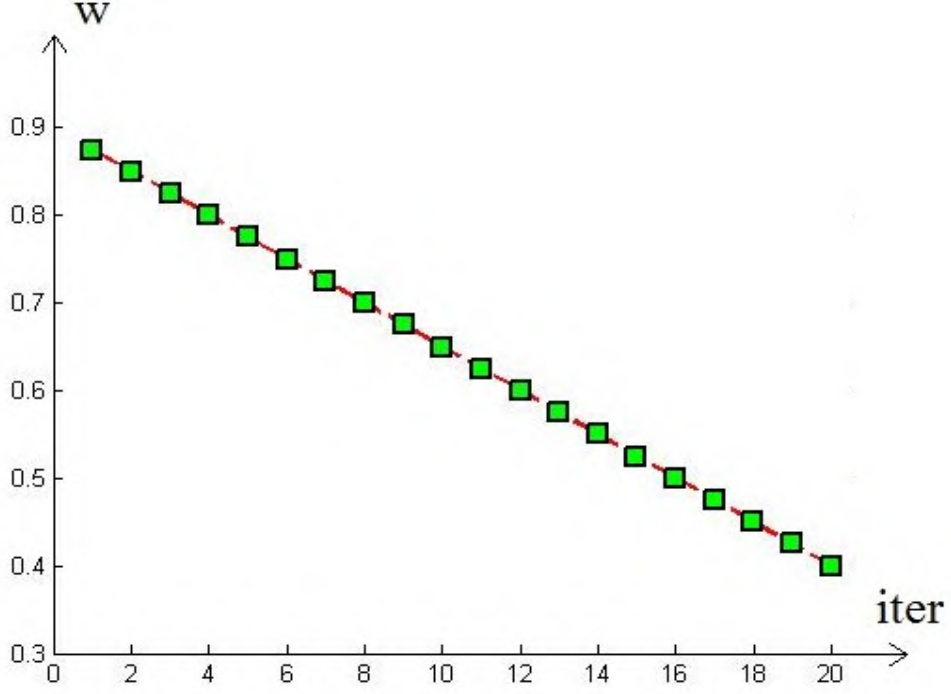
Şekil 4.3. Rastgele eylemsizlik ağırlığının stratejisine göre eylemsizlik ağırlığı değişimleri

#### 4.5.3. Eylemsizlik ağırlığının doğrusal azaltılmasının stratejisi

Eylemsizlik ağırlığının doğrusal azaltılmasının stratejisi PSO'nun verimliliğini ve performansını artırır (Xin et al., 2009). Deneysel olarak  $w_{max}=0.9$ 'dan  $w_{min}=0.4$ 'e kadar olan eylemsizlik ağırlığı mükemmel sonuçlar vermektedir. Buna göre; eylemsizlik ağırlığı aşağıdaki gibi ifade edilebilir.

$$\omega_k = \omega_{max} - (\omega_{max} - \omega_{min}) * \frac{iter}{iter_{max}} \quad (4.11)$$

$\omega_{max}, \omega_{min}$ , maksimum ve minimum eylemsizlik ağırlığı olarak tanımlanır. iter veya k şimdiki iterasyon sayısıdır.  $iter_{max}$  ise maksimum iterasyon sayısıdır. Şekil 4.4'deki her iterasyonda eylemsizlik ağırlığı değişimlerini eylemsizlik ağırlığının doğrusal azaltılmasının stratejisine göre göstermektedir.



Şekil 4.4. Eylemsizlik ağırlığının doğrusal azaltılmasının stratejisine göre eylemsizlik ağırlığı değişimleri

PSO algoritması yüksek eylemsizlik ağırlıklı başlatılır, bu nedenle global arama yeteneği güçlüdür ancak yerel arama yeteneği zayıftır. Birkaç iterasyondan sonra eylemsizlik ağırlığı azalır, bu nedenle global arama yeteneği zayıflar ve yerel arama yeteneği güçlenir. Algoritmanın sonunda eylemsizlik ağırlığı düşük olur, bu nedenle global arama yeteneği zayıf ancak yerel arama yeteneği güçlüdür.

#### 4.5.4. Sigmoid ile eylemsizlik ağırlığının azaltılmasının veya artırılmasının stratejileri

Bu strateji, PSO'nun performansını artırmak için sigmoid fonksiyon ile modüle edilmiş yeni bir eylemsizlik ağırlığı önermektedir (Maliket al., 2007). Sigmoid fonksiyonu temel olarak şu şekilde verilir:

$$f(t) = \frac{1}{1 + e^{-t}} \quad (4.12)$$

Bu stratejide eylemsizlik ağırlığı bu denklemde sunulmuştur.

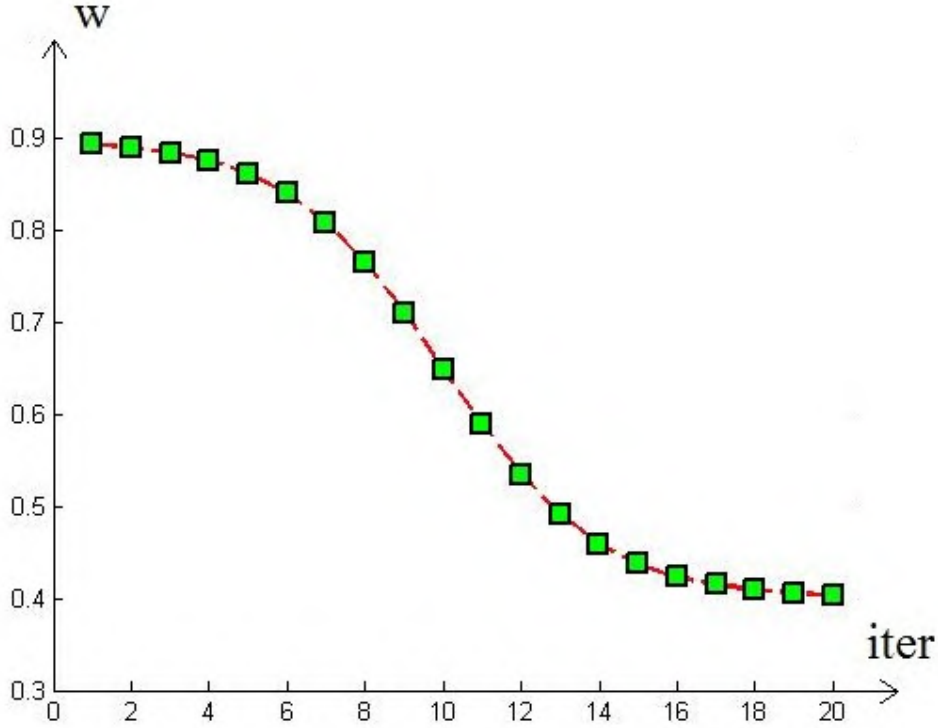
$$w_k = w_{end} + (w_{start} - w_{end}) * \left( \frac{1}{1 + e^{-u*(k-n*gen)}} \right) \quad (4.13)$$

$$w_k = w_{end} + (w_{start} - w_{end}) * \left( \frac{1}{1 + e^{u*(k-n*gen)}} \right) \quad (4.14)$$

$$u = 10^{\log(gen)-2}$$

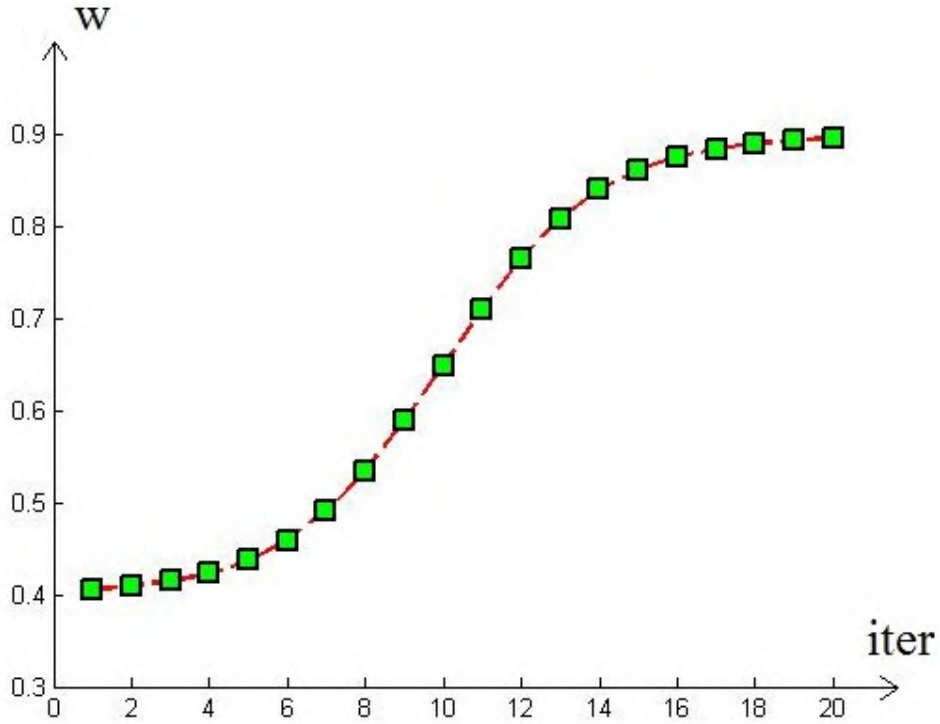
$w_{start}, w_{end}$ , başlangıç ve son eylemsizlik ağırlığının değerleridir.  $k$  ve  $gen$ , sırasıyla şimdiki ve maksimum iterasyon sayısıdır.  $u$  sigmoid fonksiyonun netliğini ayarlamak için sabittir.  $n$ , sigmoid fonksiyonunun ayırma noktasını ayarlamak için bir sabittir.

Sigmoid ile eylemsizlik ağırlığının azaltılmasının yönteminde, eylemsizlik ağırlığı denklem 4.13'e göre değişir. Şekil 4.5, her iterasyonda eylemsizlik ağırlığı değişimlerini sigmoid ile eylemsizlik ağırlığının azaltılmasının stratejisine göre göstermektedir.



Şekil 4.5. Sigmoid ile eylemsizlik ağırlığının azaltılmasının stratejisine göre eylemsizlik ağırlığı değişimleri

Sigmoid ile eylemsizlik ağırlığının artırılmasının yönteminde, eylemsizlik ağırlığı denklem 4.14'e göre değişir. Şekil 4.6, her iterasyonda eylemsizlik ağırlığı değişimlerini sigmoid ile eylemsizlik ağırlığının artırılmasının stratejisine göre göstermektedir.



Şekil 4.6. Sigmoid ile eylemsizlik ağırlığının artırılmasının stratejisine göre eylemsizlik ağırlığı değişimleri

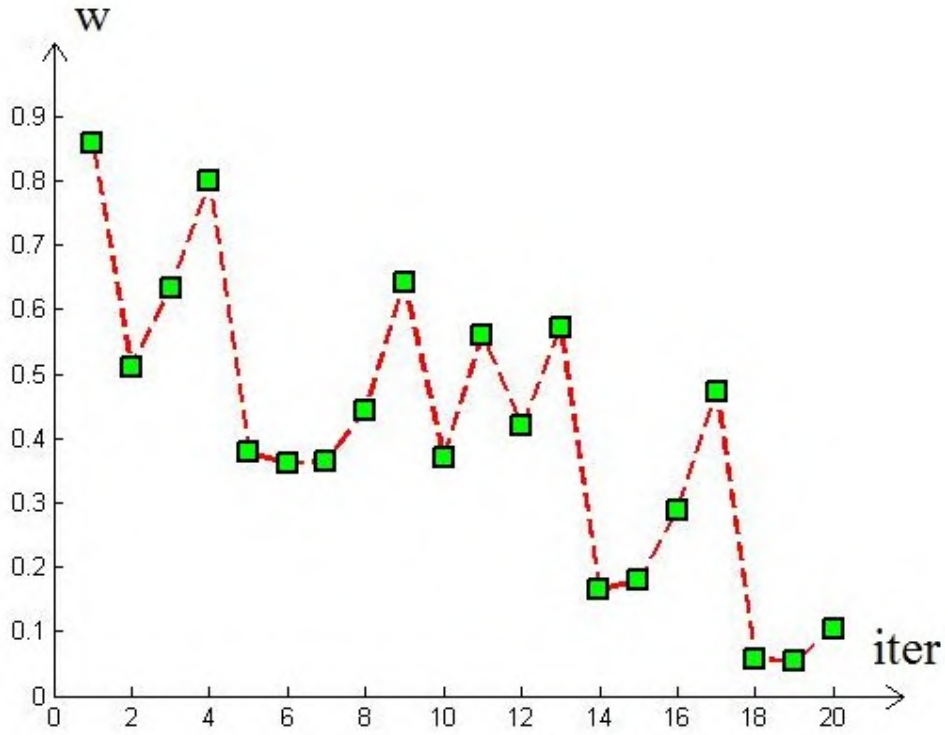
#### 4.5.5. Kaotik eylemsizlik ağırlığı stratejisi

Bu strateji, PSO algoritmasının hızlı bir yakınsama ve daha iyi global arama yeteneğine sahip olmasını sağlar (Fenget al., 2007). Eylemsizlik ağırlığı aşağıdaki denklemler ile ifade edilebilir.

$$z = 4 * z * (1 - z)$$

$$\omega = (\omega_1 - \omega_2) * \frac{iter_{max} - iter}{iter_{max}} + \omega_2 * z \quad (4.15)$$

$\omega_1, \omega_2$ , başlangıç ve son eylemsizlik ağırlığının değerleridir.  $iter$  ve  $iter_{max}$ , sırasıyla şimdiki ve maksimum iterasyon sayısıdır.  $z$ , '0' ile '1' arasında değişen rastgele bir sayıdır. Kaotik sistem kullanılarak dinamik olarak değiştirilir. Şekil 4.7'deki her iterasyonda eylemsizlik ağırlığı değişimlerini kaotik eylemsizlik ağırlığı stratejisine göre göstermektedir.



Şekil 4.7. Kaotik eylemsizlik ağırlığı stratejisine göre eylemsizlik ağırlığı değişimleri

Eylemsizlik ağırlığı denklem 4.15'e göre rastgele bir sayı olarak üretilir. Kaotik eylemsizlik ağırlığı stratejisi rastgele eylemsizlik ağırlığı stratejisi ile aynı işe yaramaktadır.

## 4.6. Adaptif Parçacık Sürüsü Optimizasyonu

Adaptif parçacık sürüsü optimizasyonu (APSO), klasik parçacık sürüsü optimizasyonundan daha iyi özelliklere sahiptir. APSO, daha hızlı yakınsama hızı ile tüm arama alanında küresel bir arama gerçekleştirebilir. APSO, arama verimliliğini ve yakınsama hızını iyileştirmek için eylemsizlik ağırlığı, hızlanma katsayıları ve diğer algoritmik parametrelerin çalışma zamanında otomatik olarak kontrol edilmesini sağlar. Bazı adaptif parçacık sürüsü optimizasyonu yöntemleri aşağıda sunulmuştur.

### 4.6.1. Parçacıkların başarısına dayalı adaptif PSO

Bu yöntemde eylemsizlik ağırlığı sürü parçacıklarının başarı oranlarının bir fonksiyonudur. Kendi yerel en iyi parçacıkların (Pbest) şimdiki uygunluk değeri önceki uygunluk değerinden daha iyi olursa, bu parçacıklar doğru yoldan hareket ediyor ve başarılıdır. Parçacıkların başarısı yüksek olduğunda eylemsizlik ağırlığı yüksek olmalıdır. Global arama yeteneği, yüksek eylemsizlik ağırlığını arttıracaktır. Parçacıkların başarısı düşük olduğunda eylemsizlik ağırlığı düşük olmalıdır. Yerel arama yeteneği, düşük eylemsizlik ağırlığını arttıracaktır. Eylemsizlik ağırlığı aşağıdaki denklemde sunulmuştur.

$$\omega(t) = (\omega_{\max} - \omega_{\min}) * P_s(t) + \omega_{\min}$$
$$P_s(t) = \frac{\sum_{i=1}^{n_s} S_i(t)}{n_s} \quad (4.16)$$
$$S_i(t) = \begin{cases} 1 & \text{if } f(x_{i,best}(t)) < f(x_{i,best}(t-1)) \\ 0 & \text{else} \end{cases}$$

$P_s$  parçacığın başarısını belirlemek için bir fonksiyonudur.  $n_s$  parçacıkların sayısıdır.  $t$  şimdiki iterasyon sayısıdır. Tüm parçacıklar başarılı olursa  $P_s$  en yüksek değeri '1' alır ve eylemsizlik ağırlığı maksimum değeri ' $w_{\max}$ ' olur. Bu durumda, parçacıklar hızlıdır ve arama alanında çok uzakta uçarlar. Tüm parçacıklar başarılı değilse,  $P_s$  en düşük değeri '0' alır ve eylemsizlik ağırlığı minimum değeri ' $w_{\min}$ ' alır. Bu durumda, parçacıklar çok yavaştır ve küçük bir alanda uçarlar.

#### 4.6.2. Global-yerel en iyi eylemsizlik ağırlığı adaptif PSO

Bu yöntemde eylemsizlik ağırlığı parçacıkların uygunluk değerlerine göre değişir (Arumugam ve Rao, 2006). Eylemsizlik ağırlığı pbest ve gbest parçacıkların uygunluk değerinin bir fonksiyonu olarak tanımlanır. Eylemsizlik ağırlığı şöyle bir ifade edilebilir.

$$\omega_i = 1.1 - \frac{F(gbest)}{F(pbest_i)} \quad (4.17)$$

F(gbest), tüm parçacıkların en iyi parçacığının uygunluk değeridir. F(pbest<sub>i</sub>) kendi yerel en iyi parçacığın uygunluk değeridir. Kendi yerel en iyi parçacığın uygunluk değeri global en iyi parçacığın uygunluk değerine çok yakınsa, eylemsizlik ağırlığı düşük olacaktır. Bu durumda, parçacıkların hızı sıfıra yakın ve oldukça küçük bir alanda uçmaktadır. Bunun tersi, kendi yerel en iyi parçacığın uygunluk değeri global en iyi parçacığın uygunluk değerinden çok büyükse, eylemsizlik ağırlığı yüksek olacaktır. Bu durumda, parçacıkların hızı yüksektir ve arama alanında daha hızlı uçmaktadır. Bu yöntemde, eylemsizlik ağırlığının otomatik olarak kontrol edilmesini sağlanmaktadır.

Eğer gbest ve pbest parçacıkları iterasyonlarla güncellenmezse, eylemsizlik ağırlığının değeri değişmez. Bu durumda eylemsizlik ağırlığının değeri sabittir ve parçacıklar sabit bir hızda hareket eder.

#### 4.7. Önerilen Adaptif Parçacık Sürü Optimizasyon Algoritması

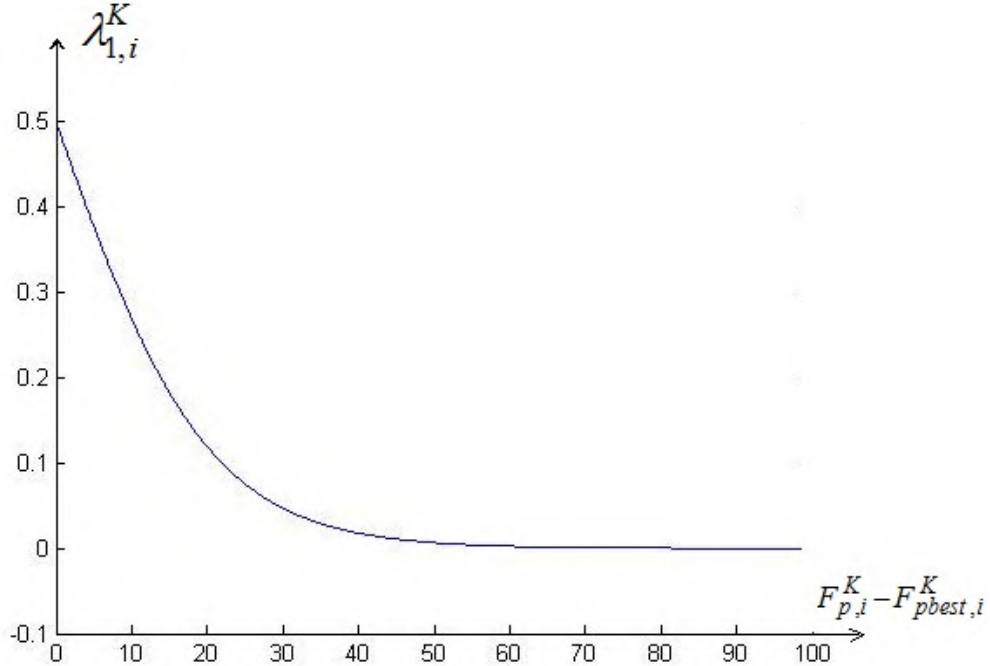
Bu çalışmada, parçacıkların uygunluk değerlerinin karşılaştırılmasına dayanılarak eylemsizlik ağırlığını ayarlamak için yeni bir strateji önerilmiştir. Birçok araştırmacı eylemsizlik ağırlığının değerinin global arama durumunda büyük olması ve yerel arama durumunda küçük olmasını tavsiye etmiştir. Eylemsizlik ağırlığının doğru bir şekilde uyarlanması PSO'nun verimliliğini artırır. Bu çalışmada, eylemsizlik ağırlığının değerinin önerilen denklem ile ifade edilebilir.

$$\lambda_{1,i}^K = \frac{1}{1 + e^{\alpha(F_{p,i}^K - F_{pbest,i}^K)}}$$

$$\lambda_{2,i}^K = \frac{1}{1 + e^{\alpha(F_{pbest,i}^K - F_{gbest}^K)}}$$

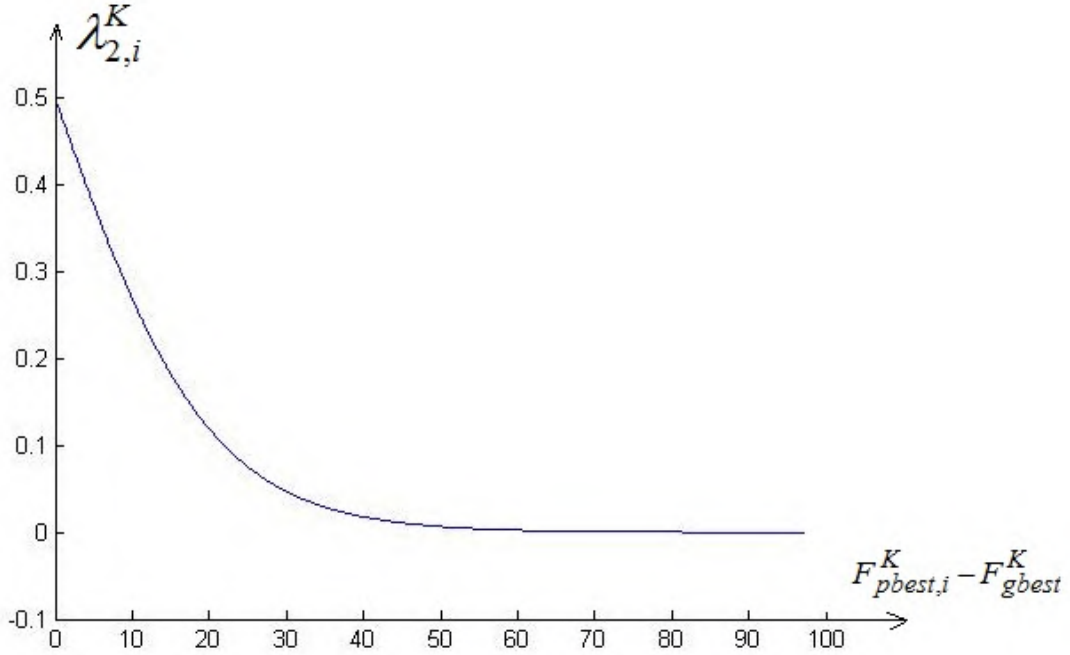
$$w_i^K = w_{\max} - (w_{\max} - w_{\min}) \times (\lambda_{1,i}^K + \lambda_{2,i}^K)$$
(4.18)

Eylemsizlik ağırlığı ( $\omega$ ), her iterasyondaki ( $K$ ) değerlerine ( $\lambda_1, \lambda_2$ ) göre ' $\omega_{\max} = 0.9$ 'dan ' $\omega_{\min} = 0.3$ 'e kadar değişir.  $F_{p,i}^K$ ,  $k$  iterasyonundaki parçacık ( $i$ )'nin uygunluk değeridir.  $F_{pbest,i}^K$ ,  $k$  iterasyonundaki yerel en iyi parçacık ( $i$ )'nin uygunluk değeridir.  $F_{gbest}^K$ ,  $k$  iterasyonundaki global en iyi parçacığın uygunluk değeridir. ( $\alpha$ ) sigmoid fonksiyonun netliğini ayarlamak için sabittir. Bu sabitin ( $\alpha < 1$ ) bu koşulu sağlamasını öneriyoruz. Bu yazıda ( $\alpha = 0.1$ ) öneririz. Şekil 4.8'da sürüdeki parçacıkların uygunluk değerleri ile Pbest parçacıklarının uygunluk değerleri arasındaki farka göre  $\lambda_1$ 'in değerini gösterir. Şekil 4.9'da Pbest parçacıklarının uygunluk değerleri gbest parçacığın uygunluk değeri arasındaki farka göre  $\lambda_2$ 'in değerini gösterir.



Şekil 4.8. Sürüdeki parçacıkların uygunluk değerleri ile Pbest parçacıklarının uygunluk değerleri arasındaki farka göre  $\lambda_1$ 'in değeri





Şekil 4.9. Pbest parçacıklarının uygunluk değerleri gbest parçacığının uygunluk değeri arasındaki farka göre  $\lambda_2$ 'in değeri

Sürüdeki parçacıklar gbest parçacığından çok uzaktaysa, parçacıkların uygunluk değerleri Pbest parçacıklarının uygunluk değerlerinden daha büyüktür ve Pbest parçacıklarının uygunluk değerleri gbest parçacığının uygunluk değerinden daha büyüktür ( $F_{p,i} > F_{pbest,i} > F_{gbest}$ ). Bu durumda  $(\lambda_1, \lambda_2)$  değerleri sıfıra yakındır. Sonuç olarak eylemsizlik ağırlığı yüksek bir değeri alır. Bu durumda, global arama yeteneğini geliştirmek için eylemsizlik ağırlığı daha yüksek bir değere ayarlanmalıdır.

Sürüdeki parçacıklar gbest parçacığına çok yakınsa, parçacıkların uygunluk değerleri Pbest parçacıklarının uygunluk değerlerine yakındır ve Pbest parçacıklarının uygunluk değerleri gbest parçacığının uygunluk değerine yakındır ( $F_{p,i} \approx F_{pbest,i} \approx F_{gbest}$ ). Bu durumda  $(\lambda_1, \lambda_2)$  değerleri (0.5)'e yakındır. Sonuç olarak eylemsizlik ağırlığı düşük bir değeri alır. Bu durumda, yerel arama yeteneğini geliştirmek için eylemsizlik ağırlığı daha düşük bir değere ayarlanmalıdır.

Bu çalışmada, parçacıkların uygunluk değerlerinin karşılaştırılmasına dayanılarak eylemsizlik ağırlığı otomatik olarak kontrol edilir.

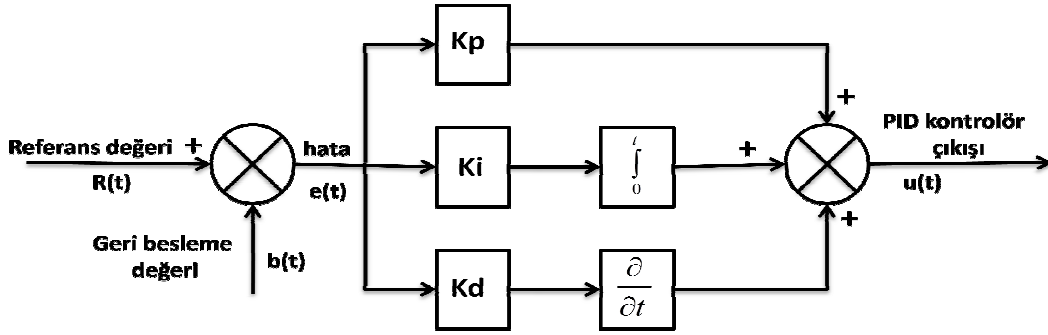
## 5. PID DENETİMİ

Tüm kontrol sistemlerinin amacı, sistemin değişkenlerini izleyerek sistemi istenen duruma getirmektir. Kontrolör görevi, sistem çıkışının değeri istenen değerden farklıysa düzeltici işlemi yapmaktır. Kontrolör giriş verileri, sistemin ölçülen çıkışı ve referans değeridir. Giriş verileri kontrolörde işlenerek kontrol edilen sisteme istenen çıkışı sağlamaktadır.

PID (Proportional-Integral-Derivative) denetim, oransal, integral ve türevsel kontrolün avantajlarını tek bir üniteye birleştirilen bir kontrolör çeşididir. PID günümüzde basit yapısı nedeniyle endüstride yaygın olarak kullanılan bir kontrol tekniğidir. Kontrol alanındaki PID kontrolör parametrelerinin ayarlanması önemli bir problemdir. PID kontrolörün ayarlanması kolay görünür, ancak kontrol sisteminin en iyi performansını sağlayan kazançları bulmak karmaşık bir görevdir. PID kontrolörünü ayarlamak için en yaygın klasik yöntemi Ziegler-Nichols'tur. Bu yöntemi uygulamak için yapılan deneyler uzun sürebilir ve optimal sonuçlarını vermez.

PID kontrolörü etkin bir şekilde kullanmak için, parametrelerinin optimum ayarı önemli bir araştırma alanı haline gelmiştir. En uygun PID parametrelerini elde etmek için bir çok optimizasyon yöntemi kullanılmaktadır. Araştırmacılar tarafından PID kontrolörü ayarlamak için genetik algoritma (GA), karınca koloni algoritması (KKA), yapay arı koloni algoritması (YAK) ve parçacık sürüsü optimizasyonu algoritması (PSO) kullanılmıştır. Bu çalışmada, PID kontrolörün ayar işlemi, adaptif parçacık sürüsü optimizasyonu (PSO) algoritması ile yapılmıştır.

PID kontrolörleri oluşturan kısımların her biri birer parametre ile yönetilirler. Bu parametreler ( $K_p$ ,  $K_d$ ,  $K_i$ ) her sistem için ayrı değerler almaktadırlar. Şekil 5.1'de PID kontrolörün iç yapısına ait blok diyagramı verilmiştir. Blok diyagramdan görüldüğü gibi PID kontrolörün yapısını oransal kazanç, integral alıcı ve türev alıcı devrelerin birleşiminden oluşmaktadır.



Şekil 5.1. PID kontrolörün blok diyagramı

Buna göre PID kontrolörün genel ifadesi aşağıdaki gibidir.

$$U(t) = K_p * e(t) + \frac{1}{T_i} \int e(t) dt + T_d * \frac{de(t)}{dt} \quad (5.1)$$

Burada  $K_p$ , oransal kazancıdır.  $T_i$ , integral zamanıdır.  $T_d$ , türev zamanıdır.  $e(t)$ , referans değeri ile çıkış arasındaki hata olarak ifade edilir. PID denetimin transfer fonksiyonu şöyle bir ifade edilebilir.

$$\frac{U(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \quad (5.2)$$

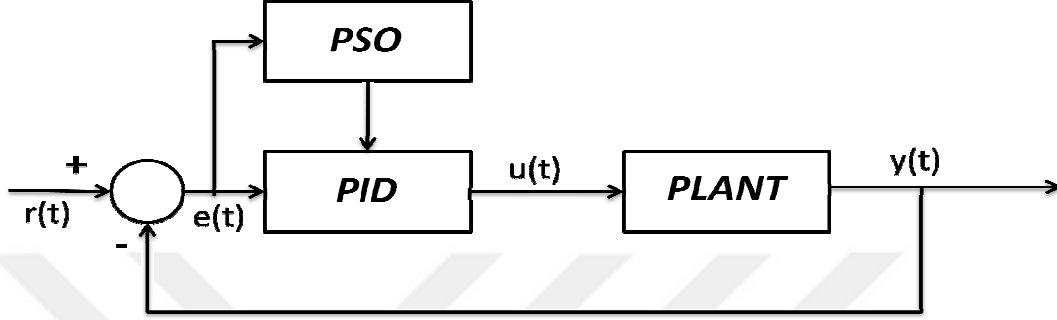
PID denetim; üç temel denetim etkisinin avantajlarını tek bir ünite de birleştiren bir denetimdir. Oransal kontrol ( $K_p$ ) aşımı artırır fakat yükselme zamanını ve kararlı hal hatasını azaltır. İntegral kontrol ( $K_i$ ) kararlı hal hatasını yok eder fakat geçici durum yanıtını kötüleştirir. Diğer bir deyişle, İntegral kontrol aşımı ve yerleşme zamanını artırır fakat yükselme zamanını azaltır. Diferansiyel kontrol ( $K_d$ ), yerleşme zamanını ve aşımı azaltır. Kapalı çevrimli bir kontrol sistemin  $K_p$ ,  $K_d$ , ve  $K_i$  katsayılarının etkileri çizelge 5.1'de detaylandırılmıştır.

Çizelge 5.1. PID katsayılarının sistem üzerine etkisi

PID	Yükselme zamanı	Aşım	Yerleşme zamanı	Kararlı hal hatası
$K_p$	Azalır	Artar	Az Değişir	Azalır
$K_i$	Azalır	Artar	Artar	Yok olur
$K_d$	Az Değişir	Azalır	Azalır	Az Değişir

## 6. SİMULASYON ÇALIŞMALARI

Bu çalışmada kullanılan PSO algoritmasının PID sistemine uygulanmasına ait blok diyagramı şekil 6.1’de verilmiştir.



Şekil 6.1. PSO-PID kontrolör blok diyagramı

$r(t)$ , referans sinyalidir veya istenen motor hızıdır.  $y(t)$ , sistemin çıkışı veya motorun gerçek hızıdır.  $e(t)$ , referans sinyali ve sistemin çıkışı arasındaki hatadır.  $u(t)$ , kontrol sinyalidir veya sistemin girişidir.

Burada plant kontrol edilmesi gereken sistemdir. PID, referans sinyali ve sistemin çıkışı arasındaki hatayı gidermek için kullanılan denetleyicidir. PSO ise çıkışın optimum değerlerini vermesini sağlayacak olan  $K_p$ ,  $K_i$  ve  $K_d$  değerlerini elde etmeyi sağlayan bir algoritma olarak ifade edilmiştir.

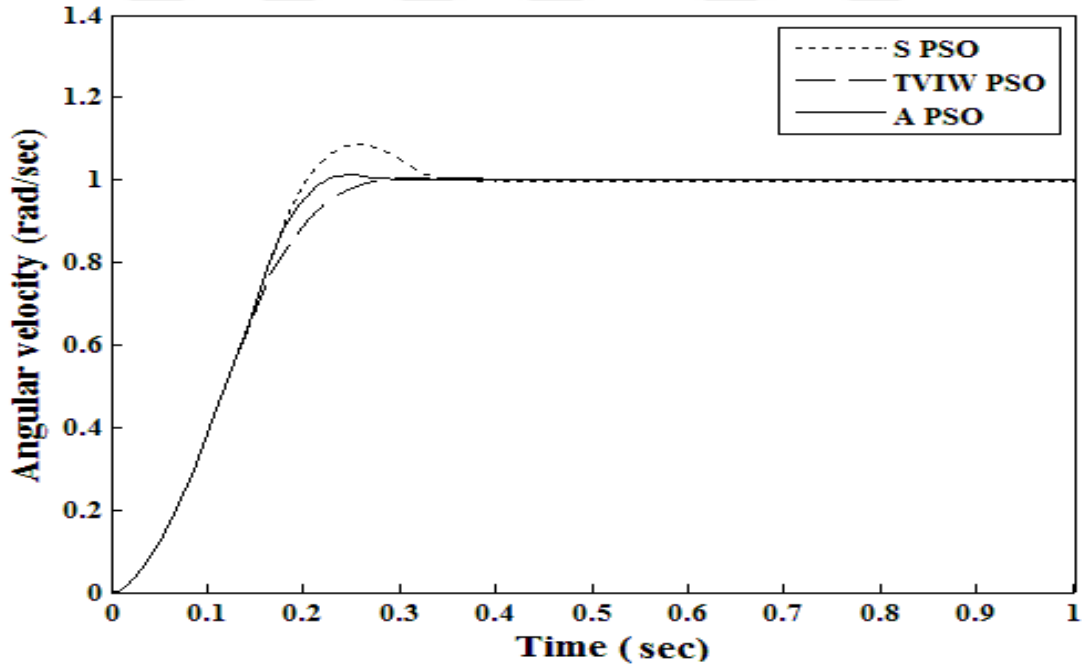
PID kontrolörün manuel ayarlanması optimum bir performans sağlamadığı gibi ayarlanması zor ve zaman alıcıdır. Bu çalışmada, PID kazançlarının ayarlanması işlemi, standart PSO algoritması (SPSO), eylemsizlik ağırlığının doğrusal azaltmasının PSO algoritması (TVIW-PSO) ve önerilen adaptif PSO algoritması (APSO) aracılığıyla gerçekleştirilmiştir.

DA motor sisteminin PID kontrolörü, MATLAB / Simulink ortamında modellenmiştir ve simüle edilmiştir. Simülasyonda kullanılan DA motor parametreleri ve sabitleri aşağıda çizelge 6.1’de verilmiştir.

Çizelge 6.1. Doğru akım DA motoru ile ilgili parametreleri.

Parametre tanımı	Sembol	Değer	Birim
Armatür direnci	R	2	$\Omega$
Armatür endüktansı	L	0.5	Henry
Motor moment katsayısı	$K_m$	0.1	
Hız volt katsayısı	$K_b$	0.1	
Sürtünme katsayısı	$K_f$	0.2	N.m.s
Atalet momenti	J	0.02	$\text{Kg.m}^2$

SPSO, TVIW-PSO ve APSO tarafından ayarlanmış PID kontrolörün hız tepkisi şekil 6.2'te gösterilmektedir.



Şekil 6.2. DA motor için üç PSO stratejilerine göre adım cevabı

Standart PSO ilk algoritmadır. Eylemsizlik ağırlığı '1' olarak ayarlanmış ve hızlanma katsayıları '2' olarak ayarlanmışlar. Maksimum iterasyon sayısı 20'dir. Standart parçacık

sürüsü optimizasyonunu kullanarak PID kazançlarını ayarlayarak, elde edilen PID kazançları şunlardır:

Çizelge 6.2. Standart PSO algoritması ile DA motorun PID değerleri

Metot	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Standart PSO algoritması	2181	-1.74	33.59

İkinci yöntemde, PID kazançlarını ayarlamak için eylemsizlik ağırlığının doğrusal azaltmasının PSO algoritması kullanılmıştır. Eylemsizlik ağırlığı denklem 4.11'e göre ayarlanmıştır. Hızlanma katsayıları '2' olarak ayarlanmıştır. Maksimum iterasyon sayısı 20'dir. Bu algoritmanın sonunda, elde edilen PID kazançları:

Çizelge 6.3. Eylemsizlik ağırlığının doğrusal azaltmasının PSO algoritması ile DA motorun PID değerleri

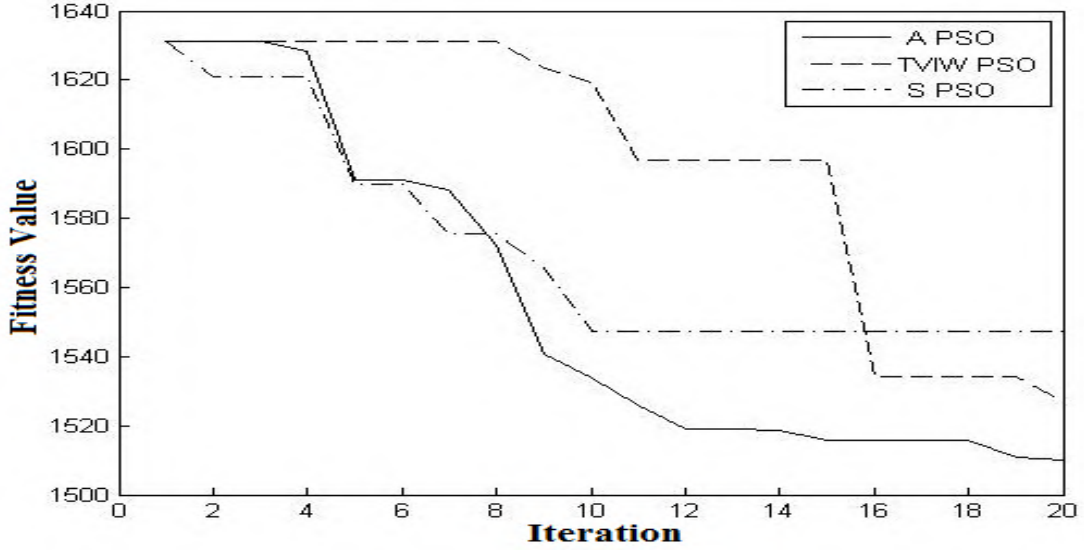
Metot	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Eylemsizlik ağırlığının doğrusal azaltmasının PSO algoritması	39.41	33.66	1.83

Son kullanılan algoritma, önerilen adaptif PSO algoritmasıdır. Eylemsizlik ağırlığı denklem 4.18'e göre ayarlanmıştır. Hızlanma katsayıları '2' olarak ayarlanmıştır. Algoritmanın sonunda, elde edilen PID kazançları:

Çizelge 6.4. Önerilen adaptif PSO algoritması ile DA motorun PID değerleri

Metot	K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>
Önerilen adaptif PSO algoritması	485.55	34	14.24

Belirli iterasyon sayısı boyunca üç PSO algoritmasının en iyi uygunluk değeri şekil 6.3'te sunulmaktadır. Bu çalışmada, kontrolörün performansını belirlemek için (IAE) değerlendirme kriterini kullanılmıştır. Denklem 4.5'de açıklanan uygunluk fonksiyonuna göre mutlak hatanın azaltılması amaçlanmıştır.



Şekil 6.3. '20' iterasyon için üç PSO algoritmasının en iyi uygunluk değeri

Standart PSO, eylemsizlik ağırlığının doğrusal azaltan PSO ve önerilen adaptif PSO algoritması kullanarak öz ayarlamalı PID kontrolör ile DA motor sisteminin açılma hızının performans endeksleri çizelge 6.5'de verilmektedir.

Çizelge 6.5. Üç farklı PSO yöntemi kullanarak öz ayarlamalı PID kontrolör ile DA motor sisteminin açılma hızının performans değerleri

Metot	Standart PSO algoritması	Eylemsizlik ağırlığının doğrusal azaltılmasının PSO algoritması	Önerilen adaptif PSO algoritması
Yükselme zamanı (s)	0.13	0.15	0.13
Yerleşme zamanı (s)	0.32	0.25	0.21
Aşım (%)	8.53	0.61	1.27
Kararlı durum hatası	0	0	0

Ek 1'deki adaptif PSO kullanarak PID kontrolörün tasarımı için matlab program kodları verilmiştir. Aşağıda matlab program sonuçları adım adım verilmiştir.

İterasyon 1'deki parçacıkların başlangıç konumları (PID kazançları) arama alanında rastgele olarak belirlenir. Parçacıkları değerlendirmek için sürüdeki tüm parçacıkların uygunluk değerleri hesaplanır. bu çalışmada denklem 4.5'de açıklanan uygunluk fonksiyonuna göre mutlak hatanın azaltılması amaçlanmıştır. Parçacıkların seçilen başlangıç konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.6'de verilmiştir.

Çizelge 6.6. 1'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 1				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	175	575	5	20.33
2	17	12	25	47.45
3	103	70	24	31.66
4	45	143	18	28.31
5	13	250	5	28.83
6	400	305	12	17.44
7	25	354	8	27.33
8	80	341	15	25.34
9	175	185	25	27.20
10	18	42	3	20.72

Eğer parçacık konumunun uygunluk değeri Pbest parçacık konumunun uygunluk değerinden daha iyi ise, parçacık konumu yerel en iyi konumu Pbest olarak güncellenir. Pbest parçacık vektörünün en iyi uygunluk değerine sahip parçacık global en iyi parçacık gbest olarak güncellenir. Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.7'de verilmiştir.

Çizelge 6.7. 1'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu

İterasyon 1					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	175	575.	5	20.33	
2	17	12	25	47.45	
3	103	70	24	31.66	
4	45	143	18	28.31	
5	13	250	5	28.83	
6	400	305	12	17.44	gbest
7	25	354	8	27.33	
8	80	341	15	25.34	
9	175	185	25	27.20	
10	18	42	3	20.72	



Parçacıkların hız ve konumları, denklem 4.1 ve 4.2'ye göre güncellenir. İterasyon 2'deki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.8'de verilmiştir.

Çizelge 6.8. 2'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 2				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	446.14	173.41	8.52	16.44
2	378.64	175.47	18.61	16.72
3	50	50	14.42	24.95
4	227.13	343.60	8.08	18.80
5	435.95	254.41	10.05	17.08
6	400	305	12	17.44
7	633.19	342.60	13.08	17.08
8	661.40	338.13	10.88	17.09
9	401.94	196.72	6.76	16.80
10	462.87	131.74	19.55	16.01

İterasyon 2'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.9'de verilmiştir.

Çizelge 6.9. 2'inci iterasyondaki Pbest parçacıkların konumları ve gbest parçacık konumu

İterasyon 2					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	446.14	173.41	8.52	16.44	
2	378.64	175.47	18.61	16.72	
3	50	50	14.42	24.95	
4	227.13	343.60	8.08	18.80	
5	435.95	254.41	10.05	17.08	
6	400	305	12	17.44	
7	633.19	342.60	13.08	17.08	
8	661.40	338.13	10.88	17.09	
9	401.94	196.72	6.76	16.8	
10	462.87	131.74	19.55	16.01	gbest

İterasyon 3'teki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.10'de verilmiştir.

Çizelge 6.10. 3'üncü iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 3				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	541.33	43.35	27.98	16.20
2	500	200.19	18.28	16.50
3	50	50	19.42	30.02
4	346.58	278.84	22.85	18.63
5	606.28	98.08	23.38	15.59
6	475.57	178.73	24.01	16.81
7	704.66	239.13	25.26	16.73
8	698.28	0.01	19.47	15.38
9	589.81	117.65	24.79	15.83
10	596.33	158.67	24.52	16.15

İterasyon 3'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.11'de verilmiştir.

Çizelge 6.11. 3'üncü iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 3					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	541.33	43.35	27.98	16.20	
2	500	200.19	18.28	16.50	
3	50	50	14.42	24.95	
4	346.58	278.84	22.85	18.63	
5	606.28	98.08	23.38	15.59	
6	475.57	178.73	24.01	16.81	
7	704.66	239.13	25.26	16.73	
8	698.28	0.01	19.47	15.38	gbest
9	589.81	117.65	24.79	15.83	
10	462.87	131.74	19.55	16.01	

İterasyon 4'teki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.12'de verilmiştir.

Çizelge 6.12. 4'üncü iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 4				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	793.83	0.01	17.70	15.51
2	500	2.36	18.57	15.5
3	50	50	21.73	31.86
4	618.04	54.44	21.82	15.54
5	779.13	0.01	23.38	15.47
6	669.43	76.88	19.10	15.48
7	724.72	175.51	24.48	16.06
8	709.35	0.01	22.05	15.42
9	705.21	49.07	25.54	15.24
10	750.02	0.01	10.48	15.70

İterasyon 4'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.13'de verilmiştir.

Çizelge 6.13. 4'üncü iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 4					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	793.83	0.01	17.70	15.51	
2	500	2.36	18.57	15.5	
3	50	50	14.42	24.95	
4	618.04	54.44	21.82	15.54	
5	779.13	0.01	23.38	15.47	
6	669.43	76.88	19.10	15.48	
7	724.72	175.51	24.48	16.06	
8	698.28	0.01	19.47	15.38	
9	705.21	49.07	25.54	15.24	gbest
10	750.02	0.01	10.48	15.70	

İterasyon 5'teki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.14'de verilmiştir.

Çizelge 6.14. 5'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 5				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	706.4	51.72	24.10	15.35
2	500	0.01	24.18	16.56
3	50	50	18.44	28.84
4	797.22	0.01	25.98	15.48
5	747.19	35.21	25.98	15.34
6	776.87	41.41	28.78	15.18
7	712.85	0.01	25.63	15.45
8	705.55	4.46	22.75	15.43
9	739.82	28.49	25.77	15.34
10	716.24	0.01	20.19	15.38

İterasyon 5'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.15'de verilmiştir.

Çizelge 6.15. 5'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 5					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	706.4	51.72	24.1	15.35	
2	500	2.36	18.57	15.5	
3	50	50	14.42	24.95	
4	797.22	0.01	25.98	15.48	
5	747.19	35.21	25.98	15.34	
6	776.87	41.41	28.78	15.18	gbest
7	712.85	0.01	25.63	15.45	
8	698.28	0.01	19.47	15.38	
9	705.21	49.07	25.54	15.24	
10	716.24	0.01	20.19	15.38	

İterasyon 6'teki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.16'de verilmiştir.

Çizelge 6.16. 6'ıncı iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 6				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	793.88	63.96	32.51	15.36
2	500	28.98	16.32	15.4
3	50	49.12	22.01	32.6
4	813.07	38.19	29.2	15.19
5	767.02	53.7	30.35	15.29
6	809.11	30.77	31.69	15.3
7	719.15	0.01	31.06	15.94
8	729.11	10.48	26.31	15.46
9	747.92	32.98	31.51	15.2
10	822.11	77.55	32.16	15.43

İterasyon 6'daki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.17'de verilmiştir.

Çizelge 6.17. 6'ıncı iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 6					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	706.4	51.72	24.1	15.35	
2	500	28.98	16.32	15.4	
3	50	50	14.42	24.95	
4	813.07	38.19	29.2	15.19	
5	767.02	53.7	30.35	15.29	
6	776.87	41.41	28.78	15.18	gbest
7	712.85	0.01	25.63	15.45	
8	698.28	0.01	19.47	15.38	
9	747.92	32.98	31.51	15.20	
10	716.24	0.01	20.19	15.38	

İterasyon 7'deki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.18'de verilmiştir.

Çizelge 6.18. 7'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 7				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	639	33.24	11.94	15.33
2	500	38.99	26.13	15.62
3	50	40.28	23.19	34.59
4	754.2	50.4	30.03	15.29
5	791.28	37.55	30.47	15.2
6	745.38	52.87	22.96	15.22
7	747.15	61.53	21.68	15.53
8	736.82	56.04	28.38	15.28
9	751.86	49.88	28.69	15.24
10	757.70	0.01	28.15	15.48

İterasyon 7'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.19'de verilmiştir.

Çizelge 6.19. 7'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 7					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	639	33.24	11.94	15.33	
2	500	28.98	16.32	15.4	
3	50	50	14.42	24.95	
4	813.07	38.19	29.2	15.19	
5	791.28	37.55	30.47	15.2	
6	776.87	41.41	28.78	15.18	gbest
7	712.85	0.01	25.63	15.45	
8	736.82	56.04	28.38	15.28	
9	747.92	32.98	31.51	15.2	
10	716.24	0.01	20.19	15.38	

İterasyon 8'deki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.20'de verilmiştir.

Çizelge 6.20. 8'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 8				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	701.77	40.2	37.55	15.99
2	500	30.19	20.73	15.27
3	50	48.05	22.14	32.44
4	759.27	31.78	26.52	15.67
5	787.7	36.87	30.37	15.18
6	846.19	36.47	23.55	15.22
7	780.88	21.94	35.77	15.85
8	785.68	49.13	29.64	15.22
9	746.15	14.1	33.5	15.77
10	686.46	0.01	15.8	15.49

İterasyon 8'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.21'de verilmiştir.

Çizelge 6.21. 8'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 8					
Pbest Parçacık	Kp	Ki	Kd	Uygunluk değeri	Gbest Parçacık
1	639	33.24	11.94	15.33	
2	500	30.19	20.73	15.27	
3	50	50	14.42	24.95	
4	813.07	38.19	29.2	15.19	
5	787.70	36.87	30.37	15.18	gbest
6	776.87	41.41	28.78	15.18	
7	712.85	0.01	25.63	15.45	
8	785.68	49.13	29.64	15.22	
9	747.92	32.98	31.51	15.2	
10	716.24	0.01	20.19	15.38	

İterasyon 9'daki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.22'de verilmiştir.

Çizelge 6.22. 9'uncu iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 9				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	789	33.18	16.82	15.36
2	500	37.68	31.49	16.47
3	50	35.39	27.38	44.02
4	772.82	40.81	31.84	15.21
5	786.62	36.67	30.34	15.17
6	718.46	36.92	29.23	15.26
7	759.45	9.67	19.31	15.4
8	802.8	38.32	31	15.19
9	755.18	60.09	30.71	15.32
10	692.67	46.71	21.1	15.18

İterasyon 9'daki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.23'de verilmiştir.

Çizelge 6.23. 9'uncu iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 9					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	639	33.24	11.94	15.33	
2	500	30.19	20.73	15.27	
3	50	50	14.42	24.95	
4	813.07	38.19	29.2	15.19	
5	786.62	36.67	30.34	15.17	gbest
6	776.87	41.41	28.78	15.18	
7	759.45	9.67	19.31	15.4	
8	802.8	38.32	31	15.19	
9	747.92	32.98	31.51	15.2	
10	692.67	46.71	21.1	15.18	



İterasyon 10'daki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.24'de verilmiştir.

Çizelge 6.24. 10'uncu iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 10				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	713.87	34.7	31.3	15.43
2	500	36.84	24.41	16.06
3	50	43.18	13.25	24.43
4	792.99	31.92	31.59	15.3
5	786.3	36.61	30.34	15.17
6	795.1	45.48	31.72	15.3
7	765.52	54.64	22.55	15.35
8	783.55	33.77	31.3	15.19
9	787.45	18.79	30.08	15.35
10	781.67	56.96	23.64	15.26

İterasyon 10'daki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.25'de verilmiştir.

Çizelge 6.25. 10'uncu iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 10					
Pbest parçacık	Kp	Ki	Kd	Uygunluk değeri	gbest parçacık
1	639	33.24	11.94	15.33	
2	500	30.19	20.73	15.27	
3	50	43.18	13.25	24.43	
4	813.07	38.19	29.20	15.19	
5	786.3	36.61	30.34	15.17	gbest
6	776.87	41.41	28.78	15.18	
7	765.52	54.64	22.55	15.35	
8	802.8	38.32	31	15.19	
9	747.92	32.98	31.51	15.2	
10	692.67	46.71	21.1	15.18	

İterasyon 20'deki yeni parçacıkların konumları ve bunların uygunluk değerleri aşağıdaki çizelge 6.26'de verilmiştir.

Çizelge 6.26. 20'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

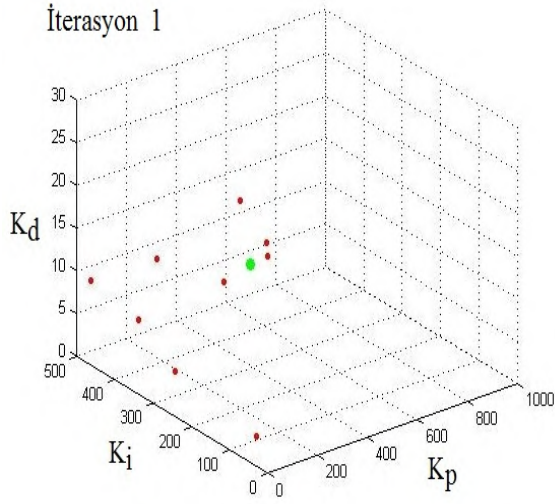
İterasyon 20				
Parçacık	Kp	Ki	Kd	Uygunluk değeri
1	795.45	39.24	27.24	15.26
2	500	44.81	16.83	15.45
3	50	35.42	39.62	53.83
4	775.61	38.97	28.67	15.13
5	797.61	38.86	27.24	15.27
6	779.43	39.05	27.62	15.12
7	759.72	37.49	27.46	15.16
8	745.07	43.01	33.58	15.49
9	774.71	40.94	26.62	15.26
10	721.24	42.77	22.74	15.31

İterasyon 20'deki yeni Pbest parçacıkların konumları ve gbest parçacık konumu aşağıdaki çizelge 6.27'de verilmiştir.

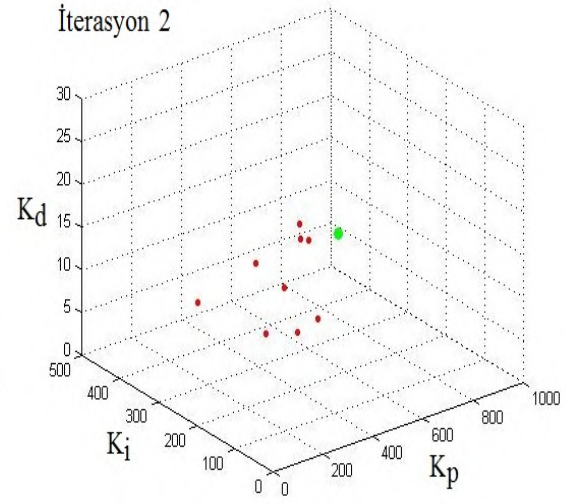
Çizelge 6.27. 20'inci iterasyondaki parçacıkların konumları ve bunların uygunluk değerleri

İterasyon 20					
Pbest Parçacık	Kp	Ki	Kd	Uygunluk değeri	Gbest Parçacık
1	795.79	39.27	28	15.1	
2	500	41.34	17.9	15.15	
3	50	36.89	0.01	16.42	
4	773.51	40.99	25.32	15.12	
5	774.92	38.88	27.25	15.08	gbest
6	779.43	39.05	27.62	15.12	
7	759.72	37.49	27.46	15.16	
8	796.68	41.36	24.1	15.13	
9	766.6	40.78	28.45	15.15	
10	692.67	46.71	21.1	15.18	

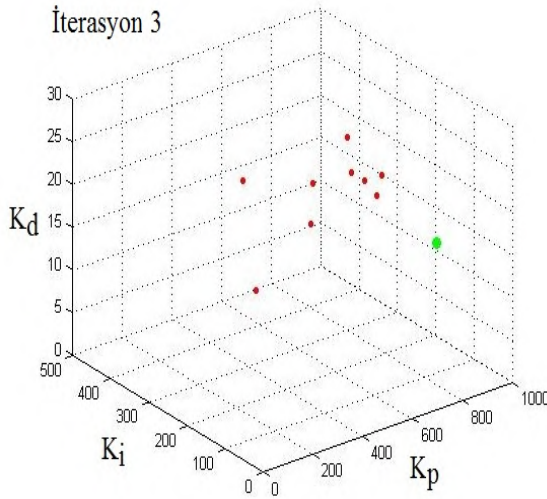
Şekil 6.4-6.14, her iterasyonda parçacıkların konumlarını göstermektedir. Şekil 6.15, DA motor için IAE kriterine göre adım cevabı göstermektedir.



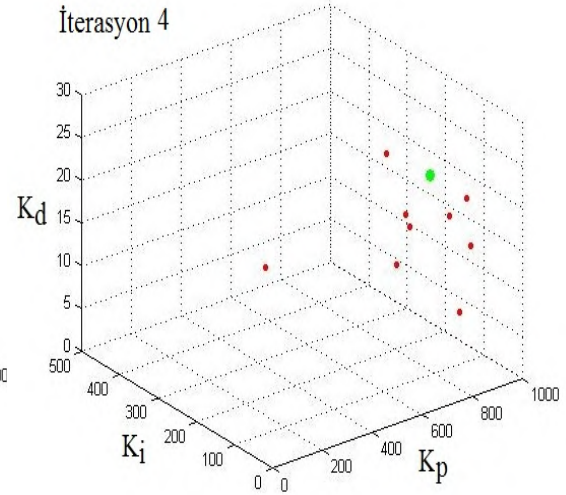
Şekil 6.4. 1'inci iterasyon sonunda parçacıkların konumları



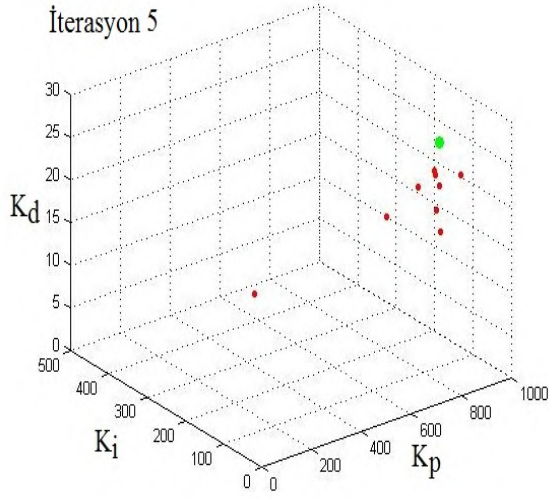
Şekil 6.5. 2'inci iterasyon sonunda parçacıkların konumları



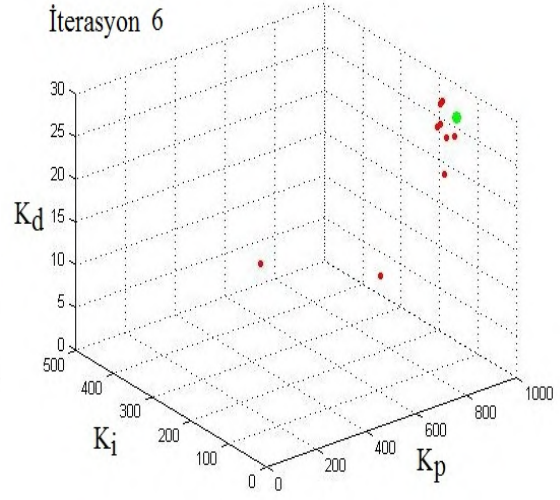
Şekil 6.6. 3'üncü iterasyon sonunda parçacıkların konumları



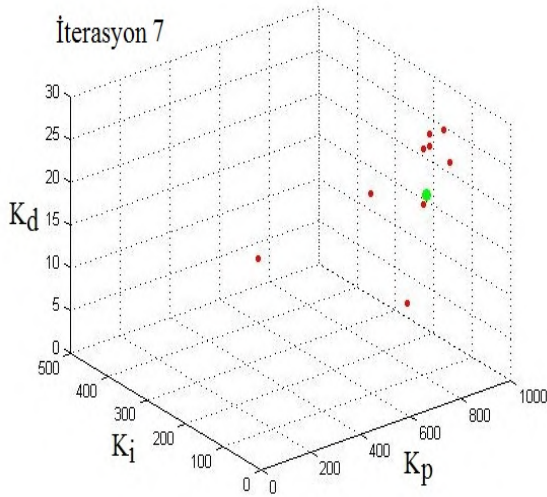
Şekil 6.7. 4'üncü iterasyon sonunda parçacıkların konumları



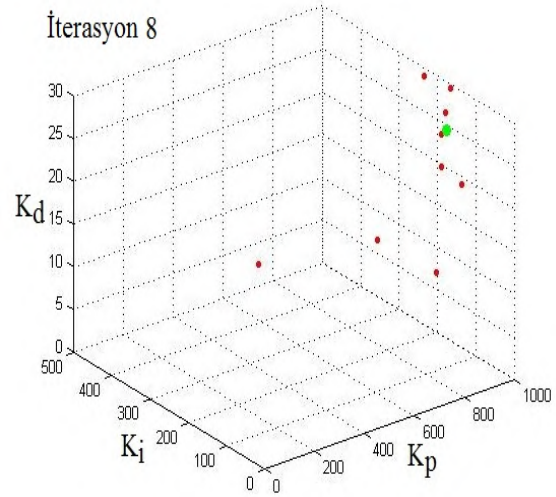
Şekil 6.8. 5'inci iterasyon sonunda parçacıkların konumları



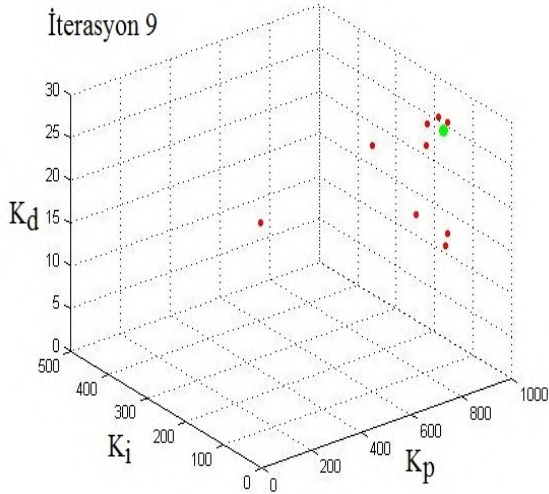
Şekil 6.9. 6'inci iterasyon sonunda parçacıkların konumları



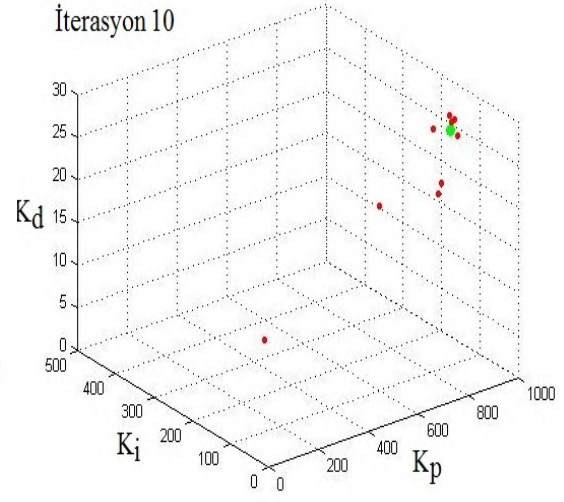
Şekil 6.10. 7'inci iterasyon sonunda parçacıkların konumları



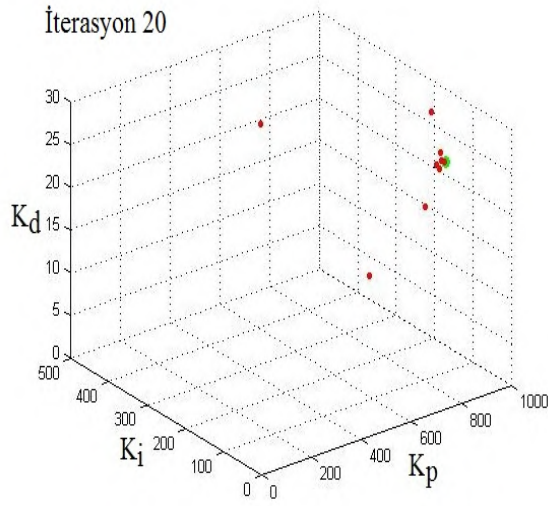
Şekil 6.11. 8'inci iterasyon sonunda parçacıkların konumları



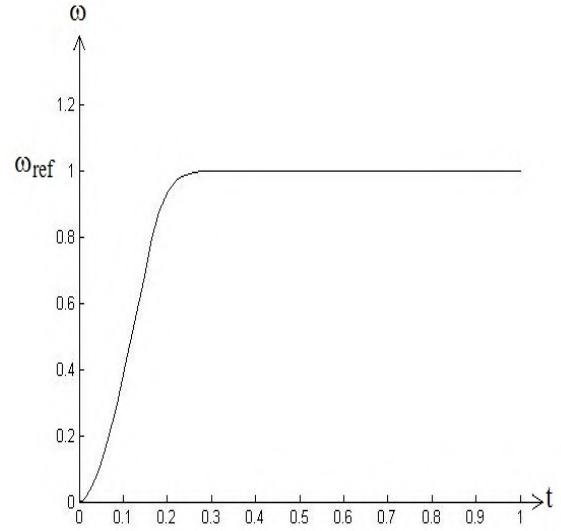
Şekil 6.12. 9'uncu iterasyon sonunda parçacıkların konumları



Şekil 6.13. 10'uncu iterasyon sonunda parçacıkların konumları



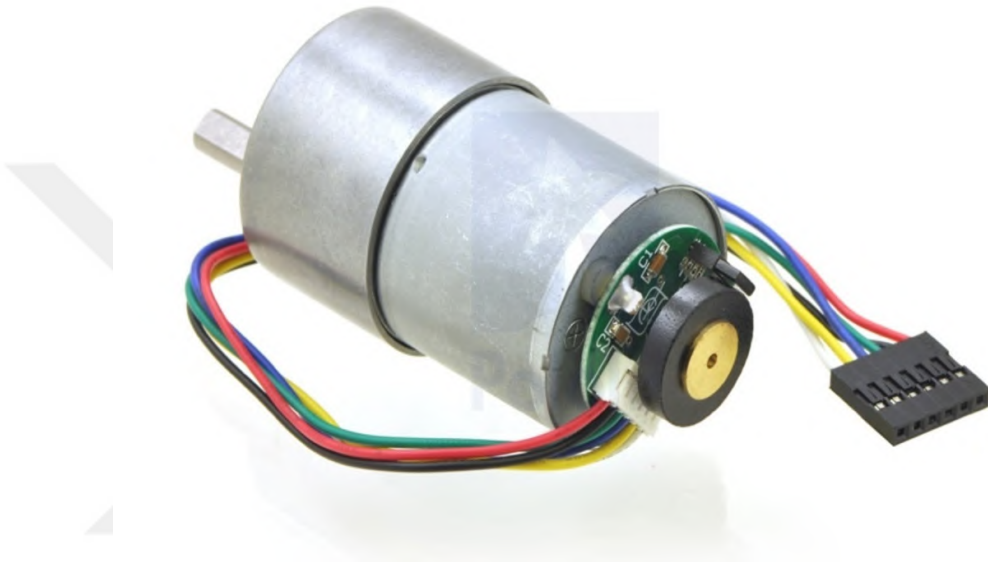
Şekil 6.14. 20'inci iterasyon sonunda parçacıkların konumları



Şekil 6.15. DA motor için IAE kriterine göre adım cevabı

## 7. DENEYSEL ÇALIŞMALARI

Bu tezde fırçalı 350 RPM (30:1) 37Dx52L mm metal dişlisi bulunan motor enkoderli aşağıdaki 7.1 şekilde görülen motor kullanılmıştır.



Şekil 7.1. Fırçalı 350 rpm (30:1) 37Dx52L mm metal dişli motor 64 CPR enkoderli

Bu motor DC 12V gerilimle çalışmakta ve motor milinde 30:1 oranında metal dişli kutusu (redüktör) bulunmaktadır. Motorun arka kısmında motor miline tutturulmuş, redüktör çıkışının her tam turunda '1920' defa pals veren enkoder bulunmaktadır. Bu motor 12V gerilimle çalışmaktadır. Genel olarak, bu tür motorlar bu nominal voltajın altında veya üstünde de çalışabilir. Düşük voltajlar uygun olmayabilir ve daha yüksek voltajlar motor ömrünü olumsuz olarak etkileyebilir. Motor 1V gibi düşük bir voltajda da dönebilir.

Motorun nominal voltajı 12V'da, serbest dönüş hızı 350 rpm, 8 kg-cm stalltorku ve 5 A stall akımı bulunmaktadır. Kullanılan metal dişlisi bulunan fırçalı doğru akım motoruna ait teknik özellikleri çizelge 7.1'de sunulmuştur.

Çizelge 7.1. Kullanılan fırçalı dişli DA motor özellikleri

Boyutlar	37D x 52L mm
Mil Çapı	6 mm
Gear ratio	30:1
Nominal Voltajı	12 V
Serbest Dönüş Hızı (6V)	175 rpm
Serbest Dönüş Akımı (6V)	250 mA
Stall Akımı (6V)	2500 mA
StallTorku (6V)	4 kg-cm
Serbest Dönüş Hızı (12V)	350 rpm
Serbest Dönüş Akımı (12V)	300 mA
Stall Akımı (12V)	5000 mA
StallTorku (12V)	8 kg-cm

Motor milinin arkada bulunan uzantısı üzerindeki manyetik diskin dönüşünü algılayan iki kanallı Hall-effect enkoder kullanılır. Quadrature enkoder, motor milinin her dönüşünde 64 sayım yapar. Dişli kutusu çıkışının dönüş sayısını bulmak için dişli oranını 64'le çarpmanız yeterlidir. Motor ve enkoderin altı adet farklı renkte kablolarının fonksiyonlarını çizelge 7.2'de sunulmuştur.

Çizelge 7.2. Motor/enkoder kablolarının fonksiyonları.

Renk	Fonksiyon
Siyah	Motor gücü
Kırmızı	Motor gücü
Mavi	Hall sensörü Vcc (3.5 – 20 V)
Yeşil	Hall sensörü GND
Sarı	Hall sensörü A çıkışı
Beyaz	Hall sensörü B çıkışı



Hall sensörü maksimum 10mA çeker ve 3.5 – 20V arası Vcc giriş voltajı ile çalışır. A ve B çıkışları 0V'tan Vcc'ye yaklaşık 90° faz farklı kare dalgalardır. Kare dalgaların frekansı motorun hızını ve kare dalgaların sırası yönü verir. Motorun hızı, denklem 7.1'de açıklanabilir.

$$RPM = \frac{Pulse * 60}{T * 60 * 30} \quad (7.1)$$

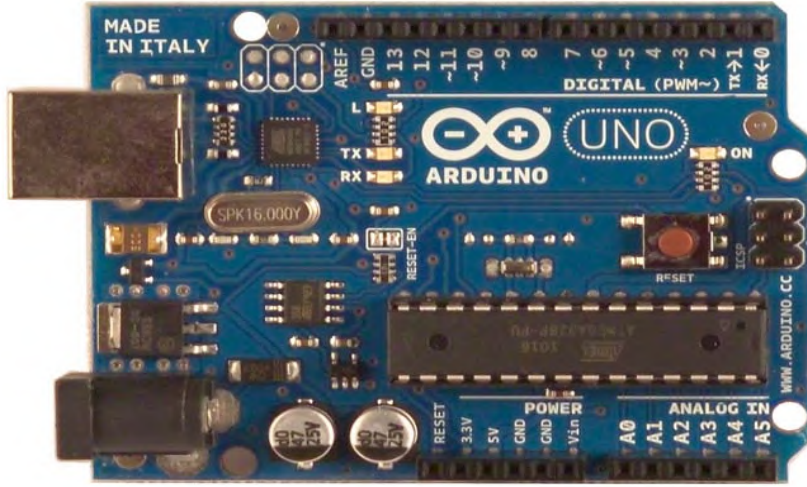
Burada RPM motorun hızı, Pulse ise iki kanal (A, B) için hem yükselen hem de düşen kenarların sayısıdır. Bu çalışmada DA motorun hızı kontrol etmek amacıyla L298N H köprülü çift motor sürücü kartı kullanılmıştır. H-köprüleri düşük akımlar ile büyük akımları iletilmesini kontrol edebilmek amacıyla kullanılan devrelerdir. Bu motor sürücü kartı ile birbirinden bağımsız olarak iki farklı DA motor kontrol edebilir. Kullanılan L298N H köprülü çift motor sürücü kartı şekil 7.2'de gösterilmiştir.



Şekil 7.2. L298N H köprülü çift motor sürücü kartı

Bu çalışmada kontrolör olarak arduino uno r3 kartı kullanılmıştır. Arduino Uno ATmega328 mikrodenetleyicisi içeren bir Arduino kartıdır. Arduino Uno'nun 14 tane dijital giriş / çıkış pini vardır. Bunlardan 6 tanesi PWM çıkışı olarak kullanılabilir. Ayrıca 6 adet analog girişi, bir adet 16 MHz kristal osilatörü, USB bağlantısı, power jakı (2.1mm), ICSP başlığı ve reset butonu bulunmaktadır. Kullanılan Arduino Uno R3 kartı şekil 7.3'da gösterilmiştir. Arduino Uno, Arduino Software (IDE) programını kullanarak programlanır.





Şekil 7.3. Kullanılan arduino uno R3 kartı

Deneyisel çalışmada, önerilen adaptif PSO algoritması arduino uno r3 kartı kullanılarak programlandı. Eylemsizlik ağırlığı denklem 4.18'e göre ayarlanmıştır. Maksimum iterasyon sayısı 10'dur. Hızlanma katsayıları '2' olarak ayarlanmıştır. Algoritmanın sonunda, elde edilen PID kazançları:

Çizelge 7.3. Deneyisel çalışmada, önerilen adaptif PSO algoritması ile DA motorun PID kazançları.

Metot	$K_p$	$K_i$	$K_d$
Önerilen adaptif PSO	0.0812	2.14	0.0036

Deneyisel çalışmada, önerilen adaptif PSO algoritması kullanarak öz ayarlamalı PID kontrolör ile DA motor sisteminin açılma hızı performans endeksleri çizelge 7.4'te verilmektedir.

Çizelge 7.4. Deneyisel çalışmada, önerilen adaptif PSO kullanarak öz ayarlamalı PID kontrolör ile DA motor sisteminin açılma hızının performans değerleri

Metot	Önerilen adaptif
Yerleşme zamanı (s)	0.3
Aşım (%)	5.66
Kararlı durum hatası	0

## 8. SONUÇLAR

Bu çalışmada hareket kontrolünün daha iyi yapılabilmesi için gerekli optimum PID parametreleri, adaptif parçacık sürüsü optimizasyonu kullanılarak tespit edilmiştir. Doğru akım motorunun transfer fonksiyonu PID denetim ile çarpılıp geri besleme yapılmıştır. Parçacık sürüsü optimizasyonu algoritması kullanarak öz ayarlamalı PID kontrolör tasarımı bu tezin temel mantığını oluşturmaktadır.

Adaptif PSO algoritması, arama verimliliğini ve yakınsama hızını iyileştirmek için eylemsizlik ağırlığı, hızlanma katsayıları ve diğer algoritmik parametrelerin çalışma zamanında otomatik olarak kontrol edilmesini sağlar. Eylemsizlik ağırlığı, global ve yerel arama süreci arasında denge kurmak için kullanılmaktadır. Bu çalışmada eylemsizlik ağırlığı, parçacıkların uygunluk değerlerinin karşılaştırılmasına dayanılarak ayarlanmıştır. Parçacıkların uygunluk değerleri, sürü durumunu belirler. Eylemsizlik ağırlığı değeri, global arama durumunda büyük olmalı ve yerel arama durumunda küçük olmalıdır.

Bu çalışmada, PID kazançlarının ayarlanması işlemi, standart PSO algoritması (SPSO), eylemsizlik ağırlığının doğrusal azaltmasının PSO algoritması (TVIW-PSO) ve önerilen adaptif PSO algoritması (APSO) aracılığıyla gerçekleştirilmiştir.

Çizelge 6.5'deki DA motorunun açılma hızının performans endeksleri, üç PSO-PID kontrolörünün yükselme zamanının birbirine çok yakın olduğunu göstermektedir. Önerilen adaptif PSO algoritması kullanan PID kontrolörü, en iyi yerleşme zamanına sahiptir. Standart PSO algoritması kullanan PID kontrolörü, maksimum aşma değerine sahiptir. APSO ve TVIW-PSO kullanan PID kontrolörleri için aşma değerleri çok düşüktür. Buna ek olarak, kararlı durum hatası tüm PID kontrolörleri için sıfırdır. Şekil 6.3'de adaptif PSO algoritmasının optimal çözüm bölgesine dolayısıyla en hızlı yakınsamaya sahip olduğu görülmektedir. Önerilen adaptif PSO algoritması kullanan PID kontrolörü, en iyi ve en düşük uygunluk değerine ulaşmıştır.

## 9. ÖNERİLER

PSO, kontrol parametrelerinin değerlerine oldukça bağımlıdır: Eylemsizlik ağırlığı ( $\omega$ ), hızlanma katsayıları ( $c_1$ ,  $c_2$ ), rastgele değerler ( $r_1$  ve  $r_2$ ). Yerel ve global arama arasındaki dengeyi artırmak için PSO kontrol parametreleri ve parçacık arama davranışına odaklanan bir çalışma yapılmalıdır. Bundan sonraki çalışmalarda önerilen adaptif PSO algoritmasının diğer algoritmalar (GA, YAK ve KKA) ile kıyaslanması hedeflenmektedir.

Eylemsizlik ağırlığı önerilen 4.18 denklemine göre ayarlanır. Denklem 4.18'deki ' $\alpha$ ' parametresinin optimum değerini bulmak için gelecekte bir çalışma yapılmalıdır. ( $\alpha$ ) parametresi sigmoid fonksiyonun netliğini ayarlamak için bir sabittir.

## KAYNAKLAR

- Dorigo, M., Maniezzo, V., Colorni, A., 1991. The Ant System: An Autocatalytic Optimizing Process. Dipartimento di Elettronica, Milano, 91-107.
- Eberhart, R. and Kennedy, J., 1995, October. A New Optimizer Using Particle Swarm Theory. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.
- Tang, K.S., Man, K.F., Kwong, S., He, Q., 1996, Genetic Algorithms And Their Applications. *Signal Processing Magazine*, 13(6), 22-37.
- Shi, Y. and Eberhart, R., 1998, May. A Modified Particle Swarm Optimizer. In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on (pp. 69-73). IEEE.
- Mitsukura, Y., Yamamoto, T., Kaneda, M., 1999. A Design Of Self-Tuning PID Controllers Using A Genetic Algorithm. American Control Conference, 2-4 Haziran, San Diego, 1361 – 1365.
- Eberhart, R.C. and Shi, Y., 2001. Tracking And Optimizing Dynamic Systems With Particle Swarms. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on (Vol. 1, pp. 94-100). IEEE.
- Lieslehto, J., 2001. PID Controller Tuning Using Evolutionary Programming. American Control Conference, 25-27 Haziran, ABD, 2828-2833.
- Karaboga, D., 2005. An İdea Based On Honey Bee Swarm For Numerical Optimization (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Arumugam, M.S. and Rao, M.V.C., 2006. On The Performance Of The Particle Swarm Optimization Algorithm With Various Inertia Weight Variants For Computing Optimal Control Of A Class Of Hybrid Systems. *Discrete Dynamics in Nature and Society*, 2006.
- Tamer, S., Karakuzu, C., 2006. Parçacık Sürüsü Optimizasyon Algoritması Ve Benzetim Örnekleri. ELECO 2006 Elektrik-Elektronik-Bilgisayar Sempozyumu, 6-10 Aralık, Bursa, 302-306.
- Chen, J., Ren, Z., Fan, X., 2006. Particle Swarm Optimization With Adaptive Mutation And Its Application Research In Tuning Of PID Parameters. *Systems and Control in Aerospace and Astronautics*, 19-21 Ocak, Çin, 994-999.
- Belgin, Ö., 2007. Haberleşme Şebekelerinin Tasarımında Sezgisel Yaklaşımlar: Değişken Komşu Arama, Kuş Sürüsü Optimizasyonu, Karınca Kolonisi Optimizasyonu. Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

- Malik, R.F., Rahman, T.A., Hashim, S.Z.M. and Ngah, R., 2007. New Particle Swarm Optimizer With Sigmoid Increasing Inertia Weight. *International Journal of Computer Science and Security*, 1(2), pp.35-44.
- Feng, Y., Teng, G.F., Wang, A.X. and Yao, Y.M., 2007, September. Chaotic Inertia Weight In Particle Swarm Optimization. In *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on* (pp. 475-475). IEEE.
- Jalilvand, A., Kimiyaghalam, A., Ashouri, A., Mahdavi, M., 2008. Advanced Particleswarm Optimization-Based PID Controller Parameters Tuning. 12th IEEE International Multitopic Conference, 23-24 Aralık, Karachi, 429-435.
- Xin, J., Chen, G. and Hai, Y., 2009, April. A Particle Swarm Optimizer With Multi-Stage Linearly-Decreasing Inertia Weight. In *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on* (Vol. 1, pp. 505-508). IEEE.
- Allaoua, B., Gasbaoui, B. and Mebarki, B., 2009. Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy. *Leonardo Electronic Journal of Practices and Technologies*, 14, pp.19-32.
- Gözde, H., Taplamacıoğlu, M.C., Kocaarslan, İ., Şenol, M.A., 2010. İki Bölgele Ara Isıtımlı Termal Güç Sisteminin Yük-Frekans Kontrolü İçin Parçacık Sürüsü Optimizasyonu Tabanlı PI-Kontrolör. *Isı Bilimi ve Tekniği Dergisi*, 30(1), 13-21.
- Özyön, S., Yaşar, C., Temurtaş, H., 2011. Parçacık Sürü Optimizasyon Algoritmasının Termik Birimlerden Oluşan Çevresel Ekonomik Güç Dağıtım Problemlerine Uygulanması. 6th International Advanced Technologies Symposium IATS'11, 16-18 Mayıs, Elazığ, 175-180.
- Yüksel, İ., 2011. Otomatik Kontrol Sistem Dinamiği Ve Denetim Sistemleri. Nobel yayınevi, Ankara, 401 s.
- Bansal, J.C., Singh, P.K., Saraswat, M., Verma, A., Jadon, S.S. and Abraham, A., 2011, October. Inertia Weight Strategies İn Particle Swarm Optimization. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on* (pp. 633-640). IEEE.

## EKLER

### Ek-1 Adaptif PSO PID Matlab Simulink Program Kodları

```
clear all
close all
open('dcPID.mdl');
model=('dcPID.mdl');
load_system(model)
set_param('dcPID','StopTime',mat2str(1));
%-----DC motor parameters -----
R = 2.0;           % Ohms
L = 0.5;          % Henrys
Km = 0.1; Kb = 0.1; % torque and back emf constants
Kf = 0.2;         % Nms
J = 0.02;         % kg.m^2/s^2
% -----PSO algorithm -----
mx_iterasyon=20 ;
% Kp   Ki   Kd
P=[175  575  5;
   17   12  25;
  103   70  24;
   45  143  18;
   13  250  5;
  400  305  12;
   25  354  8;
   80  341  15;
  175  185  25;
   18  42   3];
P_iter=zeros(10,3); Pbest_iter=zeros(10,3); gbest_iter=zeros(1,3);
Pbest=P;
V=zeros(size(P,1),size(P,2));
c1=2;c2=2;
for n=1:mx_iterasyon
    for i=1:size(P,1)
```

```

Kp=P(i,1); Ki=P(i,2); Kd=P(i,3);
set_param('dcPID/PID/Kp','Gain',mat2str(Kp));
set_param('dcPID/PID/Ki','Gain',mat2str(Ki));
set_param('dcPID/PID/Kd','Gain',mat2str(Kd));
sim(model)
Er=Error.signals.values(:,1);
Fp(i,n)=sum(abs(Er));
if isnan(Fp(i,n))
    Fp(i,n)=1e7;
end
end
%-----Update Pbest Particles-----
for i=1:size(P,1)
if n==1
    Fpbest(i,n)=Fp(i,n);
else
    if Fp(i,n)<Fpbest(i,n-1)
        Pbest(i,:)=P(i,:);
        Fpbest(i,n)=Fp(i,n);
    else
        Fpbest(i,n)=Fpbest(i,n-1);
    end
end
end
end
%-----Update gbest particle-----
for i=1:size(P,1)
if Fp(i,n)==min(Fpbest(:,n))
    gbest(1,:)=P(i,:);
    Fgbest=Fp(i,n);
    gbest_index=i;
end
end
%-----Draw particles in 3D space-----
figure(n)

```

```

S=20*ones(10,1); S(gbest_index)=50;
C = repmat([1 0 0],10,1); C(gbest_index,:)= [0 1 0];
scatter3(P(:,1),P(:,2),P(:,3),S(:),C(:,:),'filled');
axis ([0 1000 0 500 0 30])

shg

pause(0.5)
P_iter=horzcat(P_iter,P);
Pbest_iter=horzcat(Pbest_iter,Pbest);
gbest_iter=vertcat(gbest_iter,gbest);
cost(n)=Fgbest*100;
%-----Update inertia weight -----
a=0.1;wmax=0.9;wmin=0.3;
for i=1:size(P,1)
L1= 1/(1+exp((a)*(Fp(i,n)-Fpbest(i,n))));
L2= 1/(1+exp((a)*(Fpbest(i,n)-Fgbest)));
w(i,n)=wmax-(wmax-wmin)*(L1+L2);
end
%-----Update particles for next iteration-----
for i=1:size(P,1)
for j=1:size(P,2)
r1=rand(1); r2=rand(1);
V(i,j)=w(i,n)*V(i,j)+c1*r1*(Pbest(i,j)-P(i,j))+c2*r2*(gbest(j)-P(i,j));
P(i,j)=P(i,j)+V(i,j);
if P(i,j)<0
P(i,j)=0.01;
end
if P(1,j)>1000
P(1,j)=1000;
end
if P(2,j)>500
P(2,j)=500;
end
if P(3,j)>50
P(3,j)=50;

```



```

end
end
end
%----- Algorithm stopping criteria -----
if n==mx_iterasyon
    Kp=gbest(1,1)
    Ki=gbest(1,2)
    Kd=gbest(1,3)
    set_param('dcPID/PID/Kp','Gain',mat2str(Kp));
    set_param('dcPID/PID/Ki','Gain',mat2str(Ki));
    set_param('dcPID/PID/Kd','Gain',mat2str(Kd));
    sim(model)
    Er=Error.signals.values(:,1);
    Fitness=sum(abs(Er))
    figure(n+1)
    t=Output.time;
    out=Output.signals.values(:,1);
    ref=ones(size(t,1));
    hold on
    plot(t,out,'-k','LineWidth',1)
    figure(n+2)
    hold on
    n=1:mx_iterasyon;
    plot(n,cost,'-k','LineWidth',1)
    shg
break
end
end
%----- show particles 3D movement through iteration -----
for i=1:20
    figure(i); shg
    pause(0.2)
end
%----- THE END -----

```

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Adı, soyadı : Hussein ALRUİM ALHASAN  
Uyruğu : Suriye  
Doğum tarihi ve yeri : 20.11.1987, Halep  
Medeni hali : Evli  
Telefon : 0 539 391 23 10  
e-posta : husseinengine2@gmail.com

### Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Halep Üniversitesi/Elektrik-Elektronik Müh. Bölümü	2011
Lise	Isam Elnadiri Lisesi	2006

### İş Deneyimi

Yıl	Yer	Görev
2012-2013	Halep Üniversitesi/Elektrik-Elektronik Müh. Bölümü	Araştırma Görevlisi

### Yabancı Dil

İngilizce, Türkçe.

### Yayınlar

Şit, S., Kılıç, E., Özçalık, H.R., Alhasan, H.A. and Gani, A., 2016. Modelling And Performance Comparison Of PD And Takagi-Sugeno Type Fuzzy Logic Controllers For Inverted Pendulum System. *Bitlis Eren University Journal of Science and Technology*, 6(2).

ALRUİM ALHASAN, H., ATEŞ, M., BERBER, Ö., GÜNEŞ, M., 2016. Optimal Tuning of Linear Quadratic Regulators for Inverted Pendulum System Using Particle Swarm Optimization and Genetic Algorithms, *Otomatik Kontrol Ulusal Toplantısı*, pp.662-666.

BERBER, Ö., ATEŞ, M., ALRUİM ALHASAN, H., GÜNEŞ, M., 2016. Parçacık Sürü Optimizasyonu ve PID ile Mobil Robotun Optimum Yörünge Kontrolü. *Kahramanmaraş Sutcu Imam University Journal of Engineering Sciences*, 19(3), pp.165-169.

ALRUİM ALHASAN, H., GÜNEŞ, M., 2017. A New Adaptive Particle Swarm Optimization Based On Self-Tuning Of PID Controller For DC Motor System. *Çukurova University Journal of the Faculty of Engineering and Architecture*, 32(3), pp. 243-250.

**Hobiler** : Mühendislik bilimleri, Araştırma, Futbol.