



**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**YERÇEKİMSSEL ARAMA ALGORİTMASI İLE PID  
DENETLEÇ PARAMETRELERİNİN TESPİTİ**

**Semih ÇAKIR**

**Yüksek Lisans Tezi**

**Tez Danışmanı**

**Yrd. Doç. Dr. Metin KESLER**

**BİLECİK, 2012**



**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK LİSANS**

**JÜRİ ONAY FORMU**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun .../.../2012 tarih ve ..... sayılı kararıyla oluşturulan jüri tarafından .../.../2012 tarihinde tez savunma sınavı yapılan Semih ÇAKIR'ın "YERÇEKİMSSEL ARAMA ALGORİTMASI İLE PID DENETLEÇ PARAMETRELERİNİN TESPİTİ" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

### **JÜRİ**

#### **ÜYE**

**(TEZ DANIŞMANI) :** Yrd. Doç. Dr. Metin KESLER

**ÜYE:** Yrd. Doç. Dr. Cihan KARAKUZU

**ÜYE:** Yrd. Doç. Dr. Uğur YÜZGEÇ

### **ONAY**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ...../...../..... Tarih ve ...../..... sayılı kararı.

**İMZA/MÜHÜR**

## ÖZET

Doğrusal olmayan sistemlerin optimizasyon problemlerinde sezgisel optimizasyon algoritmaları oldukça sık kullanılmaktadır. Bu tez çalışmasında sezgisel optimizasyon algoritmalarından Yerçekimsel Arama Algoritması (YAA) kullanılarak Doğru Akım (DA) motor denetimi için Oransal-İntegral-Türevsel (Proportional-Integral-Derivative, PID) denetleç parametrelerinin tespiti yapılmıştır. Sezgisel algoritmaların (Tur Atan Karınca Kolonisi Algoritması (TACO), Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO) ve Diferansiyel Gelişim Algoritması (DE)) 100 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA'nın denetim başarımı, Kesler vd. (2011)'nin yaptıkları çalışmada verilen sonuçlar ile karşılaştırılmıştır. Kıyaslamasını yaptığımız YAA, birçok kategoride TACO ve GA'dan daha iyi, Parçacık Sürü Optimizasyonu (PSO) ve Diferansiyel Gelişim (DE) algoritmalarından daha düşük başarımler sergilemektedir.

### **Anahtar Kelimeler**

DA Motor, Optimizasyon, PID, Sezgisel Algoritmalar, Yerçekimsel Arama Algoritması

## **ABSTRACT**

Heuristic optimization algorithms are used widely in nonlinear optimization problems. In this thesis, Gravitational Search Algorithm (GSA), which is a heuristic optimization algorithm, is used to determine PID controller parameters for Direct Current (DC) motor control system. The heuristic algorithms (Touring Ant Colony Optimization (TACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution Algorithm (DE)) were separately run 100 times at maximum generation number in Kesler et al. (2011). Obtained average PID controller parameters are evaluated and compared with GSA's results. As a result, in many categories the performance of GSA is better than the performances of TACO and GA, but the results show that the proposed GSA's performance is lower than the performances of the PSO and DE algorithms.

### **Keywords**

DC Motor, Gravitational Search Algorithm, Heuristic Algorithms, Optimization, PID

## TEŞEKKÜR

Tez çalışmamda büyük emeđi geen, ilgi ve desteđini esirgemeyen danıřmanım, deđerli Sayın Yrd. Do. Dr. Metin KESLER hocama, alıřmalarından ve bilgilerinden yararlandığım Sayın Arř. Gör. Nesibe YALIN ve Sayın Arř. Gör. Nihan KAZAK hocalarıma ve hayatımın her safhasında bana maddi ve manevi büyük emekler vererek beni bu günlere getiren aileme teřekkürü bir bor bilirim.

Semih AKIR

## İÇİNDEKİLER

<b>ÖZET.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>TEŞEKKÜR.....</b>	<b>iii</b>
<b>İÇİNDEKİLER.....</b>	<b>iv</b>
<b>ÇİZELGELER DİZİNİ.....</b>	<b>vi</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>vii</b>
<b>SİMGELER VE KISALTMALAR DİZİNİ.....</b>	<b>ix</b>
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. SEZGİSEL ALGORİTMALAR.....</b>	<b>6</b>
2.1. Tur Atan Karınca Kolonisi Algoritması .....	7
2.2. Parçacık Sürü Optimizasyonu .....	10
2.3. Genetik Algoritma .....	12
2.4. Diferansiyel Gelişim Algoritması .....	14
2.5. Yerçekimsel Arama Algoritması .....	18
2.5.1. Newton' un hareket kanunları.....	18
2.5.1.1. Eylemsizlik yasası.....	18
2.5.1.2. İvme yasası.....	19
2.5.2. Newton' un evrensel çekim kanunu.....	19
2.5.3. YAA' nın yapısı ve çalışması.....	20
<b>3. DENETİM SİSTEMLERİ.....</b>	<b>25</b>
3.1. Denetim Sistemlerinin Türleri.....	25
3.1.1. Açık çevrim denetim sistemleri (Geri beslemesiz sistemler).....	25
3.1.2. Kapalı çevrim denetim sistemleri (Geri beslemeli sistemler).....	26
3.2. PID Denetim Sistemlerinin Tasarımı.....	27
3.2.1. P (Oransal) denetleç.....	28
3.2.2. PI denetleç.....	28
3.2.3. PD denetleç.....	29
3.2.4. PID denetleç.....	29

<b>4. YERÇEKİMSSEL ARAMA ALGORİTMASI İLE PID PARAMETRELERİNİN OPTİMİZASYONU.....</b>	<b>34</b>
<b>5. SONUÇ VE ÖNERİLER.....</b>	<b>50</b>
<b>KAYNAKLAR.....</b>	<b>51</b>
<b>ÖZGEÇMİŞ.....</b>	<b>56</b>

## ÇİZELGELER DİZİNİ

### Sayfa No

<b>Çizelge 3.1.</b>	PID parametrelerinin geçici hal parametreleri üzerine etkisi.....	31
<b>Çizelge 4.1.</b>	DA motor parametreleri.....	34
<b>Çizelge 4.2.</b>	TACO' nun 50 defa kořturulması sonucu her bir kořturmaya ait PID denetleç parametreleri $K_p$ , $K_d$ , $K_i$ ve MSE deęerleri.....	37
<b>Çizelge 4.3.</b>	GA' nın 50 defa kořturulması sonucu her bir kořturmaya ait PID denetleç parametreleri $K_p$ , $K_d$ , $K_i$ ve MSE deęerleri.....	38
<b>Çizelge 4.4.</b>	PSO' nun 50 defa kořturulması sonucu her bir kořturmaya ait PID denetleç parametreleri $K_p$ , $K_d$ , $K_i$ ve MSE deęerleri.....	39
<b>Çizelge 4.5.</b>	DE' nin 50 defa kořturulması sonucu her bir kořturmaya ait PID denetleç parametreleri $K_p$ , $K_d$ , $K_i$ ve MSE deęerleri.....	40
<b>Çizelge 4.6.</b>	YAA' nın 50 defa kořturulması sonucu her bir kořturmaya ait PID denetleç parametreleri $K_p$ , $K_d$ , $K_i$ ve MSE deęerleri.....	41
<b>Çizelge 4.7.</b>	Algoritmaların 50 kez kořturulma sonucunda başarımların kıyaslama ölçütleri.....	42
<b>Çizelge 4.8.</b>	Algoritmaların 100 kez kořturulma sonucunda başarımların kıyaslama ölçütleri.....	48



## ŞEKİLLER DİZİNİ

	<b>Sayfa No</b>
<b>Şekil 2.1.</b> Karıncanın karar verme mekanizması.....	7
<b>Şekil 2.2.</b> TACO' nun akış diyagramı.....	8
<b>Şekil 2.3.</b> Bir karınca tarafından bulunan çözüm yolu.....	9
<b>Şekil 2.4.</b> GA tarafından çözüm üretme döngüsü.....	14
<b>Şekil 2.5.</b> DE algoritması akış diyagramı.....	17
<b>Şekil 2.6.</b> Eylemsizlik yasasına göre cisimlerin farklı durumları.....	18
<b>Şekil 2.7.</b> Cisme uygulanan kuvvet sonucu hız değişimi.....	19
<b>Şekil 2.8.</b> Birbirine etki eden kütleler.....	19
<b>Şekil 2.9.</b> Kütlelerin birbirleri ile etkileşimi.....	20
<b>Şekil 2.10.</b> YAA' nın akış diyagramı.....	21
<b>Şekil 3.1.</b> Açık çevrim denetim sistemi blok diyagramı.....	26
<b>Şekil 3.2.</b> Kapalı çevrim denetim sistemi blok diyagramı.....	26
<b>Şekil 3.3.</b> $K_p$ , $K_i$ ve $K_d$ değerleri ile PID denetleç başarımı ( $K_p=100$ , $K_i=200$ , $K_d=1$ ).....	33
<b>Şekil 3.4.</b> PID denetleç ile istenen denetim başarımı ( $K_p=100$ , $K_i=200$ , $K_d=10$ )....	33
<b>Şekil 4.1.</b> DA motor hız denetimi blok diyagramı.....	34
<b>Şekil 4.2.</b> YAA ile PID denetleç optimizasyonu.....	35
<b>Şekil 4.3.</b> Algoritmaların 50 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA' nın denetim başarımı .....	43
<b>Şekil 4.4.</b> Algoritmaların 50 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA' nın denetim başarımı ( $K_p \leq 100$ ).....	43
<b>Şekil 4.5.</b> Algoritmaların 50 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA' nın denetim başarımı ( $K_p \leq 200$ ).....	44

- Şekil 4.6.** TACO' nun 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri..... 45
- Şekil 4.7.** GA' nın 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri ..... 46
- Şekil 4.8.** PSO' nun 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri ..... 46
- Şekil 4.9.** DE' nin 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri ..... 47
- Şekil 4.10.** YAA' nın 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri ..... 47
- Şekil 4.11.** Algoritmaların 100 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA' nın denetim başarımı..... 49

## SİMGELER VE KISALTMALAR DİZİNİ

<b>DA:</b>	Dođru Akım
<b>DE:</b>	Diferansiyel Gelişim Algoritması (DGA, Differential Evolution)
<b>GA:</b>	Genetik Algoritma
<b>MSE:</b>	Ortalama Karesel Hata (Mean Square Error)
<b>PID:</b>	Oransal – İntegral – Türevsel Denetleç
<b>PSO:</b>	Parçacık Sürü Optimizasyonu Algoritması
<b>TACO:</b>	Tur Atan Karınca Kolonisi Algoritması
<b>YAA:</b>	Yerçekimsel Arama Algoritması (Gravitational Search Algorithm)

## 1. GİRİŞ

Teknolojinin gelişimiyle birlikte özellikle endüstriyel alanda kullanılan denetim sistemlerinin önemi gittikçe artmaya başlamıştır. Bilgisayarlı denetim sistemlerinde, ulaşım ve güç sistemlerinde, robotik, savunma, oto montaj ve benzeri sektörlerde denetim sistemlerinin kullanım oranı oldukça fazladır. Denetim sistemlerinin karmaşık ve zor ayarlanabilir parametrelere sahip olması birtakım alternatif çözüm yollarının ortaya çıkmasını sağlamıştır. Bu çözüm yollarının başında optimizasyon gelmektedir.

Optimizasyon kelime olarak bir problemi iyileştirme, daha iyi çözümü bulma anlamı taşımaktadır. Diğer bir ifade ile optimizasyon, işgücü, hammaddeler, sermaye, süreçler, zaman, kapasite, ekipman gibi bir sistemde mevcut olan kaynakların en verimli şekilde kullanılarak belirli amaçlara (maliyetin azaltılması, kârın artırılması, kapasite kullanımının en yükseltilmesi ve verimliliğin artırılması gibi) ulaşmayı sağlayan bir süreç olarak tanımlanmaktadır (Gass, 2000).

Bir sistemi optimize etmekteki amaç, bir optimizasyon probleminin belirli kısıtlar doğrultusunda uygun şartları sağlayan parametrelerin en iyi çözümünü, en kısa sürede ya da makul bir zaman dilimi içerisinde elde edilmesini sağlamaktır. Bir problemi optimize ederken çeşitli algoritmalarından yararlanılır. Bu algoritmalar, belirli kurallara ve işleyişe göre tanımlanan, birtakım matematiksel ifadeler yardımıyla problemleri çözme yöntemleridir (Akay, 2006), yakınsama özelliğine sahiptir ve kesin çözümü değil kesin çözüme yakın en iyi çözümü garanti edebilmektedir (Pinedo, 1995; Karaboğa, 2011).

Günümüzdeki optimizasyon problemlerinin çoğu doğrusal olmayan (non-linear) bir yapıya sahiptir ve bu tür problemlerin çözülebilmesi için geliştirilmiş birçok teknik ve yaklaşım mevcuttur (Kesler vd., 2011). Özellikle çok değişkenli optimizasyon süreçlerinde, değişken sayısına ve veri tiplerine bağlı olarak problemlerin zorluk dereceleri de artabilmektedir. Bu tür problemlerin çözümü için klasik optimizasyon yöntemlerinin yanı sıra en iyiye yakın çözümler elde etmeye yarayan sezgisel (heuristic) algoritmalarda kullanılmaktadır. Bu algoritmaların bazıları şunlardır: Tur Atan Karınca Koloni Algoritması (Touring Ant Colony Algorithm, TACO), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO), Genetik Algoritmalar (Genetic Algorithms, GA), Diferansiyel (Farksal) Gelişim Algoritması (Differential Evolution

Algorithm, DE) ve Yerçekimsel Arama Algoritması (Gravitational Search Algorithm, GSA).

Denetim sistemleri, belirli bir amaca yönelik olarak kendisini veya başka bir sistemi yönlendirmek, kumanda etmek veya ayarlamak amacıyla bir araya getirilen fiziksel organlar kümesi olarak tanımlanmaktadır (Ogata, 1990). Denetim sistemleri son yarım yüzyıldır önem arz ederek gelişmiştir (Ayaz, 2011). Üç terimli denetleç PID (**P**roportional, **I**ntegral, **D**erivative), pratik uygulamalarda daha fazla kullanım alanına sahiptir (Akay, 2006; Ang vd., 2005; Ayaz, 2011; Bassi vd., 2011; Dorf ve Bishop, 2005; Li vd., 2006; Wang vd., 1999). Tercih edilmesindeki en büyük sebep, PID denetleç yapısının basit ve kararlı olmasıdır. PID denetleç parametrelerinin optimizasyonunda geleneksel yöntemler arasından en bilineni Ziegler-Nichols (Z-N) metodudur (Bassi vd., 2011; Ogata, 1990). Pratik uygulamalarda denetim sistemlerinin karmaşıklığının artması nedeniyle alternatif algoritmalar geliştirilmiştir. Çeşitli bilgisayar yazılımları desteğiyle bu algoritmaların uygulanması işlevsellik kazanmıştır (Li vd., 2006).

Literatürde PID denetleç parametrelerinin ( $K_p$ ,  $K_d$ ,  $K_i$ ) optimizasyonu için sezgisel optimizasyon algoritmalarını kullanan birçok sezgisel yöntem, matematiksel ve deneysel çalışma mevcuttur (Akay, 2006; Alli ve Kaya, 2001; Ang vd., 2005; Astrom ve Hagglund, 1984; Ayaz, 2011; Bassi vd., 2011; Cihan ve Karakuzu, 2008; Dandıl vd., 2002; Donghai vd., 2007; Dorf ve Bishop, 2005; Hsiao vd., 2004; Karaboğa ve Kalınlı, 1996; Kesler vd., 2011; Li vd., 2006; Shyr vd., 2002; Thomas ve Poongodi, 2009; Tutkun ve Dinçer, 2010; Wang vd., 1999; Yıkan, 2005). Karaboğa ve Kalınlı (1996), PID denetleç tasarımında tabu araştırma (TA) algoritmasını kullanarak yeni bir yaklaşım önermiştir. Önerilen yaklaşımın performansı, bazı geleneksel yöntemlerle karşılaştırılmış ve TA algoritmasının PID denetleç tasarımında başarılı bir şekilde kullanılabileceği gösterilmiştir.

Alli ve Kaya yaptıkları çalışmada (2001), iki dinamik sistem için PID denetleç katsayılarını sezgisel algoritmalarından biri olan genetik algoritma kullanarak optimize etmişlerdir.

Dandıl ve arkadaşları (2002) yaptıkları çalışmada, doğrusal ve doğrusal olmayan çeşitli sistemler için GA ile model referans PID denetleç tasarımını gerçekleştirmişlerdir. Arzu edilen denetim kriterlerini sağlayacak şekilde denetleyen, en

uygun PID parametrelerini bulmuşlardır. GA – PID referans modeli ile klasik yöntemlerin tersine, zor ve oldukça karışık tasarım işlemlerine gerek duymadan etkili sonuçlar elde edilebileceğini göstermişlerdir.

Shyr ve arkadaşları (2002) yaptıkları çalışmada, GA ile Tepe Tırmanma Algoritmasını (HCA) kullanarak bir memetik algoritma modeli oluşturmuşlardır. Modeli, örnek bir PID optimizasyonu için test ederek elde ettikleri sonuçları genetik algoritma ile karşılaştırmışlardır.

Hsiao ve arkadaşları (2004) yaptıkları çalışmada, verilen transfer fonksiyonu üzerinden bilinen matematiksel teknikler ile PID denetleç katsayılarının optimizasyonunu gerçekleştirmişlerdir. Karınca kolonisi algoritması (KKA) ile matematiksel tekniklerden elde ettikleri PID katsayılarının performansını karşılaştırmışlardır. KKA kullanılarak yapılan optimizasyon sırasında bulunan PID katsayılarının performansını hesaplamak için maksimum aşım, sistem oturma zamanı, yükselme zamanı ve referans ile çıkış arasındaki toplam hata değerine bakılmıştır.

Akay (2006) tez çalışmasında, öncelikle genetik algoritma ve benzetilmiş tavlama algoritmasını bir arada kullanarak bir memetik algoritma modeli gerçekleştirmiştir. PID denetleç tasarımı amacıyla memetik algoritmaya dayalı bir yaklaşım önermiştir. Önerilen yaklaşımla elde edilen sonuçlar literatürde yer alan uygulamalara ait sonuçlarla karşılaştırmış ve önerilen yaklaşımın PID denetleç tasarımında başarıyla kullanılabilceğini göstermiştir.

Donghai ve arkadaşları (2007) yaptıkları çalışmada, genetik algoritma kullanılarak PI ve PID denetleçlerin başarımını kıyaslamışlardır. Önerilen yöntem literatürde bulunan bazı örnekler üzerinde uygulanmıştır. Elde edilen benzetim sonuçları PID denetleyicinin, PI denetleyiciye göre daha iyi sonuçlar verdiğini göstermiştir.

Cihan ve Karakuzu (2008) çalışmalarında, doğrusal olmayan ters sarkaç denetim problemi için bir bulanık denetleç tasarlamışlardır. Bulanık parametrelerin belirlenmesinde karşılaşılan sorunlara çözüm üretmek için yapay zeka teknikleri içerisinde yer alan DE' nin kullanılabilceğini göstermişlerdir.

Ünal (2008) tez çalışmasında, PID denetleyiciyi KKA, GA ve Z-N metodu ile optimize etmiştir. Bu algoritmaların kullanılması ile elde edilen sonuçlar; aşım, yükselme zamanı, oturma zamanı kriterlerine ve yörünge takibindeki karekök ortalama

(Root Mean Square-RMS) hatasına göre karşılaştırılmıştır. GA ve KKA' nın performanslarının, Z-N metodununkine göre daha iyi olduğu ifade edilmiştir.

Bassi ve arkadaşları (2011) yaptıkları çalışmalarında, PID denetleç parametre tespitinde sezgisel algoritmalarından PSO algoritması kullanılması ile Z-N metodunun kullanılması ile elde edilen sonuçları karşılaştırmıştır. PSO' nun Z-N metoduna göre daha başarılı olduğu ve DA motor hız denetiminde daha etkili sonuçlar verdiği belirtilmiştir.

Duman ve arkadaşları (2011) yaptıkları çalışmalarında, YAA kullanarak PID denetleç parametrelerinin DA motor hız ve pozisyon denetimi için tespitini gerçekleştirmişlerdir. Bu kapsamda YAA ile elde ettikleri sonuçları, Z-N metodunun kullanılması ile elde edilen sonuçlarla kıyaslamışlardır. Elde edilen sonuçlara göre YAA' nın Z-N metoduna göre daha etkili ve daha verimli sonuçlar verdiği görülmüştür. Maksimum aşım, sistem oturma zamanı ve yükselme zamanı değerleri kıyaslanarak YAA' nın daha iyi performans sağladığı ifade edilmiştir.

Kesler ve arkadaşlarının (2011) çalışmalarında, güncel sezgisel algoritmalarından GA, TACO, PSO ve DE yöntemlerini kullanarak PID denetleç parametrelerinin DA motor denetimi için tespiti yapılmıştır. Elde edilen sonuçlar, merkezi eğilim ölçülerine dayalı olarak irdelenmiş ve istatistikî sonuçlar doğrultusunda gerçek zamanlı uygulamalarda DE' nin ve PSO' nun kullanımının TACO' ya ve GA' ya göre daha uygun olduğu görülmüştür.

Kazak (2011) tez çalışmasında, Yerçekimsel Arama Algoritması (YAA) kullanarak daha hızlı ve daha etkin optimizasyon algoritmalarının geliştirilmesini amaçlamıştır. Bu amaçla, nesnelere deformasyonlarını belirlemek için kullanılan kütle-yay sistemleri metodu incelenmiş ve YAA içerisine entegre edilmiştir. Bunun yanı sıra, YAA yöntemi daha fazla yüzey taraması yapacak şekilde geliştirilmiştir. Geliştirilen algoritmalar test edilerek, sonuçları YAA ile kıyaslanmıştır. Test sonuçlarına göre algoritmaların daha etkin ve verimli sonuçlar verdiği ancak daha yavaş çalıştığı belirtilmiştir.

Bu tez çalışmasında, sezgisel algoritmalarından biri olan Yerçekimsel Arama Algoritması kullanılarak PID denetleç parametrelerinin DA motor denetimi için optimizasyonu amaçlanmıştır. PID denetleç parametrelerinin optimizasyonunda diğer bazı algoritmaları kullanan Kesler vd., (2011)' nin çalışması ile başarımların kıyaslanması

yapılmıştır. Bu optimizasyon problemi için en uygun algoritma belirlenmeye çalışılmıştır.

YAA ile PID denetleç parametrelerinin optimizasyonunu anlatan bu tez çalışması beş bölümden oluşmaktadır. Birinci bölümde, konunun genel bir tanımı yapılarak bu çalışmanın önemi, amacı ve literatürde yapılmış çalışmalara yer verilmiştir.

İkinci bölümde, güncel sezgisel algoritmalar ile ilgili tanımlar yapılmış, YAA ayrıntılı olarak açıklanmıştır. Üçüncü bölümde, denetim sisteminin temel öğeleri, denetim sistemi tasarımı, PID denetleyiciler ve sistem tasarımında kullanılan klasik metotlardan bazıları tanıtılmıştır. Dördüncü bölümde, YAA kullanılarak PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve ölçüt değerlerinin tespiti yapılmıştır. Beşinci ve son bölümde ise YAA kullanılması ile elde edilen sonuçlar Kesler vd., (2011)' nin yaptıkları çalışmada elde edilen sonuçlarla kıyaslanmıştır. Tez çalışmasının kısa bir değerlendirilmesi yapılarak elde edilen sonuçlar yorumlanmıştır.



## 2. SEZGİSEL ALGORİTMALAR

Birtakım problemlerin üstesinden gelmek için uygun çözüm yollarını değerlendirerek, bu çözüm yollarından etkili olanlara karar vermek amacıyla geliştirilen algoritmalara sezgisel algoritmalar denir. Bu algoritmalar kesin çözümü değil, en iyi çözüme yakın bir çözümü garanti edebilirler ve yakınsama özelliğine sahiptirler (Pham ve Karaboğa, 2000).

1940'lı yılların sonuna doğru matematik profesörü George Polya ile sezgisel yaklaşımlı çalışmalar başlamıştır. Optimizasyon teknikleri kullanılarak yapılan çalışmalar ise 1960'lı yıllarda görülmüş ve günümüzde de devam etmektedir (Polya, 1948; Romanycia vd., 1985).

Sezgisel algoritmalar, başlangıç çözümü için en iyi çözümü kabul eder ve sonra sezgisel yöntemler kullanarak bunu geliştirmeye çalışır. Sezgisel algoritmalara gerek duyulması şu şekilde açıklanabilir (Karaboğa, 2011):

- ❖ Optimizasyon problemi, en iyi çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- ❖ Karar verici için sezgisel algoritmalar anlaşılabilirlik bakımından çok daha basit olabilir.
- ❖ Sezgisel algoritmalar, kesin çözümü elde etme ve öğrenme amaçlı kullanılabilir.
- ❖ Matematiksel ifadelerle tanımlanan gerçek dünya problemlerinde amaçların belirlenmesi, sınırlamaların oluşturulması, test edilecek alternatiflerin seçilmesi ve problem verilerinin bir araya getirilmesi amaçlı toplanma biçimi ihmal edilebilir. Herhangi bir ihmal nedeniyle model parametrelerinin belirlenmesinde karşılaşılan hata, sezgisel algoritmalarla elde edilen alt optimal çözümden daha büyük olabilir.

Sezgisel algoritmaların değerlendirilmesi için kriterler (Karaboğa, 2011):

- ❖ Herhangi bir algoritmanın etkinliğinin değerlendirilmesi için çözüm kalitesi ve hesaplama zamanı arasındaki ilişki kontrol edilebilmelidir.
- ❖ Algoritma prensipleri oldukça basit olmalı ve genel olarak uygulanabilir olmalıdır.

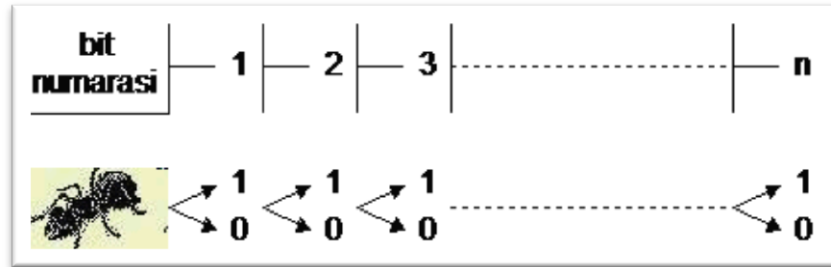
- ❖ Algoritmalar, yapılan değişiklikleri kolayca karşılayabilmelidir.
- ❖ Her zaman yüksek kaliteli ve kabul edilebilir çözümleri üretebilme kabiliyetine sahip olmalıdır.
- ❖ Algoritma kolayca analiz edilebilmelidir.
- ❖ Algoritma içinde insan-makine etkileşimi olmalıdır. İyi bir kullanıcı arayüzü kullanılarak grafiksel olarak sergilenebilmelidir.

Sezgisel algoritmalara duyulan ihtiyaç ve değerlendirme kriterleri doğrultusunda yaygın olarak kullanılan sezgisel optimizasyon algoritmalarının genel özellikleri şu şekildedir:

### 2.1. Tur Atan Karınca Kolonisi Algoritması

TACO (Touring Ant Colony Optimization) algoritması, Hiroyasu ve arkadaşları (2000) tarafından nümerik problemler için geliştirilmiş karınca koloni algoritmalarından biridir. Mühendislik alanında özellikle sürekli optimizasyon problemleri için önerilmiştir. TACO' da çözümler, ikili (binary) sayılarla temsil edilmekte ve karıncalar bu bit değerlerinin bir veya sıfır olup olmadığını araştırmaktadır. Bir ikili sayının değeri araştırılırken karıncalar yalnızca feromon maddesi bilgisini kullanır. Bir karınca, dizide bulunan ikili sayıların değeri için karar verdiğinde problem için çözüm üretmiş olur. Bu üretilen çözüm, problem için değerlendirilir ve uygunluk fonksiyonu ile çözüme dair bir ölçüt değeri elde edilir (Hsiao vd., 2004; Karaboğa, 2011).

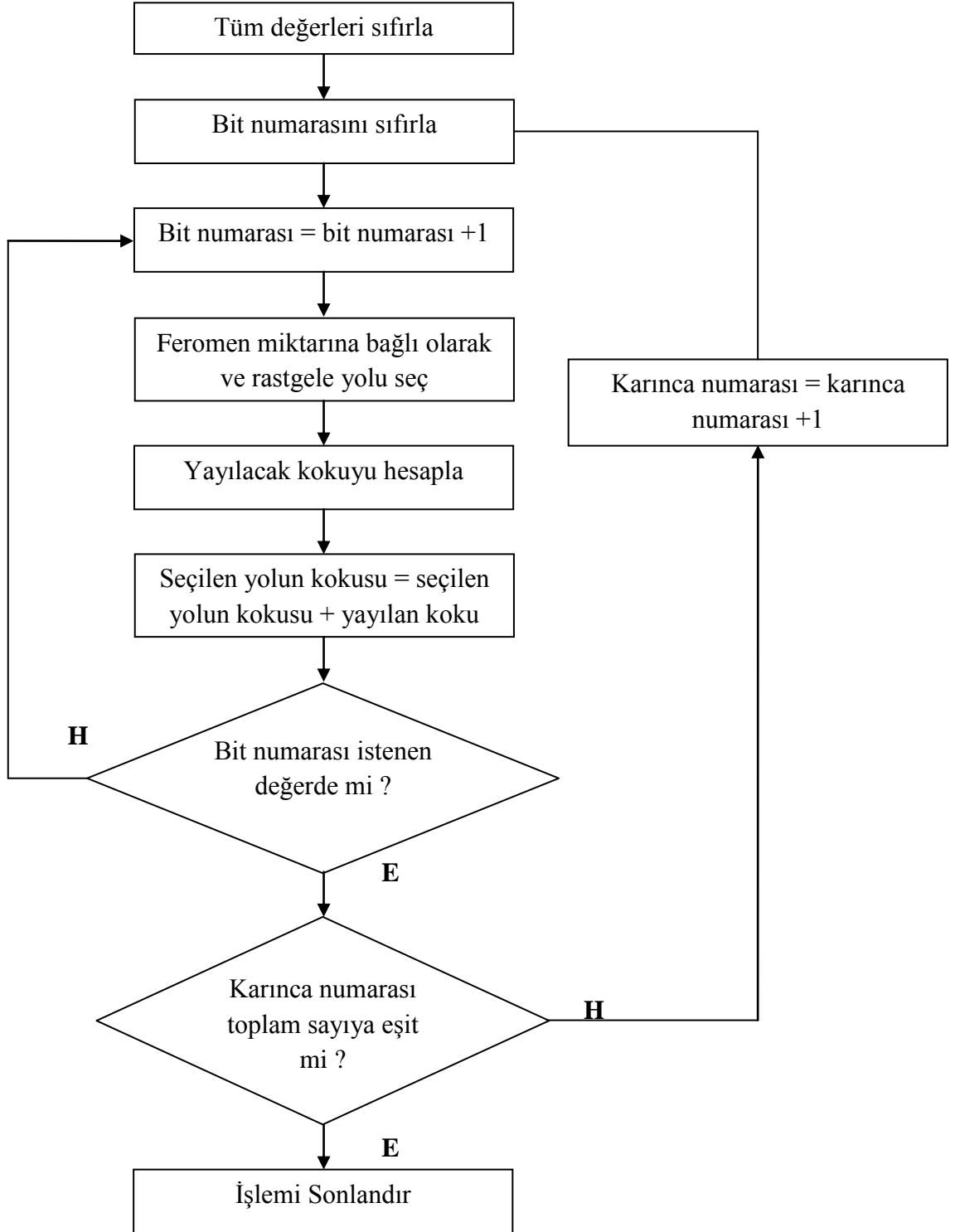
Ölçüt değerine göre karıncanın geçmiş olduğu yollara bıraktığı yapay feromon maddesi miktarı ikili sayılar arasındaki alt yollara yapıştırılır (Karaboğa, 2011; Karaboğa vd., 2002; Kesler vd., 2011).



Şekil 2.1. Karıncanın karar verme mekanizması (Saynak, 2002).

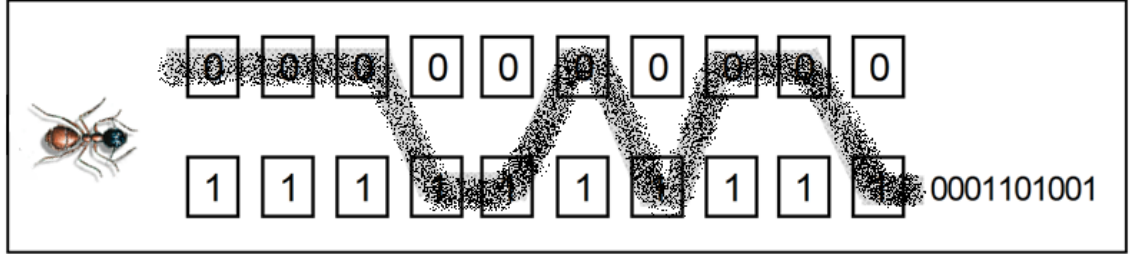
Şekil 2.1' de görüldüğü gibi TACO yönteminde karınca n adet seçim yapar, her seçim sonrası bir önceki seçtiği yolların ikili sistemdeki değerliklerini onluk sisteme

dönüştürerek feromen miktarını hesaplar ve seçtiği yola feromen bırakır. En basit haliyle karınca kolonilerinin davranışını modelleyen algoritmanın adımları Şekil 2.2' de gösterilmiştir.



Şekil 2.2. TACO' nun akış diyagramı (Saynak, 2002).

Şekil 2.3' de bir karınca tarafından oluşturulan temsili çözüm yolu verilmektedir.



Şekil 2.3. Bir karınca tarafından bulunan çözüm yolu (Kesler vd., 2011).

E. 2.1' de karıncanın (0 → 1) yolunu tercih etme olasılığı verilmektedir. E. 2.2' de ise seçilen yola karınca tarafından eklenen yapay feromen miktarı verilmektedir.

$$P_{01} = \frac{\tau_{01}}{\tau_{01} + \tau_{00}} \quad (\text{E.2.1})$$

$$\Delta\tau_{01}^k(t, t+1) = \begin{cases} \frac{Q}{C_k} & \text{eğer karınca } (0 \rightarrow 1) \text{ seçerse} \\ 0 & \text{diğer durum} \end{cases} \quad (\text{E.2.2})$$

Burada  $\Delta\tau_{01}^k$  değeri (0 → 1) yoluna eklenen yapay feromen miktarını, Q pozitif sabit sayıyı,  $C_k$  ise karınca tarafından hesaplanan çözümün ölçüt değerini ifade etmektedir. M tane karıncanın çözüm üretmesi sonrasında t ve t+1 zamanları arasında (0 → 1) yolunda ki toplam feromen miktarının bulunması E. 2.3' de tanımlanmıştır. E. 2.4' de alt yollardaki t+1 anındaki feromen miktarı hesaplanmaktadır. Burada,  $\rho$  buharlaşma katsayısıdır ve  $1 - \rho$  feromenin buharlaşma miktarını ifade eder.

$$\Delta\tau_{01}(t, t+1) = \sum_{k=1}^M \Delta\tau_{01}^k(t, t+1) \quad (\text{E.2.3})$$

$$\tau_{01}(t+1) = \rho\tau_{01}(t) + \Delta\tau_{01}(t, t+1) \quad (\text{E.2.4})$$

## 2.2. Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), balık ve kuş sürülerinin yiyecek bulma anındaki davranışlarından esinlenerek Kennedy ve Eberhart (1995) tarafından geliştirilen popülasyon temelli sezgisel yöntemlerden biridir.

Algoritma, arama uzayında rastgele atanmış parçacıklarla başlatılır. Bu parçacıkların her biri bireysel bir çözüme karşılık gelir. Dolayısıyla parçacıkların tamamı alternatif çözümler kümesini oluşturur. Parçacıklar arama uzayında bir hareket içerisinde, her bir parçacığın konumu kendi ve diğerlerinin pozisyonlarına göre ayarlanmaktadır. Arama uzayında sürü en iyi parçacığı izleyerek uçuşunu ve araştırmasını sürdürür (Bağış, 2009; Kesler vd., 2011; Yalçın, 2012).

Her bir parçacık bir hafızaya sahiptir ve uygunluk fonksiyonundan elde edilen değerler ile parçacığın hafızasında var olan değerler karşılaştırılır. Karşılaştırma sonrasında en iyi değer elde edilir. Parçacığın o ana kadar elde ettiği en iyi uygunluğa sahip parçacık pbest olarak ifade edilir. Arama uzayındaki bütün parçacıkların o ana kadar elde ettiği küresel en iyi uygunluğa sahip parçacık ise gbest olarak işaretlenir. Her jenerasyonda bu pbest ve gbest parçacık değerlerine bakılarak parçacıkların hızları belirlenir. Arama uzayında buldukları konumlar güncellenir ve sonlandırma kriteri sağlandığı takdirde en iyi çözüm elde edilir (Bağış, 2009; Yalçın, 2012).

E. 2.5 ve E. 2.6' da sürüde bulunan parçacıklar ve parçacıkların hızları d-boyutlu dizi ile gösterilmektedir. N, sürüde bulunan parçacık sayısını belirtmektedir (Yalçın, 2012).

$$x_i = (x_{i1} \ x_{i2} \ x_{i3} \ \dots \ x_{id}) \quad i= 1, 2, 3, 4 \dots N \quad (\text{E.2.5})$$

$$v_i = (v_{i1} \ v_{i2} \ v_{i3} \ \dots \ v_{id}) \quad i= 1, 2, 3, 4 \dots N \quad (\text{E.2.6})$$

Eşitliklere göre d-boyutlu arama uzayında akış içerisinde bulunan N tane parçacığın konumları E. 2.7 ve hızları E. 2.8 ile gösterilir:

$$x = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1d} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{Nd} \end{bmatrix} \quad (\text{E.2.7})$$

$$V = \begin{bmatrix} V_{11} & V_{12} & V_{13} & \dots & V_{1d} \\ V_{21} & V_{22} & V_{23} & \dots & V_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ V_{N1} & V_{N2} & V_{N3} & \dots & V_{Nd} \end{bmatrix} \quad (\text{E.2.8})$$

E. 2.9 ve E. 2.10 ile her parçacığın hız ve konumu güncellenir. Eşitliklerdeki;  $k$ , jenerasyon sayısını,  $r_1$  ve  $r_2$  değerleri  $[0,1]$  aralığında rastgele sayıları,  $c_1$  ve  $c_2$  değerleri öğrenme sabitlerini,  $v_{ij}$ ,  $i$ . parçacığın  $j$ . boyuttaki hızını,  $x_{ij}$ , ise parçacığın konumunu ifade etmektedir (Gözde vd., 2010; Yalçın, 2012).

$$V_{ij}^{k+1} = v_{ij}^k + c_1 r_1 \times (pbest_{ij} - x_{ij}^k) + c_2 r_2 \times (gbest_i - x_{ij}^k) \quad (\text{E.2.9})$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (\text{E.2.10})$$

E. 2.11' de her parçacık için  $d$ -boyutlu yerel en iyi konum, E. 2.12' de arama uzayındaki her bir parçacık için  $N \times d$  boyutlu bir matris ve E. 2.13' de küresel en iyi konum  $d$ -boyutlu ile gösterilir (Gözde vd., 2010; Yalçın, 2012).

$$pbest_{ij} = (pbest_{i1} \ pbest_{i2} \ pbest_{i3} \ \dots \ pbest_{id}) \ i=1, 2, 3, \dots, N \quad (\text{E.2.11})$$

$$pbest = \begin{bmatrix} pbest_{11} & pbest_{12} & pbest_{13} & \dots & pbest_{1d} \\ pbest_{21} & pbest_{22} & pbest_{23} & \dots & pbest_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ pbest_{N1} & pbest_{N2} & pbest_{N3} & \dots & pbest_{Nd} \end{bmatrix} \quad (\text{E.2.12})$$

$$gbest = (gbest_1 \ gbest_2 \ gbest_3 \ \dots \ gbest_d) \ i=1, 2, 3, \dots, N \quad (\text{E.2.13})$$

PSO algoritmasının kaba kodu aşağıda özetlenmiştir:

**For** Arama uzayında bulunan her parçacık için

    Parçacık için başlangıç konumu ata

**End**

**Do**

**For** Arama uzayında bulunan her parçacık için

Uygunluk deęerini hesapla

**Eęer** uygunluk deęeri, pbest' ten daha iyi ise;

řimdiki paręacıęı yeni pbest olarak ayarla

**End**

Bütün paręacıkların bulunduęu pbest deęerlerinin en iyisini, bütün paręacıkların gbest'i olarak ayarla

**For** Arama uzayındaki her paręacık için

E. 2.9' a göre paręacık hızını hesapla

E. 2.10' a göre paręacık pozisyonunu güncelle

**End**

**While** maksimum jenerasyon sayısına ulařana veya minimum hata kořulu saęlanana kadar devam et (Baęıř ve etin, 2009; Yalın, 2012).

PSO algoritması hesaplamadaki avantajlarından ve fazla miktarda hafızaya gerek duymamasından dolayı tercih edilmektedir. Sınır aęları eęitimi, elektronikte g ve voltaj kontrol, PID denetle parametre tespiti, gezgin satıcı problemleri ve akıř tipi izelgeleme problemleri kullanım alanlarına rnek olarak verilebilir (Kazak, 2011).

### 2.3. Genetik Algoritma

Genetik algoritma fikri ilk olarak J. Holland ve arkadaşları tarafından 1960' lı yıllarda Michigan niversitesi' nde ortaya atılmıřtır. Genetik algoritma doęal yařamda canlılar arasında var olan rekabet sonucu daha iyi ve daha gl bireylere kazanma řansının verildięi, biyolojik olaylar rnek alınarak geliřtirilmiř bir sezgisel arama teknięidir (Holland, 1975; Kesler vd., 2011; Tan vd.,1995). zellikle doęrusal olmayan, poplasyon temelli ve zor problemlerin zm iin geliřtirilmiřtir (Goldberg, 1989).

Genetik algoritmada, arama uzayında bulunan her nokta, kromozom olarak adlandırılan ikili bit dizisi ile kodlanır. zm iin "iyi" yi belirleyen bir uygunluk (fitness) fonksiyonu ve yeni zmler meydana getirmek iin yeniden kopyalama (recombination), deęiřtirme (mutation), tabii seme (natural selection) ve aprazlama (cross-over) operatrleri kullanılır (Emel ve Tařkın, 2002; Jang, 1997; Kesler vd., 2011). En iyi zm, hayatta kalabilmeyi bařaran zmdr (Mitchell, 1998). Genetik algoritmaların bir grup zmle uęrařması nemli zellięinden biridir. Bu da ktlerin

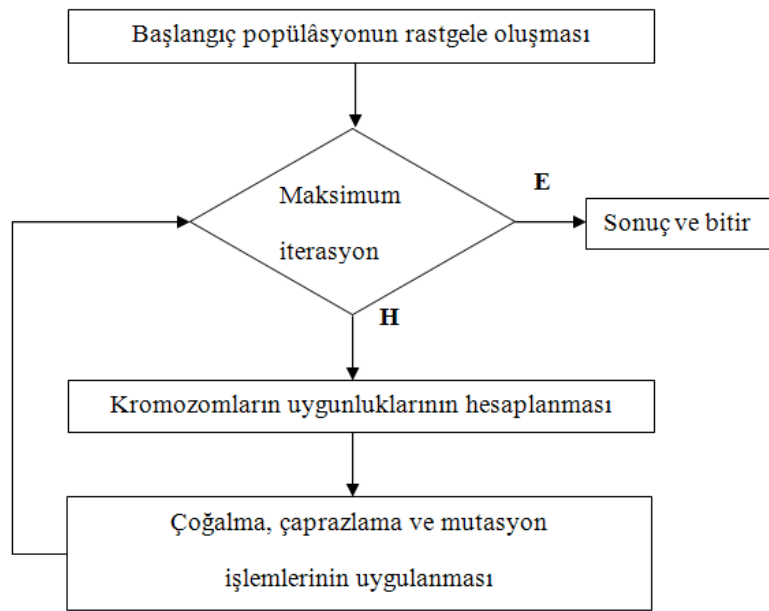
elenip, iyilerin seçilmesini sağlamaktadır. Diğer algoritmalarından farklı olarak seçme özelliği, çözüme uygun bireyin kalitesini artırır ancak optimal çözümü garanti etmez. Çözümlerin uygunluğuna göre seçim işlemi rastgele gerçekleşir.

Çözüm için işlem basamakları aşağıda verilmiştir (Emel ve Taşkın, 2002; Engin, 2001; Kesler vd., 2011):

1. Arama uzayındaki olası çözümlerin dizi olarak kodlandığı popülasyon oluşturulur.
2. Rastgele bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir.
3. Her dizi için uygunluk değeri hesaplanarak, her kromozomun ne derece iyi olduğu bulunur.
4. Uygunluk değerlerine göre rastsal olarak seçilen kromozomlar eşleşme havuzunda kopyalanır ve havuzda bireyler çift olarak gruplara ayrılır.
5. Yeni bireylerin uygunluk değerleri hesaplanarak iki ebeveyn kromozomun arasında belirlenen parçalar değiş tokuş edilir.
6. Meydana gelen yeni çözümlerin önceki çözümleri kopyalaması ihtimaline karşı ve genetik farklılık yaratmak amacıyla yeni bireyin bir bitini (ikili düzende ise) rastgele değiştirir.
7. Önceden belirlenen kuşak sayısı boyunca 2. adımdan itibaren yukarıdaki işlemler devam ettirilir.
8. İterasyon sona erdiğinde amaç fonksiyonuna göre en uygun olan kromozom seçilir.

Şekil 2.4' de GA' nın akış şeması verilmektedir (Kesler vd., 2011).





Şekil 2.4. GA tarafından çözüm üretme döngüsü.

GA, optimizasyon, bilgi sistemleri ve otomatik programlama, mekanik öğrenme, finans ve pazarlama, endüstride montaj işlemlerinde, çizelgeleme problemlerinde, gezgin satıcı problemlerinde, araç rotalama problemlerinde ve PID denetleç parametreleri tespitinde başarılı olarak kullanılmaktadır.

#### 2.4. Diferansiyel Gelişim Algoritması

Diferansiyel Gelişim Algoritması (DE), 1995 yılında Price ve Storn tarafından geliştirilmiş, popülasyona dayalı sezgisel optimizasyon tekniğidir (Storn ve Price, 1995). Gerçek parametrelili optimizasyon problemlerinin çözümünde etkin sonuçlar verebilen basit bir algoritmadır. İşleyiş ve operatörleri itibarıyla genetik algoritmaya benzerlikleri bulunsa da yapı ve uygulanışı açısından birtakım farklılıklar bulunmaktadır (Karaboğa, 2011; Keskintürk, 2006; Kesler vd., 2011).

GA' daki mutasyon, çaprazlama ve seçim operatörleri DE' de de bulunmaktadır. Farklı olarak DE çaprazlama olasılık oranı (crossover probability constant), ölçekleme faktörü (scaling factor) gibi farklı operatörlere sahiptir. DE' nin temelini popülasyonda bulunan iki birey arasındaki farkın diğer üçüncü bireye aktarılması oluşturmaktadır. Çaprazlama ve mutasyon operatörleri tüm popülasyona sırayla uygulanmayıp, her jenerasyonda tüm bireylere teker teker uygulanır. Daha sonra seçim operatörü kullanılarak elde edilen yeni birey ile mevcut bireyin uygunluğu karşılaştırılır ve

uygunluđu daha iyi olan yeni bir birey olarak sonraki jenerasyona aktarılır. Aksi durumda eskisi gelecek jenerasyona aktarılır (Karabođa, 2011; Keskintürk, 2006; Özyön vd., 2011).

DE' de bulunan ve D ile ifade edilen optimize edilecek problemin deđişken sayısıdır. D boyutlu çözüm vektörü, birey olarak adlandırılır. Birey sayısı (popülasyon büyüklüğü)  $(NP) \geq 4$  olmalıdır. Nedeni ise, algoritmanın işlem dışında üç adet harici bireye daha gereksinimi vardır.

Çözüm için işlem basamakları aşağıda verilmiştir (Kesler vd., 2011):

1. Algoritma parametrelerinin tespiti. (maksimum jenerasyon ( $G_{max}$ ), çaprazlama olasılık oranı (CR), ölçekleme faktörü (F), popülasyon büyüklüğü ( $NP \geq 4$ ) gibi)
2. Başlangıç popülasyonun rastgele oluşturulması.
3. Her bireyin maliyet fonksiyonunun hesaplanması ve en iyi bireyin belirlenerek muhafaza edilmesi.
4. Popülasyon içerisinde farklı üç bireyin seçilmesi ve seçim faktörüne göre mutasyon ve çaprazlama işlemleri ile yeni popülasyonun oluşturulması.
5. Eski popülasyon ve yeni oluşturulan popülasyondaki bireylerin maliyet fonksiyonu değerlerine göre kıyaslanarak yeni popülasyon bireylerinin seçilmesi.
6. Popülasyon içerisindeki en iyi bireyin güncellenmesi.
7. Maksimum jenerasyona ulaşıldığında aramanın sonlandırılması aksi takdirde 4. adıma geri dönülmesi.
8. Çözüm olarak son adımdaki en iyi bireylerin kullanılması.

Bir başka ifadeyle DE kodlama, mutasyon, çaprazlama ve seçim olmak üzere dört işlem basamağına sahiptir. İşlem basamaklarının açıklanmalı hali aşağıda verildiđi üzere (Özyön vd., 2011):

#### ❖ Kodlama ve başlangıç popülasyonu

Yeni bireylerin üretilebilmesi için mevcut olanların dışında üç adet daha bireye ihtiyaç duyulduđu bir önceki sayfada da belirtilmişti. Bu nedenle popülasyonumuzun

büyüklüğü  $(NP) > 3$  ya da  $(NP) \geq 4$  olmalıdır.  $NP$  adet  $D$  boyutlu çözüm vektöründen oluşan başlangıç popülasyonu E. 2.14 ile oluşturulur.

$$x_{j,i,g=0} = x_j^{(l)} + rand_j[0,1].(x_j^{(u)} - x_j^{(l)}) \quad (E.2.14)$$

E. 2.14' de  $x_{j,i,g}$ ,  $g$  neslindeki  $i$  bireyinin  $j$  parametresini,  $x_j^{(l)}, x_j^{(u)}$  değişkenlere ait alt ve üst değerleri ifade etmektedir.

#### ❖ Mutasyon

Mutasyon, birey üzerinde rastgele değişiklik yapmaktadır. DE' de mutasyon işlemi için seçilen bireyin dışında birbirinden farklı üç birey seçilir ( $r_1, r_2, r_3$ ). İlk ikisinin farkı alınıp  $F$  ile çarpılır.  $F$  ise genellikle 0-2 arasında rastgele bir değer alınmaktadır. Ardından üçüncü birey ile toplanır.

$$n_{j,i,g+1} = x_{j,r_3,g} + F.(x_{j,r_1,g} - x_{j,r_2,g}) \quad (E.2.15)$$

E. 2.15' de  $n_{j,i,g+1}$   $g+1$  mutasyon işlemi uygulanmış ara bireyi,  $r_1, r_2, r_3$  yeni birey üretiminde kullanılacak rastgele bireyleri  $r_{1,2,3} \in \{1,2,3,4 \dots NP\}$ ,  $r_1 \neq r_2 \neq r_3 \neq i$  göstermektedir.

#### ❖ Çaprazlama

Mutasyon sonucunda elde edilen fark bireyi ve  $x_{i,g}$  bireyi kullanılarak yeni birey ( $u_{i,g+1}$ ) üretilir. Deneme bireyi için  $CR$  ihtimali ile fark kromozomundan,  $(1-CR)$  ihtimalle mevcut bireyden seçilir.  $j=j_{rand}$  şartı, en az bir tane çözüm vektörünün üretilen yeni bireyden alınması için kullanılmaktadır. Rastgele seçilen  $j=j_{rand}$  gen  $CR$  değerini dikkate almaksızın  $n_{j,i,g+1}$  ' den seçilir.

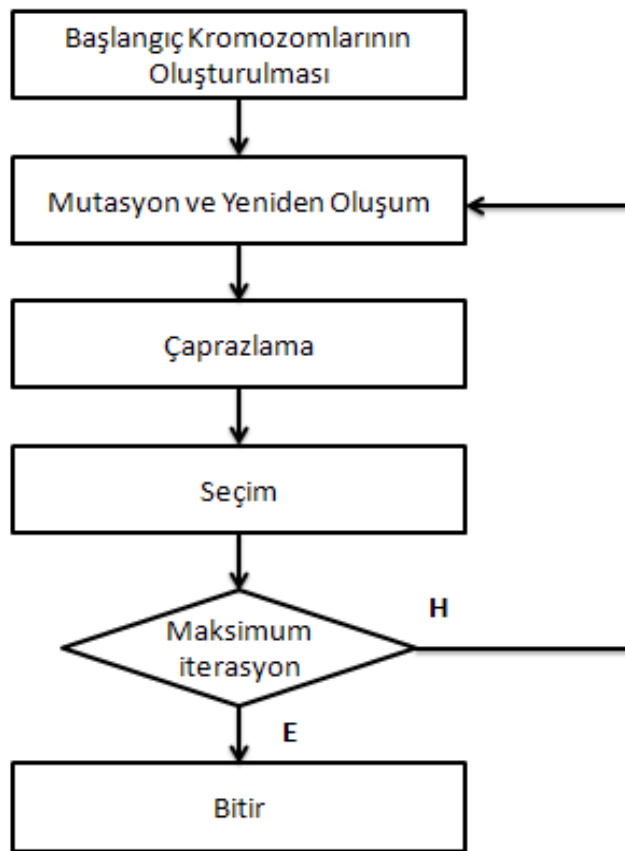
$$x_{j,u,g+1} = \begin{cases} x_{j,u,g+1} & \text{eğer } rand [0,1] \leq CR \text{ veya } j = j_{rand} \\ x_{j,i,g} & \text{diğer durumda} \end{cases} \quad (E.2.16)$$

❖ Seçim

Uygunluğu yüksek olan birey yeni nesle aktarılır. İşlem  $g=g_{max}$  oluncaya kadar devam eder,  $g_{max}$  sağlandığında mevcut en iyi birey çözüm olarak alınır.

$$x_{i,g+1} = \begin{cases} x_{u,g+1} & \text{eğer } f(x_{u,g+1}) \leq f(x_{i,g+1}) \\ x_{i,g} & \text{diğer durumda} \end{cases} \quad (\text{E.2.17})$$

Algoritmanın durdurulması belirlenen jenerasyon sayısına bağlıdır. İşlem basamakları Şekil 2.5' de gösterilmiştir.



Şekil 2.5. DE Algoritması akış diyagramı (Özyön, 2011).

DE birçok alanda başarılı olarak uygulanmaktadır. Yapay sinir ağı tasarımı, PID denetleç parametrelerinin tespiti, bulanık mantık, filtre tasarımı ve gezgin satıcı problemleri kullanım alanlarına örnek verilebilir.

## 2.5. Yerçekimsel Arama Algoritması (YAA)

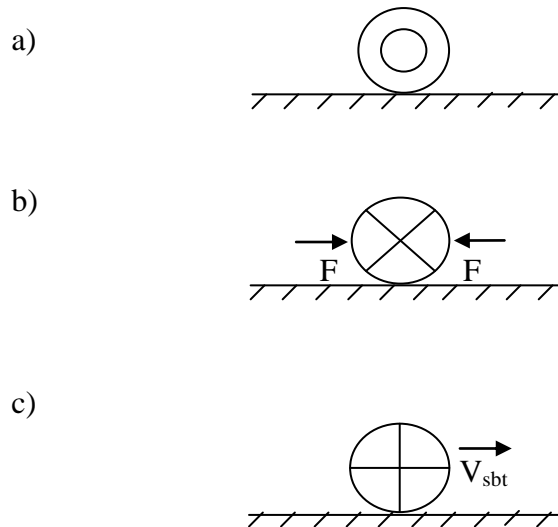
Rashedi ve arkadaşları (2009) tarafından Newton'un hareket ve yerçekimi kanunlarından esinlenerek geliştirilmiş sürü temelli sezgisel optimizasyon algoritmasıdır. YAA, Newton'un hareket kanunlarından ikincisi olan ivme kanunu ve yine Newton'un evrensel çekim kanunu temel alınarak tasarlanmış meta sezgisel optimizasyon yöntemlerinden biridir.

### 2.5.1. Newton' un hareket kanunları

#### 2.5.1.1. Eylemsizlik yasası

Evrende bulunan bir cisim üzerine etki eden kuvvetlerin vektörel toplamı sıfır ise ya da cismin üzerine etkiyen kuvvet yok ise Şekil 2.6.a ve Şekil 2.6.b' de gösterildiği gibi cisim mevcut durumunu korur. Bir başka ifade ile Şekil 2.6.c' de gösterildiği gibi eğer cisim belli bir hızla hareket ediyorsa hareketine devam eder, hareket etmiyor ise mevcut durağanlığını korur (Kazak, 2011; Serway ve Beichner, 2007). Eylemsizlik yasası:

- Cismin ivmesinin (a) sıfır olması demektir.
- Cismin durağan hareketinin değişmesini engellemeye çalışan kuvvettir.
- Durağan hareketin değişmesi demek, cismin ivme kazanması demektir. O halde, eylemsizlik kuvveti ivmeye karşı koyan bir kuvvettir.

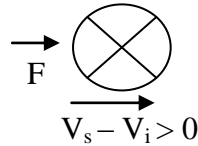


Şekil 2.6. Eylemsizlik yasasına göre cisimlerin farklı durumları.

### 2.5.1.2. İvme yasası

Cismin üzerine etkiyen kuvvet sonucu cisimde hız değişimi meydana gelir. Bu hızdaki değişim ivme olarak ifade edilir . Cisimi hareket ettiren kuvvet ile cismin ivmesi aynı doğrultudadır. Herhangi iki cisme aynı yönde aynı büyüklükte kuvvet uygulandığında kütlesi büyük olanın ivmesi daha küçük, kütlesi küçük olan cismin ivmesinin ise daha büyük olduğu görülmüştür. (Kazak, 2011; Rashedi vd., 2009; Serway ve Beichner, 2007).

Cisme uygulanan kuvvet ile oluşan ivme durumu Şekil 2.7' de gösterilmiştir. Bir cismin ivmesi, üzerine uygulanan kuvvet ile doğru, kütlesi ile ters orantılıdır. Bu bağıntı E. 2.18' de gösterilmiştir.



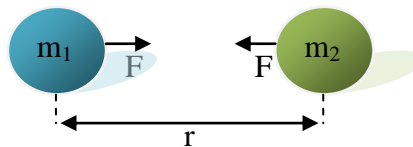
Şekil 2.7. Cisme uygulanan kuvvet sonucu hız değişimi.

E. 2.18' de ifade edilen  $F$ , cisme uygulanan kuvveti,  $m$ , cismin kütlesini,  $a$  ise cisme uygulanan kuvvet sonrası hız değişimini yani ivmeyi ifade eder.

$$F = m \times a \quad (\text{E.2.18})$$

### 2.5.2. Newton' un evrensel çekim kanunu

Evrende bulunan cisimler yerçekimi kuvvetine bağlı olarak birbirlerine çekim kuvveti uygularlar. Uygulanan bu çekim kuvveti, cisimlerin kütlelerinin çarpımıyla doğru, cisimlerin arasındaki mesafeyle ters orantılıdır. Bu oran Şekil 2.8' de gösterilmiştir.



Şekil 2.8. Birbirine etki eden kütleler.

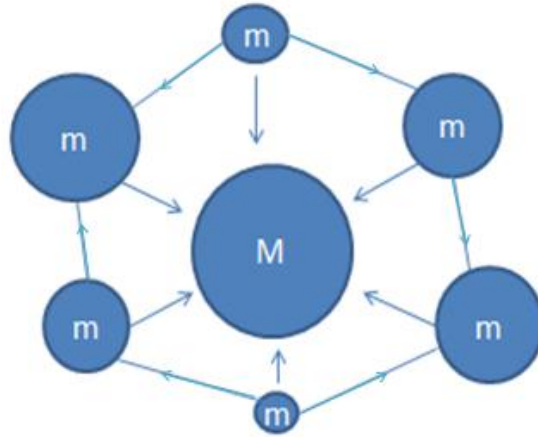
Newton' un yerçekimi kanununa göre kütleler arasındaki kuvvet:

$$F = G \times \frac{m_1 \times m_2}{r^2} \quad (\text{E.2.19})$$

eşitliği ile ifade edilir. E. 2.19' da G, yerçekimi sabitini ifade eder (Kazak, 2011; Rashedi vd., 2009; Serway ve Beichner, 2007).

### 2.5.3. YAA' nın yapısı ve çalışması

Yerçekimsel arama algoritmasında arama uzayındaki her bir parçacık bir kütle olarak kabul edilir. Bu tanımla YAA' yı bir yapay kütle sistemi olarak ifade etmek mümkündür (Kazak, 2011; Rashedi vd., 2009). Yerçekimi kuvvetine maruz kalan kütleler arama uzayı içerisinde hareket ederek optimum çözüme ulaşırlar. Bu durum Şekil 2.9' da gösterilmiştir.



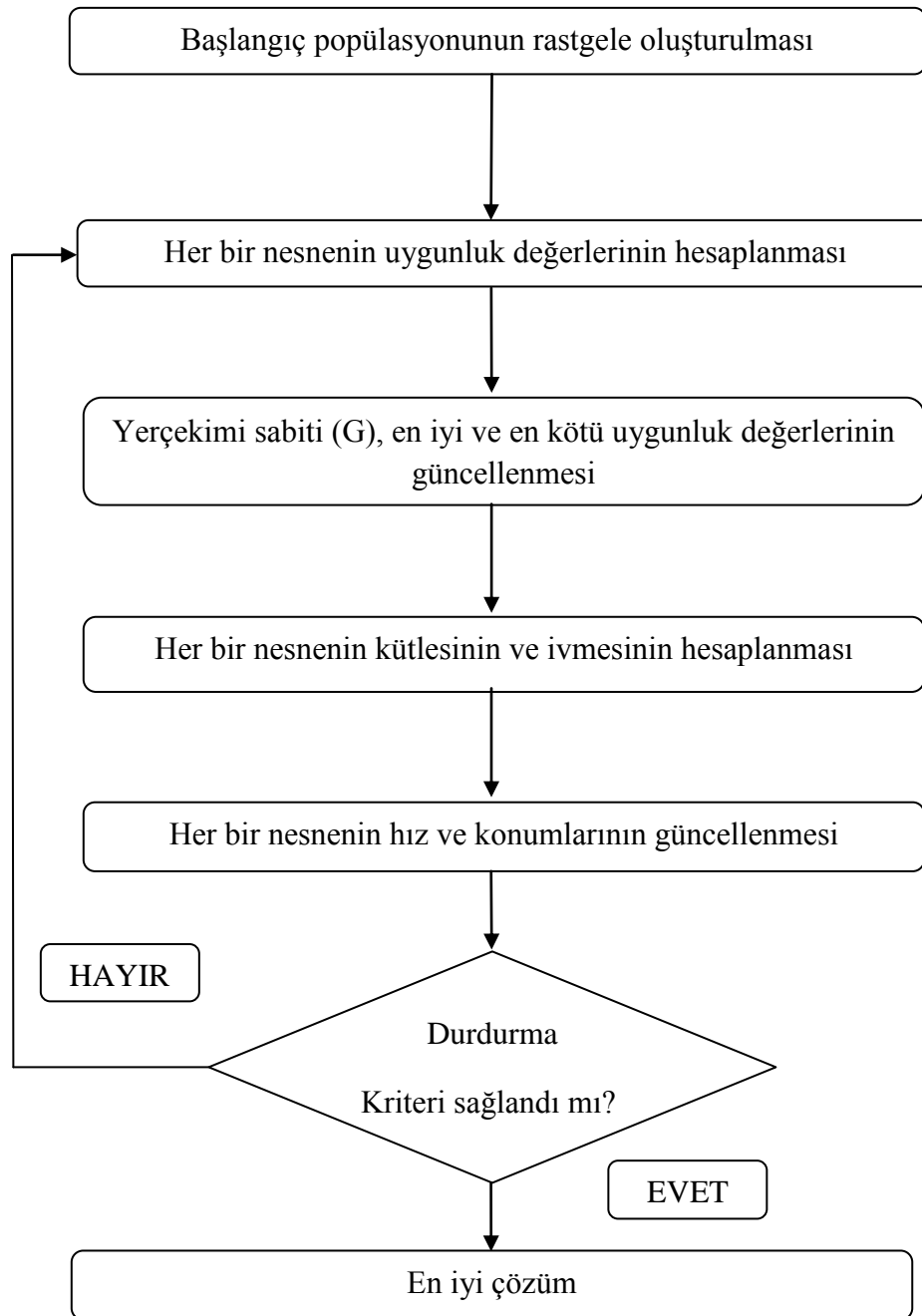
Şekil 2.9. Kütlelerin birbirleri ile etkileşimi.

Noktasal kütlelerin bu şekildeki hareketi Newton'un ikinci hareket kanununa uygun olarak gerçekleşir. Arama uzayındaki her bir kütle, konumu, eylemsizlik kütlesi, aktif yerçekimsel kütlesi ve pasif yerçekimsel kütlesi olmak üzere dört özelliği vardır. Her kütle arama uzayında belli bir pozisyonda bulunur. Buldukları konum, çözülmesi beklenen problem için birer çözüm alternatifidir. Kütleler, üzerlerine herhangi bir kuvvet etki ettiğinde bu kuvvete karşı koymak isteyerek bir direnç gösterir. Bu direnç eylemsizlik kütlesi olarak adlandırılır. (Kazak, 2011).

Eylemsizlik kütlesi fazla olan kütle daha yavaş hareket etmek isteyecek, az olanı ise daha hızlı hareket edecektir. Bu durumda oluşan yerçekimi alanı gücüne aktif yerçekimsel kütle denir. Yerçekimi alanı ile bir nesnenin iletişim gücüne pasif yerçekimsel kütle denir (Kazak, 2011).

YAA' da her bir nesnenin kütle miktarı o nesnenin performansını gösterir. Algoritma süresince en ağır olan kütle diğer kütlelere göre daha yavaş hareket edecek ve diğerlerini kendine çekecek bir çekim kuvveti uygulayacaktır. Jenerasyon sayısı bitiminde ya da herhangi bir sonlandırma eylemi olduğunda kütlesi en fazla olan nesne, problemin optimum çözümünü oluşturmuş olacaktır.

Rashedi ve arkadaşlarının yaptıkları çalışmaya dayanarak YAA' nın akış şeması Şekil 2.10' da gösterilmiştir (Kazak, 2011; Rashedi vd., 2009).



Şekil 2.10. YAA' nın akış diyagramı.



Algoritmanın adımları;

**1. Başlangıç:** YAA' da kullanılan yerçekimi sabiti (G), maksimum jenerasyon sayısının, bir kütle için diğer kütlelere yapacağı etkiyi hesaplamak için kullanılan küçük bir sabitin ( $\epsilon$ ) ilk değerlerinin atıldığı kısımdır.

**2. Arama Uzayının Tanımlanması:** YAA için her olası çözüm bir küttedir. Sistemin kaç kütteden oluşacağını tasarlandığı kısımdır. Kütleler arama uzayına rastgele yerleştirilir. E. 2.20' de tanımlanan  $x_i^d$ , i. kütle için d. boyuttaki konumunu gösterir.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i = 1, 2, 3, \dots, N \quad (E.2.20)$$

**3. Yerçekimi Sabitinin Hesaplanması:** Başlangıç kısmında atanan sabit değerin her jenerasyonda azaltıldığı kısımdır. Zamanla azalan yerçekimi sabiti ile arama hızı kontrol edilmektedir. Yerçekimi sabitinin jenerasyon ile değişiminin hesaplanmasında E. 2.21 eşitliğinden yararlanılır.

$$G(t) = G_0 \times e^{(-\alpha \frac{k}{K})} \quad (E.2.21)$$

$G_0$ , yerçekimi sabitinin başlangıç değerini,  $\alpha$ , kullanıcının belirlediği sabit bir değeri, k, o andaki jenerasyon sayısını ve K maksimum jenerasyon sayısını göstermektedir.

#### **4. Uygunluk Fonksiyonu ile Uygunluk Değerlerinin Hesaplanması:**

Hesaplanan uygunluk değerinin en iyi ve en kötülerini seçilir (best, worst).

Problem minimize yapılmak isteniyorsa;

$$\text{best}(k) = \min_{j \in \{1, \dots, N\}} \text{fit}_j(k) \quad (E.2.22)$$

$$\text{worst}(k) = \max_{j \in \{1, \dots, N\}} \text{fit}_j(k) \quad (E.2.23)$$

Problem maksimize edilmek isteniyorsa;

$$\text{best}(k) = \max_{j \in \{1, \dots, N\}} \text{fit}_j(k) \quad (\text{E.2.24})$$

$$\text{worst}(k) = \min_{j \in \{1, \dots, N\}} \text{fit}_j(k) \quad (\text{E.2.25})$$

E. 2.22, E. 2.23, E. 2.24 ve E. 2.25 denklemlerinde yer alan  $j = \{1, \dots, N\}$  dir.  $\text{fit}_j(k)$ ,  $j$ . kütlenin  $k$  anındaki uygunluk değeri,  $\text{best}(k)$ ,  $k$  anındaki en iyi çözüm,  $\text{worst}(k)$  ise  $k$  anındaki en kötü çözümü ifade etmektedir.

**5. Kütle Hesabı:** Arama uzayında bulunan bir kütlenin aktif yerçekimsel kütlesi, pasif yerçekimsel kütlesi ve eylemsizlik kütlesi birbirlerine eşit alınarak tüm kütleler hesaplanır.

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad i=1,2,3,\dots,N \quad (\text{E.2.26})$$

$$m_i(k) = \frac{\text{fit}_i(k) - \text{worst}(k)}{\text{best}(k) - \text{worst}(k)} \quad (\text{E.2.27})$$

$$M_i(k) = \frac{m_i(k)}{\sum_{j=1}^N m_j(k)} \quad (\text{E.2.28})$$

$M_{ai}$  kütlenin aktif yerçekimsel kütlesi,  $M_{pi}$  kütlenin pasif yerçekimsel kütlesi,  $M_{ii}$   $i$ . kütlenin eylemsizlik kütlesi,  $M_i$  eylemsizlik kütesidir,  $m_i(k)$   $k$  anında  $i$ . kütlenin değeridir. Bu değişkenlerin hesaplanmasında E. 2.26, E. 2.27 ve E. 2.28 denklemleri birlikte kullanılır. Bu işlemde normalizasyon yapılmaktadır.

**6. Kuvvet Hesabı:** Newton'un evrensel yerçekimi kanunundan esinlenerek iki kütlenin birbirine uyguladığı kuvvet; kütleler arası mesafenin kütleler çarpımına bölünmesiyle hesaplanır. İki kütle arasındaki kuvvet hesaplandıktan sonra bir kütleyle etkiyen toplam kuvvet hesaplanır.

$$R_{ij}(k) = \|x_i(k), x_j(k)\|_2 \quad (\text{E.2.29})$$

E. 2.29' da tanımlanan  $R_{ij}$ ,  $k$  anında  $i$  ve  $j$  kütleleri arasındaki mesafeyi vermektedir. Bu kütleler arasındaki kuvvet E. 2.30 ile hesaplanır.

$$F_{ij}^d(k) = G(k) \frac{M_{pi}(k) M_{aj}(k)}{R_{ij}(k) + \varepsilon} (x_j^d(k) - x_i^d(k)) \quad (\text{E.2.30})$$

$F_{ij}^d(k)$  k anında d. boyutta i ve j kütleleri arasındaki kuvveti,  $G(k)$  yerçekimi sabitini,  $M_{pi}(k)$  k anında i. kütlelerin pasif yerçekimsel kütlelerini,  $M_{aj}(k)$  k anında j. kütlelerin aktif yerçekimsel kütlelerini,  $\varepsilon$  kullanıcı tarafından atanan sabit bir değeri,  $x_j^d(k)$  ve  $x_i^d(k)$  ise i ve j kütlelerinin d. boyuttaki konumunu belirtir.

Bir kütleyle etkiyen toplam kuvvet E. 2.31 yardımı ile hesaplanır.

$$F_i^d(k) = \sum_{j \in K_{best}, j \neq i}^N rand_j F_{ij}^d(k) \quad (E.2.31)$$

$rand_j$ ,  $[0,1]$  aralığında verilen rastgele bir sayıdır.  $K_{best}$  başlangıçta  $K_0$  kadar kütle ile başlayan lineer olarak azalan bir değerdir.

**7. İvme Hesabı:** Newton'un ikinci yasası olan ivme yasasına dayanarak kütlelerin ivmeleri E. 2.32 ile hesaplanır.

$$a_i^d(k) = \frac{F_i^d(k)}{M_{ii}(k)} \quad (E.2.32)$$

$a_i^d(k)$  k anında d. boyuttaki i. kütlelerin ivmesini verir.

**8. Hız ve Konum Güncellemesi:** Kütlelerin o andaki hızı ile o anda oluşan hız değişiminin (ivme) toplamına eşittir. E. 2.33' de gösterilmiştir.

$$V_i^d(k+1) = rand_i \times V_i^d(k) + a_i^d(k) \quad (E.2.33)$$

$rand_i$ ,  $[0,1]$  aralığında atanan rastgele bir sayıdır. İvme ile bir kütlelerin sistemdeki yeri değişmektedir. Bu sebepten kütlelerin konum güncellenmesi E. 2.34' de yapılmaktadır.

$$x_i^d(k+1) = x_i^d(k) + V_i^d(k+1) \quad (E.2.34)$$

**9. Sonlandırma Kriteri:** Maksimum jenerasyon sayısına ulaşıldığında ya da minimum hata koşulu sağlandığında, amaç fonksiyonuna en uygun olan kütle seçilir ve algoritma sonlanır.

### 3. DENETİM SİSTEMLERİ

Son yıllarda denetim sistemleri, teknolojik araç ve gereçlerin gelişmesinde önemli rol oynayan bir bilim dalı olmuştur. Teknolojinin gelişmesi ile denetim sistemlerinin önemi gittikçe artmıştır. Günümüzde neredeyse yapılan tüm etkinliklerde denetim sisteminin etkileri görülmektedir. Denetim sistemleri, üretilen ürünlerin kalite denetiminde, silah ve savunma sistemlerinde, uzay teknolojilerinde, bilgisayarlı denetim sistemlerinde, otomatik montaj hatlarında, makine ve aletlerin denetiminde, ulaşımda, güç sistemlerinde, evlerde ve ofislerde kullanılan ısıtma ve havalandırma sistemlerinde, ilaç endüstrisinde, robotik ve benzeri endüstriyel sektörlerde yaygın olarak kullanılmaktadır (Boz, 2012-a; Kuo, 2006).

Denetim sistemlerinin amacı, sistemdeki değişkenlerin değerlerini istenilen seviyede tutarak kararlı bir çalışma ortamı sağlamaktır (Hurma, 1998). Denetim sistemleri, denetim ve sistem olarak adlandırılan yapılardan meydana gelmektedir. Denetim kelime anlamı olarak; düzenleyen, yöneten, kumanda eden demektir. Sistem ise, belli bir işlevi gerçekleştirmek için bir araya getirilen elemanlar kümesi olarak ifade edilebilir (Boz, 2012-a). Bu tanımlamalardan yola çıkılarak denetim sistemleri, denetim sistemleri elemanları ile girişleri kullanarak, çıkışları önceden belirlenmiş bir şekilde sistemin denetlenmesi şeklinde tanımlanabilir (Boz, 2012-a; Kuo, 2006).

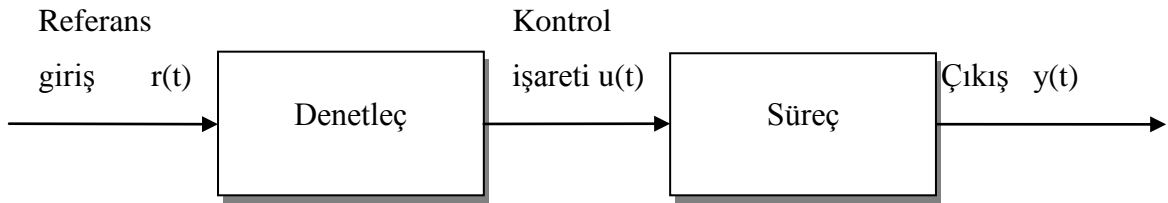
#### 3.1. Denetim Sistemlerinin Türleri

Denetim sistemleri çıkış denetimi veya denetlenen büyüklüğün kumanda edilmesi bakımından iki türe sahiptir. Bunlar açık çevrim denetim sistemleri (geri beslemesiz sistemler) ve kapalı çevrim denetim sistemleridir (geri beslemeli sistemler). Bölüm 3.1.1 ve bölüm 3.1.2' de tanımlamaları yapılmıştır.

##### 3.1.1. Açık çevrim denetim sistemleri (Geri beslemesiz sistemler)

Açık çevrimli denetim sistemleri genellikle zaman temelli olup belirli zaman dilimlerinde belirli işlemler yapar. Denetlenen sistemde bozucu etkenler bulunmadığı takdirde açık çevrimli denetim sistemleri etkili kullanılabilir. Bu denetim türünde herhangi bir ölçme ve karşılaştırma işlemi yoktur. Bu sebeple sistemin hassasiyeti başlangıçtaki kalibrasyona bağlıdır. Çok sayıda karmaşık olmayan uygulamada açık çevrim denetim sistemlerine rastlamak mümkündür. Trafik sinyalizasyonu açık çevrim

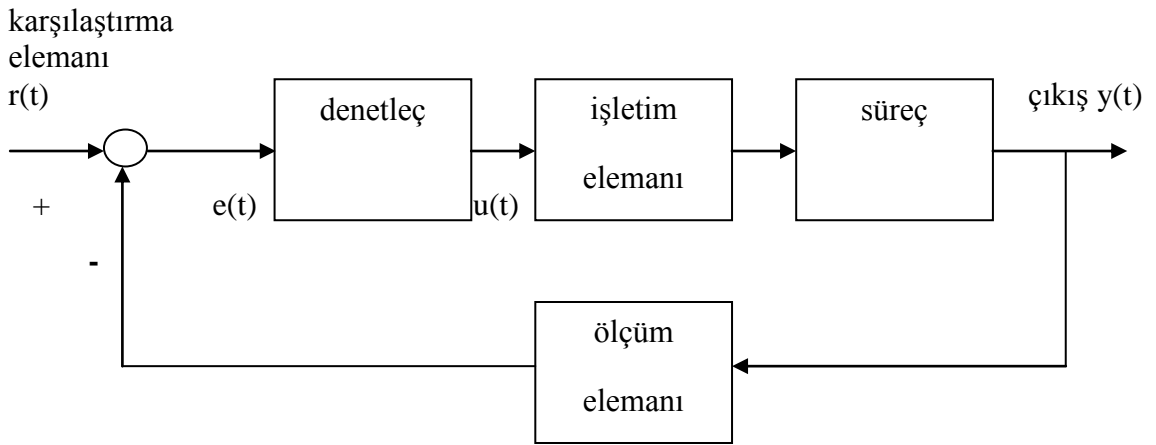
denetime örnek verilebilir. Şekil 3.1' de açık çevrim denetim sistemi blok diyagramı gösterilmiştir (Baba, 1989; Kuo, 2006). Açık çevrim denetim sistemlerinin Şekil 3.1' de gösterildiği gibi denetleç ve süreç olmak üzere iki bileşeni bulunmaktadır.



Şekil 3.1. Açık çevrim denetim sistemi blok diyagramı.

### 3.1.2. Kapalı çevrim denetim sistemleri (Geri beslemeli sistemler)

Bir veya birden fazla geri beslemeye sahip sistemler kapalı çevrim sistem olarak adlandırılır. Şekil 3.2' de geri beslemeli denetim sisteminin blok diyagramı gösterilmiştir (Kuo, 2006).



Şekil 3.2. Kapalı çevrim denetim sistemi blok diyagramı.

Kapalı çevrimli denetim sistemlerinde sistem çıkışı ölçülür. Ölçme işlemini ölçüm elemanı gerçekleştirir. Denetim cihazı ölçüm elemanından ölçüm değerini alır ve sistemin ne derece iyi çalıştığına karar verir. Çıkışta ölçülen değer ile referans değeri karşılaştırılarak sistem geri beslenir. Çıkış değeri ile referans değeri arasındaki fark bulunur. Fark değeri sistemdeki hata değerini ifade eder. Hata değerine göre denetim mekanizması devreye girer ve hatayı düzeltmek için denetim işaretini üretir. Şekil 3.2' de  $r(t)$  referans değerini,  $y(t)$  çıkış değerini,  $u(t)$  denetim değişkenini,  $e(t)$  sistem

cevabının olması istenilen değeri ile sistem çıkışı arasındaki farkı yani hata değerini ifade eder (Hurma, 1998).

### 3.2. PID Denetim Sistemlerinin Tasarımı

Bir denetim sisteminin tasarımı üç aşamada gerçekleştirilir (Kuo, 2006).

1. Sistemin neyi, nasıl yapması gerektiğinin belirlenmesi (tasarım kriterleri).
2. Denetleyicinin sisteme nasıl bağlanacağıının belirlenmesi.
3. Tasarım hedeflerine göre denetleç parametrelerinin tespit edilmesi.

Sistemlerin tasarlanma amacı belirli görevleri yerine getirmek olduğundan bu görevleri hangi doğrulukta ve ne kadar sürede yapmaları konusu önem teşkil etmektedir. Belirli amaç için tasarlanması düşünülen sistemin, başarımını etkileyecek olan kriterlerin bilinmesi ve bu kriterlerin sistem tasarımı sürecinde sisteme uygulanması gereklidir.

Sistemin amacının ne olduğu, nasıl yapması gerektiği ve nasıl yapacağını değerlendirmek tasarım kriterleri ile belirlenir. Sistemden beklenen sistemin kararlı olması, giriş değişkeni  $r(t)$  ve çıkış değişkeni  $y(t)$ ' ye bağlı olarak denetim hatasının küçük olmasıdır.

Denetim sistemleri tasarımının amacı, çıkış değerinin istenilen ya da olması beklenen şekilde gerçekleşmesidir. Bir sistemde kararlılık, güvenilirlik, duyarlılık, gözlenebilirlik ve denetlenebilirlik oldukça önemlidir. Tüm bu önem arz eden özellikler için denetleçler kullanılır. Denetleçler, sistemin düzgün işlemlerini, dış bozucu etkenler ile gürültülerin giderilmesini sağlamak için çalışırlar (Boz, 2012-a).

Günümüzde en sık kullanılan denetleç çeşitlerinden biri PID denetleçtir. PID denetleç, oransal (P = Proportional), integral (I = Integral) ve türevsel (D = Derivative) işlevlerini belirten kelimelerin baş harfleri bir araya getirilerek ifade edilir. Denetim sistemleri tasarımında her üç kısım kullanılabileceği gibi yalnızca birinin ya da ikisinin de kullanılması yeterli olabilir (Hurma, 1998). Yapılan uygulamalarda en çok tercih edilen PID denetleyicinin türev ve integral kısmının denetim sistemine farklı etkileri vardır (Kuo, 2006).

### 3.2.1. P (Oransal) denetleç

Oransal denetleç, sabit K kazançlı denetim sistemlerinde denetleç çıkışındaki işaretin, denetleç girişinde bulunan işarete oranı olarak ifade edilebilir. Bu tanıma dayanarak oransal denetleç E. 3.1 ile ifade edilebilir.

$$u(t) = K_p \times e(t) \quad (E.3.1)$$

E. 3.1' de ifade edilen  $u(t)$  denetleç çıkışını,  $K_p$  oransal kazancı,  $e(t)$  hatayı göstermektedir. Doğrusal olmayan sistemlerde, çıkış değeri sabit değere yaklaşırsa sistemin kararlı hale geldiği kabul edilir ve  $K_p$  ile denetlenen sistemde  $K_p$  arttıkça cevap hızı artar. Ancak hata değerinde bir değişim olacağından geri beslemeli sistemler kararsız hale gelebilir (Hurma, 1998).

Oransal denetleyicinin dezavantajı  $K_p$ ' nin sıfır olması durumunda E. 3.1' de belirtilen bağıntıya bağlı olarak denetleç çıkışının da sıfır olmasıdır. Bu duruma kalıcı durum hatası adı verilir (Baba, 1989).

### 3.2.2. PI (Oransal İntegral) denetleç

Oransal denetleyicide ifade edilen kalıcı durum hatasının oluşumunu engellemek için denetim elemanına integral eklenir. PI denetleç yapısında kullanılan I denetleç hata değerini azaltır. Uygulamalarda P ve I denetleçlerin birlikte kullanılması tercih edilmektedir. Bunun nedeni, denetim sisteminde giriş verilen cevap istenenden farklı ise sistemden önce kazanç sağlanmak istenmesidir. Bu yüzden oransal kazanç sistemlerde vazgeçilmezdir (Yıkan, 2005).

I ve PI denetleçlerin transfer fonksiyonları E. 3.2 ve E. 3.3' de verilmiştir (Yıkan, 2005).

I denetleç için:

$$u = K_i \int e(t)dt \quad \text{ve} \quad C(s) = \frac{K_i}{s} \quad (E.3.2)$$

PI denetleç için:

$$u = K_p e(t) + K_i \int e(t) dt \quad \text{ve} \quad C(s) = \left( K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \right) \quad (\text{E.3.3})$$

### 3.2.3. PD (Oransal – Türevsel) denetleç

Denetim sisteminde, hata olmadığı durumlarda integral kısım denetim işareti üretmeyi sürdürür, bunun sonucunda sistemde salınımlar meydana gelir. Salınımları engellemek için, hatanın sifıra yakın bir değerde olduğu denetleçe iletilmelidir. Bu da hata değerinin türevi alınarak sağlanabilir. Sistemin kararlılığında düzelmeler D denetleç kullanıldığında görülür. D denetleç, P denetleç ile birlikte kullanılmaktadır (Hurma, 1998).

D ve PD denetleçlerin transfer fonksiyonları E. 3.4 ve E. 3.5’ de verilmiştir.

D denetleç yapısı:

$$u = K_d \frac{de(t)}{dt} \quad \text{ve} \quad C(s) = K_d s \quad (\text{E.3.4})$$

PD denetleç yapısı:

$$u = K_p e(t) + K_d \frac{de(t)}{dt} \quad \text{ve} \quad C(s) = K_p + K_d s \quad (\text{E.3.5})$$

### 3.2.4. PID (Oransal-İntegral-Türevsel) denetleç

Önceki bölümlerde, PD denetleyicinin sisteme bir zayıflık getirdiği ancak kararlı hal davranışı üzerine etki etmediği belirtilmiştir. PI denetleyicinin ise, kararlı hal hatalarını düzelttiği ve buna karşın sistemin yükselme zamanını artırdığı belirtilmiştir. Bu bilgiler kullanıcıları, PI ve PD denetleçlerinin avantajlarını içerisinde barındıran, PID denetleyicisini kullanmaya yönlendirir (Kuo, 2006).

PID denetleç, geri beslemeli denetleç sistemlerinden en etkili kullanım alanına sahip olan ve düşünüldüğünde ilk akla gelen türdür. Endüstriyel uygulamalarda yaygın olarak kullanılmaktadır.



PID denetleyicinin yaygın kullanım alanına sahip olmasının nedenleri şunlardır (Ayaz, 2011):

- ❖ Basit bir yapıya sahip olduğundan kolayca anlaşılabilir.
- ❖ Farklı uygulamalarda etkili sonuçlar vermiştir.
- ❖ Maliyeti oldukça düşüktür.
- ❖ Algoritma olarak gömülü sistemlere kolaylıkla entegre edilebilmektedir.

PID denetleyicinin işlevi E. 3.6 ile ifade edilebilir (Ayaz, 2011).

$$u(t) = K_p \left[ e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right] \quad (\text{E.3.6})$$

$u(t)$  : Denetim sinyali,

$e(t)$  : Denetim hatası

$K_p$  : Oransal kazanç,

$T_I$  : İntegral zamanı,

$T_D$  : Türev zamanı,

} PID denetleç parametreleri

PID denetleyiciye ait transfer fonksiyonu E. 3.7 ile ifade edilebilir (Ayaz, 2011).

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left[ 1 + \frac{1}{T_I s} + T_D s \right] \quad (\text{E.3.7})$$

E. 3.6' da belirtilen PID denetleyiciye ait  $K_p$  orantı kazanç parametresi,  $T_I$  integral parametresi ve  $T_D$  türev parametresi denetlenen sistem içerisinde önemli bir role sahiptir. Çizelge 3.1' de endüstride kullanılan sistemlerde de geçerli olan PID parametrelerinin geçici hal parametrelerine etkisi belirtilmiştir (Ayaz, 2011).

Çizelge 3.1. PID parametrelerinin geçici hal parametreleri üzerine etkisi.

Parametre	Yükselme Zamanı	Sistem Ani Tepkisi	Oturma Zamanı	Kararlı Hal Hatası
$K_p$ artarsa	Azalır	Artar	Çok Az Etkilenir	Azalır
$T_I$ azalır	Azalır	Artar	Artar	Yok Olur
$T_D$ artarsa	Çok Az Etkilenir	Azalır	Azalır	Çok Az Etkilenir

PID denetleyicinin sistem içerisindeki başarımı,  $K_p$ ,  $K_d$  ve  $K_i$  katsayılarının uygun ve doğru değerlerle seçilmiş olmasına bağlıdır. Eğer bu katsayılar uygun değerde olmaz ise sistemi istenilen şekilde denetlemek mümkün olamaz, hatta sistem kararsız bir hal alabilir.

Bu katsayıların sistem üzerindeki etkileri aşağıdaki gibi açıklanabilir (Alli, 2001; Dandıl vd., 2002; Ünal, 2008).

- ❖  $K_p$  artarsa sistemde aşım artar, cevap süresi kısalır.
- ❖  $K_d$  artarsa sönüm değeri artar, aşım ve salınım azalır.
- ❖  $K_i$  artarsa sönüm değeri azalır, salınım ve aşım artar.

DA motoru hız denetimi için PID denetleyicinin kazanç katsayılarının ( $K_p$ ,  $K_d$  ve  $K_i$ ) sisteme olan etkileri, motor parametreleri ve transfer fonksiyonu verilen aşağıdaki örnekle açıklanmıştır (Boz, 2012-b).

DA motor parametreleri:

- Rotor eylemsizlik momenti ( $J$ ) = 0.01 kg.m<sup>2</sup>
- Mekanik sistemin sönüm oranı ( $b$ ) = 0.1 Nm/s
- Elektromotor kuvvet sabiti ( $K=K_m=K_b$ ) = 0.01 Nm/Amp
- Rezistans ( $R$ ) = 1  $\Omega$
- İndüktans ( $L$ ) = 0.5 H
- Giriş ( $V$ ): kaynak voltajı
- Çıkış ( $\theta$ ): mil durumu

- Rotor ve milin sert olmadığı kabul edilir.

PID denetleç transfer fonksiyonu E. 3.8 ile verilmiştir.

$$K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s} \quad (\text{E.3.8})$$

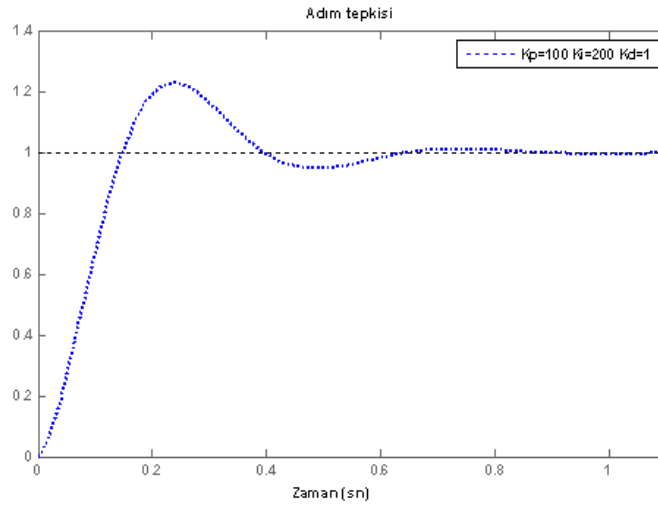
Kontrol sistemlerinde sadece oransal denetleyicinin kullanılmasıyla sistemde kararlı hal hatasının ve aşmanın büyük olduğu görülmektedir. Bu sorunu önlemek için sistemlerde integral yapı ile türev halinin kullanılması gerektiği önceki bölümlerde açıklanmıştır. İntegral yapı ile kararlı hal hatasının yok edildiği ve türev halinin de aşmayı azalttığı bilinmektedir.

$K_p$ ,  $K_d$  ve  $K_i$  değerlerinin denetim sistemine etkilerini belirlemek amacıyla DA motor parametreleri ve transfer fonksiyonu aşağıda gösterildiği gibi Matlab programında kodlanır.

```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
num=K;
trfonk=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];
Kp=100; Ki=200; Kd=1;
numc=[Kd, Kp, Ki];
denc=[1 0];
numa=conv(num,numc);
dena=conv(trfonk,denc);
[numac,denac]= cloop (numa,dena);
t=0:0.01:1;
step (numac,denac,t) ;
```

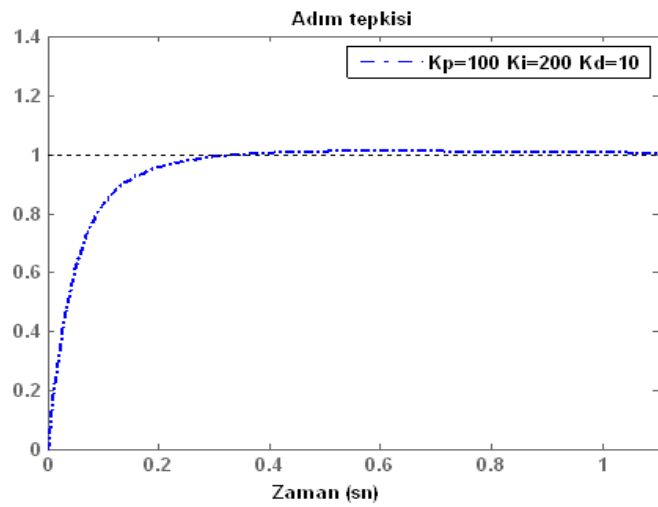
$K_p$ ,  $K_d$  ve  $K_i$  kazanç katsayı değerleri, *cloop* kapalı çevrimli denetim sistemlerinde transfer fonksiyonu çözümlene komutu, *numac* ve *denac* kapalı çevrimli transfer fonksiyonu pay (numaratör) ve paydadır (denumaratör).

$K_p=100$ ,  $K_i=200$  ve  $K_d=1$  değerlerini aldığıında elde edilen sonuç Şekil 3.3' de gösterilmiştir.



Şekil 3.3.  $K_p$ ,  $K_i$  ve  $K_d$  değerleri ile PID denetleç başarımı ( $K_p=100$ ,  $K_i=200$ ,  $K_d=1$ ).

Şekil 3.3' de sistemin referans değerine göre aşma yaptığı görülmektedir. Bu büyük aşmayı düşürmek için 1 olan  $K_d$  değeri 10 olarak değiştirildiğinde, oluşan yeni durum Şekil 3.4 ile gösterilmiştir.



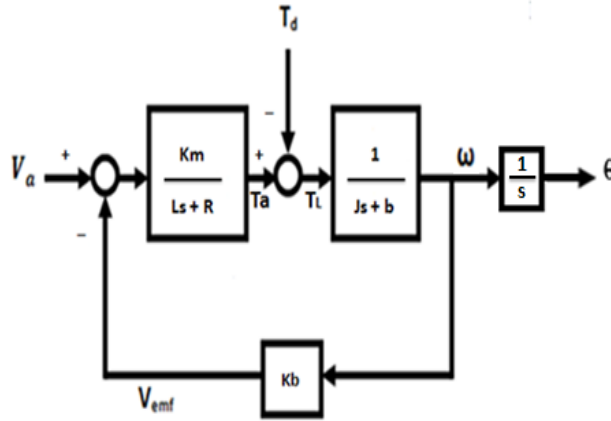
Şekil 3.4. PID denetleç ile istenen denetim başarımı ( $K_p=100$ ,  $K_i=200$ ,  $K_d=10$ ).

#### 4. YAA İLE PID PARAMETRELERİNİN OPTİMİZASYONU

PID denetleyicinin katsayıları olarak ifade edilen  $K_p$ ,  $K_i$  ve  $K_d$  denetim sisteminin performansını etkileyen unsurların başında gelmektedir. Bu katsayılar tasarım kriterlerine uygun olarak belirlendiği sürece, sistemin istenilen şekilde denetlenmesini sağlamaktadır.

Bu bölümde, sezgisel yöntemlerden biri olan YAA kullanılarak DA motor denetimi için PID denetleç parametrelerinin tespiti yapılmıştır. Endüvi sargısı denetimli DA motorunun hızı, endüvi sargısı uçlarına uygulanan gerilimle kolayca denetlenebilir.

Şekil 4.1’de DA motora ait blok diyagram verilmiştir (Ogata, 1990).



Şekil 4.1. DA motor hız denetimi blok diyagramı.

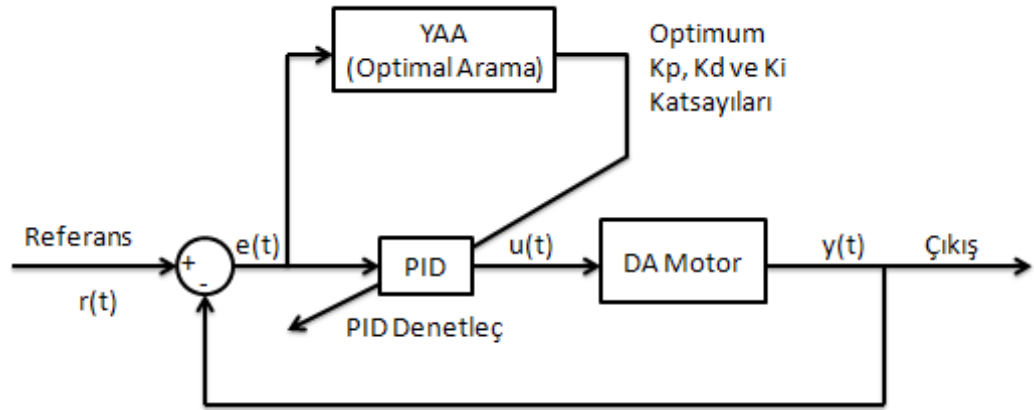
Çizelge 4.1’ de çalışmada kullanılan DA motora ait parametreler verilmiştir (Kesler vd., 2011).

Çizelge 4.1. DA motor parametreleri.

Parametreler			
Sembol	Değer	Sembol	Değer
<b>P</b>	93.3 kW	<b>R<sub>a</sub></b>	0.0125Ω
<b>V<sub>a</sub></b>	210 V	<b>b</b>	1.89 Nm/s
<b>Hız</b>	1150 dev/dk	<b>J</b>	2.83 kgm <sup>2</sup>
<b>L<sub>a</sub></b>	1.2 mH	<b>K<sub>m</sub>=K<sub>b</sub>=K</b>	1.78 Nm/amp

Şekil 4.1 ve Çizelge 4.1 ile verilen DA motor diyagramı ve parametrelerinde,  $J$  rotor eylemsizlik momentini,  $b$  mekanik sistemin sönüm oranını,  $K_b$  elektromotor kuvvet sabitini,  $K_m$  motor sabitini,  $R_a$  motor sargı direncini,  $L_a$  indüktansı,  $P$  gücü,  $V_a$  kaynak voltajını,  $\theta$  çıkışı ifade eder.

YAA kullanılarak DA motor denetimi için PID denetleç parametrelerinin tespiti için tasarlanan blok şema Şekil 4.2 (Duman vd., 2011)' de verilmiştir.



Şekil 4.2. YAA ile PID denetleç optimizasyonu.

Şekil 4.2' de  $r(t)$  referans değerini,  $y(t)$  çıkış değerini,  $u(t)$  denetim değişkenini,  $e(t)$  hata değerini ifade eder.

YAA kullanılarak PID denetleç optimizasyonu için kullanılan algoritmanın kaba kodu aşağıdaki gibidir:

*YAA ve PID için başlangıç değerlerini ata*

*PID parametreleri ( $K_p$ ,  $K_d$  ve  $K_i$ ) değerlerinden oluşan  $N$  adet üyeyi rastgele konumlandır*

**DO**

*FOR her bir kütle için*

*PID parametrelerini uygunluk fonksiyonuna gönder ve uygunluk değerini hesapla*

*Sırasıyla kütle, kuvvet, ivme değerlerini hesapla*

*Hız ve konum güncellemesini gerçekleştir*

**END**

*En iyi, en kötü uygunluk değerlerini güncelle*

**WHILE** sonlandırma kriteri sağlanana kadar

PID parametrelerinin optimizasyonu için ölçüt fonksiyonu olarak E. 4.1' de verilen ortalama karesel hata (MSE) ölçütü kullanılmıştır.

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N (\omega_{set} - \omega_{(k)})^2 \quad (\text{E.4.1})$$

E. 4.1' de ifade edilen N toplam örnek sayısını,  $\omega_{set}$  istenilen değeri,  $\omega_{(k)}$  ise çıkış değerini belirtmektedir.

Kullanılan algoritmada örnek sayısı 10, maksimum jenerasyon sayısı olarak 100 alınmıştır. Algoritmada kullanılan parametreler;  $\varepsilon = 10^{-12}$ ,  $\alpha = 10$  ve  $G_c = 50$  olarak seçilmiştir.  $G_c$ , yerçekimi sabitidir. PID parametrelerinin sınır değerleri (Kesler vd.,2011)' nin kullandığı gibi E. 4.2' de verildiği biçimde kullanılmıştır.

$$\begin{aligned} 0 &\leq K_p \leq 10, \\ 0 &\leq K_d \leq 3, \\ 0 &\leq K_i \leq 10 \end{aligned} \quad (\text{E.4.2})$$

Yerçekimsel Arama Algoritması' na ait kodlar MATLAB programlama dilinde yazılmış ve Genuine Intel® CPU, T2300 @ 1,66 GHz, 1GB RAM donanımlı 32 bit işletim sistemine sahip bir dizüstü bilgisayarda çalıştırılmıştır.

Kesler vd., (2011)' nin çalışmalarında kullanılan algoritma kodları yazarlarının izni alınarak çalıştırılmıştır. Optimizasyon algoritmalarından TACO, GA, PSO, DE algoritmaları ile YAA' nın 50' şer defa koşturulması ile her bir algoritmaya ait PID denetleç parametre değerlerinin ve MSE' nin son nesilde bulunan değerleri sırasıyla Çizelge 4.2, Çizelge 4.3, Çizelge 4.4, Çizelge 4.5 ve Çizelge 4.6' da gösterilmiştir.

Çizelge 4.2. TACO' nun 50 defa koşturulması sonucu her bir koşturmaya ait PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri.

Koşma Zamanı	$K_p$ Değeri	$K_i$ Değeri	$K_d$ Değeri	MSE Değeri
1	7,30	7,25	2,94	178,017
2	6,77	9,84	3,00	186,141
3	6,72	6,67	2,73	180,844
4	8,10	7,25	2,94	163,613
5	7,67	9,26	2,90	164,522
6	7,30	8,68	2,70	161,630
7	8,10	8,31	2,86	157,033
8	9,15	8,15	2,95	146,212
9	7,04	6,72	2,81	178,393
10	7,20	6,88	2,54	160,242
11	8,31	9,21	3,00	158,701
12	6,72	8,89	2,87	182,117
13	7,88	8,25	3,00	167,697
14	6,83	8,15	2,73	173,740
15	7,67	8,10	2,54	148,558
16	7,51	7,35	2,98	176,060
17	7,46	9,74	2,97	170,732
18	7,51	7,20	3,00	177,374
19	7,51	8,57	2,84	165,513
20	6,67	9,95	2,60	165,806
21	6,72	6,77	2,97	193,829
22	7,04	7,72	2,95	182,775
23	6,67	8,89	2,76	176,969
24	6,98	9,84	2,79	170,346
25	6,72	8,36	2,92	186,083
26	6,67	8,94	2,73	175,050
27	7,41	7,57	2,94	174,985
28	6,67	7,30	2,49	166,123
29	7,41	8,89	2,79	164,257
30	6,67	9,79	2,92	184,090
31	6,77	6,83	2,70	177,315
32	7,35	6,67	2,86	174,846
33	8,04	7,78	2,71	152,205
34	7,72	7,25	2,87	166,838
35	7,67	6,67	2,83	167,326
36	8,04	8,41	2,86	157,649
37	7,25	9,84	2,63	156,905
38	6,98	8,52	2,65	165,304
39	6,98	8,47	2,59	161,965
40	6,93	7,25	2,71	173,426
41	8,41	7,67	2,63	143,171
42	7,51	7,25	2,68	160,856
43	6,77	7,57	2,46	161,418
44	6,93	6,93	2,76	177,157
45	7,88	7,30	2,95	167,850
46	7,67	7,41	2,68	157,654
47	6,98	6,98	2,86	181,090
48	6,67	7,46	2,60	171,888
49	6,88	7,88	2,70	171,646
50	7,41	6,98	2,98	179,308



Çizelge 4.3. GA' nın 50 defa koşturulması sonucu her bir koşturmaya ait PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri.

Koşma Zamani	KP Değeri	Ki Değeri	Kd Değeri	MSE Değeri
1	8,89	9,15	2,08	108,79
2	9,05	9,63	2,40	120,43
3	9,15	9,05	2,33	117,41
4	8,31	9,58	2,11	115,91
5	9,42	9,15	2,30	113,08
6	9,52	9,42	2,63	125,69
7	8,15	9,26	2,02	113,75
8	8,78	9,52	2,25	117,18
9	8,62	9,68	2,33	122,38
10	8,41	8,99	2,32	125,39
11	8,78	7,72	2,44	129,35
12	7,83	9,58	2,03	117,76
13	8,62	9,52	2,10	111,69
14	8,68	7,46	2,11	116,11
15	8,25	6,72	2,38	136,18
16	9,58	8,47	2,00	99,99
17	8,89	7,30	2,24	119,89
18	9,79	7,04	2,10	105,26
19	9,05	9,63	2,17	110,62
20	7,20	7,14	2,21	141,56
21	9,31	6,83	2,37	122,07
22	7,99	7,67	2,29	132,33
23	7,67	7,57	2,22	133,90
24	7,62	10	2,43	139,91
25	9,31	6,93	2,10	110,13
26	9,95	6,83	2,67	127,69
27	9,15	9,84	2,43	120,33
28	9,15	7,51	2,75	138,64
29	8,36	8,36	2,13	118,34
30	9,79	9,15	2,75	127,79
31	9,68	9,95	2,33	110,60
32	9,21	8,99	2,25	113,48
33	9,26	8,25	2,06	106,07
34	9,15	9,63	2,19	110,23
35	9,95	8,84	2,51	116,85
36	9,10	8,62	2,56	128,52
37	10	7,57	2,27	109,13
38	9,31	9,31	2,46	120,75
39	8,89	9,58	2,46	125,17
40	7,41	7,99	2,21	135,79
41	8,04	9,89	2,02	113,90
42	10	7,35	2,57	121,87
43	9,74	9,42	2,35	111,49
44	8,73	8,04	2,22	119,15
45	9,58	8,68	2,68	128,33
46	9,37	7,88	2,62	129,67
47	7,62	9,05	2,16	127,89
48	9,05	7,20	2,30	121,20
49	8,62	9,37	2,21	117,06
50	9,84	8,78	2,56	119,97

Çizelge 4.4. PSO' nun 50 defa koşturulması sonucu her bir koşturmaya ait PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri.

Koşma Zamanı	KP Değeri	Ki Değeri	Kd Değeri	MSE Değeri
1	5,46	10	0,28	41,64
2	5,59	10	0,52	55,61
3	10	10	0,99	53,30
4	10	10	0,99	53,37
5	10	10	0,42	30,99
6	10	4,15	0,73	61,13
7	10	7,54	0,86	52,19
8	10	10	0,99	53,21
9	6,38	10	0,54	51,12
10	10	10	1,14	59,04
11	10	10	0,98	52,64
12	10	10	0,66	40,30
13	10	8,85	0,85	49,17
14	10	10	1,47	72,73
15	7,05	10	0,61	51,19
16	10	10	1,00	53,59
17	10	4,66	0,49	48,51
18	10	6,10	1,83	95,45
19	3,16	10	0,97	126,12
20	3,69	6,83	1,00	129,21
21	10	10	1,04	55,42
22	1,14	4,82	0,33	166,57
23	10	10	1,18	60,73
24	10	10	1,75	83,98
25	10	10	0,58	36,85
26	10	8,13	0,93	53,86
27	10	10	1,55	75,74
28	10	10	1,22	62,48
29	3,32	10	1,07	131,25
30	9,89	10	0,46	32,87
31	10	10	1,21	62,24
32	10	10	0,60	37,80
33	1,03	5,73	0,38	162,08
34	10	8,98	0,62	39,97
35	1,71	5,46	0,31	125,67
36	10	10	0,60	37,60
37	10	10	0,80	45,46
38	10	10	1,67	80,46
39	10	8,04	0,77	47,78
40	3,89	6,69	1,05	130,09
41	2,41	10	0,72	121,07
42	6,16	9,04	0,46	50,28
43	2,88	5,16	1,03	171,09
44	3,65	9,76	0,41	66,56
45	10	10	1,13	58,65
46	10	10	1,36	67,96
47	10	10	0,57	36,51
48	10	10	1,67	80,82
49	1,46	7,25	0,47	136,45
50	4,57	10	0,96	96,84

Çizelge 4.5. DE' nin 50 defa koşturulması sonucu her bir koşturmaya ait PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri.

Koşma Zamanı	KP Değeri	Ki Değeri	Kd Değeri	MSE Değeri
1	10	10	0,12	22,32
2	10	10	0,12	22,32
3	10	10	0,12	22,32
4	10	10	0,12	22,32
5	10	7,45	0,12	26,78
6	10	10	0,12	22,32
7	10	10	0,12	22,32
8	10	10	0,12	22,32
9	10	10	0,12	22,32
10	10	10	0,12	22,32
11	10	10	0,12	22,32
12	10	9,20	0,11	23,47
13	10	10	0,12	22,32
14	10	10	0,13	22,33
15	10	10	0,12	22,32
16	10	10	0,12	22,32
17	9,62	10	0,12	22,87
18	10	10	0,12	22,32
19	10	10	0,12	22,32
20	10	10	0,12	22,32
21	9,99	10	0,12	22,32
22	9,96	8,99	0,12	23,85
23	10	10	0,12	22,32
24	10	10	0,12	22,32
25	10	10	0,12	22,32
26	10	10	0,12	22,32
27	10	10	0,12	22,32
28	10	10	0,12	22,32
29	8,22	8,91	0,10	27,12
30	10	10	0,12	22,32
31	10	10	0,12	22,32
32	10	10	0,12	22,32
33	10	10	0,12	22,32
34	10	10	0,12	22,32
35	10	10	0,12	22,32
36	9,64	9,95	0,13	22,98
37	10	10	0,12	22,32
38	10	10	0,12	22,32
39	10	10	0,12	22,32
40	10	10	0,12	22,32
41	10	10	0,12	22,32
42	10	10	0,12	22,32
43	10	10	0,12	22,32
44	10	10	0,12	22,32
45	10	9,90	0,12	22,45
46	10	10	0,12	22,32
47	10	10	0,12	22,32
48	10	10	0,12	22,32
49	10	10	0,12	22,32
50	10	10	0,12	22,32

Çizelge 4.6. YAA' nın 50 defa koşturulması sonucu her bir koşturmaya ait PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri.

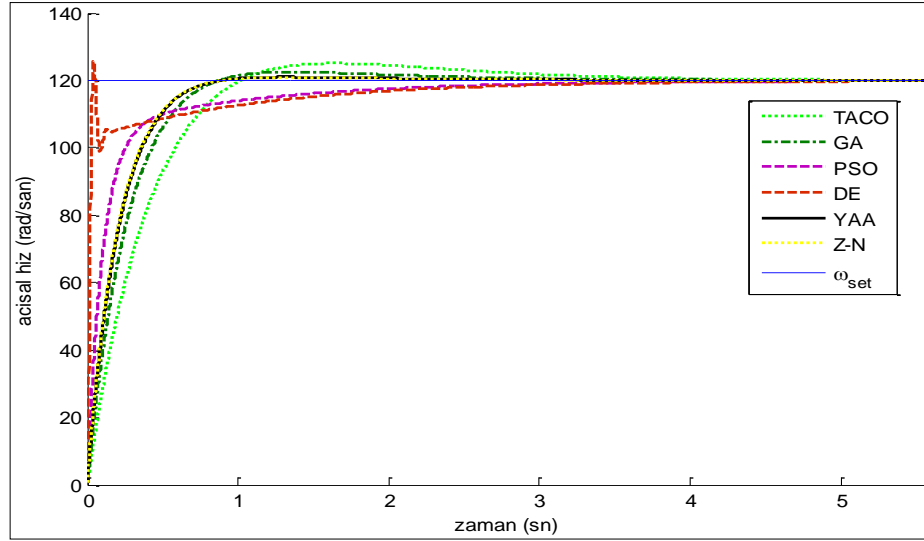
Koşma Zamanı	KP Değeri	Ki Değeri	Kd Değeri	MSE Değeri
1	10	9,57	2,39	110,57
2	9,51	8,81	1,68	86,52
3	9,01	8,37	1,79	96,22
4	10	8,26	2,44	114,50
5	10	8,88	1,89	91,03
6	10	10	2,20	101,95
7	10	6,67	2,12	105,45
8	8,61	8,12	1,65	94,04
9	10	8,02	2,37	112,09
10	9,66	8,17	2,05	101,75
11	8,59	8,97	1,97	107,30
12	9,81	8,43	2,23	107,43
13	8,35	8,59	1,53	89,62
14	9,93	6,36	2,09	105,89
15	10	8,45	1,93	93,57
16	8,44	6,74	1,62	98,19
17	10	8,04	2,18	104,58
18	10	10	2,65	120,21
19	10	9,58	2,13	100,02
20	10	10	2,08	97,36
21	10	8,33	2,06	98,87
22	10	10	1,87	88,72
23	10	8,88	2,18	102,91
24	10	8,93	1,93	92,64
25	8,54	9,64	1,84	100,72
26	10	8,68	2,17	102,70
27	8,17	8,18	1,58	94,89
28	10	8,51	1,73	85,39
29	9,80	7,50	2,06	102,52
30	9,77	8,71	2,11	102,54
31	10	8,40	2,13	101,82
32	9,87	9,14	2,00	96,16
33	10	10	2,29	105,70
34	10	10	2,29	105,60
35	10	8,90	1,79	87,17
36	8,68	6,54	1,64	97,31
37	8,88	8,37	2,02	107,66
38	10	8,09	2,24	106,72
39	10	8,78	1,89	91,48
40	10	9,46	1,97	93,47
41	8,71	8,75	1,89	102,77
42	10	10	1,95	92,00
43	8,90	8,07	1,70	93,73
44	10	9,84	2,27	105,16
45	10	9,75	2,26	104,95
46	9,82	8,44	2,04	99,69
47	10	10	1,98	93,34
48	8,41	9,79	1,82	100,92
49	8,86	8,40	1,76	96,02
50	9,65	10	2,04	98,60

Her bir algoritmanın 50 defa kořturulması sonucunda Çizelge 4.7’ de aritmetik ortalaması verilen PID parametrelerinin denetim başarımını gösteren grafik Şekil 4.3’ de verilmiştir.

Çizelge 4.7’ de en iyi başarımı gösteren algoritmaya ait veri koyu mavi ve altı çizgili olarak, en kötü başarımı gösteren algoritmaya ait veri koyu kırmızı yazı tipi ile gösterilmiştir. Çizelge 4.7’ de ki veriler dikkate alınarak DE algoritmasının parametre değerlerinin ve MSE’ nin son nesilde bulunan değerlerinin diğer algoritmalara (TACO, GA, PSO, YAA) göre en iyi sonucu, TACO’ nun ise en kötü sonucu verdiği görülmektedir.

Çizelge 4.7. Algoritmaların 50 kez kořturulma sonucunda başarımların kıyaslama ölçütleri.

Parametre	Merkezi Eğilim Ölçütü	Algoritma Türleri				
		TACO (Kesler vd.,2011)	GA (Kesler vd.,2011)	PSO (Kesler vd.,2011)	DE (Kesler vd.,2011)	YAA
Ölçüt Değeri	Ortalama	<b>169,2654</b>	120,5326	72,9129	<u>22,5821</u>	99,8092
	Ort. Sapma	8,9034	7,6136	<b>29,8510</b>	<u>0,4631</u>	5,9755
	Varyans	10,7782	9,4509	<b>37,2579</b>	<u>0,9369</u>	7,3433
Kp değeri (Son Nesilde Bulunan)	Ortalama	<u>7,3048</u>	8,9164	7,8683	<b>9,9482</b>	9,5997
	Ort. Sapma	0,4563	0,5883	<b>2,8092</b>	<u>0,0952</u>	0,5089
	Varyans	0,5573	0,7223	<b>3,1528</b>	<u>0,2583</u>	0,5976
Kd değeri (Son Nesilde Bulunan)	Ortalama	<b>2,7981</b>	2,3130	0,8846	<u>0,1195</u>	2,0099
	Ort. Sapma	0,1287	0,1647	<b>0,3227</b>	<u>0,0015</u>	0,1931
	Varyans	0,1491	0,2007	<b>0,3940</b>	<u>0,0042</u>	0,2391
Ki değeri (Son Nesilde Bulunan)	Ortalama	<u>7,9915</u>	8,5820	8,9438	<b>9,8877</b>	8,7626
	Ort. Sapma	0,8488	0,8957	<b>1,4319</b>	<u>0,2005</u>	0,7454
	Varyans	0,9869	1,0052	<b>1,7693</b>	<u>0,4188</u>	0,9502
Koşma Zaman	Ortalama	1,8160	18,3955	2,7954	1,7329	2,4185



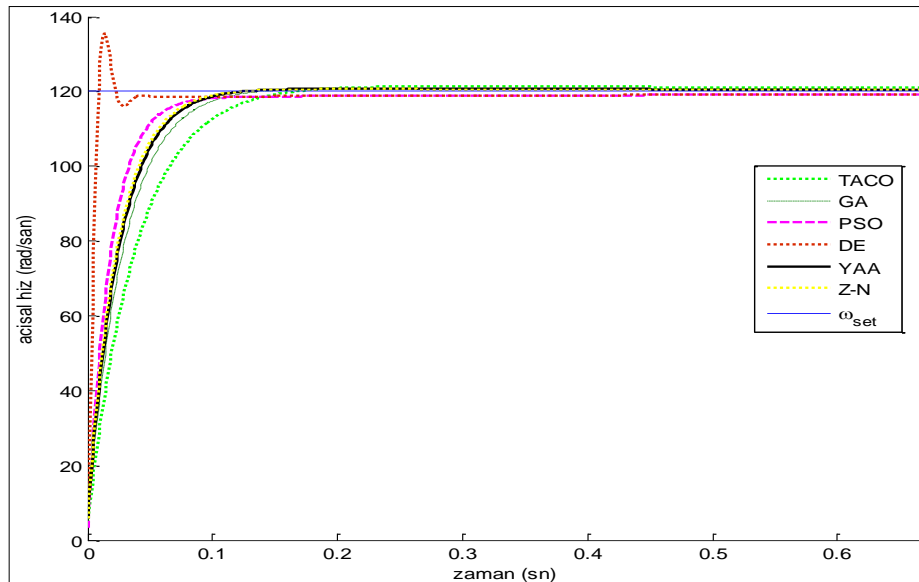
Şekil 4.3. Algoritmaların 50 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA'nın denetim başarımı (Kesler vd., 2011).

E.4.2' de verilen PID parametrelerinin sınır değerleri,  $K_p$  değeri için 0 ile 100 aralığında,  $K_d$  değeri için 0 ile 3 aralığında ve  $K_i$  değeri için 0 ile 10 aralığında kullanıldığında Şekil 4.4 elde edilmektedir.

$$0 \leq K_p \leq 100, \quad (E.4.2)$$

$$0 \leq K_d \leq 3,$$

$$0 \leq K_i \leq 10$$



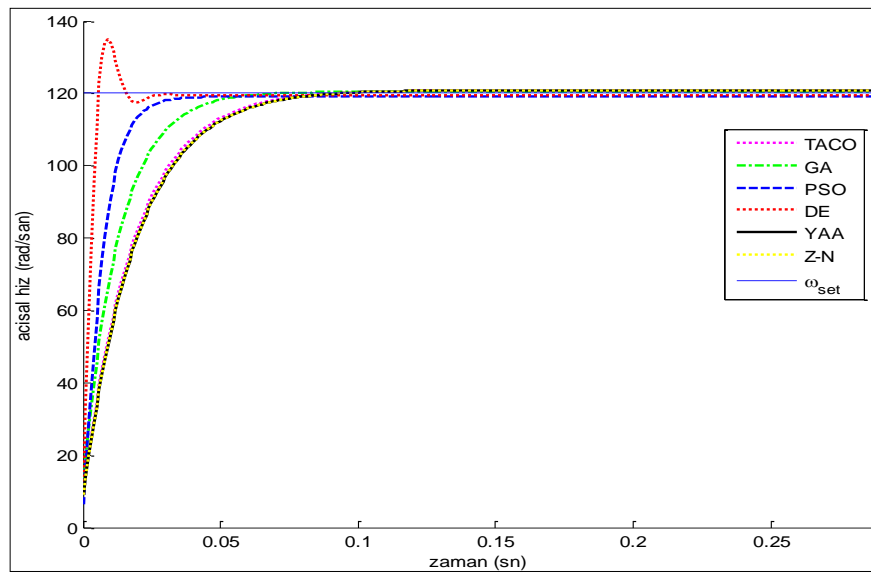
Şekil 4.4. Algoritmaların 50 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA'nın denetim başarımı ( $0 \leq K_p \leq 100$ ) (Kesler vd., 2011).

E.4.3' de verilen PID parametrelerinin sınır deęerleri,  $K_p$  deęeri için 0 ile 200 aralıęında,  $K_d$  deęeri için 0 ile 3 aralıęında ve  $K_i$  deęeri için 0 ile 10 aralıęında kullanıldıęında Őekil 4.5 elde edilmektedir.

$$0 \leq K_p \leq 200, \quad (E.4.3)$$

$$0 \leq K_d \leq 3,$$

$$0 \leq K_i \leq 10$$

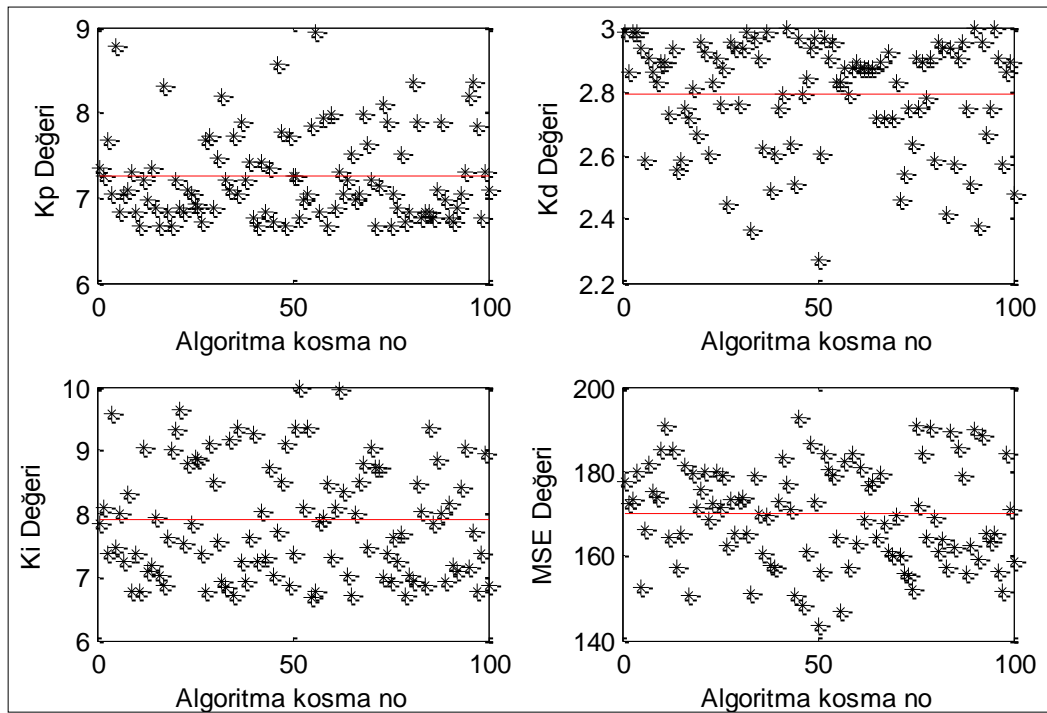


Şekil 4.5. Algoritmaların 50 defa kořturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA'nın denetim bařarımı ( $0 \leq K_p \leq 200$ ) (Kesler vd., 2011).

Elde edilen sonuçlar dikkatle incelendięinde 50 defa kořturulan algoritmaların  $K_p$  sınır aralıkları  $0 \leq K_p \leq 10$ ,  $0 \leq K_p \leq 100$  ve  $0 \leq K_p \leq 200$  olarak deęiřtirildięinde ortalama PID denetleç parametreleri ile kıyaslamasını yaptığımız YAA'nın denetim bařarımı  $K_p$  sınır deęeri artırıldıęında düşmektedir.

Kıyaslanan algoritmalarından DE algoritmasının sistemin referans deęerini ařtıęı görölse de algoritmalar içerisinde en iyi bařarımı gösterdięi açıkça görölmektedir.

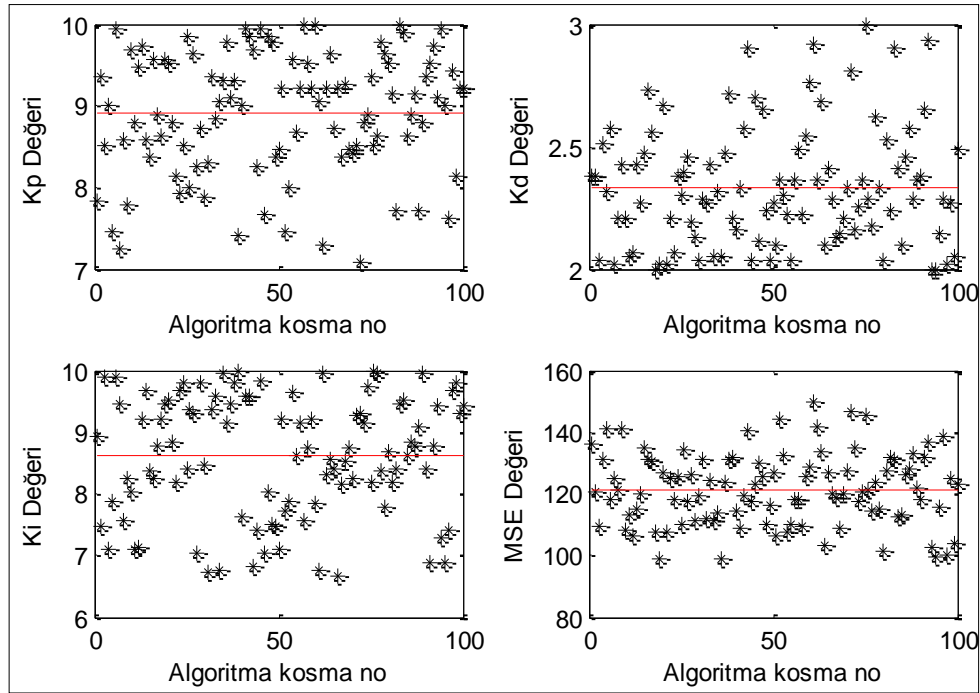
Kesler vd., (2011)' nin çalışmalarında kullanılan algoritma kodları yazarlarının izni alınarak çalıştırılmıştır. Her bir optimizasyon algoritması (TACO, GA, PSO, DE) ile YAA' nın 100' er defa koşturulması ile optimizasyonu gerçekleştirilen her bir algoritmaya ait PID denetleç parametre değerlerinin ve MSE' nin son nesilde bulunan değerlerinin optimum değerleri sırasıyla Şekil 4.6, Şekil 4.7, Şekil 4.8, Şekil 4.9 ve Şekil 4.10' da verilmiştir.



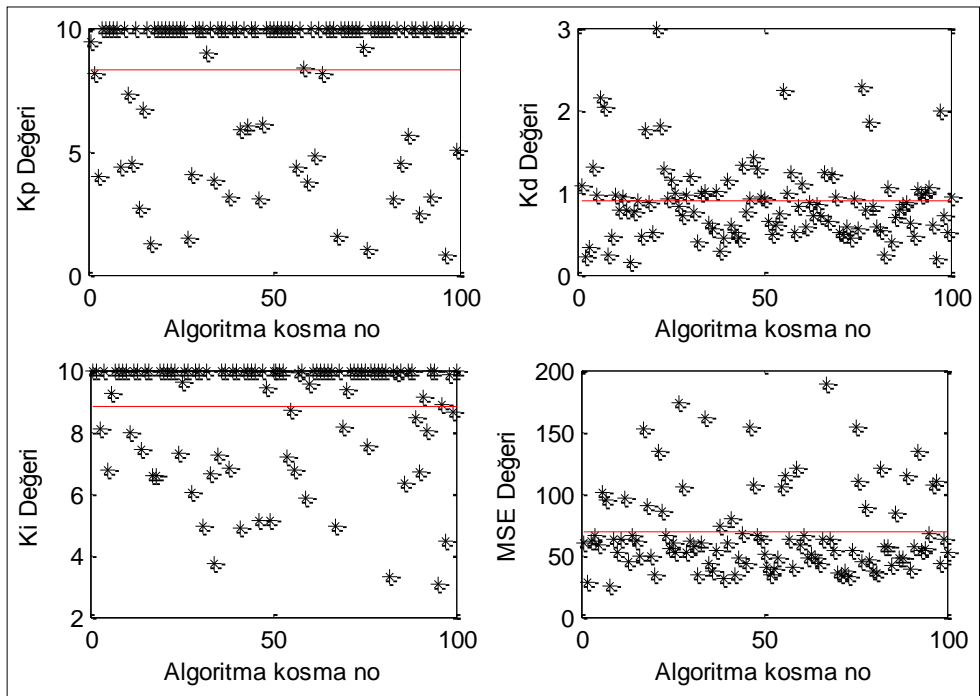
Şekil 4.6. TACO' nun 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri (Kesler vd., 2011).

Şekillerde ki kırmızı çizgi referans olarak adlandırılır. Her bir yıldız ise her koşturmada elde edilen parametre değerlerini sembolize etmektedir. Elde edilmesi beklenen yıldızların bir değerde yoğunlaşmış bir bütün oluşturmasıdır.

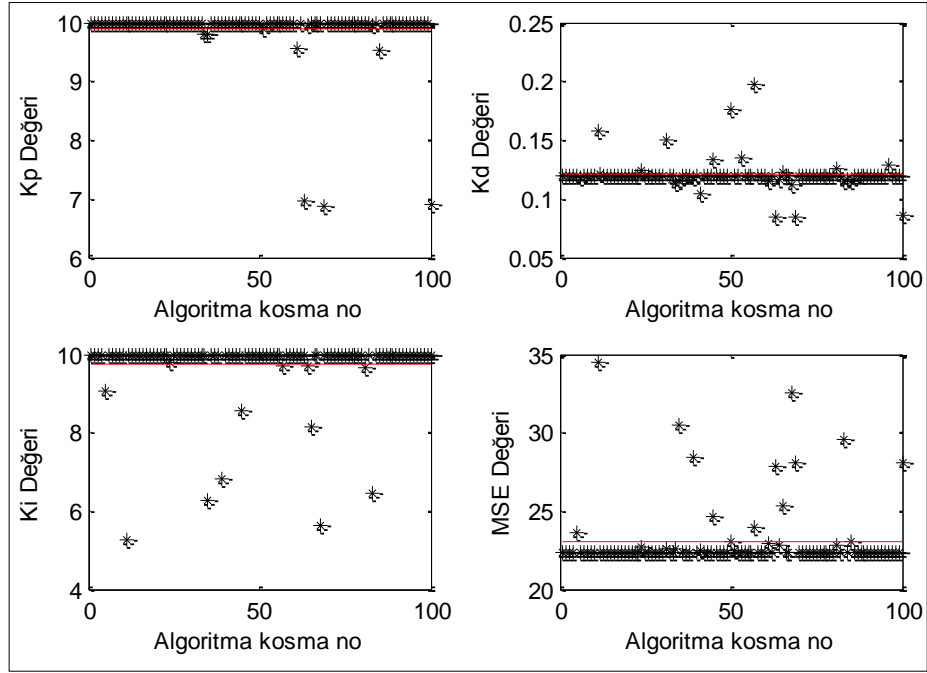




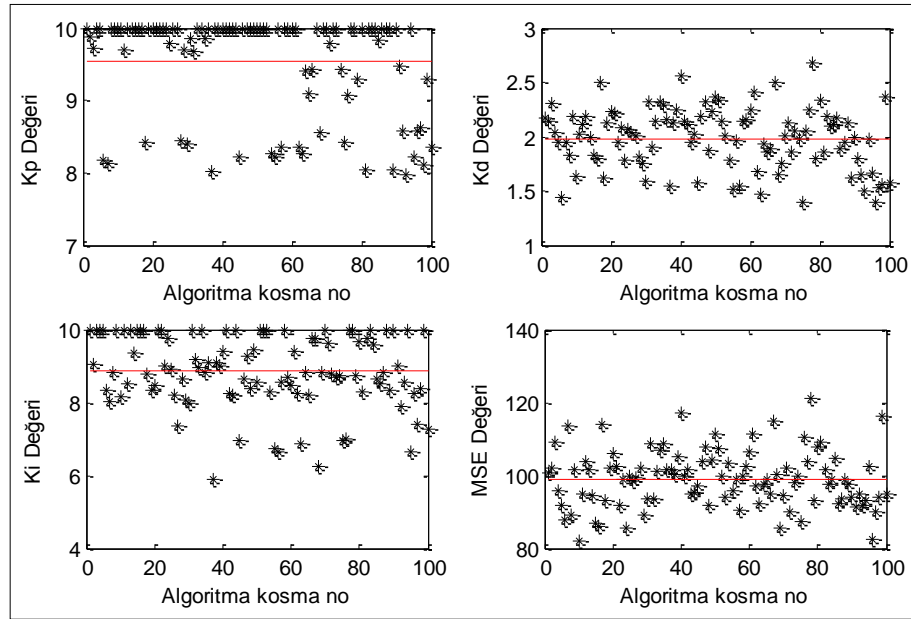
Şekil 4.7. GA' nın 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri (Kesler vd., 2011).



Şekil 4.8. PSO' nun 100 defa koşturulması sonucu her bir koşturma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri (Kesler vd., 2011).



Şekil 4.9. DE' nin 100 defa çalıştırılması sonucu her bir çalıştırma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri (Kesler vd., 2011).



Şekil 4.10. YAA' nın 100 defa çalıştırılması sonucu her bir çalıştırma sonunda elde edilen PID denetleç parametreleri  $K_p$ ,  $K_d$ ,  $K_i$  ve MSE değerleri.

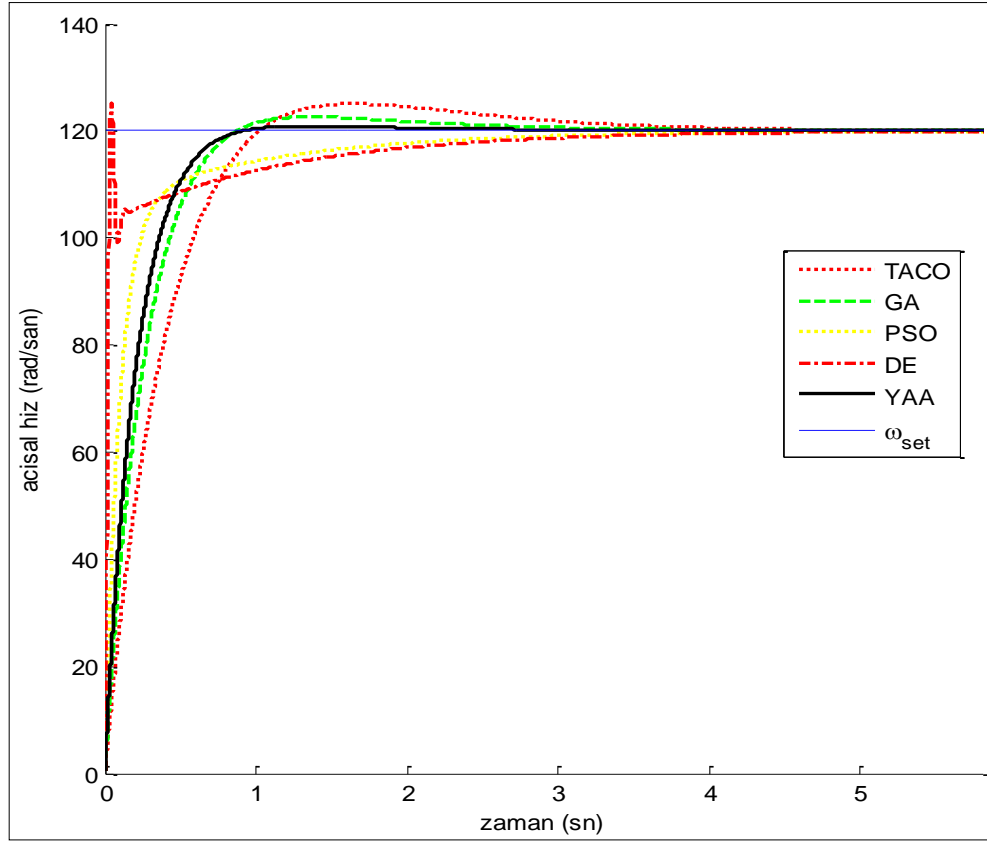
Şekillerden görüldüğü üzere, birbirine yakın değerler en çok DE algoritmasının kullanılması durumunda elde edilmiştir. Daha sonra PSO ve YAA gelmektedir. Şekillerde, GA ve TACO' nun her çalıştırmada dağınık ve birbirine uzak değerler verdiği görülmüştür.

Çizelge 4.8. Algoritmaların 100 kez koşturulma sonucunda başarımları kıyaslama ölçütleri.

Parametre	Merkezi Eğilim Ölçütü	Algoritma Türleri				
		<i>TACO</i> (Kesler vd.,2011)	<i>GA</i> (Kesler vd.,2011)	<i>PSO</i> (Kesler vd.,2011)	<i>DE</i> (Kesler vd.,2011)	<i>YAA</i>
Ölçüt Değeri	Ortalama	<b>169,9577</b>	121,5293	68,2663	<u>23,0508</u>	99,0745
	Ortalama Sapma	9,8496	9,5671	<b>26,3428</b>	<u>1,2089</u>	6,1617
	Varyans	11,6536	11,6442	<b>34,7459</b>	<u>2,1760</u>	7,7941
Kp değeri (Son Nesilde Bulunan)	Ortalama	<u>7,2556</u>	8,9090	8,2644	<b>9,8917</b>	9,5301
	Ortalama Sapma	0,4192	0,6242	<b>2,3826</b>	<u>0,1963</u>	0,6038
	Varyans	<u>0,5233</u>	0,7477	<b>2,8441</b>	0,5275	0,7084
Kd değeri (Son Nesilde Bulunan)	Ortalama	<b>2,7903</b>	2,3335	0,8942	<u>0,1206</u>	1,9823
	Ortalama Sapma	0,1422	0,1973	<b>0,3425</b>	<u>0,0049</u>	0,2279
	Varyans	0,1717	0,2465	<b>0,4879</b>	<u>0,0127</u>	0,2794
Ki değeri (Son Nesilde Bulunan)	Ortalama	<u>7,8947</u>	8,6138	8,8419	<b>9,7487</b>	8,8785
	Ortalama Sapma	0,7720	0,8400	<b>1,4952</b>	<u>0,4419</u>	0,8183
	Varyans	0,8974	0,9799	<b>1,8365</b>	<u>0,8905</u>	1,0165
Koşma Zaman	Ortalama	15,84	18,4725	3,0168	1,7332	3,1191

Çizelge 4.7 ve 4.8’ de en iyi başarımları gösteren algoritmaya ait veri koyu mavi ve altı çizgili olarak, en kötü başarımları gösteren algoritmaya ait veri koyu kırmızı yazı tipi ile gösterilmiştir.

Her bir algoritmanın 100 defa koşturulması sonucunda bulunan Çizelge 4.8’ de aritmetik ortalaması verilen PID parametrelerinin denetim başarımlarını gösteren grafik Şekil 4.11’ de verilmiştir.



Şekil 4.11. Algoritmaların 100 defa koşturulması sonucu elde edilen ortalama PID denetleç parametreleri ile YAA'nın denetim başarımı.

Şekil 4.11' de verilen grafikten elde edilen sonuçlara göre algoritmalar içerisinde en iyi başarımı DE algoritmasının, en kötü başarımı ise TACO algoritmasının gösterdiği anlaşılmaktadır. Kıyaslaması yapılan YAA'nın ise TACO'dan daha iyi, DE'den daha kötü ortalama bir başarımla sergilediği görülmektedir.

## 5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, güncel optimizasyon algoritmalarından Rashedi ve arkadaşları tarafından önerilen YAA kullanılarak DA motor denetimi için PID denetleç parametrelerinin tespiti yapılmıştır. PID denetleç parametrelerinin optimizasyonunda diğer bazı algoritmaları (TACO, GA, PSO, DE) kullanan Kesler vd., (2011)' nin çalışması ile YAA' nın başarımlarını kıyaslaması yapılmıştır. Bu optimizasyon problemi için en uygun algoritma belirlenmeye çalışılmıştır.

Bu tez çalışmasının, Çizelge 4.7. ve 4.8.' de ki verilere dayalı olarak Kesler vd., (2011)' nin çalışmasına kıyasla YAA' nın PID denetleç parametreleri ile denetim başarımının, TACO ve GA' ya göre daha iyi, PSO ve DE' ye göre daha kötü olduğu anlaşılmaktadır.

Çalışma sonucunda elde edilen verilere dayanarak YAA' nın PSO ve DE' ye göre kötü sonuçlar verdiği görülsede PID denetleç parametrelerinin optimizasyonunda YAA' nın kullanılabilir olduğuna varılabilir. Kıyaslaması yapılan sezgisel optimizasyon algoritmaları dışında ileride YAA' nın PID denetleç parametrelerini elde etmedeki başarımını diğer güncel sezgisel optimizasyon algoritmaları (Yapay Arı Kolonisi Algoritması - YAKA, Kedi Sürüsü Optimizasyonu – KSO, Ateş Böceği Algoritması – ABA, Kurt Kolonisi Algoritması ) ile de kıyaslanabilir.

## KAYNAKLAR

- Akay, R., “Memetik Algoritma Kullanarak PID Denetleyici Tasarımı”, Yüksek Lisans Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 2006.
- Alli, H. ve Kaya, M., “Genetik Algoritma Kullanarak PID Kontrol Parametrelerinin Bulunması”, *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 13: 1-8 (2001).
- Ang, K. H., Chong, G. and Li, Y., “PID Control System Analysis, Design, and Technology”, *IEEE Transactions on Control Systems Technology*, 13(4): 559-576 (2005).
- Astrom, K. J. and Hagglund, T., “Automatic tuning of simple regulators with specifications on phase and amplitude margins”, *Automatica*, 20(5): 645-651 (1984).
- Ayaz, B., “PID Kontrolörlerinin Optimal Parametrelerinin Belirlenmesi Amacıyla Bir Bulanık Mantık Karar Mekanizması Tasarımı”, Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 2011.
- Baba, A. F., “Bilgisayar Destekli PID Kontrol”, Yüksek Lisans Tezi, *Marmara Üniversitesi Elektrik Elektronik Mühendisliği*, İstanbul, 1989.
- Bağış, A. ve Çetin, O., “Doğrusal Olmayan Bir Sistemin Bulanık Kural Parametrelerinin PSO Algoritması ile Optimize Edilmesi”, *Elektronik Letters on Science & Engineering*, 5(1) (2009).
- Bağış, S., “Yapay Zeka Algoritmaları Kullanılarak Sistem Modelleme”, Yüksek Lisans Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 2009.
- Bassi, S. J., Mishra, M. K. and Omizegba, E.E., “Automatic Tuning of Proportional-Integral-Derivative (PID) Controller Using Particle Swarm Optimization (PSO) Algorithm”, *International Journal of Artificial Intelligence & Applications (IJAIA)*, 2(4): 25-34 (2011).
- Boz, A. F.,a-) <http://web.sakarya.edu.tr/~afboz/control/bolum1.html>, (Erişim Tarihi: 19 Mayıs 2012).
- Boz, A. F.,b-) <http://web.sakarya.edu.tr/~afboz/control/bolum10.html>, (Erişim Tarihi: 27 Mayıs 2012).
- Cihan, A. ve Karakuzu, C., “Bulanık-PID Kontrolör Parametrelerinin Diferansiyel Gelişim Algoritması ile En Uygulanması”, *Electrical, Electronics and Computer Engineering (ELECO)*, Bursa, (2008).

## KAYNAKLAR(Devam Ediyor)

- Dandıl, B., Gökbulut, M. ve Ata, F., “Genetik Algoritma ile Model Referans – PID Denetleyici Tasarımı”, *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 14(2): 33-43 (2002).
- Donghai, L., Furong G., Yali X. and Chongde L., “Optimization of Decentralized PI/PID Controllers Based on Genetic Algorithm”, *Asian Journal of Control*, 9(3): 306-316 (2007).
- Dorf, R. C. and Bishop, R. H., “Modern Control Systems”, *Prentice Hall 10<sup>th</sup> Edition*, USA, 2005.
- Duman, S., Maden, D. ve Güvenç, U., “Determination of the PID Controller Parameters for Speed and Position Control of DC Motor Using Gravitational Search Algorithm”, *Electrical, Electronics and Computer Engineering (ELECO)*, Bursa, 225-229 (2011).
- Emel, G. G. ve Taşkın, Ç., ”Genetik Algoritmalar ve Uygulama Alanları”, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 11(1): 129-152 (2002).
- Engin, O., “Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu, Yayınlanmamış Doktora Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 2001.
- Gass, S., “Making Decisions with Precision”, <http://www.businessweek.com/stories/2000-10-29/making-decisions-with-precision#B3705139>, 2000.
- Goldberg, D. E., “Genetic Algorithms in Search Optimization and Machine Learning”, *Addison Wesley Publishing Company*, USA, 1989.
- Gözde, H., Taplamacıoğlu, M. C., Kocaarslar, İ. ve Şenol, M. A., “İki Bölge Ara-Isıtmalı Termal Güç Sisteminin Yük-Frekans Kontrolü için Parçacık Sürüsü Optimizasyonu Tabanlı PI-Kontrolör”, *Isı Bilimi ve Tekniği Dergisi*, 30(1): 13-21 (2010).
- Hiroyasu, T., Miki, M., Ono, Y. and Minami, Y., “Ant colony for continuous functions”, *The Science and Engineering*, Doshiska University, (2000).
- Holland, J. H., “Adaptation in Natural and Artificial Systems”, *University of Michigan Press*, Ann Arbor, MI, 1975.

## KAYNAKLAR(Devam Ediyor)

- Hsiao, Y. T., Chuang, C. L. and Chien, C. C., “Ant Colony Optimization for Designing of PID Controllers”, *International Symposium on Computer Aided Control Systems Design (IEEE)*, Taipei, Taiwan, 321-326 (2004).
- Hurma, H. N., “PID Kontrolör ve PLC Uygulaması”, Yüksek Lisans Tezi, *İstanbul Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 1998.
- Jang, J. S. R., “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, *Prentice-Hall*, USA, 173-196 (1997).
- Karaboğa, D., “Yapay Zeka Optimizasyon Algoritmaları”, *Nobel Yayın Dağıtım*, Ankara, 2011.
- Karaboğa, D. ve Kalınlı, A., “Tuning PID Controller Parameters Using Tabu Search Algorithm”, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 134-136 (1996).
- Karaboğa, N., Güney, K. and Akdağlı, A., “Null Steering of Linear Antenna Arrays with Use of Modified Touring Ant Colony Optimization Algorithm”, *International Journal of RF and Microwave Computer-Aided Engineering*, 12(4): 375-383 (2002).
- Kazak, N., “Geliştirilmiş Yerçekimsel Arama Algoritması”, Yayınlanmamış Yüksek Lisans Tezi, *Bilecik Üniversitesi Fen Bilimleri Enstitüsü*, Bilecik, 2011.
- Kennedy, J. and Eberhart, R. C. “Particle Swarm Optimization”, *IEEE International Conference Neural Network IV*, Perth, Australia, 1942-1948, (1995).
- Keskintürk, T., “Diferansiyel Gelişim Algoritması”, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 5(9): 85-99 (2006).
- Kesler, M., Karakuzu, C. ve Yüzgeç U., “Güncel Sezgisel Arama Algoritmalarının Denetleyici Parametrelerinin Optimizasyonunda Başarım Kıyaslaması”, *6<sup>th</sup> International Advanced Technologies Symposium (IATS'11)*, Elazığ, 281-286 (2011).
- Kuo, B. C., “Otomatik Kontrol Sistemleri”, Çeviren: Prof. Dr. Atilla Bir, *Literatür Yayıncılık*, İstanbul, 2006.



## KAYNAKLAR(Devam Ediyor)

- Li, Y., Ang, K. H. and Chong, G. C. Y., “Patents, software, and hardware for PID control: an overview and analysis of the current art”, *in IEEE Control Systems*, 26: 42-54 (2006).
- Mitchell, M., “Introduction to Genetic Algorithms”, *Bradford Pres*, London, 10-48 (1998).
- Ogata, K., “Modern Control Engineering 2<sup>nd</sup> Ed.”, *Prentice-Hall*, Inc., USA, 1990.
- Özyön, S., Yaşar, C. ve Temurtaş, H.,”Diferansiyel Gelişim Algoritmasının Valf Nokta Etkili Konveks Olmayan Ekonomik Güç Dağıtım Problemlerine Uygulanması”, *6<sup>th</sup>International Advanced Technologies Symposium (IATS’11)*, Elazığ, 181-186 (2011).
- Pham, D. T. and Karaboğa, D., “Intelligent Optimisation Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Neural Networks, Advanced Manufacturing Series”, *Springer-Verlag*, London, 2000.
- Pinedo, M., “Scheduling: theory, algorithms, and systems”, *Prentice-Hall*, Englewood Cliffs, NJ., 1995.
- Polya, G., “How to Solve it”, *Princeton University Pres*, Princeton, 1948.
- Rashedi, E., Nezamabadi, H. and Saryazdi, S., “GSA: A Gravitational Search Algorithm”, *Information Sciences*, 179(13): 2232-2248 (2009).
- Romanycia, M. H. J. and Pelletier, F. F., “What is a heuristic”, *Computer Intelligence*, 1: 47-58 (1985).
- Saynak, U., “Karıncı Kolonisi Optimizasyon Algoritmaları”, <http://bilimselkonular.com/index.php/ueye-bloglarından/KarA-nca-Kolonisi-Optimizasyon-AlgoritmalarA-.html>, 2002.
- Serway, R. A. ve Beichner, R. J., “Fen ve Mühendislik için Fizik 1”, Prof. Dr. Kemal Çolakoğlu, *Palme Yayıncılık*, Ankara, 2007.
- Shyr, W. J., Wang, B. W., Yeh, Y. Y. and Su, T. J., “Design of Optimal PID Controllers Using Memetic Algorithm”, *Proceedings of the American Control Conference*, Anchorage, 2130-2131 (2002).
- Storn, S. and Price, K., “Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces”, *ICSI Tech. Rep.*, TR-95-012 (1995).

## KAYNAKLAR(Devam Ediyor)

- Tan, K. C., Li, Y., Murray-Smith, D. J., Sharman, K. C., "System identification and linearization using genetic algorithms with simulated annealing", *in Proc. of First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA)*, 164-169, (1995).
- Thomas, N. and Poongodi, P., "Position control of DC motor using genetic algorithm based PID controller", *Proceedings of the World Congress on Engineering (WCE)*, London, 1618-1622 (2009).
- Tutkun, N. ve Dinçer, M., "DA Şönt Motoru için PID Denetleyici Parametrelerinin Genetik Algoritmalar Yöntemi ile Belirlenmesi", *Electrical, Electronics and Computer Engineering (ELECO)*, Bursa, 330-333 (2010).
- Ünal, M., "PID Kontrolörün Karınca Kolonisi / Genetik Algoritma Tabanlı Optimizasyonu ve Gunt RT 532 Basınç Prosesinin Kontrolü", Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 2008.
- Wang, Q., Lee, T., Fung, H., Bi, Q. and Zhang, Y., "PID Tuning for Improved Performance", *IEEE Transactions on Control Systems Technology*, 7: 457-465 (1999).
- Yalçın, N., "Sezgisel Algoritma Öğrenmeli Yapay Sinir Ağları ile Epilepsi Hastalığının Teşhisi", Yayınlanmamış Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya, 2012.
- Yıkan, F., "Dayanımlı PID Kontrolör Tasarım Metotlarının Araştırılması ve Geliştirilmesi", Yüksek Lisans Tezi, *İnönü Üniversitesi Fen Bilimleri Enstitüsü*, Malatya, 2005.

## ÖZGEÇMİŞ

### Kişisel Bilgiler

**ADI SOYADI** : Semih ÇAKIR  
**DOĞUM YERİ VE TARİHİ** : KDZ. EREĞLİ / 19.08.1986

### Eğitim Durumu

**Lisans Öğrenimi** : Anadolu Üniversitesi Eğitim Fakültesi  
Bilgisayar ve Öğretim Teknolojileri Öğretmenliği  
**Bildiği Yabancı Diller** : İngilizce, İspanyolca (Başlangıç), Rusça(Başlangıç)

### İş Deneyimi

**Stajlar** : Mehmet Akif Ersoy İ.Ö.O. /ESKİŞEHİR  
Fatih Sultan Mehmet İ.Ö.O. /ESKİŞEHİR  
**Çalıştığı Kurumlar** : Eskişehir Tepebaşı Halk Eğitim Merkezi  
Bilecik Üniversitesi Meslek Yüksekokulu (31.Madde)

### **İletişim**

Adres: Yeni Bağlar Mh. Beraberlik Sk. İter Apt. 12/6 Tepebaşı / ESKİŞEHİR

Tel: 0 506 642 93 40

E-Posta Adresi: semihc@anadolu.edu.tr