

**DÖRT ROTORLU BİR İNSANSIZ HAVA ARACININ  
KESİR DERECELİ DENETLEYİCİ İLE KONTROLÜ**

Hasan SARİBAŞ  
Yüksek Lisans Tezi

Havacılık Elektrik ve Elektronik Anabilim Dalı  
Temmuz-2015

## JÜRİ VE ENSTİTÜ ONAYI

Hasan Saribaş'ın “**Dört Rotorlu Bir İnsansız Hava Aracının Kesir Dereceli Denetleyici İle Kontrolü**” başlıklı **Havacılık Elektrik ve Elektronik** Anabilim Dalındaki, Yüksek Lisans Tezi 20.07.2015 tarihinde aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Adı Soyadı	İmza
Üye (Tez Danışmanı) : Yrd. Doç. Dr. Sinem KAHVECİOĞLU	.....
Üye : Yrd. Doç. Dr. Yasemin IŞIK	.....
Üye : Yrd. Doç. Dr. Ayşe KAHVECİOĞLU	.....

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

## ÖZET

Yüksek Lisans Tezi

### DÖRT ROTORLU BİR İNSANSIZ HAVA ARACININ KESİR DERECELİ DENETLEYİCİ İLE KONTROLÜ

Hasan SARİBAŞ

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Havacılık Elektrik ve Elektronik Anabilim Dalı

Danışman: Yrd. Doç. Dr. Sinem KAHVECİOĞLU

2015, 71 sayfa

Bu çalışmada, dört rotorlu, olduğu yerden dikey kalkış/iniş yapabilen, quadrotor tipi bir insansız hava aracı klasik *PID* ve kesir dereceli *PID* denetleyicileri kullanılarak kontrol edilmiştir. Kesir dereceli *PID* denetleyicisi, klasik *PID* denetleyicisinde bulunan  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerine ek olarak,  $\lambda$  ve  $\mu$  parametrelerine de sahiptir. Bu parametrelerin belirlenmesinde sezgisel optimizasyon yöntemlerinden olan Genetik Algoritma ve Parçacık Sürü Optimizasyonu kullanılmıştır. Sezgisel yöntemleri ile minimum yapılmaya çalışılan farklı amaç fonksiyon kriterleri için kesir dereceli *PID* ile klasik *PID* denetleyicilerinin performansları kıyaslanmıştır. ITSE kriterinin ve parçacık sürü optimizasyonu yönteminin kesir dereceli *PID* denetleyicilerde daha iyi sonuç verdiği gözlenmiştir.

**Anahtar sözcükler:** Kesirli Dereceli *PID*, Quadrotor, Sezgisel Optimizasyon, Genetik Algoritma, Parçacık Sürü Optimizasyonu.

## ABSTRACT

Master of Science Thesis

### CONTROL OF AN QUADROTOR UNMANNED AERIAL VEHICLE WITH A FRACTIONAL ORDER CONTROLLER

Hasan SARİBAŞ

Anadolu University  
Graduate School of Sciences  
Department of Avionics

Supervisor: Yrd. Doç. Dr. Sinem KAHVECİOĞLU

2015, 71 pages

In this thesis, four-rotors quadrotor (quadrocopter) type unmanned aerial vehicle with the capability of vertical take-off/landing (VTOL) is controlled by both classical *PID* and fractional order *PID* controllers. In addition to classical *PID* controller parameters  $K_p$ ,  $K_i$  and  $K_d$ , fractional order *PID* controllers have also  $\lambda$  and  $\mu$  parameters. Heuristic optimization tools genetic algorithm and particle swarm optimization are used to determine these five different parameters. Fractional order *PID* and classical *PID* controllers are compared in terms of the objective function criteria values which are tried to be minimized by using heuristic optimization methods. Observations showed that using ITSE criteria and particle swarm optimization in fractional order *PID* controllers give best results.

**Key Words:** Fractional Order *PID*, Quadrotor, Heuristic Optimization, Genetic Algorithm, Particle Swarm Optimization.

## TEŐEKKÜR

Tezin hazırlanmasında fikir, görüş ve desteklerini paylaşan değerli hocam ve danışmanım Yrd. Doç. Dr. Sinem KAHVECİOĐLU'na, tezin her aşamasında manevi desteđiyle yanımda bulunan eşim Araş. Gör. Sevcan SARİBAŐ'a teşekkür ederim.

TÜBİTAK/Bideb programına tez sürecinde verdiği destekten dolayı teşekkürlerimi sunarım.

Hasan SARİBAŐ

## 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>ŞEKİLLER DİZİNİ .....</b>	<b>vi</b>
<b>ÇİZELGELER DİZİNİ .....</b>	<b>viii</b>
<b>SİMGELER DİZİNİ .....</b>	<b>ix</b>
<b>KISALTMALAR DİZİNİ .....</b>	<b>ixi</b>
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. LİTERATÜR TARAMASI .....</b>	<b>5</b>
2.1. Kesir Dereceli Sistemler .....	5
2.2. Kesir Dereceli Denetleyiciler ile İnsansız Hava Araçlarının Kontrolü ....	9
2.3. Sezgisel Optimizasyon Yöntemlerinin Kontrol Problemlerine Uygulanması.....	11
<b>3. KESİR DERECELİ SİSTEMLER .....</b>	<b>13</b>
3.1. Kesir Dereceli Sistemlerin Farklı Tanımları .....	13
3.1.1. Riemann-Liouville tanımı.....	13
3.1.2. Grünwald-Letnikov tanımı .....	14
3.1.3. Caputo tanımı.....	15
3.2. Kesir Dereceli Sistemlerin Laplace Dönüşümleri.....	15
3.3. Kesir Dereceli Denetleyiciler.....	18
<b>4. SEZGİSEL OPTİMİZASYON .....</b>	<b>21</b>
4.1. Genetik Algoritma .....	21
4.2. Parçacık Sürü Optimizasyonu.....	25

4.3. Denetleyici Parametrelerinin Belirlenmesinde PSO ve GA Kullanımı.....	29
<b>5. QUADROTOR .....</b>	<b>30</b>
5.1. Quadrorotor Hareket Tanımları.....	30
5.1.1. Düşey eksenindeki hareket ve havada asılı kalma.....	31
5.1.2. Yalpalama açısı (roll angle, $\phi$ ).....	31
5.1.3. Yunuslama açısı (pitch angle, $\theta$ ).....	32
5.1.4. Sapma açısı (yaw angle, $\psi$ ).....	32
5.2. Quadrorotorun Matematiksel Modeli.....	33
5.3. QBall-X4 Quadrorotor .....	36
<b>6. YAPILAN ÇALIŞMALAR VE BENZETİMLER .....</b>	<b>39</b>
6.1. QBall-X4 Yükseklik Kontrolü.....	41
6.2. $x$ ve $y$ Pozisyon Kontrolü .....	51
6.2.1. Yunuslama ve yalpalama açılarının kontrolü.....	52
6.3. Sapma Açısı Kontrolü.....	57
6.4. Quadrorotorun $x$ , $y$ , $z$ Eksenlerinde Kontrolü.....	60
<b>7. SONUÇLAR .....</b>	<b>63</b>
<b>KAYNAKLAR .....</b>	<b>65</b>

## ŞEKİLLER DİZİNİ

2.1. Bilinen ilk quadrotor .....	9
3.1. $y=100$ fonksiyonunun kesir dereceli türev ifadeleri .....	17
3.2. $y=100$ fonksiyonunun kesir dereceli integral ifadeleri .....	317
3.3. $\lambda$ ve $\mu$ değişimine göre denetleyici tipleri.....	19
3.4. $PI^\lambda D^\mu$ denetleyicinin dahil olduğu sistemin blok diyagramı.....	20
3.5. $PI^\lambda D^\mu$ denetleyici iç yapısını blok diyagramı .....	20
4.1. Çaprazlama öncesi kromozomlar.....	22
4.2. Çaprazlama sonrası kromozomlar.....	23
4.3. Mutasyon öncesi kromozom .....	23
4.4. Mutasyon sonrası kromozom .....	23
4.5. Genetik algoritma yöntemi akış şeması .....	24
4.6. Parçacık sürü optimizasyonu yöntemi akış şeması .....	26
4.7. Hız ve konumun vektörel güncellenmesi.....	28
5.1. Quadrotor üzerine etkiyen kuvvetler , açılar ve hareket eksenleri .....	30
5.2. Havada asılı kalma .....	31
5.3. Yalpalama hareketi.....	31
5.4. Yunuslama hareketi.....	32
5.5. Sapma hareketi .....	32
5.6. QBall-X4 .....	37
6.1. QBall-X4 kontrol çevrimi blok diyagramı .....	40
6.2. Yükseklik kontrolü blok diyagramı .....	42
6.3. GA $PID$ ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi .....	43
6.4. GA $PID$ ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması.....	44
6.5. GA $KD PID$ ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi .....	45
6.6. GA $KD PID$ ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması .....	46
6.7. PSO $PID$ ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi .....	47



6.8. PSO <i>PID</i> ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması .....	48
6.9. PSO KD <i>PID</i> ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi .....	49
6.10. PSO KD <i>PID</i> ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması .....	50
6.11(a). $x$ eksenli kontrol blok diyagramı .....	51
6.11(b). $y$ eksenli kontrol blok diyagramı .....	51
6.12. Yunuslama ve yalpalama açılarının kontrol blok diyagramı .....	52
6.13. Açı kontrolünde farklı denetleyici ve optimizasyon yöntemleri için amaç fonksiyon değerlerinin değişimi.....	54
6.14. Açı kontrolünde farklı denetleyici ve optimizasyon yöntemleri için yunuslama ve yalpalama açılarının karşılaştırılması.....	54
6.15. $x$ , $y$ pozisyon kontrol blok diyagramı.....	55
6.16. Pozisyon kontrolünde farklı denetleyici ve optimizasyon yöntemleri için amaç fonksiyon değerlerinin değişimi.....	56
6.17. Pozisyon kontrolünde farklı denetleyici ve optimizasyon yöntemleri için $x$ , $y$ pozisyonlarının karşılaştırılması .....	57
6.18. Sapma açısı kontrol blok diyagramı.....	58
6.19. Sapma açısı kontrolünde farklı denetleyici ve optimizasyon yöntemleri için amaç fonksiyon değerlerinin değişimi.....	59
6.20. Sapma açısı kontrolünde farklı denetleyici ve optimizasyon yöntemlerinin karşılaştırılması.....	59
6.21. QBall-X4 Simulink blok diyagramı.....	60
6.22. Quadrotorun $x$ , $y$ , $z$ eksenlerinde istenen referans yörüngeyi izlemesi – Senaryo 1.....	61
6.23. Quadrotorun $x$ , $y$ , $z$ eksenlerinde istenen referans yörüngeyi izlemesi – Senaryo 2.....	62

## ÇİZELGELER DİZİNİ

3.1. Kesir dereceli denetleyici için parametre tanımları .....	18
3.2. $\lambda$ ve $\mu$ değişimine göre denetleyici tipleri .....	19
5.1. QBall-X4 parametreleri (Quanser Inc., 2010).....	38
6.1. Quadrotor kontrolü için optimizasyon parametreleri.....	40
6.2. Yükseklik kontrolünde farklı kriter değerleri için GA <i>PID</i> denetleyicisine ait performansların karşılaştırılması .....	43
6.3. Yükseklik kontrolünde farklı kriter değerleri için GA <i>KD PID</i> denetleyicisine ait performansların karşılaştırılması.....	45
6.4. Yükseklik kontrolünde farklı kriter değerleri için PSO <i>PID</i> denetleyicisine ait performansların karşılaştırılması.....	47
6.5. Yükseklik kontrolünde farklı kriter değerleri için PSO <i>KD PID</i> denetleyicisine ait performansların karşılaştırılması.....	49
6.6. ITSE kriteri için açı kontrolünde farklı kriter değerleri için PSO <i>KD PID</i> denetleyicisine ait performansların karşılaştırılması.....	53
6.7. ITSE kriteri için pozisyon kontrolünde farklı optimizasyon ve denetleyicilere ait performansların karşılaştırılması .....	56
6.8. Sapma açısı kontrolünde ITSE kriteri için farklı optimizasyon ve denetleyicilere ait performansların karşılaştırılması .....	58
6.9. ITSE kriteri için PSO <i>KD PID</i> denetleyicisine ait parametreler.....	61

## SİMGELER DİZİNİ

$K_p$	: Oransal denetleyici katsayısı
$K_i$	: İntegral denetleyici katsayısı
$K_d$	: Türevsel denetleyici katsayısı
$\lambda$	: İntegral denetleyici katsayısının derecesi
$\mu$	: Türevsel denetleyici katsayısının derecesi
$\Gamma$	: Euler-Gama fonksiyonu
$G_{Best}$	: O zamana kadar yiyeceğe en yakın olan bireyin konumu
$P_{Best}$	: Sürü bireyinin kendinin o zamana kadar yiyeceğe en yakın olduğu konum
$c_1, c_2$	: Öğrenme faktörleri
$w$	: Atalet ağırlığı
$r_1, r_2$	: 0-1 arasında rastgele üretilen sayı
$P_c$	: Çaprazlama oranı
$P_m$	: Mutasyon oranı
$C$	: Kosinüs
$S$	: Sinüs
$R(s)$	: Laplace domeninde referans giriş sinyali
$E(s)$	: Laplace domeninde hata sinyali
$U_i$	: $i$ 'nci denetleyiciye ait kontrol sinyali
$Y(s)$	: Laplace domeninde çıkış sinyali
$T_i$	: $i$ 'nci motorun ürettiği itki kuvveti ( $N$ )
$b$	: İtki kuvveti katsayısı ( $N$ )
$a$	: İvme ( $m/s^2$ )
$d$	: Sürüklenme kuvveti katsayısı ( $N$ )
$u$	: Eyleyici girişi
$v$	: Eyleyici dinamiği
$g$	: Yerçekim ivmesi
$W$	: Eyleyici bant genişliği ( $rad/sn$ )
$K$	: Pozitif kazanç ( $N$ )
$m$	: Quadrotorun toplam ağırlığı ( $kg$ )
$l$	: Rotor ile quadrotor merkezi arası mesafe ( $m$ )
$x$	: $x$ -eksenindeki pozisyon ( $m$ )

- $y$  :  $y$ -eksenindeki pozisyon ( $m$ )  
 $z$  :  $z$ -eksenindeki pozisyon ( $m$ )  
 $I_X$  :  $x$ -eksenindeki atalet momenti ( $kgm^2$ )  
 $I_Y$  :  $y$ -eksenindeki atalet momenti ( $kgm^2$ )  
 $I_Z$  :  $z$ -eksenindeki atalet momenti ( $kgm^2$ )  
 $\Omega$  : Açısal hız ( $rad/sn$ )  
 $\Phi$  : Yalpa açısı (roll angle), ( $rad$ )  
 $\theta$  : Yunuslama açısı (pitch angle), ( $rad$ )  
 $\Psi$  : Sapma açısı (yaw angle), ( $rad$ )

## KISALTMALAR DİZİNİ

ACO	:	Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
CRONE	:	Controle Robuste D'Ordre Non Entier
FOMCON	:	Kesir Dereceli Modelleme ve Kontrol (Fractional Order Modeling and Control)
GA	:	Genetik Algoritma (Genetic Algoritim)
IAE	:	Integral Absolute Error
ISE	:	Integral Squadred Error
ITAE	:	Integral Time-Weighted Absolute Error
ITSE	:	Integral Time-Weighted Squared Error
İHA	:	İnsansız Hava Aracı (Unmanned Aerial Vehicle, UAV)
KD	:	Kesir Dereceli (Fractional order)
KDD	:	Kesir Dereceli Denetleyici (Fractional Order Controller)
KDS	:	Kesir Dereceli Sistem (Fractional Order System)
LQR	:	Doğrusal Kuadratik Regülatör (linear Quadratic Regulator)
$PI^\lambda D^\mu$	:	Kesir Dereceli Oransal-İntegral-Türevsel (Fractional Order Proportional – Integral – Derivative)
PID	:	Oransal-İntegral-Türevsel (Porportional – Integral – Derivative)
PSO	:	Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
TID	:	Tilted Proportional and Integral
VTOL	:	Dikey Kalkış/İniş (Vertical Take-Off/Landing)

## 1. GİRİŞ

İnsansız Hava Araçları (İHA), belirli bir uçuş güzergâhı üzerinde sahip oldukları sensörler vasıtasıyla otonom uçabilme yeteneğini kullanarak ya da uzaktan kumanda edilerek, genellikle insan kullanımı için sıkıcı, kirli ve tehlikeli olarak tanımlanan görevleri yerine getirebilen, üzerlerinde insan bir pilot taşımayan hava araçlarıdır. Askeri amaçlı olarak genelde gözetleme, güvenlik, düşman takibi gibi görevlerde kullanılan İHA'lar gelişen teknolojiyle birlikte bilimsel araştırmalar, dizi/film setleri, maç yayınları, ürün teslimi, arama kurtarma çalışmaları, trafik yoğunluğu kontrolü gibi birçok alanda da sivil amaçlı olarak kullanılmaya başlanmıştır. Zor arazi koşullarına uygun, gerek yüksek manevra kabiliyeti gerekse olduğu yerden dikey kalkış/iniş (vertical take-off/landing, VTOL) yapabilme özelliğine sahip İHA'ların kullanımı da her geçen gün artmaktadır.

VTOL yapabilen hava taşıtları iki rotora sahip olan helikopter, üç rotorlu trikopter, dört rotorlu quadrokopter ya da diğer adıyla quadrotor, altı rotorlu hezkakopter gibi birçok farklı konfigürasyona sahiptir. Bu hava taşıtlarından insansız olarak en yaygın kullanılan model ise simetrik olarak yerleştirilmiş dört rotora sahip quadrotorlardır.

Dört rotorlu İHA'larda karşılıklı rotorların aynı yönlü dönmesi ve komşu rotorların zıt yönlü dönmesi prensibiyle, rotorların dönüş hızları ayarlanarak havada hareket sağlanır. Rotorların pervaneleri döndürmesiyle oluşan taşıma kuvveti (lift) quadrotorun ağırlığına eşit olduğunda quadrotor havada asılı kalır; eğer taşıma kuvveti quadrotorun ağırlığından küçükse quadrotor alçalır, büyükse yükselir. Quadrotorun istenen hareketi yapması için pervanelerin istenen hızda birlikte dönmeleri gerekmektedir. Bunun için rotorların devir sayılarının kontrol edilmesi gerekir. Genelde kontrol işleminde kolay tasarlanabilir olması ve yaygın olmasından dolayı *PID* denetleyiciler kullanılmaktadır. Bu tez kapsamında quadrotorun istenen hareketi yapması için gerekli olan devir sayılarını kontrol etmek üzere klasik *PID* denetleyicisi ve klasik *PID* denetleyicisinin bir versiyonu olan Kesir Dereceli *PID* denetleyicisi (Fractional Order PID Controller,  $PI^\lambda D^\mu$ ) kullanılmıştır.

$PI^\lambda D^\mu$  denetleyicisi, 20.yüzyılın sonlarında Podlubny tarafından ortaya konmuştur. Bu denetleyici, aslında 1695 yılında Leibniz ile L'Hospital arasındaki mektuplaşmalarda yer alan diyaloglarla literatüre kazandırılan kesir dereceli sistemlerin bir uyarlamasıdır. Bahsi geçen mektuplarda anlatılmak istenen türevin ya da integralin derecesinin reel sayı olması, örneğin, bir sistemin  $0.6^{\text{nci}}$  dereceden türev ya da integralinin sonucunun ne olduğudur. Kesir dereceli denetleyicilere (KDD) bu fikir uyarlandığında ise anlatılmak istenen, aynı mantıkla, klasik  $PID$  denetleyicisinin integral ve türev ifadelerinin derecelerinin reel sayı olmasıdır.  $PI^\lambda D^\mu$  denetleyicisinde  $\lambda$  integralin derecesini,  $\mu$  ise türevin derecesini göstermektedir. Parametre sayısının klasik  $PID$  denetleyicisine göre artması, katsayı belirleme işlemi açısından zorluğa neden olsa da çözüm uzayını genişletmesinden dolayı daha dayanıklı bir kontrol sistemi tasarlanmasını sağlamaktadır. Bu katsayıların belirlenmesinde sistematik bir yöntem olmamakla birlikte, genelde Genetik Algoritma (Genetic Algorithm, GA), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO) ya da Karınca Kolonisi Optimizasyonu (Ant Colony Optimization, ACO) gibi sezgisel optimizasyon yöntemleri kullanılmaktadır.

Optimizasyon problemlerinin çözümü için kullanılan sezgisel yöntemler doğayı ilham olarak almaktadırlar. GA evrimi, PSO kuş ve balık sürülerinin bir bölgedeki yiyeceği bulmaları anındaki davranışlarını, ACO ise karıncaların yiyeceğe giderken en kısa yolu seçmede feromon adı verilen kokuyu yayma ve sonrasında algılama davranışlarını modelleyerek oluşturulan optimizasyon yöntemleridir. Sezgisel algoritmalar amaç fonksiyonunu en küçük ya da en büyük yapan değeri bulmayı hedefler. Sezgisel optimizasyon yöntemlerinin klasik optimizasyon yöntemlerinden en büyük farkı, sistemi matematiksel olarak modelleme gereksiniminin bulunmamasıdır. Modelleme ihtiyacının olmamasının yanı sıra bu yöntemlerin sistemin sürekliliği, o noktada türevlenebilirliği ya da dış bükey (convex), olması gibi matematiksel gereklilikleri yerine getirme gibi bir sorumluluğu da bulunmamaktadır.

Evrin sürecini esas alan GA, evrimin işlem basamakları olan seçilimi ya da diğer adıyla doğal seleksiyonu, çaprazlamayı ve mutasyonu modellemiştir. Amaç iyi bireylerin hayatta kalma olasılıklarının artırılmasıdır. Seçilim

aşamasında iyi bireyler bir sonraki nesle daha fazla sayıda aktarılır ve bu bireyler kendi aralarında çaprazlanarak hem daha iyi bireyler elde edilmeye çalışılır hem de çeşitlilik artırılır. Son aşama olan mutasyon da yine çeşitliliği arttırmak üzere uygulanan bir işlemdir.

Parçacık sürü optimizasyonu kuş ya da balık sürülerinin bilgi paylaşımı esasına dayanmaktadır. Tehlike anında sürüdeki bireylerin birbirlerinden uzaklaşarak küçük kümeler halinde hareket ettikleri, yiyecek bulma anında ise birlikte hareket ettikleri gözlenmiştir. 1994-1995 yıllarında Dr. Eberhart ve Dr. Kennedy tarafından bu hayvanların bilgi paylaşımı esnasında kullandıkları yöntem üç tür davranış üzerine kurgulanarak modellenmiştir. Davranış türlerinin birincisi olan keşifsel (exploratory) davranış biçimi ile hayvanların rastgele hareket etme davranışı, ikincisi olan bilişsel (cognitive) davranış biçimi ile hayvanların hafızalarında yer etmiş olan en iyi bölgeyi ziyaret etme eğilimi ve son olarak üçüncüsü olan sosyal (social) davranış biçimi ile de sürüdeki diğer bireyleri takip etme eğilimi modellenmektedir. Bir bölgeye rastgele başlangıç koşullarıyla bırakılarak yayılan sürü bireyleri, kendi hafızalarında hem yiyeceğe o zamana kadar en yakın olan bireyin konumunu hem de kendilerinin o ana kadar yiyeceğe en yakın olduğu konum bilgisini tutarlar. Yöntem, bu bilgilerle birlikte o anki hız ve konum bilgilerini de kullanarak bir sonraki konumuna gitme prensibine dayanmaktadır.

Bu çalışmada, olduğu yerden dikey iniş/kalkış yapabilen dört rotorlu quadrotor tipi bir İHA'nın istenen referans yörüngede hareketini sağlamak üzere kesir dereceli denetleyici tasarımı kullanılmaktadır. Kesir dereceli denetleyiciye ait parametrelerin belirlenmesinde ise sezgisel optimizasyon yöntemlerinden olan genetik algoritma ve parçacık sürü optimizasyonu kullanılmaktadır.

İkinci bölümde kesir dereceli sistemler ve denetleyiciler, quadrotor kontrolü ve sezgisel optimizasyon yöntemleriyle ilgili olarak literatürde yapılmış çalışmalardan bahsedilmiştir.

Üçüncü bölümde kesir dereceli sistemler tanımlanarak, bu sistemlerin çözümlerine yönelik yapılmış matematiksel yaklaşımlar anlatılmakta ve kesir dereceli türev ile integrallere örnekler verilmektedir. Bu bölümde aynı zamanda,



çalışmada kullanılan denetleyici tipi olan kesir dereceli *PID* denetleyicisi de incelenmektedir.

Dördüncü bölümde ilk olarak sezgisel optimizasyon yöntemleri ile ilgili kısa bir bilgi verilmektedir. Sonrasında ise çalışmada kesir dereceli denetleyicinin parametrelerini belirlemek üzere kullanılan genetik algoritma ve parçacık sürü optimizasyonu hakkında detaylı bilgi sunulmaktadır.

Beşinci bölümde quadrotorun tanımı ve hareketleri detaylandırılmıştır. Quadrotorun matematiksel modeli belirli varsayımlar altında çıkarılarak çalışmada kullanılan QBall-X4 quadrotoruna ait denklemler sunulmuştur.

Altıncı bölümde, tez çalışması kapsamında quadrotorun kontrolünde kullanılan kesir dereceli *PID* ve klasik *PID* denetleyicilerinin parametreleri GA ve PSO algoritmaları ile belirlenmiştir. Benzetim çalışmaları MATLAB/Simulink ortamında yapılarak ilgili grafikler elde edilmiştir.

Yedinci bölümde ise altıncı bölümde elde edilen sonuçlar irdelenip genetik algoritma ve parçacık sürü optimizasyonu algoritmaları altında klasik ve kesir dereceli *PID* denetleyicileri karşılaştırılmıştır.

## 2. LİTERATÜR TARAMASI

Bu bölümde kesir dereceli sistemler ve denetleyiciler, quadrotor kontrolü ve sezgisel optimizasyon yöntemleriyle ilgili olarak literatürde yapılmış çalışmalar ele alınmıştır.

### 2.1. Kesir Dereceli Sistemler

Klasik matematik kadar geçmişi olan kesir dereceli sistemler (KDS), 30 Eylül 1695 tarihinde Leibniz ile L'Hospital arasında geçen mektuplaşmalarda yer alan diyaloglarla literatüre kazandırılmıştır (Agrawal, 2008). Bahsi geçen yazışmalarda Leibniz, L'Hospital'e,

*“Tam dereceli olan türevin anlamı tam dereceli olmayan türev için de genellenebilir mi?”*

sorusunu sormuştur. L'Hospital ise cevaben,

*“Farz edelim bu derece  $1/2$ , bu durumda ne olacak?”*

sorusunu yöneltmiştir. Leibniz ise bu soruya,

*“Bu gelecekte faydalı sonuçlar doğuracak bir paradoksa dönüşecek”*

cevabını vermiştir.

Kesir dereceli sistemlerin fikir olarak ortaya konmasından günümüze değin yaklaşık 300 yıl geçmiş olsa da, hesaplamalarının karmaşık oluşu ve fiziksel manasının zor kavranması sebebiyle gelişimi klasik matematiğin gerisinde kalmıştır (Das, 2008). Kesir dereceli sistemlerin kullanılmasıyla doğadaki fiziksel sistemlerin modellenmesi tam sayı dereceli sistemlere göre daha doğru sonuçlar vermiştir (Agrawal, 2008). 1695 yılında ortaya çıkan kesir dereceli sistemler ve formülasyonları üzerine o tarihten itibaren 1800'lerin sonuna kadar L. Euler, J.B.J. Fourier, N.H. Abel ve J. Liouville gibi ünlü matematikçiler de çalışmıştır. Bu çalışmaların ışığında günümüzde KDS ve operatörler için kullanılan tanımlar olan Riemann-Liouville, Grunwald-

Letnikov, M. Caputo, Oldham-Spanier, K.S. Miller- B. Ross ve son olarak Kolwankar-Gangal yaklaşımları literatüre kazandırılmıştır (Oldham ve Spainer, 1974).

1900'lü yıllardan itibaren birçok bilim adamı KDS'leri kullanarak sistemlerin modellenmesi üzerine çalışma yürütmüştür. Heaviside (1971) uzun elektrik hatlarının modellenmesinde, Ichise ve ark. (1971) ile Sun ve ark. (1984) elektrokimyasal süreçlerin modellenmesinde üzerine çalışmışlardır. Tsao ve ark. (1971) dielektrik polarizasyon (dielectric polarization) üzerine, Hartley ve ark. (1995) ise kaos teorisi üzerine kesir dereceli sistemlerle ilgili olarak çalışmalar yapmıştır. Bu çalışmaların yanı sıra viskoelastik materyaller, elektromanyetik kesirli kutuplar, yarı sonsuz kayıplı (semi-infinite lossy) iletim hattı, renkli gürültü (colored noise) gibi birçok alanda da kesirli modelleme tekniği kullanılmıştır (Mandelbrot, 1967; Bagley, 1989; Engheta, 1998). KDS üzerine yapılan bu çalışmaların devamında 1994 yılında Podlubny (1994), *PID* denetleyicisinin genelleştirilmiş ifadesi olan kesirli dereceli *PID* denetleyicileri literatüre kazandırmıştır. Bu denetleyici ile klasik *PID* denetleyicisine ait parametrelere  $\lambda$  ve  $\mu$  parametreleri de eklenmiştir (Podlubny, 1994; Podlubny, 1995). Eklenen bu parametrelerle birlikte  $PI^\lambda D^\mu$  denetleyicisinin toplamda belirlenmesi gereken  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  olmak üzere beş farklı parametresi bulunmaktadır. Podlubny 1997 yılında  $PI^\lambda D^\mu$  denetleyicilerinin uygulamalarına değinirken (Podlubny, 1997), 1999 yılında ise KDS'lerin kontrolünde  $PI^\lambda D^\mu$  denetleyicisinin kullanılmasının uygun olduğu sonucuna varmıştır (Podlubny, 1999).

Bir sistemi kesir dereceli olarak modellemek için sistemin geçmişten günümüze kadar gösterdiği tüm davranışların bilinmesi gerekmektedir. Bu sebepten kesir dereceli sistemlerin zaman bölgesindeki analizi için tamsayı dereceli yaklaşımları belirli frekans değerleri arasında elde edilmektedir (Aoun ve ark., 2004). Özyetkin (2013), doktora tez çalışmasında sürekli kesir açılımı metodunu kullanarak KDS'lerin tamsayı dereceli yaklaşımlarını elde etmiştir.

Kesir dereceli *PID* denetleyicilerin parametrelerinin belirlenmesine yönelik olarak literatürde Ziegler-Nichols yöntemi (Valerio ve Costa, 2006), simetrik optimum yöntemi (Maione ve Lino, 2006) gibi tekniklerin yanı sıra

günümüzde daha çok GA ve PSO gibi sezgisel optimizasyon yöntemleri de kullanılmaktadır (Cao ve ark., 2005; Cao ve Cao, 2006a). Cao ve Cao (2006b) tarafından yapılan çalışmada, parametrelerin kestirimi için parçacık sürü optimizasyonu ile genetik algoritma karşılaştırılmıştır ve belirlenen amaç fonksiyonu kriterlerine göre, parçacık sürü optimizasyonu algoritmasının genetik algoritmaya göre daha iyi sonuç verdiği belirlenmiştir. Li ve ark. (2010), parametrelerin belirlenmesi kapsamında  $PI^\lambda D^\mu$  denetleyicisini frekans bölgesinde inceleyerek yeni bir parametre kestirim yöntemi kullanmışlar ve DC motor üzerinde gerçek zamanlı sonuçları incelemiştir. Hamamci (2007) zaman gecikmeli sistemlerde KDS'in kararlılığı üzerine çalışmıştır. Bu çalışmada kesir dereceli kontrol sistemi parametreleri olan  $\lambda$  ve  $\mu$  değerleri için belirli bir aralıkta global olarak en kararlı bölgeyi tarayan bir algoritma geliştirmiştir. Wen ve ark. (2008) ise doğrusal olmayan (nonlinear) KDS'lerin kararlılığı üzerine çalışmıştır. KDS'lerin kararlılığı üzerine diğer çalışmalara Wang ve ark. (2008) ile Lu ve Chen (2009) kaynaklarından ulaşılabilir.

Literatürde  $PI^\lambda D^\mu$ , TID (Tilted Proportional and Integral), CRONE (Controle Robuste D'Ordre Non Entier) ve Kesir Dereceli Lead-Lag Kompansatör olmak üzere dört farklı türde kesir dereceli denetleyici yapısı bulunmaktadır. Xue ve Chen (2002) çalışmalarında bu dört tip kesirli dereceli denetleyicinin karşılaştırmasını yapmış ve,

- gerçek uygulamalar için CRONE metodunun daha doğru seçim olduğu,
- kesir dereceli  $PID$  denetleyicisinin, klasik  $PID$  denetleyicisine göre daha gülbüz sonuçlar verdiği,
- kesir dereceli  $PID$  denetleyicisinin özel bir formu olan TID denetleyicisinin daha geniş bir uygulama alanına sahip olduğu, ve
- kesir dereceli lead-lag kompansatörün  $PI^\lambda D^\mu$  ya da CRONE kadar iyi sonuç verdiği ancak parametre kestirim yöntemlerinin sistematikleştirilmeye ihtiyacı olduğu

sonuçlarına varmışlardır.

Kesir dereceli denetleyicilerin kullanım alanı sadece kesirli dereceli sistemlerle sınırlandırılmamıştır. Tamsayı dereceli sistemlerin kontrolü için de kesir dereceli denetleyiciler kullanılabilir ya da kesir dereceli sistem tamsayı

dereceli denetleyiciler ile kontrol edilebilir. Zhao ve ark. (2005) yaptıkları çalışma ile kesir dereceli sistemlerin kontrolünde hem  $PI^\lambda D^\mu$  denetleyicisini hem de klasik  $PID$  denetleyicisini kullanmışlar ve kesir dereceli denetleyiciler ile daha iyi sonuçlar elde etmişlerdir.

Kesir dereceli sistemlerin MATLAB ortamında uygulanması ve sonuçların değerlendirilmesi için ilk olarak Oustaloup ve Commande (1991) ile Oustaloup ve Bansard (1993) tarafından yapılan çalışmalarla CRONE adlı uygulama birinci nesil olarak tanıtılıp, sonrasında ikinci ve üçüncü nesil çalışmalarla (Oustaloup ve ark, 1993a; 1993b) uygulama geliştirilmiştir. Daha sonra Valerio ve Costa (2005) “Toolbox Ninteger for MATLAB v.2.3” uygulamasını geliştirmiştir. Son olarak Tepljakov ve ark. (2011) tarafından günümüzde diğerlerine göre daha yaygın ve popüler bir kullanım alanına sahip olan FOMCON (Fractional-Order Modeling and Control) Toolbox’ı geliştirilmiştir. FOMCON Toolbox’ı sayesinde kesir dereceli sistemlerin zaman bölgesinde ve frekans bölgesinde analizlerinin yanı sıra denetleyici performansları da kolayca incelenebilir hale getirilmiştir.

## 2.2. Kesir Dereceli Denetleyiciler ile İnsansız Hava Araçlarının Kontrolü

Üzerinde taşıdığı pilot tarafından kontrol edilen bilinen ilk quadrotor 1922 yılında Bothezat (Taylor, 1977) tarafından tasarlanmış ve önerilmiştir. Günümüz helikopterlerinin ilk örneği ise 1939 yılında Sikorsky tarafından tasarlanmıştır. Daha sonra Sikorsky ilk ticari helikopter modelini üretmiştir (Anonim, 2008). Geçmişten günümüze kadar rotor sayısı değiştirilerek ya da rotorların konumları değiştirilerek helikopter gibi dikey iniş/kalkış yapma özelliğine sahip birçok hava aracı tasarlanmıştır.



Şekil 2.1. Bilinen ilk quadrotor (Taylor, 1977)

Helikopterlerin türevlerinden birisi olan quadrotorlar, en basit şekliyle, askeri ve sivil alandaki ihtiyaçlar doğrultusunda kullanılan, dört rotora sahip, olduğu yerden dikine kalkış/iniş yapabilen döner kanatlar olarak tanımlanabilirler. Bu hava araçları rotorların farklı hızlarda dönmesiyle havada istediği yöne doğru hareketini sağlamaktadır. Yaygınlaşan uygulama alanıyla günümüzün popüler araştırma platformlarından olan quadrotorun kontrolü birçok farklı denetleyici ile sağlanabilmektedir. Altuğ ve ark. (2002) görüntü tabanlı geri besleme kullanarak quadrotorun kontrolünü geri beslemeli doğrusallaştırma (feedback linearization) ve geri adımlamalı (backstepping) denetleyicilerle sağlayarak sonuçları karşılaştırmıştır. Waslander (2005), quadrotorun yükseklik kontrolünü integral kayma modlu takviyeli öğrenme (integral sliding mode with reinforcement learning) yöntemini kullanarak sağlamıştır. Bouabdallah ve Siegwart (2007) quadrotoru klasik *PID* denetleyicisi ve Doğrusal Kuadratik Regülatör (Linear Quadratic Regulator, LQR) yöntemleri ile kontrol etmişlerdir.

Son yıllarda KDD'ler kullanılarak İHA'ların kontrol edilmesi üzerine çalışmalar yapılmış olup bu çalışmaların sonucunda KDD'lerin klasik  $PID$  denetleyicisine göre daha iyi sonuç verdiği gözlenmiştir. Changmao ve ark. (2010) yüksek hızlı hava araçlarının zamanla değişen doğrusal olmayan modeli ile yükseklik kontrolünü kesir dereceli  $PID$  denetleyicisini kullanarak yapmıştır.  $PI^\lambda D^\mu$  denetleyicisi ile hem kontrol kalitesinin arttığı hem de sistemin parametre değişimlerine karşı hassasiyetinin azaldığı gözlenmiştir. Luo ve ark. (2011), kesir dereceli  $PI$  kontrolörün farklı tipleri olan  $(PI)^\alpha$  ve  $PI^\alpha$  denetleyicileri ile insansız hava aracının yanal doğrultuda kontrolünü gerçek zamanlı ve benzetimli olarak sağlamıştır ve  $(PI)^\alpha$  denetleyicisinin daha iyi olduğu sonucuna varmışlardır. Bir diğer çalışmada Na ve ark. (2012), uçağın nötral hız kararlılığının kontrolünü  $PI^\lambda D^\mu$  denetleyicisi türlerinden olan  $PI^\lambda$  kullanarak sağlamıştır ve tamsayı dereceli  $PI$  denetleyicisi ile karşılaştırmıştır. Çalışmalarının sonucunda KDD'nin performansının daha iyi olduğunu gözlemişlerdir.

### 2.3. Sezgisel Optimizasyon Yöntemlerinin Kontrol Problemlerine Uygulanması

Çözüm uzayında, fonksiyonları minimum ya da maksimum yapan değerleri nümerik olarak tarayan sezgisel optimizasyon yöntemlerinin uygulamalarından birisi de, sistemi kontrol eden denetleyicinin katsayılarının bulunmasıdır. Çözüm uzayında hatayı minimum yapan denetleyici katsayılarının belirlenmesi için kullanılan sezgisel yöntemler herhangi bir matematiksel fonksiyona ya da modele ihtiyaç duymazlar.

Genetik algoritma evrimi esas alan sezgisel yöntemlerden birisidir. Evrimsel süreçler olan seçim (doğal seleksiyon), çaprazlama ve mutasyon tekniklerinin modellenmesiyle oluşturulmuştur. İyi olanın hayatta kalması felsefesiyle çalışır. Bu alandaki ilk çalışma 1975 yılında Holland (1975) tarafından yapılmıştır. Bu algorithmada parametreler değerleri ile değil kodlarla ifade edilir. 1986 yılında yaptığı çalışmada Grefenstette (1986), GA'ları kontrol parametrelerinin bulunması için kullanmıştır. Ota (1996) GA kullanarak *PID* parametreleri olan  $K_p$ ,  $K_i$  ve  $K_d$  katsayılarını belirlemiştir. Bu alanda yapılan diğer çalışmalar Tang ve ark., (2001), Wu ve ark. (2007), Fan ve Joo (2009) ile Sharma ve ark., (2014) tarafından yapılanları içeren kaynaklarda mevcuttur.

Parçacık sürü optimizasyonu ise 1995 yılında Eberhart ve Kennedy (1995a; 1995b) tarafından kuş ve balık sürülerinin davranışlarının modellenmesiyle oluşturulan bir sezgisel optimizasyon yöntemidir. Bahsi geçen çalışmalarda PSO tekniğinin amaç fonksiyonları üzerinde durulmuş ayrıca uygulama alanlarından da bahsedilmiştir. PSO yöntemi ile kuş ve balık sürülerinin tehlike anında dağılarak küçük gruplara ayrılması yiyecek bulma konusunda ise birlikte hareket etmeleri modellenmiştir. Algoritmanın oluşum sürecinde sürüdeki bireylerin (parçacıkların) keşifsel, bilişsel ve sosyal davranışları esas alınmıştır. Keşifsel davranış hayvanların rastgele hareket etme isteğini, bilişsel davranış her bir parçacığın kendi hafızasında yer eden en iyi konuma gitme isteğini, sosyal davranış ise sürüdeki tüm bireyler içinde o ana kadar yiyeceğe en yakın olan parçacığın konum bilgisini hafızasında tutmasını



tanımlar. PSO yönteminde, bu üç davranış bileşeninin belirli ağırlıklar ve katsayılar ile birleştirmesiyle parçacıkların konumu ve hızı güncellenmektedir (Kennedy ve Eberhart, 1995a).

Klasik *PID* denetleyicisinde olduğu gibi KDD'lerin parametrelerinin kestiriminde de sezgisel optimizasyon yöntemleri kullanılmaktadır. Maiti ve ark. (2008) çalışmalarında PSO yöntemi kullanarak  $PI^\lambda D^\mu$  denetleyicisinin parametrelerini ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) belirlemiştir. Benzer çalışmalara Meng ve Xue (2009), Priya ve Lakshmi (2011) ile Gupta ve ark. (2014) tarafından yapılan yayınlardan ulaşılabilir.

### 3. KESİR DERECELİ SİSTEMLER

1695 yılında Leibniz ile L'Hospital arasındaki mektupla ilk kez ortaya çıkan kesir dereceli sistemlerin günümüzde popülerliğinin artmasının sebepleri, son yıllarda kesirli hesaplamalar üzerine yapılan tanımlar sayesinde mühendislik, ekonomi gibi alanlara uygulanabilir olması ve doğadaki gerçek sistemleri daha iyi modelleme olanağı sağlamasıdır (Das, 2008).

Örneğin  $2^n$ ,  $n$  tamsayı için kolayca hesaplanabilirken  $n$  eğer  $\pi$  ise,  $2^\pi$  sayısı pek akılda bir şeyler çağrıştırmaz fakat bir sonucu vardır. Benzer mantıkla bir fonksiyonun da kesirli dereceden türevi ve integrali vardır.

Kesir dereceli sistemler  ${}_aD_x^\alpha$  operatörü ile gösterilmektedir. Bu operatörde  $a$  ve  $x$  sınırları,  $\alpha$  ise operatörün derecesini göstermektedir. Genelde  $\alpha \in R$  şeklinde tanımlanmakla birlikte karmaşık sayı da olabilmektedir (Oustaloup ve ark., 2000) göstermektedir.

$${}_aD_x^\alpha = \begin{cases} \frac{d^\alpha}{dx^\alpha}, & \alpha > 0 \\ 1, & \alpha = 1 \\ \int_a^x (d\tau)^{-\alpha}, & \alpha < 0 \end{cases} \quad (3.1)$$

#### 3.1. Kesir Dereceli Sistemlerin Farklı Tanımları

Literatürde kesir dereceli sistemlerin hesaplanmasına yönelik farklı yöntemler bulunmaktadır. Bu yöntemlerden bir kısmı klasik tamsayı matematiğinin genişletilmesiyle oluşturulmuştur. En popüler olanları Riemann-Liouville, Grünwald-Letnikov ve Caputo tanımlarıdır (Monje ve ark., 2010).

##### 3.1.1. Riemann-Liouville tanımı

Riemann-Liouville tanımına göre kesir dereceli sistemlerde integral ifadesi (3.2) ile verilmektedir.

$${}_a D_x^\alpha f(x) = \frac{1}{-\alpha} \int_a^x (x - \tau)^{-\alpha-1} f(\tau) d\tau, \quad \alpha < 0 \quad (3.2)$$

$(n-1) \leq \alpha < n$  olmak kaydıyla denklem (3.2)'nin  $n$ . dereceden türevi alınırsa,

$${}_a D_x^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_a^x (x - \tau)^{n-\alpha-1} f(\tau) d\tau, \quad n > 0 \quad (3.3)$$

elde edilmiş olur. Denklem (3.3) ile  $f(x)$  fonksiyonunun  $(n-\alpha)$ 'inci dereceden kesir dereceli türevi ifade edilmektedir.

### 3.1.2. Grünwald-Letnikov tanımı

Bir diğer tanım olan Grünwald-Letnikov tanımı ile,  $\alpha$  reel sayı dereceli sistemi (3.4) denkleminde vermektedir.

$${}_a D_x^\alpha f(x) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\frac{x-\alpha}{h}} (-1)^j \binom{\alpha}{j} f(x - jh) \quad (3.4)$$

Denklem (3.4)'de  $h$  adım sayısını göstermektedir. Denklem (3.4)'de yer alan toplam ifadesi ise,

$$\sum_{j=0}^r (-1)^j \binom{\alpha}{j} = \frac{1}{\Gamma(1-\alpha)} \frac{\Gamma(r+1-\alpha)}{\Gamma(r+1)} \quad (3.5)$$

olarak verilmektedir.

Denklem (3.5)'te kullanılan ve  $\Gamma(x)$  ile ifade edilen Euler-Gama fonksiyonu,

$$x^\alpha = \underbrace{x \cdot x \cdot x \cdot \dots \cdot x}_\alpha, \quad \alpha \text{ Tamsayı} \quad (3.6)$$

$$x^\alpha = e^{\alpha \ln x} \quad (3.7)$$

$$\alpha! = 1.2.3. \dots .(\alpha-1). \alpha , \quad \alpha \text{ Tamsayı} \quad (3.8)$$

$$\alpha! = \Gamma(\alpha+1) , \quad \alpha \text{ Reelsayı} \quad (3.9)$$

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt \quad (3.10)$$

formülasyonu ile elde edilir. Denklem (3.10)'da,  $x \geq 0$  olup bir tamsayıdır. Denklem (3.4)'te,  $\binom{\alpha}{j}$  binom katsayısını göstermektedir. Binom katsayısı,

$$\binom{\alpha}{j} = \frac{\alpha(\alpha-1)(\alpha-2)\dots(\alpha-j+1)}{j!} \quad (3.11)$$

ile ifade edilir.

### 3.1.3. Caputo tanımı

Caputo tanımı (3.12) denklemi ile ifade edilir. Bu denklemde,  $n$  tamsayı,  $\alpha$  ise reel sayıdır.

$${}_a D_x^\alpha f(x) = \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{f^n(\tau)}{(x-\tau)^{\alpha+1-n}} d\tau , \quad (n-1) \leq \alpha < n \quad (3.12)$$

## 3.2. Kesir Dereceli Sistemlerin Laplace Dönüşümleri

En genel halde doğrusal kesir dereceli diferansiyel denklemler, (3.13)'te verildiği şekilde ifade edilebilir.

$$\begin{aligned} a_1 D^{\alpha_1} y(t) + a_2 D^{\alpha_2} y(t) + \dots + a_{n-1} D^{\alpha_{n-1}} y(t) + a_n D^{\alpha_n} y(t) \\ = b_1 D^{\beta_1} r(t) + b_2 D^{\beta_2} r(t) + \dots + b_m D^{\beta_m} r(t) \end{aligned} \quad (3.13)$$

Denklem (3.13)'de  $a_k$  ( $k=0, \dots, n$ ),  $b_k$  ( $k=0, \dots, m$ ) sabit sayılar,  $\alpha_k$  ( $k=0, \dots, n$ ),  $\beta_k$

( $k=0, \dots, m$ ) ise rastgele reel sayılardır.

Giriş ve çıkışların tüm başlangıç değerlerinin sıfır olduğu kabul edilirse sistemin  $s$  bölgesinde transfer fonksiyonu,

$$G(s) = \frac{\mathcal{L}[y(t)]}{\mathcal{L}[r(t)]} = \frac{b_1 s^{\beta_1} + b_2 s^{\beta_2} + \dots + b_m s^{\beta_m}}{a_1 s^{\alpha_1} + a_2 s^{\alpha_2} + \dots + a_n s^{\alpha_n}}, \quad (3.14)$$

ile ifade edilir.

Kesir dereceli integral ifadesinin Laplace dönüşümü,

$$\mathcal{L}[{}_0D_x^{-\alpha} f(x)] = s^{-\alpha} \mathcal{L}[f(x)] = s^{-\alpha} F(s), \quad (3.15)$$

ile ifade edilirken, kesir dereceli türev ifadesinin Laplace dönüşümü de,

$$\mathcal{L}[{}_aD_x^{\alpha} f(x)] = s^{\alpha} F(s) - \sum_{k=0}^{n-1} s^k [{}_aD_x^{\alpha-k-1} f(x)]_{x=a} \quad (3.16)$$

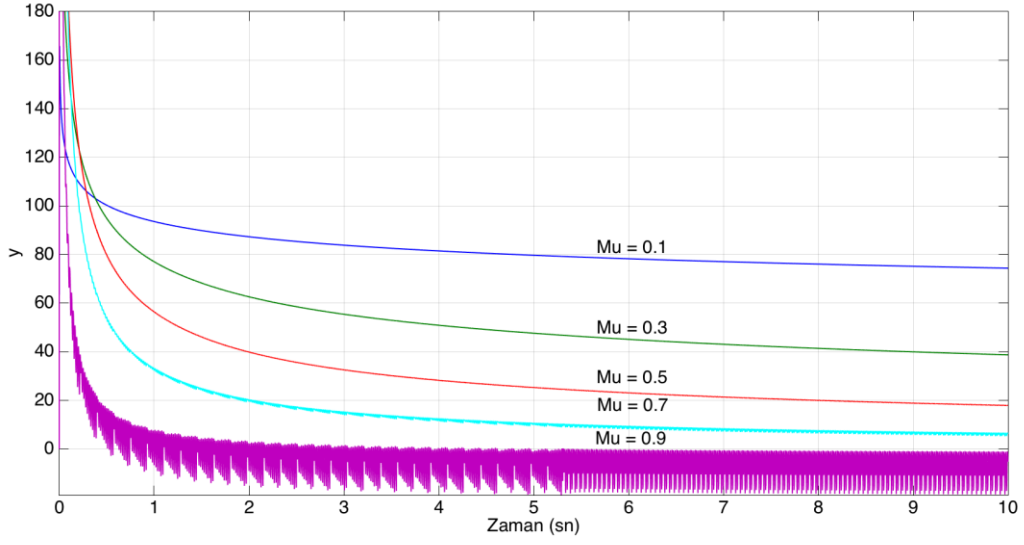
şeklinde oluşturulur.  $(n-1) \leq \alpha < n$  olmak üzere, eğer  $x = a$  noktasında  $f(x)$  fonksiyonunun türevi 0 ise, kesir dereceden türev ifadesi,

$$\mathcal{L}[{}_aD_x^{\alpha} f(x)] = s^{\alpha} F(s) \quad (3.17)$$

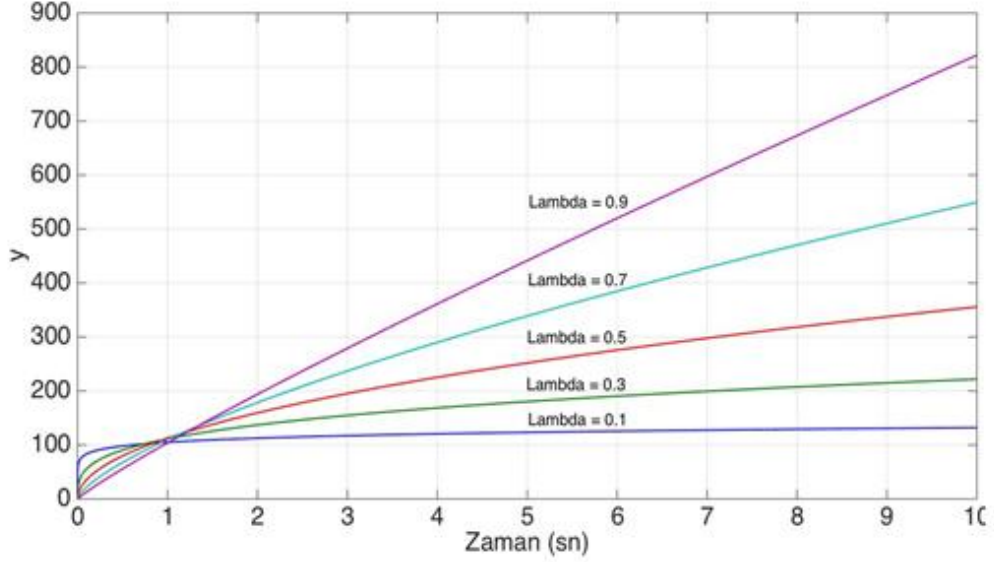
denklemini ile ifade edilir.

Laplace dönüşümünde  $s$  değişkeni yerine  $j\omega$  yazılarak frekans bölgesine geçiş sağlanabilir. Bu geçiştten yararlanılarak sistemin Bode, Nyquist, Nichols gibi eğrileri çizdirilerek frekans bölgesinde de gerekli analizler yapılabilir.

Örnek olarak  $y=100$  sabit fonksiyonunun kesir dereceli türev ve integrallerine MATLAB ortamında bakıldığında, Şekil 3.1 ve Şekil 3.2'de verilen eğriler elde edilebilir.



Şekil 3.1.  $y=100$  fonksiyonunun kesir dereceli türev ifadeleri



Şekil 3.2.  $y=100$  fonksiyonunun kesir dereceli integral ifadeleri

Şekil 3.1'e bakıldığı zaman  $y=100$  fonksiyonunun  $\mu = 0.1, 0.3, \dots, 0.9$  değerleri için kesir dereceli türev ifadeleri verilmiştir.  $\mu$  parametresinin değeri  $I'$ 'e yaklaştıkça  $y$  fonksiyonunun kesir dereceli türev ifadesinin sonucunun da birinci dereceden türevinde olduğu gibi  $0'$ a yaklaştığı gözlenebilir. Şekil 3.2'ye bakıldığı zaman,  $y=100$  fonksiyonunun  $\lambda = 0.1, 0.3, \dots, 0.9$  değerleri için kesir dereceli integral ifadeleri verilmiştir.  $\lambda$  parametresinin değeri  $I'$ 'e yaklaştıkça  $y$  fonksiyonunun kesir dereceli integral ifadesinin sonucunun da birinci dereceden integralinde olduğu gibi  $y = 100t$  sonucuna yaklaştığı gözlenebilir.

### 3.3. Kesir Dereceli Denetleyiciler

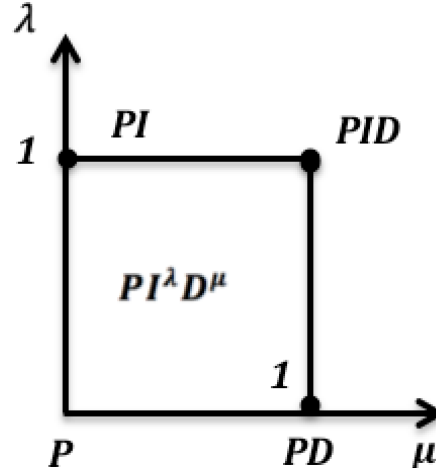
Kesir dereceli sistemlerin kullanımı sistemleri modelleme adına, benzetim çalışmalarını gerçek dünyaya bir adım daha yaklaştırmıştır (Ichise ve ark., 1971). Kontrol etmek istediğimiz sistem kesir dereceli olarak modellenmiş ise bu sistemi kontrol etmek için kesir dereceli denetleyici kullanmak da kontrolcü performansını geliştirmiştir. Eğer sistem tam sayı dereceli olarak yani klasik matematik kullanılarak modellenmişse yine çoğu sonuç genelleştirilmiş *PID* denetleyicisi kullanılmasının çıkış performansını iyileştirdiğini göstermiştir (Podlubny, 1999; Monje ve ark., 2010).

Podlubny tarafından klasik *PID* denetleyicisinin genelleştirilmesi ile  $PI^\lambda D^\mu$  denetleyicisi elde edilmiştir (Podlubny, 1994; Podlubny ve ark., 1995).  $PI^\lambda D^\mu$  denetleyicisi, klasik *PID* denetleyicisindeki  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerine ilave olarak  $\lambda$  ve  $\mu$  parametrelerini de içermektedir. Bu parametrelerin tanımları Çizelge 3.1’de verilmektedir.

$PI^\lambda D^\mu$  denetleyicisindeki  $\lambda$  ve  $\mu$  parametrelerinin değerlerinin değişimine göre klasik *PID* ve kesir dereceli *PID* denetleyicilerinin kullandığı uzay Şekil 3.3 ile gösterilmektedir. Şekil üzerinde yer alan parametrelerin değerlerine göre denetleyicinin tipi ise Çizelge 3.2’de verilmektedir.

Çizelge 3.1. Kesir dereceli denetleyici için parametre tanımları

Parametre	Tanım
$K_p$	Oransal (proportional) katsayısı
$K_i$	İntegral (integral) katsayısı
$K_d$	Türev (derivative) katsayısı
$\lambda$	Kesir dereceli denetleyicinin integral ifadesinin derecesi
$\mu$	Kesir dereceli denetleyicinin türev ifadesinin derecesi



Şekil 3.3.  $\lambda$  ve  $\mu$  değişimine göre denetleyici tipleri

Çizelge 3.2.  $\lambda$  ve  $\mu$  değişimine göre denetleyici tipleri

Parametre değeri	Denetleyici tipi
$\lambda = 0$ ve $\mu = 0$	$P$
$\lambda = 1$ ve $\mu = 0$	$PI$
$\lambda = 0$ ve $\mu = 1$	$PD$
$\lambda = 1$ ve $\mu = 1$	$PID$
$0 \leq \lambda \leq 1$ ve $0 \leq \mu \leq 1$	$PI^\lambda D^\mu$

$\lambda$  ve  $\mu$  parametrelerinin klasik  $PID$  parametrelerine ilave olması her ne kadar matematiksel işlem yükünü arttırsa da Şekil 3.3'den görüldüğü üzere çözüm uzayını genişletmesi bakımından avantaj sağlamaktadır.

Kesir dereceli denetleyicinin Laplace bölgesindeki karşılığı (3.18) ile, zaman bölgesindeki karşılığı ise (3.19) ile verilmektedir (Podlubny, 1999).

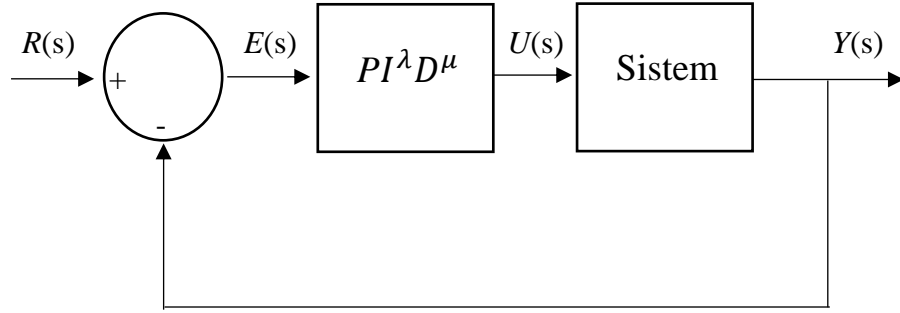
$$C(s) = K_p + K_i/s^\lambda + K_d s^\mu, \quad (\lambda, \mu \geq 0) \quad (3.18)$$

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t), \quad (\lambda, \mu \geq 0) \quad (3.19)$$

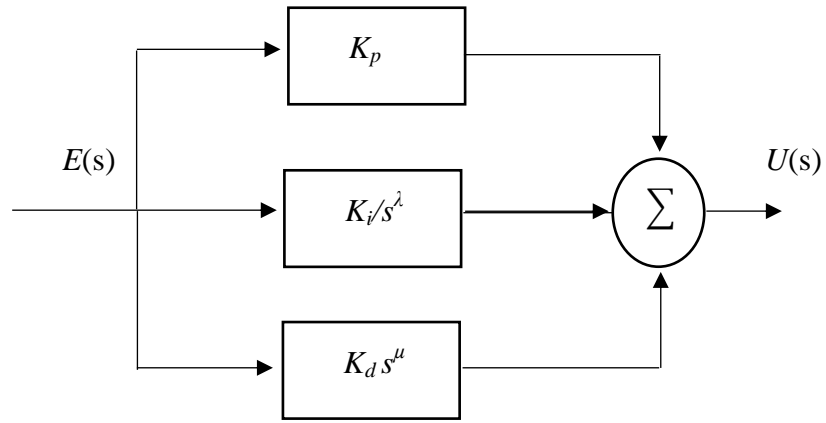
Şekil 3.4'de kesir dereceli denetleyicinin dahil edildiği sistemin blok diyagramı görülmektedir. Şekil 3.5'te ise  $PI^\lambda D^\mu$  denetleyicinin iç yapısı



gösterilmektedir.



Şekil 3.4.  $PI^\lambda D^\mu$  denetleyicinin dahil olduğu sistemin blok diyagramı



Şekil 3.5.  $PI^\lambda D^\mu$  denetleyicinin blok diyagramı iç yapısı

## 4. SEZGİSEL OPTİMİZASYON

Optimizasyon, belirli kısıtlar çerçevesinde oluşturulan çözüm uzayında amaç fonksiyonunu maksimum ya da minimum yapan karar değişkenlerinin bulunması işlemidir. Literatürde klasik ve sezgisel ana başlıkları altında birçok optimizasyon yöntemi mevcuttur. Klasik optimizasyon yöntemleri amaç fonksiyonunu oluşturmak için matematiksel modele, fonksiyonun her noktada sürekli ve türevlenebilir olması gibi matematiksel gerekliliklere ihtiyaç duyar. Ayrıca, klasik optimizasyon yöntemleri boyut sayısı arttıkça yeterliliğini yitirir. Sezgisel optimizasyon yöntemlerinde ise bahsi geçen gerekliliklere ihtiyaç yoktur. Bundan dolayı son yıllarda sezgisel optimizasyon yöntemlerinin popülerliği ve uygulama alanları artış göstermiştir. Sezgisel yöntemler işlem hızının iyi olması, başka problemlere kolay uyarlanabilir olması ya da çok amaçlı fonksiyonlara uygulanabilmesinden dolayı klasik yöntemlere göre avantajlıdır.

### 4.1. Genetik Algoritma

Evrimsel algoritmalarından birisi olan genetik algoritmanın temelleri 1975 yılında Holland tarafından atılmıştır. İlk ortaya çıkışında parametrelerin gerçek değerleri yerine kodlanmış halleri kullanılmakla birlikte sonrasında yapılan çalışmalarla parametrelerin gerçek değerlerinin de kullanılarak bu algoritmanın gerçekleştirilebileceği gösterilmiştir (Holland, 1975; Chang, 2007). Genetik algoritma yönteminin atama problemleri, çizelgeleme, fonksiyon optimizasyonu gibi birçok alana uygulanabilir olması sayesinde popülerliği her geçen gün artmıştır (Goldberg, 1989). Algoritmada uygulanan genetik kökenli işlemler seçim (doğal seleksiyon), çaprazlama ve mutasyondur. Asıl amaç iyinin hayatta kalması ve çaprazlama ve mutasyon işlemleriyle çeşitliliği sağlamaktır. Çözüm uzayını oluşturan popülasyonun her bir elemanı kromozom olarak adlandırılır ve kromozomlar genlerden oluşur (Holland, 1975).

- **Kodlama**

Genetik algoritmanın ilk adımı, çözüm kümesindeki her kromozomu ikili (binary) sisteme göre kodlamaktır. Kromozomların kaç genden oluşacağı ise olası çözümlerin alt ve üst sınırları ve hassasiyeti tarafından belirlenir. Hassasiyet çözümlerin alt ve üst sınırlar arasında ne kadar hassas bir şekilde aranması gerektiğini belirtir. Yani hassasiyet  $0.1$  ise virgülden sonra bir basamaklık bir hassasiyetle çözüm uzayı taranmalıdır.

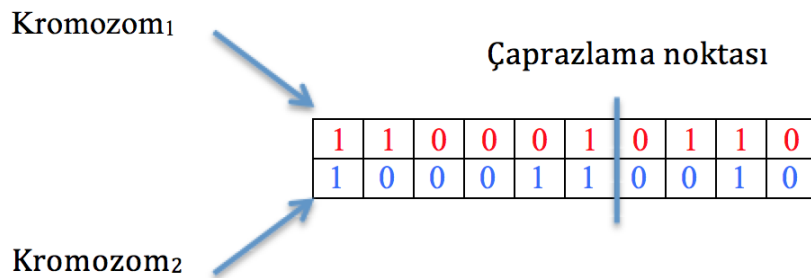
$$Bit\ sayısı = \frac{Maksimum\ deęer - Minimum\ deęer}{Hassasiyet\ oranı} \quad (4.1)$$

Örneğin çözüm  $0$  ile  $100$  arasında aranacaksa ve hassasiyet  $0.01$  ise sistemin bit sayısı,  $10000$  olarak elde edilir. Bu deęer ikili sistemde en az  $14$  bit ile ifade edilir.

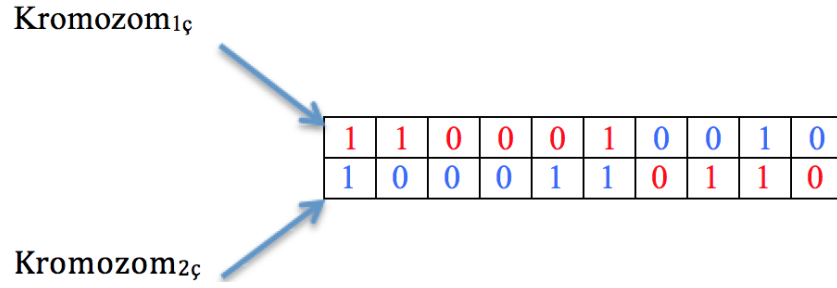
- **Çaprazlama**

Çaprazlama işlemi popülasyondan rastgele iki bireyin seçilmesi ve bu bireylerin yani kromozomların birbirleri arasında belirli bir noktadan itibaren genlerinin deęiştirilmesi işlemidir. Kromozomların hangi bitinden itibaren genlerini deęiştireceęi ise çaprazlama oranı tarafından belirlenir.

Örneğin Şekil 4.1 ile verilen  $10$  genden oluşan iki kromozomun çaprazlama noktası  $4$  ise, bu noktadan itibaren yer alan tüm genler karşılıklı olarak yer deęiştirirler yani çaprazlanırlar. Şekil 4.2'de çaprazlama işlemi sonrası oluşan yeni kromozomlar görülmektedir.



Şekil 4.1. Çaprazlama öncesi kromozomlar

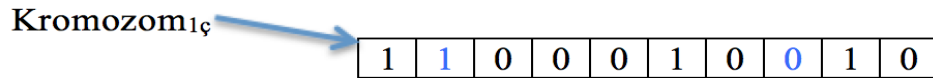


Şekil 4.2. Çaprazlama sonrası kromozomlar

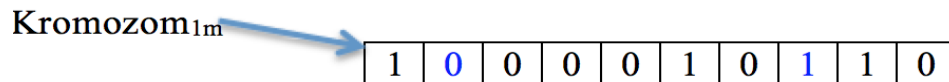
- **Mutasyon**

Mutasyon işlemi çaprazlama işleminde olduğu gibi birçok geni ve kromozomu etkilemez ama çeşitliliğin artışında önemli rol oynar. Mutasyona uğrayacak gen sayısı “mutasyon oranı” ile belirlenir. Popülasyonda mutasyona uğrayacak genler rastgele seçilir. Mutasyona uğrayacak gen belirlendikten sonra genin değeri 1 ise 0, 0 ise 1 yapılır.

Örneğin Şekil 4.3’de verilen birinci kromozomun üçüncü ve dokuzuncu genleri (bitleri) mutasyona uğratılacaksa işlem sonrasında Şekil 4.4 ile verilen yeni kromozom elde edilmiş olur.

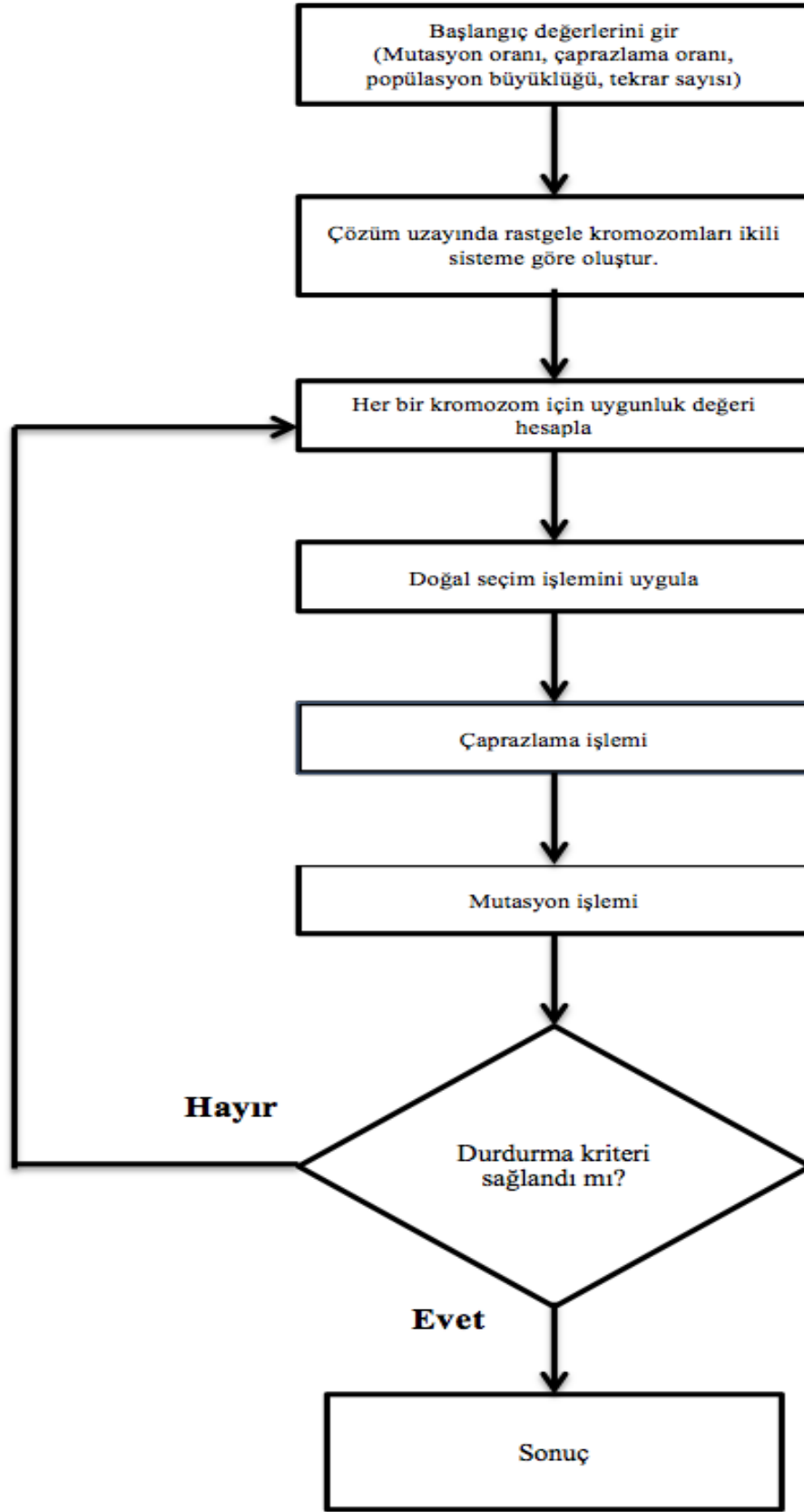


Şekil 4.3. Mutasyon öncesi kromozom



Şekil 4.4. Mutasyon sonrası kromozom

Genetik algoritma yönteminin akış diyagramı Şekil 4.5’de verilmektedir.



Şekil 4.5. Genetik algoritma yöntemi akış şeması

Şekil 4.5’de verilen akış şemasına göre genetik algoritma yöntemini uygulamak üzere takip edilmesi gereken işlem basamakları şu şekilde sıralanabilir (Portmann, 1996):

- I. Olası tüm çözümleri kapsayacak başlangıç popülasyonunu rastgele üret.
- II. Tüm kromozomların amaç fonksiyon değerini hesapla.
- III. Tekrar üreme, çaprazlama ve mutasyon işlemlerini uygula.
- IV. Yeni oluşturulan tüm kromozomların amaç fonksiyonunu hesapla.
- V. Durdurma kriterini kontrol et, eğer sağlandıysa programı sonlandır, sağlanmadıysa devam et.
- VI. Amaç fonksiyonu kötü olan kromozomları ele.
- VII. Adım III – Adım V arasındaki işlemleri tekrarla.

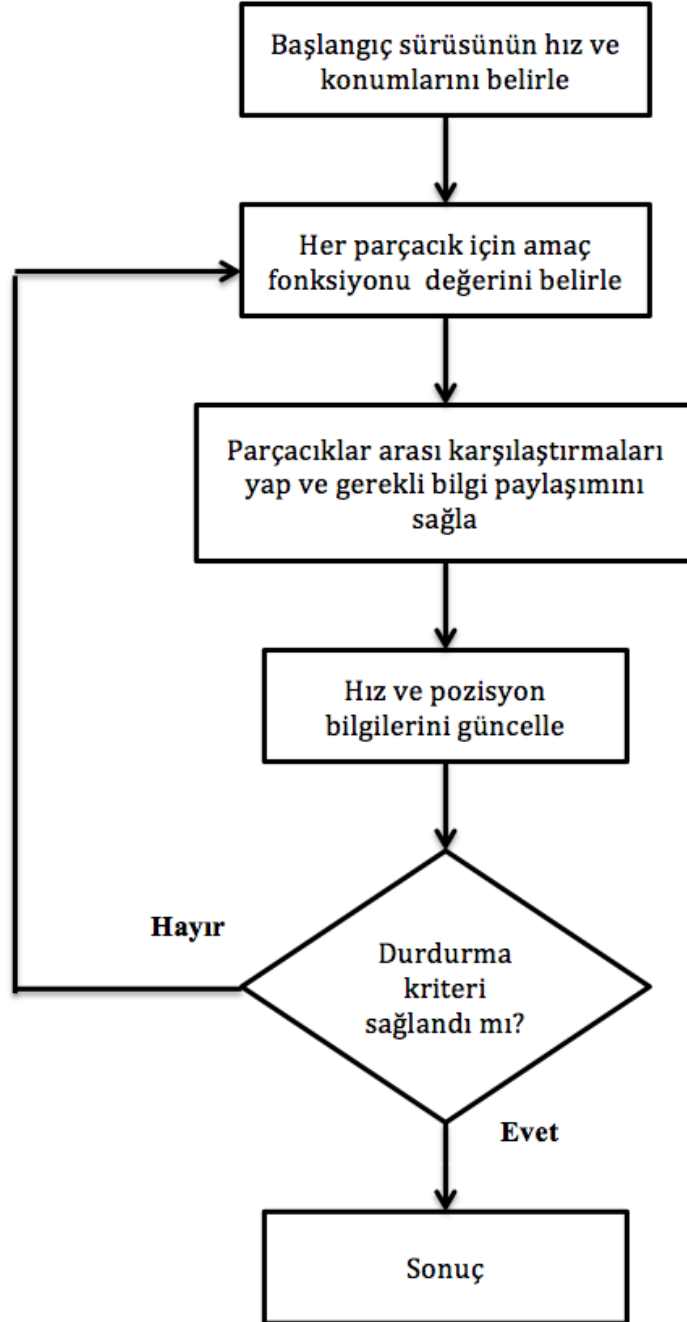
#### **4.2. Parçacık Sürü Optimizasyonu**

Sezgisel optimizasyon yöntemlerinden bir diğeri olan parçacık sürü optimizasyonu, 1995 yılında, kuş ve balık sürülerinin davranışları ilham alınarak Kennedy ve Eberhart (1995a) tarafından literatüre kazandırılmıştır. Bu yöntemde kuşların ya da balıkların yerini bilmedikleri yiyeceği bulma esnasında birbirleriyle haberleşerek yiyeceği kısa sürede bulmaları ya da tehlike anında küçük gruplara dağılarak hareket etmeleri modellenmiştir. Yani bilgi paylaşımı da esas alınarak sürünün davranışlarının benzetimi yapılmıştır. Bu modelleme, sürünün keşifsel (exploratory), bilişsel (cognitive) ve sosyal (social) davranış olarak adlandırılan üç tür davranışı üzerine kurgulanmıştır. Keşifsel davranış biçimi, hayvanların rastgele hareket etme içgüdüsünü modeller; bilişsel davranış biçimi hayvanların hafızalarında yer etmiş en iyi bölgeyi ziyaret etme eğilimini modeller, son olarak sosyal davranış biçimi ise sürüdeki diğer bireyleri takip etme eğilimini modeller.

Parçacık sürü optimizasyonu bir bölgeye rastgele başlangıç koşullarıyla bırakılarak yayılan bireylerin kendi hafızalarında hem yiyeceğe o zamana kadar en yakın olan bireyin konumunu hem de kendinin o ana kadar yiyeceğe en yakın olduğu konum bilgilerini tutar. Tüm bireyler sahip oldukları konum bilgilerini sürekli birbirleri ile paylaşarak yiyeceğe en yakın olan bireyi izler. Yiyeceğe

olan mesafe amaç fonksiyonu olarak ifade edilir. Yiyeceğe en yakın olan bireyin konumu  $G_{best}$ , bireylerin kendilerinin o zamana kadar ki yiyeceğe en yakın olduğu konum ise  $P_{best}$  olarak tanımlanır. PSO algoritması bireylerin  $G_{best}$  ve  $P_{best}$  bilgilerini o an sahip olduğu hız ve konum bilgilerine ekleyerek bir sonraki konumuna gitmesi prensibine dayanmaktadır.

Şekil 4.6’da parçacık sürü optimizasyonu algoritmasına ait akış şeması gösterilmiştir.



Şekil 4.6. Parçacık sürü optimizasyonu yöntemi akış şeması

Şekil 4.6’da akış şeması verilen parçacık sürü optimizasyonunda bireyler parçacıklar ile temsil edilir. Parçacıkların oluşturduğu popülasyon ise sürü olarak isimlendirilir. Başlangıçta rastgele çözüm uzayına dağıtılan parçacıkların o anki pozisyon bilgisine göre amaç fonksiyonunun değeri bulunur. Amaç fonksiyonu değerlerine göre parçacıklar kıyaslanır,  $G_{best}$  ve  $P_{best}$  değerleri belirlenir. Parçacıklar bu bilgileri kullanarak hız ve konum bilgilerini günceller ve sonraki konumuna gider. Bu işlemler durdurma kriteri sağlanana kadar tekrar eder. Durdurma kriteri sağlanınca program çalışmayı durdurur ve o ana kadar bulduğu en iyi çözümü sunar. Durdurma kriteri olarak ya iterasyon sayısı girilir ya da amaç fonksiyonu için sınır tanımlanır. Sonuç olarak çözüm uzayı taranır ve optimum sonuç elde edilmeye çalışılır.

Hız güncellemesi ve konum güncellemesi için her adımda kullanılan matematiksel ifadeler sırası ile denklemler (4.2) ve (4.3) ile verilmektedir.

$$V_i^{k+1} = \underbrace{wV_i^k}_{\text{Keşifsel}} + \underbrace{c_1r_1(PBest_i - X_i^k)}_{\text{Bilişsel}} + \underbrace{c_2r_2(GBest_i - X_i^k)}_{\text{Sosyal}} \quad (4.2)$$

$$X_i^{k+1} = V_i^{k+1} + X_i^k \quad (4.3)$$

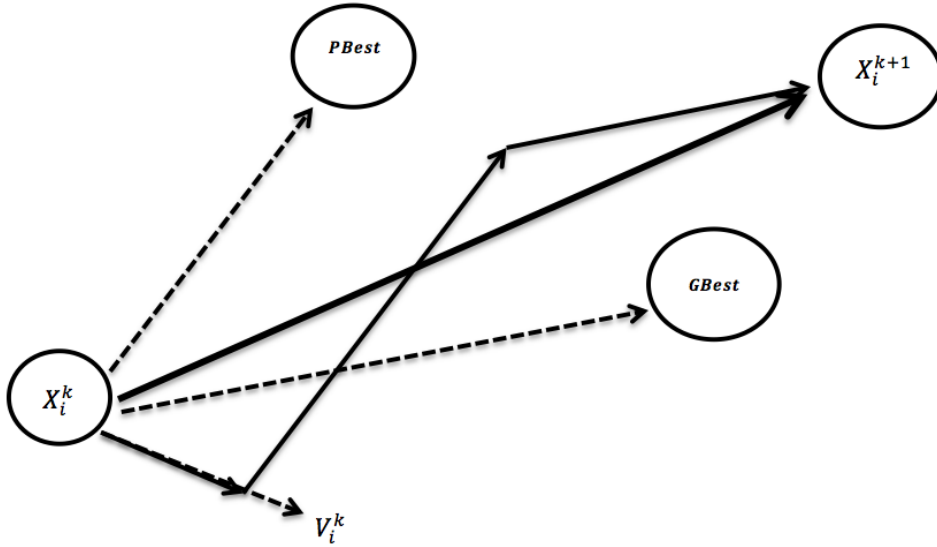
Denklemler (4.2) ve (4.3)’de yer alan tanımlamalar ise şu şekildedir:

- **Parçacık sayısı:** Optimize edilecek problemin boyutuna göre değişmekle birlikte genelde 10 ile 40 arasında alınır. Problemin boyutu arttıkça parçacık sayısı da artırılmalıdır. Her bir parçacığın numarası  $i$  indisi ile gösterilir.
- **Öğrenme Faktörleri:**  $c_1$  ve  $c_2$  öğrenme faktörleri  $G_{best}$  ve  $P_{best}$  değerlerinin bir sonraki konuma ne kadar etki edeceğini belirler. Genelde  $[0,4]$  aralığında seçilir. Düşük seçilmesi hedefe ulaşmayı geciktirmesine rağmen daha yakın olarak hedefe ulaşmayı sağlarken, yüksek seçilmesi ise hedefe ulaşmayı hızlandırırken hedefin yakınında sürekli dolaşmaya sebep olabilir.



- **Pozisyon Vektörü:** Her parçacığın o anki konumunu ifade eder ve  $X_i^k$  ile gösterilir.  $k$  değişkeni iterasyon sayısını göstermektedir.
- **Hız Vektörü:** Parçacığın keşifsel davranışını modelleyerek bir sonraki konumu bulmada etkilidir.  $V_i^k$  ile gösterilir.  $k$  değişkeni iterasyon sayısını göstermektedir.
- **Atalet Ağırlığı:**  $w$  ile gösterilir.  $0.95$  ile  $0.4$  arasında bir değerde seçilir. Parçacığın keşifsel davranışının bir sonraki konumunu belirlemede ne kadar etkili olacağını belirler. Başlangıçta yüksek seçilip sonrasında değeri azaltılabilirken, tüm algoritma boyunca sabit olarak da seçilebilir.

Denklemler (4.2) ve (4.3)'de yer alan  $r_1$  ve  $r_2$  değerleri  $0$  ile  $1$  arasında rastgele seçilmiş değerleri gösterir. Denklemler (4.2) ve (4.3) verilen hız ve konum güncellemeleri vektörel olarak Şekil 4.7 ile gösterilmiştir.



Şekil 4.7. Hız ve konumun vektörel olarak güncellenmesi

### 4.3. Denetleyici Parametrelerinin Belirlenmesinde PSO ve GA Kullanımı

Parçacık sürü optimizasyonu ve genetik algoritmanın uygulama alanlarından birisi de denetleyici parametrelerinin bu yöntemler ile belirlenmesidir. PSO ve GA sezgisel optimizasyon yöntemleri, denetleyici hata değerini kullanarak çıkış değerinin geçici hal cevabının aşma miktarını (overshoot), oturma süresini (settling time) ve kararlı hal hatasını (steady state error) minimum yapacak denetleyici katsayılarını çözüm uzayında tarar ve istenen performansı veren katsayıları seçer.

$PI^\lambda D^\mu$  denetleyicisinde  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  olmak üzere belirlenmesi gereken beş farklı parametre bulunmaktadır. Zaman bölgesindeki hata değeri kullanılarak kesir dereceli  $PID$  denetleyicisine ait katsayıların kestirimi yapılabilir. Hata değeri,  $e(t)$ , denklem (4.4) ile verilmektedir.  $y(t)$  çıkış değerini,  $r(t)$  ise istenen referans değeri göstermektedir.

$$e(t) = y(t) - r(t) \quad (4.4)$$

Sistemin çıkış performansının artırılmasında sıklıkla kullanılan dört farklı amaç fonksiyonu kriteri bulunmaktadır. ISE (Integral Squared Error), IAE (Integral Absolute Error), ITAE (Integral Time-Weighted Absolute Error) ve ITSE (Integral Time-Weighted Squared Error) olarak adlandırılan bu kriterler (4.4) denklemleri ile tanımlanmıştır.

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

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

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

$$ITSE = \int_0^{\infty} te^2(t) dt$$

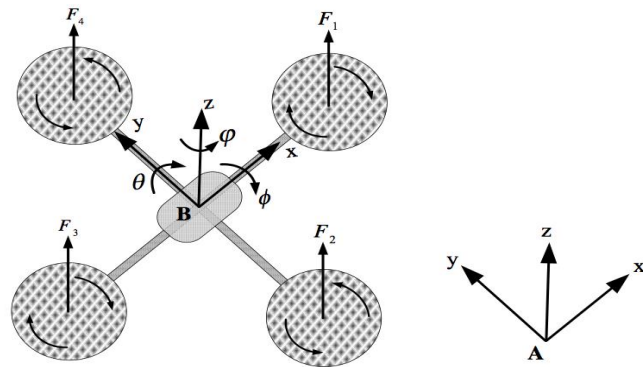
## 5. QUADROTOR

Döner kanatlı İHA grubunda yer alan quadrotorlar, manevra kabiliyetleri, basit tasarımları, dikey kalkış/iniş yapabilmenin yanı sıra havada asılı kalabilme kabiliyetleri ve boyutlarından dolayı dış ortamlara ilaveten iç ortamlarda da rahatlıkla kullanılabilir olmaları sebebiyle özellikle deneysel çalışmalarda oldukça öne çıkan platformlardır (Henriques, 2011; Michini, 2011; Azfar and Hazry,2011).

### 5.1. Quadrotor Hareket Tanımları

Quadrotorda simetrik olarak yerleştirilmiş, pervaneleri sabit hücum açılı ya da değişken hücum açılı olabilen dört adet rotor bulunur. Quadrotorun davranışı, pervanelerinin birbirlerine göre devir sayılarının ayarlanmasına bağlı olarak dört rotorun dönüş hızlarının (açısal hızlarının) değiştirilmesi ile kontrol edilir (Domingues, 2009; Raza and Gueaieb, 2010). Quadrotorda, karşılıklı yerleşmiş bir rotor çifti saat yönünde dönerken, diğer karşılıklı yerleşmiş rotor çifti ise saat yönünün tersine dönmektedir.

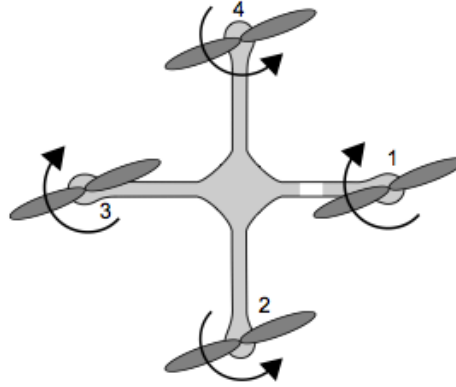
Rotorların hızları ayarlanarak rotora bağlı pervanelerin açısal hızı ve rotorların ürettiği itki kuvvetleri değiştirilir. Quadrotorun  $x$ ,  $y$ ,  $z$  yönlerinde hareket etmesi için yalpalama (roll,  $\phi$ ), yunuslama (pitch,  $\theta$ ) ve sapma (yaw,  $\psi$ ) açılarının rotorların hızları kontrol edilerek değiştirilmesi gerekmektedir. Şekil 5.1'de en genel halde rotorların dönüş yönü, açılar ve itki kuvvetlerinin yönleri gösterilmiştir.



Şekil 5.1. Quadrotor üzerine etkiyen kuvvetler, açılar ve hareket eksenleri (Friis ve ark., 2009)

### 5.1.1. Düşey eksenindeki hareket ve havada asılı kalma

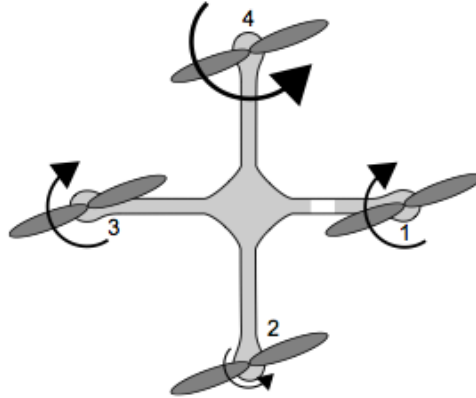
Quadrotorun havada asılı kalabilmesi için dört rotorun ürettiği toplam itki miktarının quadrotorun toplam ağırlığına eşit olması gerekir. İrtifa kazanmak için rotorların dönüş hızları aynı oranda artırılmalı, irtifa kaybetmek için ise rotorların dönüş hızları aynı oranda azaltılmalıdır. Bu sayede quadrotor z-ekseninde hareket eder (Şekil 5.2).



Şekil 5.2. Havada asılı kalma (Friis ve ark., 2009)

### 5.1.2. Yalpalama açısı (roll angle, $\phi$ )

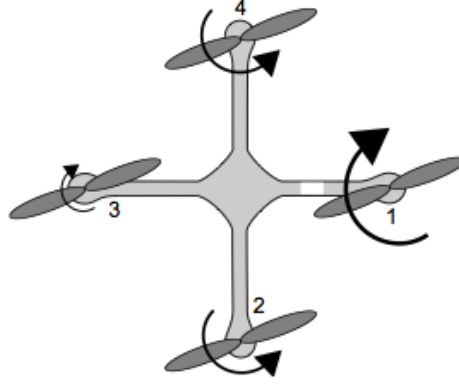
Yalpalama açısı,  $\phi$ , quadrotorun x-ekseni etrafında yaptığı açıdır ve saat yönünde pozitif olarak kabul edilir. 1 ve 3 numaralı rotorların hızları sabit olmak koşuluyla, 4 numaralı rotorun hızı artırılıp 2 numaralı rotorun hızı azaltıldığı zaman yalpalama açısı oluşur ve bu sayede quadrotor y ekseninde hareket eder (Şekil 5.3).



Şekil 5.3. Yalpalama hareketi (Friis ve ark., 2009)

### 5.1.3. Yunuslama açısı (pitch angle, $\theta$ )

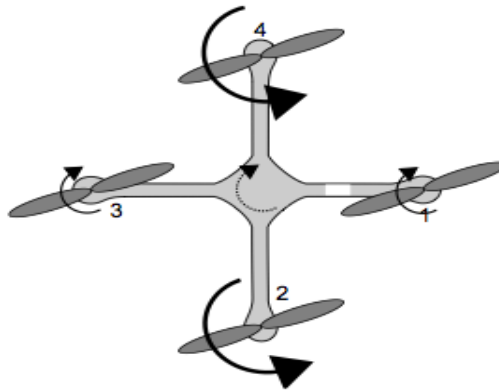
Yunuslama açısı,  $\theta$ , quadrotorun  $y$ -ekseni etrafında yaptığı açıdır ve saat yönünde pozitif olarak kabul edilir. 2 ve 4 numaralı rotorların hızları sabit olmak koşuluyla, 3 numaralı rotorun hızı azaltılıp 1 numaralı rotorun hızı artırıldığı zaman yunuslama açısı oluşur ve bu sayede quadrotor  $x$ -ekseninde hareket eder (Şekil 5.4).



Şekil 5.4. Yunuslama hareketi (Friis ve ark., 2009)

### 5.1.4. Sapma açısı (yaw angle, $\psi$ )

Sapma açısı,  $\psi$ , quadrotorun  $z$ -ekseni etrafında yaptığı açıdır, saat yönünde pozitif olarak kabul edilir. 1 ve 3 numaralı pervane çifti ile 2 ve 4 numaralı pervane çifti arasında hız farkı varsa quadrotor sapma hareketi yapar. 1 ve 3 numaralı pervane çiftinin açısal hızı 2 ve 4 numaralı pervane çiftinden büyükse saat yönünün tersi yönünde sapma hareketi, aksi halde ise saat yönünde sapma hareketi oluşur (Şekil 5.5).



Şekil 5.5. Sapma hareketi (Friis ve ark., 2009)

## 5.2. Quadrotorun Matematiksel Modeli

Quadrotorun benzetimli kontrolünü sağlamak için matematiksel modelinin çıkarılması gerekmektedir. Matematiksel modeli elde etmek üzere yapılan kabuller şu şekildedir:

- Quadrotorun gövdesi simetrik yapıdadır.
- Quadrotor ve pervaneler sabit kütleli katı (rijit) yapıdadır.
- Quadrotorun ağırlık merkezi orijindedir.
- Rotorların ürettiği itki pervanelerin açısal hızının karesiyle doğru orantılıdır.
- Yer etkisi ihmal edilmektedir.

Quadrotor  $x$ ,  $y$ ,  $z$  eksenlerinde ve  $\theta$ ,  $\phi$ ,  $\psi$  açısal koordinatlarında hareket edebildiğinden 6 serbestlik derecesine sahiptir. 6 serbestlik derecesindeki hareketini tanımlamak için Newton-Euler denklemi kullanılabilir. Bu yöntemde dünya yerçekimi merkezine göre ve quadrotorun merkezine göre olmak üzere iki eksen takımı kullanılmaktadır. Bu eksenler arasındaki geçişi sağlayan rotasyon matrisi Euler denklemine göre (5.1)'de verilmektedir. İşlemler dünya yerçekimi merkezi eksenine göre yapılır (Shames, 1997).

$$R = \begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi - S\psi S\phi & C\psi S\theta C\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi - C\psi C\phi & S\psi S\theta C\phi + C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \quad (5.1)$$

Denklem (5.1)'de  $C$  kosinüs,  $S$  ise sinüs işlemini ifade etmektedir.

Quadrotorun dünya yerçekimi eksenine göre üç-boyutlu konumu, hızı ve ivmesi (5.2) - (5.4) denklemleri ile ifade edilir.

$$\zeta = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.2)$$

$$v = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (5.3)$$

$$\dot{v} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (5.4)$$

Her bir rotorun ürettiği itki kuvveti,

$$T_i = b\Omega_i^2 \quad (5.5)$$

ile ifade edilir. Denklem (5.5)'de  $\Omega_i$ ,  $i$ 'nci motorun açısal hızını,  $b$  itki kuvveti katsayısını göstermektedir. Toplam kuvvet ise,

$$T_T = b\sum_{i=1}^4 \Omega_i^2 \quad (5.6)$$

ile verilmektedir. Newton'un ikinci kanunundan dolayı oluşan ivme,

$$a_F = \frac{b}{m} \sum_{i=1}^4 \Omega_i^2 \quad (5.7)$$

denklemleri ile elde edilir.  $m$  quadrotorun toplam kütleini,  $a_F$  ivmesinin Şekil 5.1'deki A eksen takımına etkisi  $R_{aF}$  olarak ifade edilirse, düşey doğrultudaki ivme dengesi,

$$\dot{v} = -g + R_{aF} \quad (5.8)$$

ile gösterilmektedir. Denklem (5.8)'de  $g$  yer çekimi ivmesini göstermektedir. Bu denklem takımları çözülürse,

$$\ddot{x} = (C\phi S\theta C\psi + S\phi S\psi) \frac{b}{m} (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (5.9)$$

$$\ddot{y} = (C\phi S\theta C\psi - S\phi S\psi) \frac{b}{m} (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (5.10)$$

$$\ddot{z} = -g + (C\phi C\theta) \frac{b}{m} (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (5.11)$$

$x$ ,  $y$  ve  $z$  eksenlerindeki atalet momentlerine bakılırsa aracın dönüş simetrisinden dolayı  $I_{xy}$ ,  $I_{xz}$ ,  $I_{zy}$  momentleri sıfır ve  $I_x = I_y$  olmaktadır (Bedford ve Fawler, 2005).

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (5.12)$$

$$\ddot{\phi} = \dot{\psi}\dot{\theta} \left( \frac{I_y - I_z}{I_x} \right) - \frac{J}{I_x} \dot{\theta} (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{lb}{2I_x} (-\Omega_2^2 + \Omega_4^2) \quad (5.13)$$

$$\ddot{\theta} = \dot{\psi}\dot{\phi} \left( \frac{I_z - I_x}{I_y} \right) - \frac{J}{I_y} \dot{\phi} (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{lb}{2I_y} (-\Omega_1^2 + \Omega_3^2) \quad (5.14)$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta} \left( \frac{I_x - I_y}{I_z} \right) + \frac{d}{I_z} (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (5.15)$$

Denklemler (5.13) – (5.15)'deki  $l$  rotorlar ile quadrotorun merkezi arasındaki mesafeyi,  $b$  itki kuvveti katsayısını,  $d$  ise sürüklenme kuvveti katsayısını gösterir.

Açıları ve dolayısıyla  $x$ ,  $y$  ve  $z$  yönlerindeki hareketleri kontrol edecek olan değişken,  $\Omega$  ile gösterilmekte olan rotorların açısal hızıdır. Bu sebepten denetleyici çıktıları olan,  $U_1$ ,  $U_2$ ,  $U_3$ ,  $U_4$  denklemler (5.16) - (5.19) ile ifade edilebilir (Bouabdallah ve Siegwart, 1999).

$$U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (5.16)$$

$$U_2 = b(-\Omega_2^2 + \Omega_4^2) \quad (5.17)$$

$$U_3 = b(-\Omega_1^2 + \Omega_3^2) \quad (5.18)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (5.19)$$



$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (5.20)$$

$U_1$  girdisi  $z$ -eksenindeki hareketi,  $U_2$  girdisi yalpalama açısını yani  $y$  eksenindeki hareketi,  $U_3$  girdisi yunuslama açısını yani  $x$  eksenindeki hareketi son olarak  $U_4$  girdisi ise sapma açısını kontrol eder.

Quadrotora ait en genel haldeki hareket denklemleri (5.21) - (5.26) ile verilmektedir.

$$\ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{1}{m}U_1 \quad (5.21)$$

$$\ddot{y} = (\cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi)\frac{1}{m}U_1 \quad (5.22)$$

$$\ddot{z} = -g + (\cos\phi\cos\theta)\frac{1}{m}U_1 \quad (5.23)$$

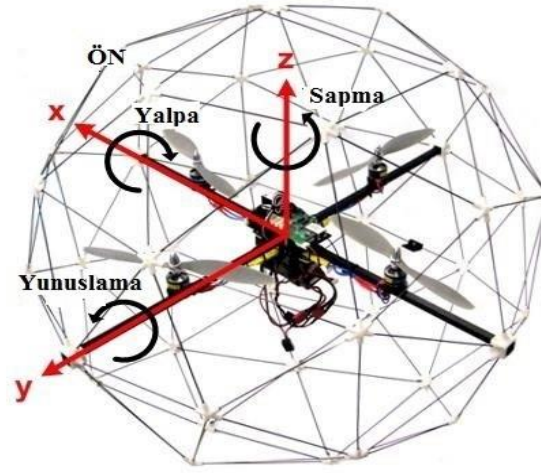
$$\ddot{\phi} = \dot{\psi}\dot{\theta}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J}{I_x}\dot{\theta}\Omega_r + \frac{l}{2I_x}U_2 \quad (5.24)$$

$$\ddot{\theta} = \dot{\psi}\dot{\phi}\left(\frac{I_z - I_x}{I_y}\right) - \frac{J}{I_y}\dot{\phi}\Omega_r + \frac{l}{2I_y}U_3 \quad (5.25)$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \quad (5.26)$$

### 5.3. QBall-X4 Quadrotor

Tez kapsamında yapılan çalışmada quadrotor olarak Quanser firmasına ait QBall-X4 kullanılmıştır (Şekil 5.6). Koruyucu karbon-fiber bir kafes ile çevrelenmiş olan bu quadrotor iç mekânlarda yaklaşık  $4x4m^2$  lik alanda, üzerinde taşıdığı kamera ve sensörler aracılığıyla otonom veya uzaktan kumanda edilerek uçabilmektedir.



Şekil 5.6. QBall-X4 (Quanser Inc., 2010)

QBall-X4 eyleyici dinamikleri

$$T = K \frac{W}{s+W} u \quad (5.27)$$

$$v = \frac{W}{s+W} u \quad (5.28)$$

denklemleri ile verilmektedir. Denklemler (5.27) ve (5.28)'de  $u$  eyleyici girişindeki PWM (Pulse-Width Modulation) değerini,  $v$  ise eyleyici dinamiğini göstermektedir. Bu parametreler çeşitli deneyler sonucunda hesaplanmış değerlerdir (Quanser Inc., 2010).

QBall-X4 parametrelerinin değerleri Çizelge 5.1'de verilmektedir.

**Çizelge 5.1.** QBall-X4 parametreleri [64]

<b>Parametre</b>	<b>Açıklama</b>	<b>Değer</b>
$K$	Pozitif kazanç	120 $N$
$W$	Eyleyici bant genişliği	15 $rad/sn$
$I_y$	Atalet momenti	0.03 $kgm^2$
$I_x$	Atalet momenti	0.03 $kgm^2$
$m$	Toplam ağırlık	1.4 $kg$
$K_y$	Pozitif kazanç	4 $Nm$
$I_z$	Atalet momenti	0.03 $kgm^2$
$l$	Rotor-quadrotor merkezi arası mesafe	0.2 $m$

QBall-X4'e ait doğrusallaştırılmış hareket denklemleri, denklemler (5.29) - (5.34) ile verilmektedir (Quanser Inc., 2010).

$$\ddot{\theta} = \frac{U_3}{I_y} \quad (5.29)$$

$$\ddot{\phi} = \frac{U_2}{I_x} \quad (5.30)$$

$$\ddot{\psi} = \frac{U_4}{I_z} \quad (5.31)$$

$$\ddot{X} = \frac{U_1}{m} \sin \theta \quad (5.32)$$

$$\ddot{Y} = -\frac{U_1}{m} \sin \phi \quad (5.33)$$

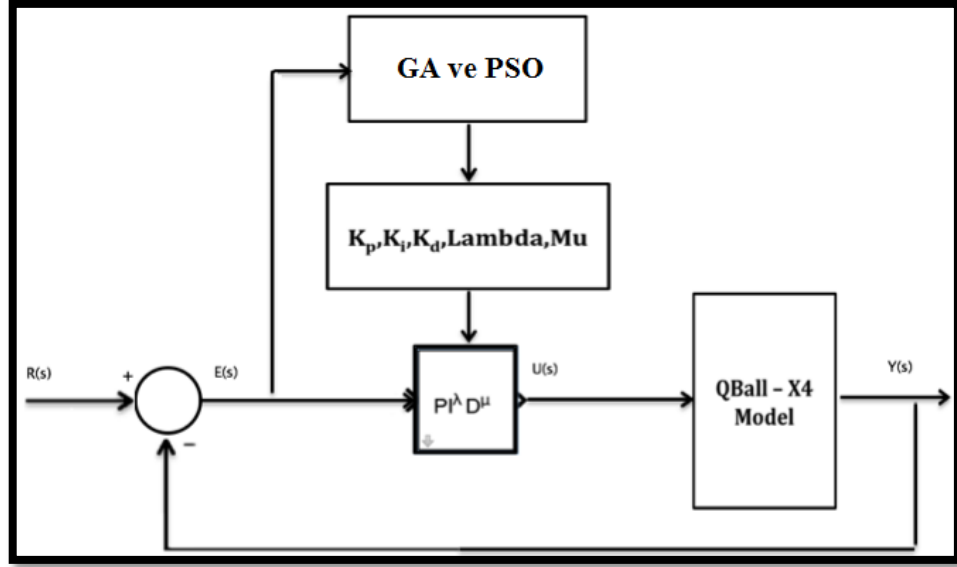
$$\ddot{Z} = \frac{U_1}{m} - g \quad (5.34)$$

## 6. YAPILAN ÇALIŞMALAR VE BENZETİMLER

Tez kapsamında olduğu yerden dikine kalkış/iniş yapabilen, 6 serbestlik derecesine sahip ( $x$ ,  $y$ ,  $z$  ve  $\theta$ ,  $\phi$ ,  $\psi$ ), dört rotorlu quadrotor tipi İHA'nın doğrusal olmayan hareket denklemleri çıkarılmıştır. Bu hareket denklemleri doğrusallaştırıldıktan sonra bu denklemlere ait matematiksel model MATLAB/Simulink ortamında oluşturulmuştur. Quadrotorun 6 serbestlik derecesindeki hareketinin kontrolü hem kesir dereceli *PID* denetleyicisi ile hem de klasik *PID* denetleyicisi ile sağlanmış ve performansları karşılaştırılmıştır. Şekil 6.1'de kontrol çevrimi blok diyagramı verilen yapıda,  $PI^\lambda D^\mu$  denetleyicisine ait  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  parametreleri ve klasik *PID* denetleyicisine ait  $K_p$ ,  $K_i$  ve  $K_d$  parametreleri sezgisel optimizasyon yöntemleri olan parçacık sürü optimizasyonu ve genetik algoritma kullanılarak belirlenmiştir.  $PI^\lambda D^\mu$  denetleyicisinin Simulink ortamında benzetimi için FOMCON Toolbox'ı (Cao ve Cao, 2006a) kullanılmıştır.

Parametrelerin belirlenmesinde amaç fonksiyonu kriteri olarak daha önceki bölümlerde bahsi geçen dört kriter olan ISE, IAE, ITAE ve ITSE eşitlikleri kullanılmıştır.

Benzetimler için yazılan kodlar MATLAB/Script dosyası ve MATLAB/Simulink dosyası olmak üzere iki kısımda ele alınmıştır. Script dosyasında yazılan GA ve PSO kodları sayesinde taranan çözüm uzayının sınırları içerisinde oluşturulan başlangıç popülasyon değerleri kullanılıp  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  değerleri belirlenir. Bu değerleri kullanmak üzere modelin de yer aldığı Simulink dosyası, “*sim(dosya adı)*” komutu sayesinde çağrılır ve çalıştırılır. İstenen kriter matematiksel formülasyonundan yararlanılarak aynı Simulink dosyası içerisinde oluşturulur. Dosya çalıştırıldıktan sonra istenen kriter değeri elde edilir. Bu değer “To Workspace” bloğu kullanılarak “Workspace” penceresine taşınır ve Script dosyası bu değeri amaç fonksiyonu olarak kullanır ve program kaldığı yerden devam eder. Program tekrar ettikçe bu fonksiyonu minimum yapan katsayılar bulunmaya çalışılır.



Şekil 6.1. QBall-X4, kontrol çevrimi blok diyagramı

Parçacık sürü optimizasyonu ve genetik algoritmanın MATLAB ortamında koşturulmaları anında kullanılan parametreler Çizelge 6.1 ile verilmiştir. Farklı kriterler için koşturulan algoritmalar sonucunda elde edilen denetleyici parametreleri ve performans ölçütleri olan maksimum aşma miktarı (maximum overshoot), kararlı hal hatası (steady-state error), titreşim (oscillation), yükselme zamanı (rising time) ve oturma zamanına (settling time) ait değerleri içeren tablolar, çıkış grafikleri ve amaç fonksiyon değerleri bu bölümde verilmektedir.

Çizelge 6.1. Quadrotor kontrolü için optimizasyon parametreleri

		<i>PID</i>	<i>KD PID</i>
<b>PSO</b>	<b>Popülasyon</b>	60	60
	<b>Tekrar sayısı</b>	100	100
	<i>W</i>	0.9,0.8,...0.1	0.9,0.8,...0.1
	<i>c<sub>1</sub></i>	0.12	0.12
	<i>c<sub>2</sub></i>	2.5	2.5
<b>GA</b>	<b>Popülasyon</b>	60	60
	<b>Tekrar sayısı</b>	100	100
	<i>P<sub>c</sub></i>	0.9	0.9
	<i>P<sub>m</sub></i>	0.01	0.01

Çizelge 6.1’de popülasyon değerleri ve tekrar sayıları sonuçların kıyaslanabilmesi için tüm denetleyici ve optimizasyon yöntemlerinde aynı seçilmiştir. PSO yönteminde güncelleme sırasında bir önceki hız bileşeninin katsayısı olan  $W$  (atalet ağırlığı) başlangıçta  $0.9$  seçilip gittikçe tekrar sayısına bağlı olarak  $0.1$ ’e doğru doğrusal şekilde azalacak şekilde seçilmiştir.  $c_1$  ve  $c_2$  öğrenme faktörleri ise  $0.12$  ve  $2$  olarak seçilmiştir. Bu parametreler genelde  $0$  ile  $4$  arasında seçilmektedir. Parametrelerin değerleri arttıkça parçacıklar uzaya daha iyi yayılır fakat optimum çözüme yaklaşma süresi artar.  $P_c$  ve  $P_m$  ile GA’ya ait çaprazlama ve mutasyon oranları verilmiştir.

Quadrotoru  $x$ ,  $y$ ,  $z$  yönlerinde istenen güzergâhta hareket ettirmek motorları farklı hızlarda döndürmekle mümkün olacaktır. Ayrıca  $x$  ve  $y$  yönlerinde ki hareket için öncelikle quadrotorun yunuslama ve yalpalama açılarının kontrol edilmesi gerekmektedir.

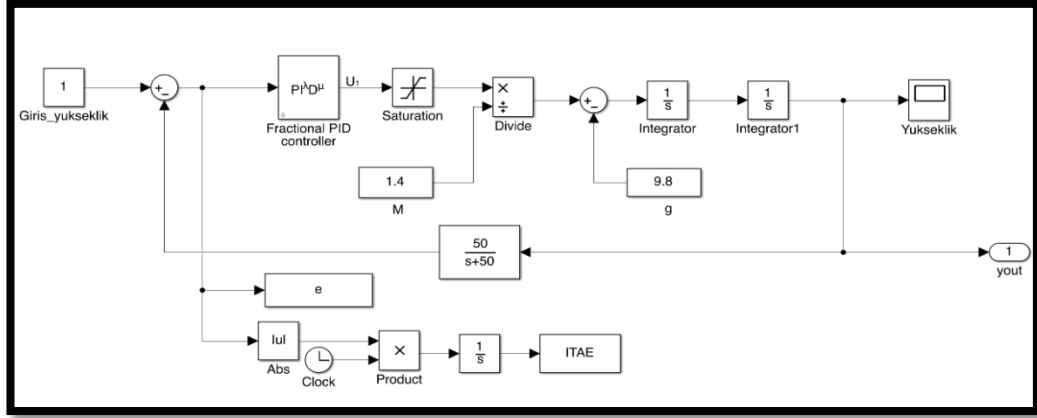
Bu bölümde sırasıyla  $z$  (yükseklik) kontrolü yapılmış, daha sonra  $x$  ve  $y$  yönlerindeki kontrol için de öncelikle yunuslama ve yalpalama açılarının kontrolü sağlanmıştır. En son olarak ise sapma açısının kontrolü sağlanmıştır.

### 6.1. QBall-X4 Yükseklik Kontrolü

Quadrotorun yükseklik kontrolü için kullanılacak denklemler (6.2.a-b) ile ve MATLAB/Simulink ortamındaki benzetimi de Şekil 6.2 ile verilmiştir.

$$\ddot{Z} = \frac{(T_1 + T_2 + T_3 + T_4)}{M} - g, \quad (6.2.a)$$

$$T_1 + T_2 + T_3 + T_4 = U_1 \quad (6.2.b)$$



Şekil 6.2. Yükseklik kontrolü blok diyagramı

Şekil 6.2 ile verilen QBall-X4 quadrotorunun Simulink blok diyagramı kullanılarak bu model için farklı kriterlerin karşılaştırılması açısından MATLAB/Script dosyası halinde hazırlanan algoritmalar koşturularak Çizelge 6.2-6.5’de verilen performans değerleri elde edilmiştir.  $x$  ve  $y$  pozisyonlarının kontrolünde  $U_1$  kontrolcü çıkışı kullanıldığından dolayı yükseklik kontrolünde en iyi performansla çalışan denetleyici katsayıları diğer denetleyiciler tasarlanırken de kullanılmıştır.

Çizelge 6.2 ile GA kullanılarak farklı kriterler için bulunan  $PID$  denetleyicisi parametreleri ve bu parametreler kullanılarak elde edilen performanslar gösterilmektedir.

MATLAB ortamında hazırlanan genetik algoritma kodunun koşturulmasıyla farklı kriter değerlerine ait amaç fonksiyonlarının karşılaştırılması Şekil 6.3 ile verilmiştir. En düşük amaç fonksiyonunun değeri ISE kriteri kullanılarak 3.56 olarak elde edilmiştir.

Koşturulan GA koduyla elde edilen  $PID$  denetleyicisine ait en iyi parametreler kullanılarak oluşturulan çıkış değerlerinin karşılaştırılması Şekil 6.4’de verilmiştir. Grafiğe bakıldığı zaman ITSE kriteri hariç diğer kriterlerin tamamının küçük salınımlar yaptığı, ITSE kriterinin maksimum aşım miktarının çok düşük olduğu, yükselme zamanlarının tüm kriterler için benzer olduğu ISE kriterinin ise en kısa sürede istenen yüksekliğe oturduğu gözlenmiştir.

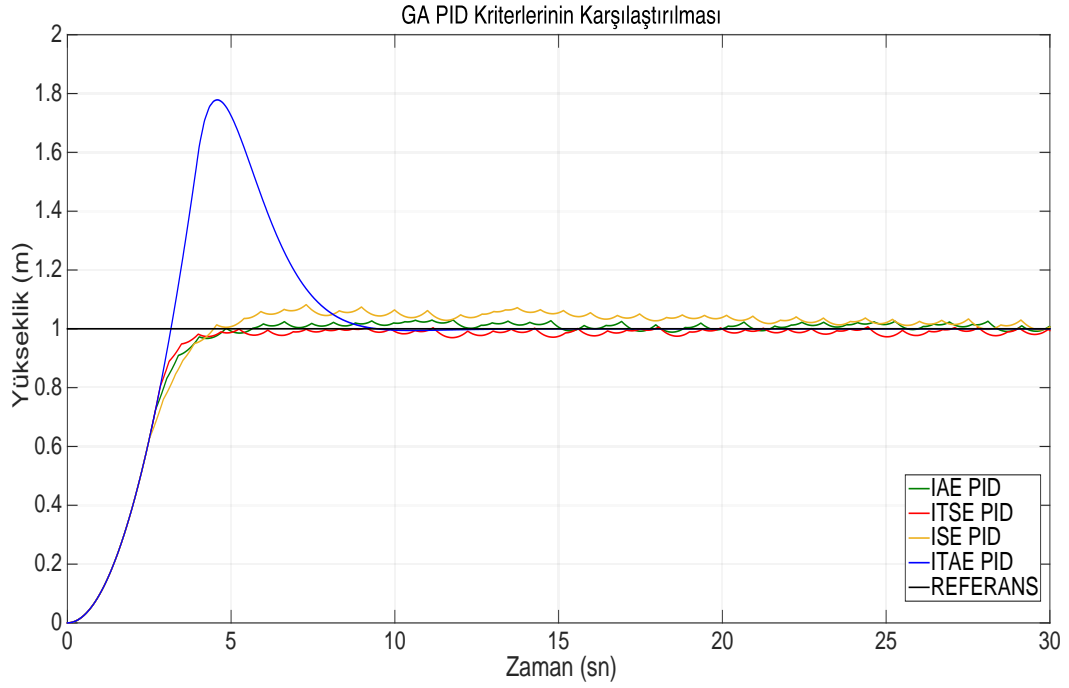
**Çizelge 6.2.** Yükseklik kontrolünde farklı kriter değerleri için GA *PID* denetleyicisine ait performansların karşılaştırılması

Parametre Değerleri	GA <i>PID</i>			
	ITAE	ITSE	ISE	IAE
$K_p$	166.5141	740.3406	860.0989	683.4524
$K_i$	97.4181	8.4234	55.1791	19.4714
$K_d$	99.6155	329.9152	981.3831	486.1747
Maksimum Aşma Miktarı	%77.8	<b>%0.6</b>	%8.2	%2.6
Kararlı hal hatası	<b>0.0001</b>	0.03	0.0352	0.0309
Titreşim	<b>Yok</b>	Var	Var	Var
Yükselme zamanı	2.51sn	2.51sn	2.52sn	2.51sn
Oturma zamanı	9.25sn	5.2sn	<b>4.5sn</b>	4.87sn
En İyi Amaç Fonksiyonu Değeri	24.76	6.859	<b>3.56</b>	5.93



**Şekil 6.3.** GA *PID* ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi





**Şekil 6.4.** GA *PID* ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması

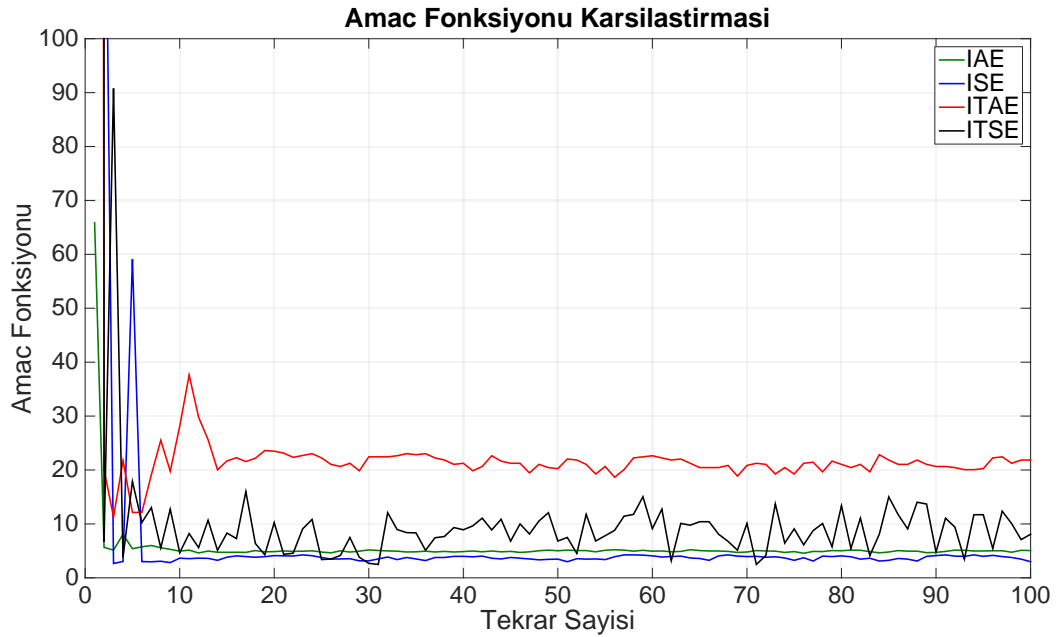
Çizelge 6.3 ile GA kullanılarak farklı kriterler için bulunan kesir dereceli *PID* denetleyicisi parametreleri ve bu parametreler kullanılarak elde edilen performanslar gösterilmiştir.

Şekil 6.5’de verilen grafikteGA kullanılarak elde edilen farklı kriterlere ait *KD PID* denetleyicisi çıktısına bakıldığı zaman en düşük amaç fonksiyonu değerini 2.482 değeriyle ITSE kriteri vermiştir. IAE kriteri ise daha kısa sürede amaç fonksiyon değerini azaltmıştır.

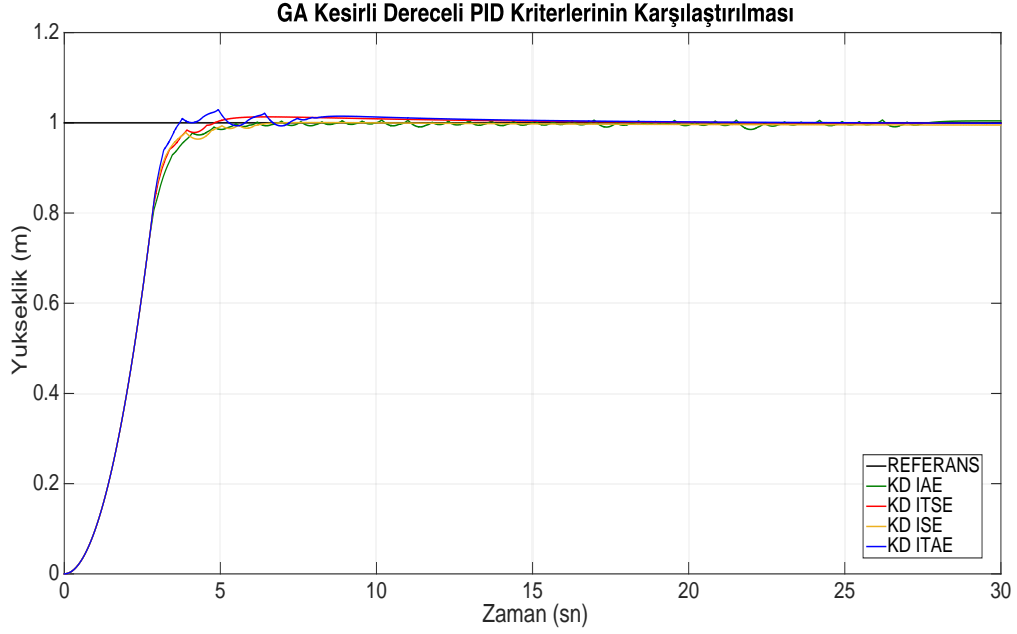
Çizelge 6.3’ de elde edilen parametre değerleri kullanılarak Şekil 6.6 grafiği elde edilmiştir. Bu grafiğe göre ISE kriterinde maksimum aşım gözlenmemiş olup, kararlı hal hatası tüm kriterler için neredeyse sıfır seviyesindedir. İstenen referans yüksekliğe en kısa sürede ITAE kriteri varmıştır.

**Çizelge 6.3.** Yükseklik kontrolünde farklı kriter değerleri için GA KD *PID* denetleyicisine ait performansların karşılaştırılması

Parametre Değerleri	GA KD <i>PID</i>			
	ITAE	ITSE	ISE	IAE
$K_p$	840.5054	499.9695	438.8085	426.7839
$K_i$	588.72	899.987	902.5209	8.3013
$K_d$	733.1990	874.9924	620.5823	232.0088
$\lambda$	0.1662	0.1210	0.0734	0.9603
$\mu$	0.8772	0.75	0.7307	0.9358
Maksimum Aşma Miktarı	%2.9	%1.6	<b>%0</b>	%3.01
Kararlı hal hatası	<b>0.003</b>	0.004	0.0036	0.0326
Titreşim	<b>Yok</b>	Yok	Yok	Yok
Yükselme zamanı	2.52sn	2.52sn	2.52sn	<b>2.5sn</b>
Oturma zamanı	<b>3.7sn</b>	3.93sn	4.76sn	4.86sn
En İyi Amaç Fonksiyonu Değeri	11.28	<b>2.482</b>	2.65	4.55



**Şekil 6.5.** GA KD *PID* ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi



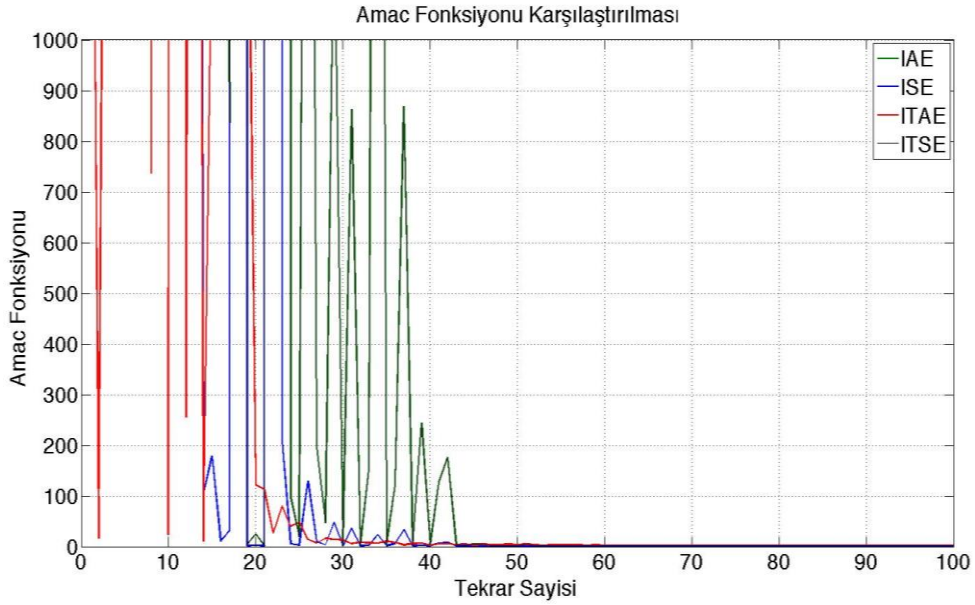
**Şekil 6.6.** GA KD *PID* ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması

Çizelge 6.4’de PSO algoritması koşturularak farklı kriterler için *PID* denetleyicisine ait parametreler ve performanslar gösterilmiştir. Çizelgeye bakıldığı zaman en iyi amaç fonksiyonu değeri *1.7006* ile ITSE kriteri kullanılarak elde edilmiştir.

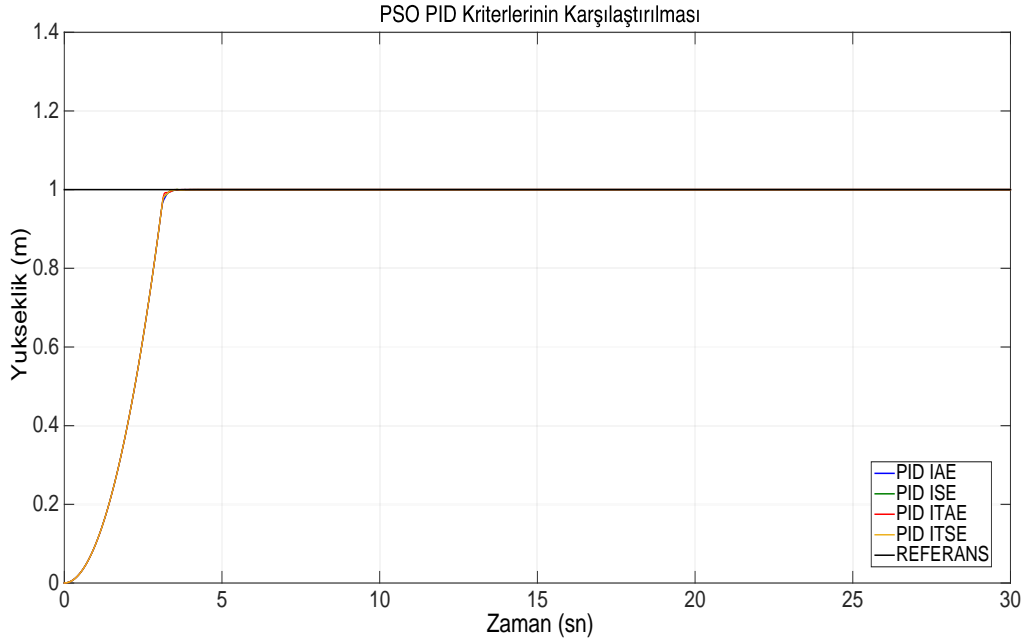
Şekil 6.7 - 6.8’de PSO algoritmasıyla elde edilen parametreler kullanılarak tasarlanan *PID* denetleyicisi farklı kriterler için çalıştırılmış ve amaç fonksiyon değerleri ile çıkış değerleri karşılaştırılmıştır. En iyi performansı ITSE ile ISE kriterleri göstermiştir.

**Çizelge 6.4.** Yükseklik kontrolünde farklı kriter değerleri için PSO *PID* denetleyicisine ait performansların karşılaştırılması

Parametre Değerleri	PSO <i>PID</i>			
	ITAE	ITSE	ISE	IAE
$K_p$	598.2593	869.6675	869.8547	502.2224
$K_i$	6.4402	6.4434	6.45	6.4355
$K_d$	52.0984	80.3969	80.4471	62.5817
Maksimum Aşma Miktarı	%0	%0	%0	%0
Kararlı hal hatası Titreşim	0 Yok	0 Yok	0 Yok	0 Yok
Yükselme zamanı	2.51sn	2.54sn	2.53sn	2.52sn
Oturma zamanı	3.39sn	3.40sn	3.41sn	3.39sn
En İyi Amaç Fonksiyonu Değeri	2.5526	1.7006	1.7064	2.13



**Şekil 6.7.** PSO *PID* ile yükseklik kontrolünde farklı kriterler için amaç fonksiyon değerlerinin değişimi



**Şekil 6.8.** PSO *PID* ile yükseklik kontrolünde farklı kriterler için çıkış performanslarının karşılaştırılması

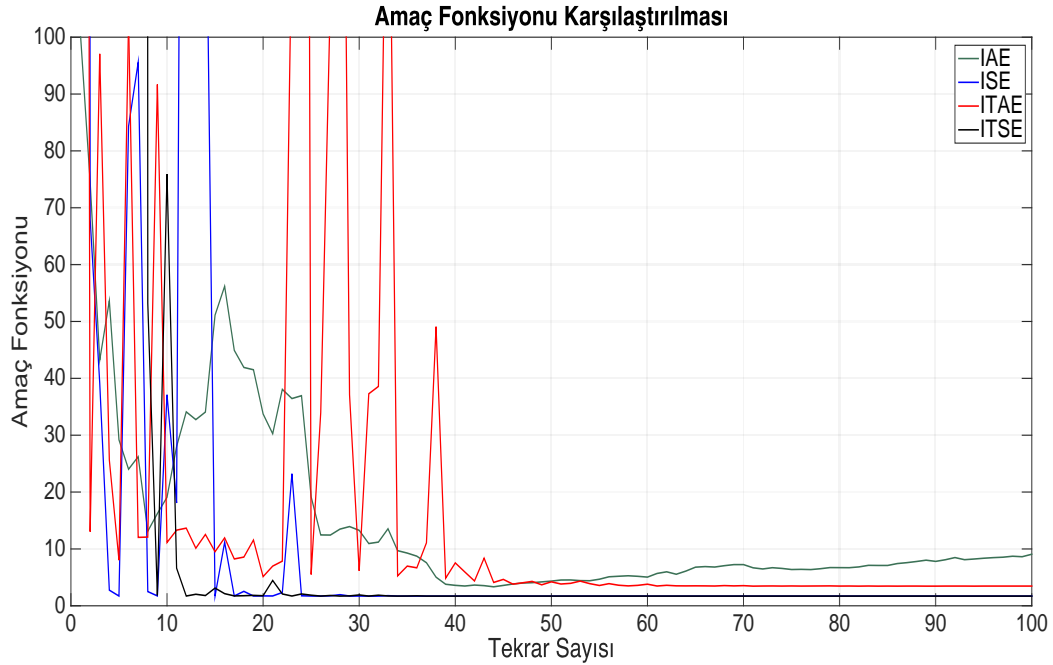
Çizelge 6.5’de QBall-X4 quadrotorunun yükseklik kontrolü için PSO algoritması koşturularak farklı kriterler için *KD PID* denetleyicisine ait parametreler ve performanslar gösterilmiştir. Çizelgeye bakıldığı zaman en iyi amaç fonksiyonu değeri  $1.6009$  ile ITSE kriteriyle elde edilmiştir.

Şekil 6.9’da farklı kriterlerin amaç fonksiyonlarının karşılaştırılması verilmiştir. En iyi performansı ITSE ve ISE kriterleri vermiştir.

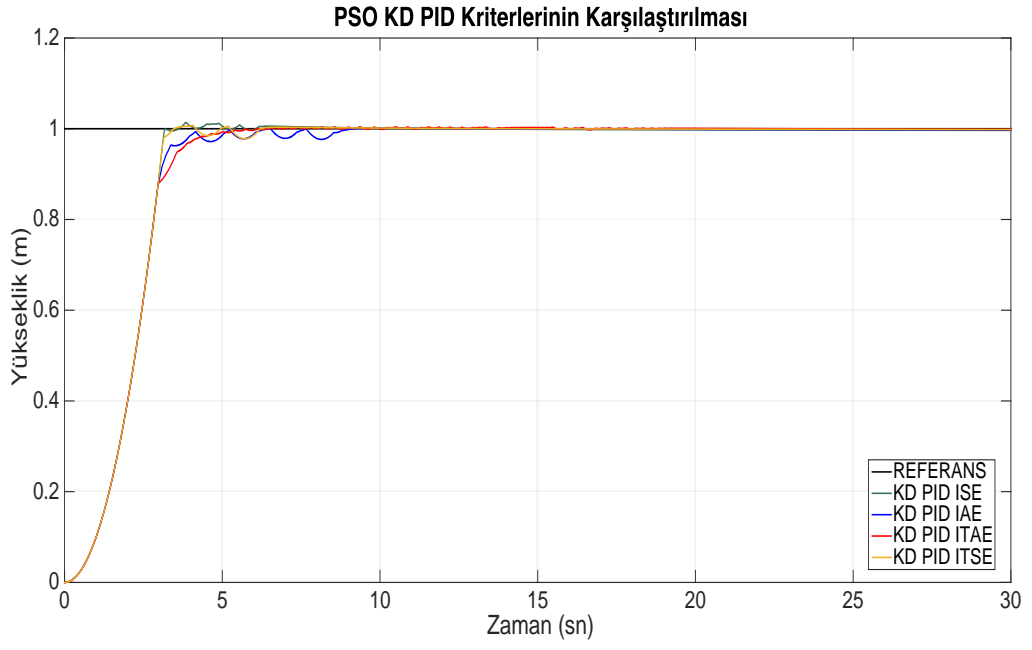
Şekil 6.10’ da ise farklı kriterler için elde edilen parametreler kullanılarak  $PI^\lambda D^\mu$  denetleyicisi tasarlanmıştır ve istenen referans yükseklik için çıkışlar karşılaştırılmıştır.

**Çizelge 6.5.** Yükseklik kontrolünde farklı kriter değerleri için PSO KD *PID* denetleyicisine ait performansların karşılaştırılması

Parametre Değerleri	PSO KD <i>PID</i>			
	ITAE	ITSE	ISE	IAE
$K_p$	852.9552	748.397	1000	3832.2
$K_i$	70.8063	223.8803	1000	1516.54
$K_d$	386.1697	406.4149	254.6835	858.81
$\lambda$	0.4647	0.2153	0.064	0.0543
$\mu$	0.6127	0.8815	0.8321	0.8131
Maksimum Aşma Miktarı	%0.4	%0.5	%1.4	%0.1
Kararlı hal hatası Titreşim	0.0003 Var	0.0001 Yok	0.0021 Var	0.002 Var
Yükselme zamanı	2.51sn	2.52sn	2.52sn	2.51sn
Oturma zamanı	4.9sn	3.2sn	3.153sn	4.197sn
En İyi Amaç Fonksiyonu Değeri	3.4503	1.6009	1.7066	3.3423



**Şekil 6.9.** PSO KD *PID* ile yükseklik kontrolünde, farklı kriterler için amaç fonksiyon değerlerinin değişimi



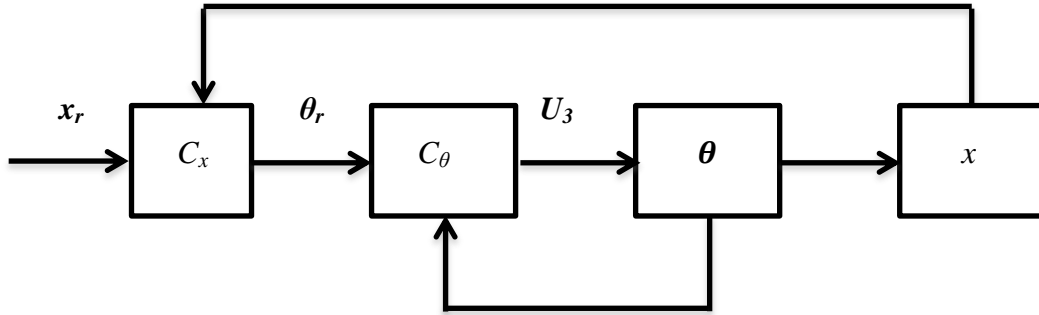
**Şekil 6.10.** PSO KD *PID* ile yükseklik kontrolünde, farklı kriterler için çıkış performanslarının karşılaştırılması

## 6.2. $x$ ve $y$ Pozisyon Kontrolü

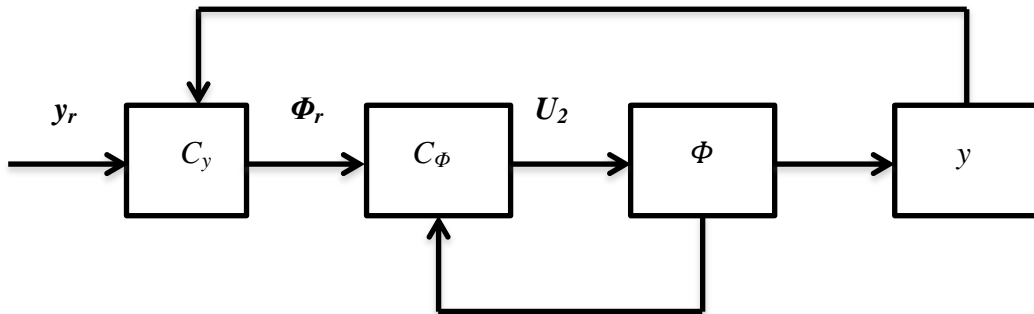
$x$  ve  $y$  pozisyon kontrolünün doğrusallaştırılmış denklemleri (6.3) ve (6.4) ile verilmiştir. Denklemlerden de anlaşılacağı üzere  $x$  ve  $y$  eksenlerinde kontrolün sağlanması için öncelikle yunuslama ve yalpalama açıları olan  $\theta$  ve  $\phi$  açılarının kontrol edilmesi gerekmektedir. Bu açıların kontrolü sağlandıktan sonra  $U_1$  yükseklik kontrolcüsünün çıkışı kullanılarak  $x$  ve  $y$  eksenlerinde quadrotorun kontrolü sağlanmalıdır. Şekil 6.11.a ve Şekil 6.11.b'de,  $x$  ve  $y$  eksenlerinde quadrotorun kontrolünün blok diyagramı gösterilmiştir.

$$\dot{X} = \frac{U_1}{M} \sin \theta, \quad (6.3)$$

$$\dot{Y} = -\frac{U_1}{M} \sin \phi, \quad (6.4)$$



Şekil 6.11(a).  $x$  eksen kontrol blok diyagramı



Şekil 6.11(b).  $y$  eksen kontrol blok diyagramı



$x$  ve  $y$  pozisyon kontrolü, iç içe iki çevrimden oluştuğundan dolayı quadrotorun  $x$  ve  $y$  eksenlerindeki kontrolünü sağlamak için öncelikle iç çevrim olan yunuslama ve yalpalama açılarının kontrol edilmesi gerektiği Şekil 6.11 ile gösterilmektedir.

### 6.2.1. Yunuslama ve yalpalama açılarının kontrolü

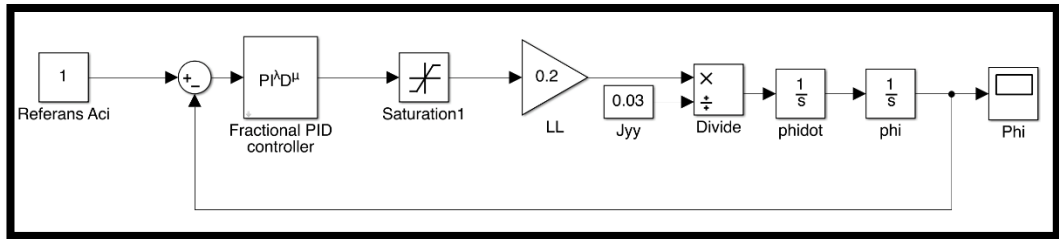
Yunuslama ve yalpalama açılarını  $PID$  ve  $KD PID$  denetleyicisi ile kontrol etmek için (6.5a-b) ve (6.6a-b) denklemleri kullanılmıştır. Bu denklemlerden yola çıkılarak Simulink blok diyagramı oluşturulmuştur (Şekil 6.12). Bu açıların kontrolünde yükseklik kontrolünde diğer kriterlere oranla daha iyi performans sergileyen ITSE kriteri kullanılmıştır ve  $GA PID$ ,  $GA KD PID$ ,  $PSO PID$ ,  $PSO KD PID$  denetleyicilerinin performansları karşılaştırılmıştır.

$$\ddot{\theta} = \frac{(T_1 - T_2)L}{I_y}, \quad (6.5.a)$$

$$T_1 - T_2 = U_3, \quad (6.5.b)$$

$$\ddot{\phi} = \frac{(T_3 - T_4)L}{I_x}, \quad (6.6.a)$$

$$T_3 - T_4 = U_4, \quad (6.6.b)$$



Şekil 6.12. Yunuslama ve yalpalama açılarının kontrol blok diyagramı

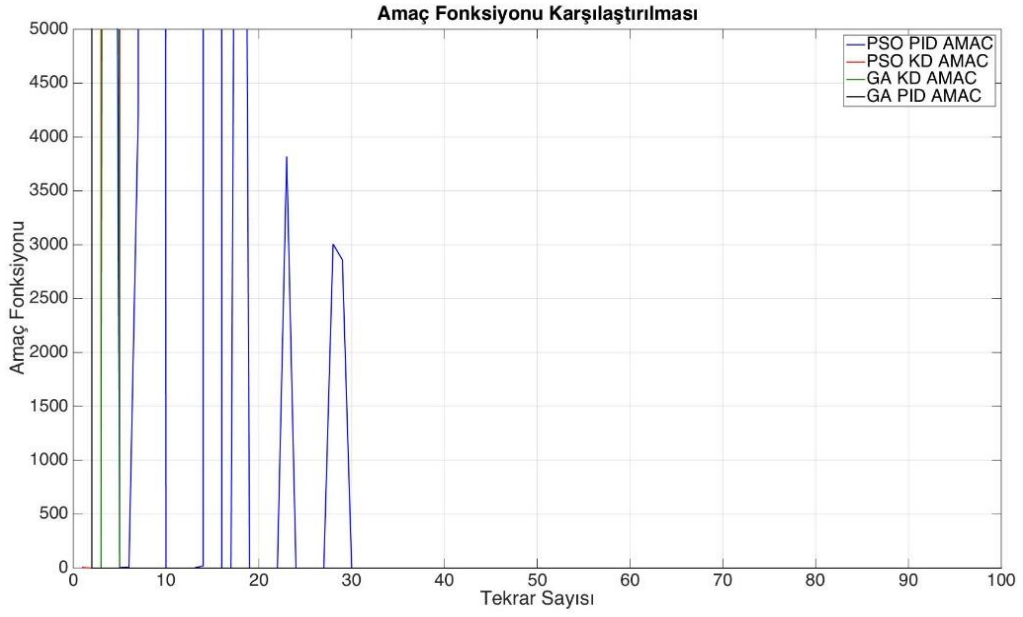
Çizelge 6.6’da yunuslama-yalpalama açılarının kontrolü için  $GA$  ve  $PSO$  algoritmaları koşturularak  $ITSE$  kriterinin  $PID$  ve  $KD PID$  denetleyicilerine ait parametreleri ve performansları gösterilmiştir. Tabloya bakıldığı zaman en iyi

amaç fonksiyonu değeri  $0.0041$  ile  $KD PID$  denetleyicisinde PSO algoritması koşturularak elde edilmiştir.

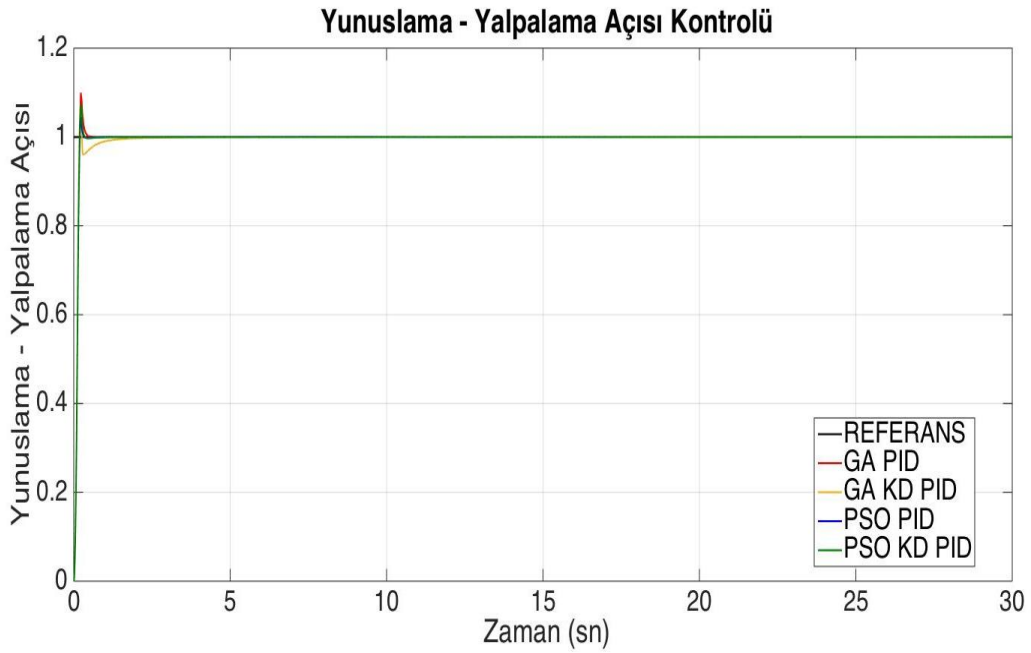
Şekil 6.13’de ITSE kriteri için amaç fonksiyonlarının karşılaştırılması verilmiştir. Şekil 6.14’de ise ITSE kriteri kullanılarak QBall-X4’ün istenen referans aç değeri yakınsamaları  $GA PID$ ,  $GA KD PID$ ,  $PSO PID$ ,  $PSO KD PID$  denetleyicileri için verilmiştir.

**Çizelge 6.6.** ITSE kriteri için aç kontrolünde farklı optimizasyon ve denetleyicilere ait performansların karşılaştırılması

Parametre değerleri	ITSE			
	$GA PID$	$GA KD PID$	$PSO PID$	$PSO KD PID$
$K_p$	963.07	846.9755	2968.3	499.2414
$K_i$	0.5493	988.8909	0.9315	0.012
$K_d$	50.357	539.5837	172.403	34.2595
$\lambda$	-	0.0309	-	0.1073
$\mu$	-	0.6287	-	0.9194
Maksimum Aşma Miktarı	%11	%4.5	<b>%2.7</b>	%7.4
Kararlı hal hatası Titreşim	0 Yok	0 Yok	0 Yok	0 Yok
Yükselme zamanı	0.13sn	0.1265sn	0.12sn	<b>0.1sn</b>
Oturma zamanı	0.6sn	1.86sn	0.5 sn	<b>0.37sn</b>
En İyi Amaç Fonksiyonu Değeri	0.0155	0.0147	0.0043	<b>0.0041</b>



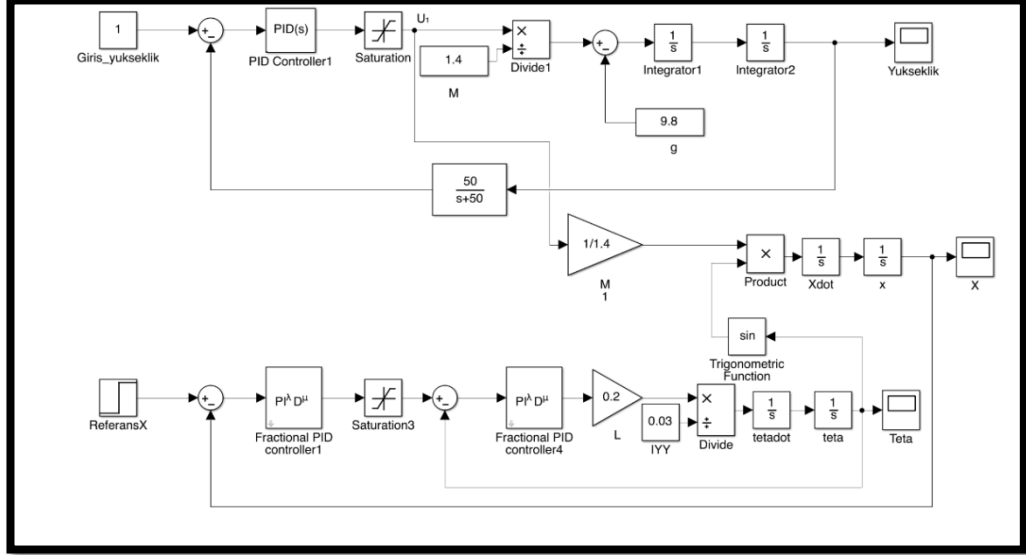
**Şekil 6.13.** Açık kontrolünde farklı denetleyici ve optimizasyon yöntemleri için amaç fonksiyon değerlerinin değişimi



**Şekil 6.14.** Açık kontrolünde farklı denetleyici ve optimizasyon yöntemleri için yunuslama ve yalpalama açılarının karşılaştırılması

En iyi sonucu veren PSO KD *PID* denetleyicisi ve bu denetleyiciye ait parametreler iç çevrimde kullanılarak,  $x$  ve  $y$  pozisyon kontrolleri ITSE kriterlerine göre farklı denetleyiciler ile dış çevrim için sağlanmıştır.

Denklem (6.3) ve (6.4)'ten yola çıkılarak  $x$  ve  $y$  eksenlerine ait Simulink blok diyagramı oluşturulmuştur (Şekil 6.15).



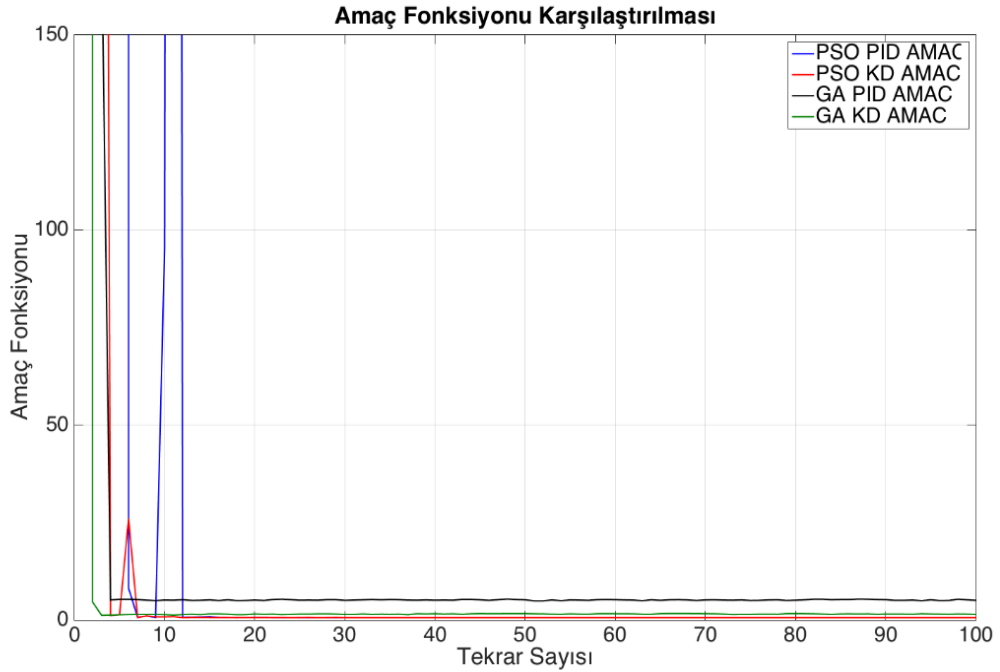
Şekil 6.15.  $x$  ve  $y$  pozisyon kontrol blok diyagramı

Çizelge 6.7’de  $x$ ,  $y$  pozisyonlarının kontrolü için GA ve PSO algoritmaları koşturularak ITSE kriterinin  $PID$  ve  $KD PID$  denetleyicilerine ait parametreleri ve performansları gösterilmiştir. Tabloya bakıldığı zaman en iyi amaç fonksiyonu değeri 0.6095 ile  $KD PID$  denetleyicisinde PSO algoritması koşturularak elde edilmiştir.

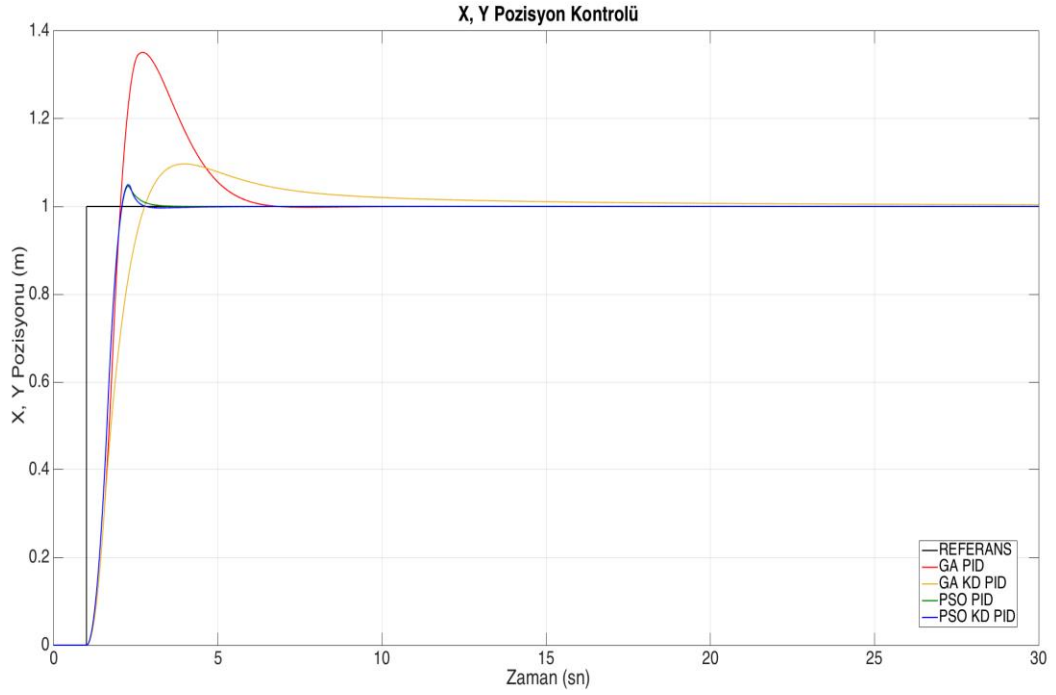
Şekil 6.16’da ITSE kriteri için amaç fonksiyonlarının karşılaştırılması verilmiştir. Şekil 6.17’de ise ITSE kriteri kullanılarak QBall-X4’ ün istenen referans  $x$  ve  $y$  pozisyon değerine yakınsamaları karşılaştırılmıştır.

**Çizelge 6.7.** ITSE kriteri için pozisyon kontrolünde farklı optimizasyon ve denetleyicilere ait performansların karşılaştırılması

Parametre değerleri	ITSE			
	GA PID	GA KD PID	PSO PID	PSO KD PID
$K_p$	998.5351	546.2370	962.8256	637.2695
$K_i$	717.6952	431.9732	0.0001	738
$K_d$	468.7176	933.5897	284.1095	774.5591
$\lambda$	-	0.1840	-	0.0116
$\mu$	-	0.5677	-	0.8942
Maksimum Aşma Miktarı	%35.1	%9.7	%4.6	<b>%3.6</b>
Kararlı hal hatası Titreşim	0.0001 Yok	0.005 Yok	0.0001 Yok	0 Yok
Yükselme zamanı	1.78sn	1.92sn	1.72sn	<b>1.71sn</b>
Oturma zamanı	6.4 sn	2.7sn	3.2sn	<b>2.4sn</b>
En İyi Amaç Fonksiyonu Değeri	4.899	1.238	0.62	<b>0.6095</b>



**Şekil 6.16.** Pozisyon kontrolünde farklı denetleyici ve optimizasyon yöntemleri için amaç fonksiyon değerlerinin değişimi



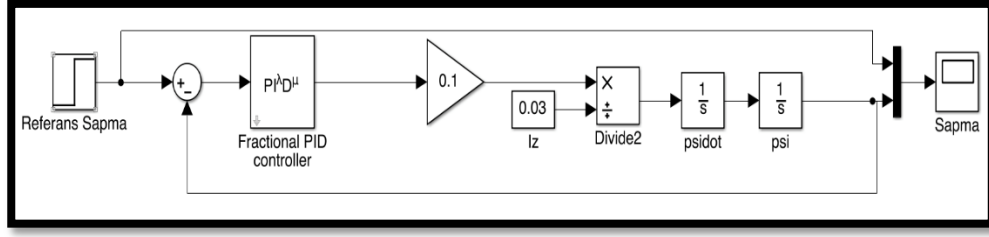
**Şekil 6.17.** Pozisyon kontrolünde farklı denetleyici ve optimizasyon yöntemleri için x, y pozisyonlarının karşılaştırılması

### 6.3. Sapma Açısı Kontrolü

Sapma açısını *PID* ve *KD PID* denetleyicisi ile kontrol etmek için (6.7.a) ve (6.7.b) ile verilen denklemler kullanılmıştır. Bu denklemlerden yola çıkılarak Simulink blok diyagramı oluşturulmuştur (Şekil 6.18). Sapma açısının kontrolünde, yükseklik kontrolünde diğer kriterlere oranla daha iyi performans sergileyen ITSE kriteri kullanılmıştır ve *GA PID*, *GA KD PID*, *PSO PID*, *PSO KD PID* denetleyicilerinin performansları karşılaştırılmıştır.

$$\ddot{\psi} = \frac{(T_1 + T_2 - T_3 - T_4)C}{I_z}, \quad (6.7.a)$$

$$T_1 + T_2 - T_3 - T_4 = U_4, \quad (6.7.b)$$



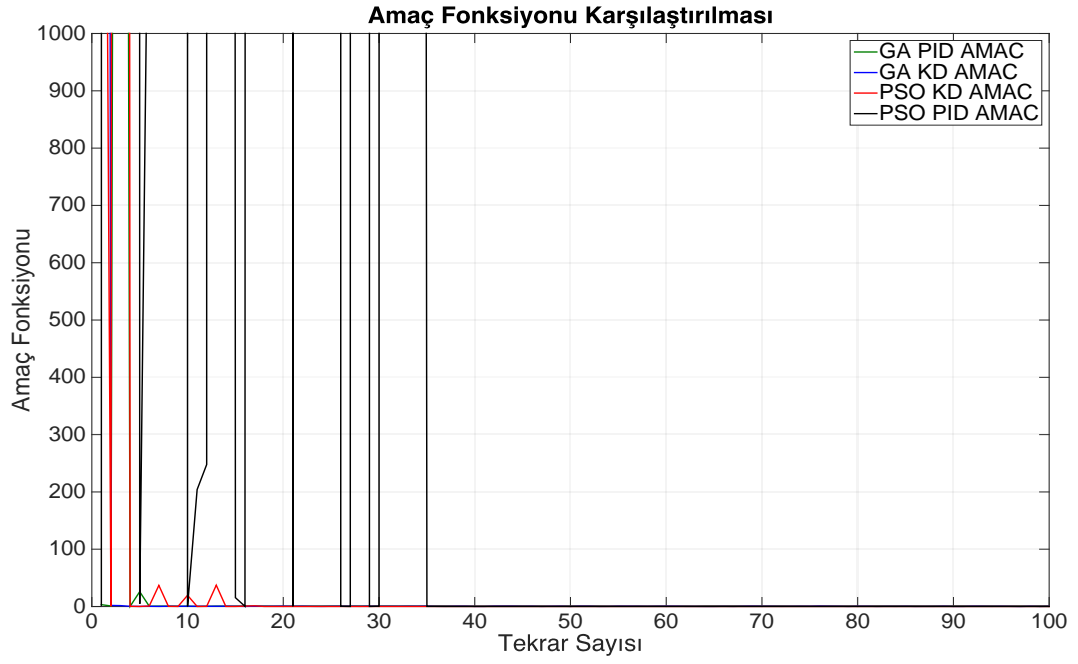
Şekil 6.18. Sapma açısı kontrol blok diyagramı

Çizelge 6.8’de sapma açısının kontrolü için GA ve PSO algoritmaları koşturularak ITSE kriterine göre elde edilen *PID* ve *KD PID* denetleyicilerine ait parametreleri ve performanslar gösterilmiştir. Tabloya bakıldığında zaman en iyi amaç fonksiyonu değeri 0.1295 ile *PID* denetleyicisinde PSO algoritması koşturularak elde edilmiştir.

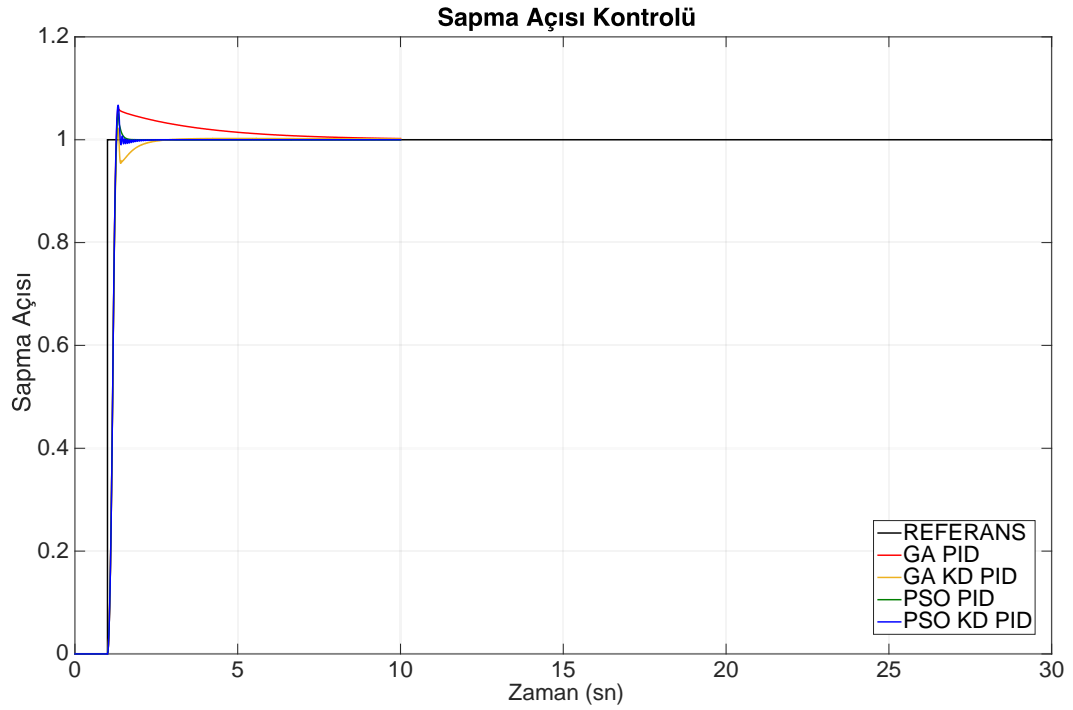
Şekil 6.19’da ITSE kriteri için amaç fonksiyonlarının karşılaştırılması verilmiştir. Şekil 6.20’de ise ITSE kriteri kullanılarak QBall-X4’ ün istenen referans sapma açısı değerine yakınsamaları karşılaştırılmıştır.

Çizelge 6.8. Sapma açısı kontrolünde ITSE kriteri için farklı optimizasyon ve denetleyicilere ait performansların karşılaştırılması

Parametre Değerleri	ITSE			
	GA <i>PID</i>	GA <i>KD PID</i>	PSO <i>PID</i>	PSO <i>KD PID</i>
$K_p$	748.397	943.0507	651.786	507.5866
$K_i$	268.326	830.9223	0.0001	197.7682
$K_d$	63.053	828.2366	50.8428	92.9837
$\lambda$	-	0.1906	-	0.0168
$\mu$	-	0.5913	-	0.9317
Maksimum Aşma Miktarı	%6.3	<b>%2.2</b>	%5	%6.7
Kararlı hal hatası	0.03	0.02	0	0
Titreşim	Yok	Yok	Yok	Yok
Yükselme zamanı	1.17sn	1.18sn	1.18sn	1.18sn
Oturma zamanı	7.8sn	2.56sn	1.55sn	<b>1.43sn</b>
En İyi Amaç Fonksiyonu Değeri	0.2148	0.1796	<b>0.1295</b>	0.1501



**Şekil 6.19.** Sapma açısı kontrolünde farklı denetleyici ve optimizasyon yöntemleri için amaç fonksiyon değerlerinin değişimi



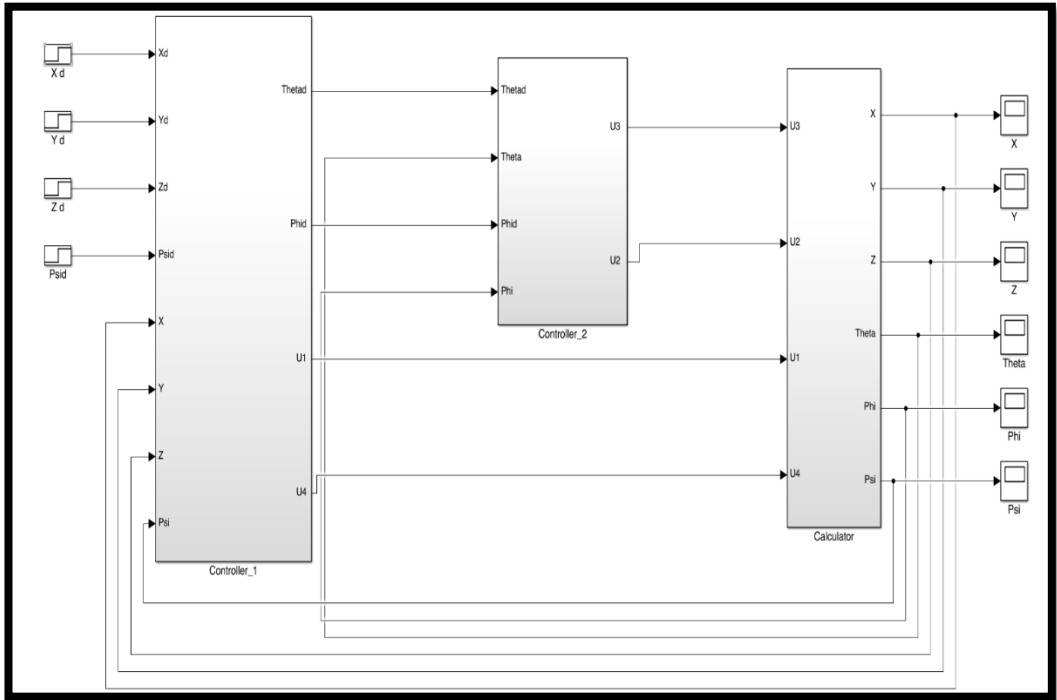
**Şekil 6.20.** Sapma açısı kontrolünde farklı denetleyici ve optimizasyon yöntemlerinin karşılaştırılması



#### 6.4. Quadrotorun $x, y, z$ Eksenlerinde Kontrolü

Altıncı bölümde bu kısma kadar quadrotorun  $x, y, z$  yörüngelerinde ve  $\theta, \Phi, \psi$  açılarında kontrolü ayrı ayrı GA ve PSO algoritmaları kullanılarak farklı denetleyiciler ve farklı kriterler kullanılarak sağlanmıştır. Bu kısımda ise şimdiye kadar bulunan en iyi parametreler kullanılarak quadrotorun tüm eksenlerde kontrolü farklı yörünge senaryoları için sağlanmıştır. Denetleyici olarak KD *PID* kullanılmıştır, kriter olarak ITSE, optimizasyon yöntemi ise parçacık sürü optimizasyonu seçilmiştir. Quadrotorun tüm eksenlerdeki kontrolü için oluşturulan Simulink blok diyagramı Şekil 6.21 ile verilmiştir.

Elde edilen değerler ve benzetim çıktıları Çizelge 6.9, Şekil 6.22 ve Şekil 6.23 ile sunulmaktadır.

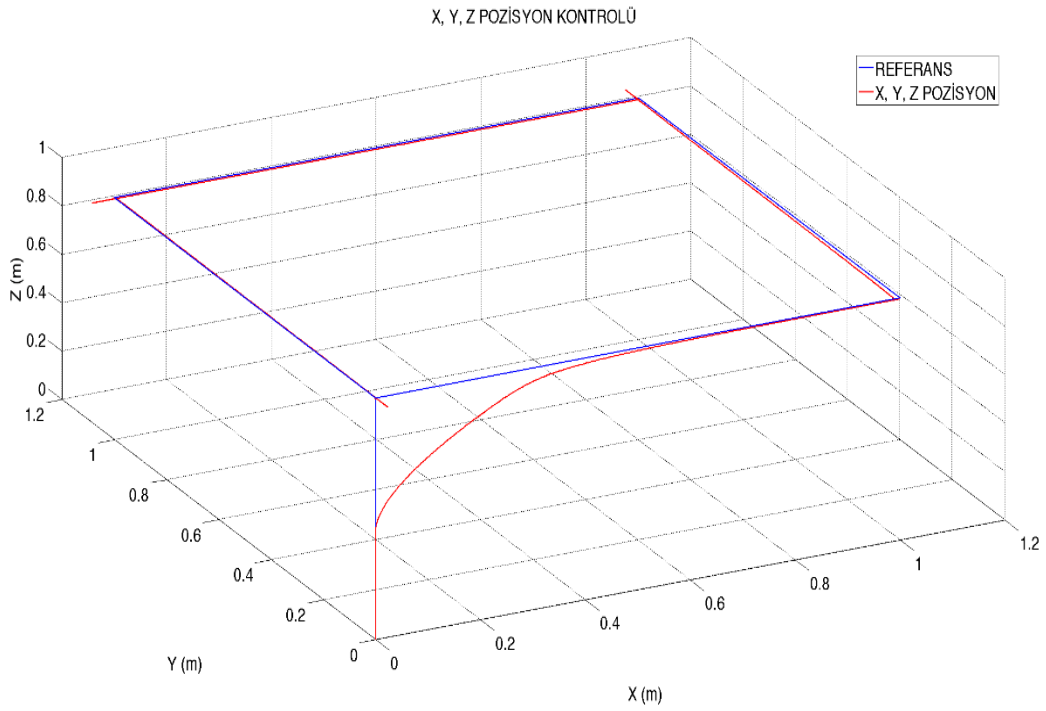


Şekil 6.21. QBall-X4 Simulink blok diyagramı

Çizelge 6.9. ITSE kriteri için PSO KD PID denetleyicisine ait parametreler

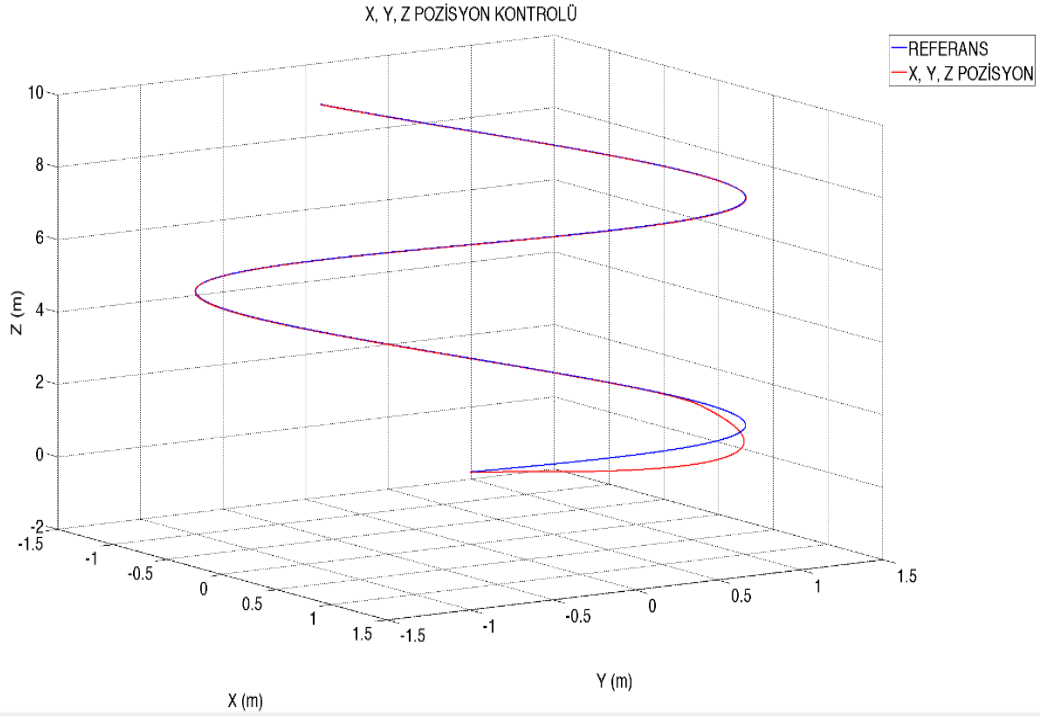
PSO KD PID, ITSE				
Parametre Değerleri	Yükseklik (z)	Yunuslama/Yalpalama	x, y	Sapma
$K_p$	748.397	499.2414	637.26	507.5866
$K_i$	223.8803	0.012	738.12	197.7682
$K_d$	406.4149	34.5952	774.5591	92.9837
$\lambda$	0.2153	0.1073	0.0116	0.0168
$\mu$	0.8815	0.9194	0.8942	0.9317

İlk senaryoda quadrotorun olduğu yerden 1 metre yükselip daha sonra x ve y eksenlerinde 1 metre uzunluğunda karesel yörünge çizmesi istenmiştir (Şekil 6.22).



Şekil 6.22. Quadrotorun x, y, z eksenlerinde istenen referans yörüngeyi izlemesi – Senaryo 1

İkinci senaryoda ise quadrotorun daha zor bir senaryo olan spiral seklinde yükselmesi istenmiştir (Şekil 6.23).



Şekil 6.23. Quadrotorun  $x$ ,  $y$ ,  $z$  eksenlerinde istenen referans yörüngeyi izlemesi – Senaryo 2

## 7. SONUÇLAR

QBall-X4 quadrotorunun istenen referans yörüngeyi takip etmesi için tasarlanan  $PI^{\lambda}D^{\mu}$  ve  $PID$  denetleyicilerinin farklı kriterler ve farklı sezgisel optimizasyon yöntemlerine göre kıyaslanması bu bölümde yapılmıştır.

Yükseklik kontrolünün değerlendirme ölçütlerine bakıldığında GA  $PID$  denetleyicisi için kullanıldığı zaman en iyi sonuç ISE kriteri kullanılarak elde edilmiştir. GA KD  $PID$  kullanıldığında ise en iyi sonuç ITSE kriteri ile elde edilmiştir. Kesir dereceli denetleyicinin klasik denetleyiciye göre hem kararlı hal hatası hem de maksimum aşım miktarı daha düşük çıkmıştır. PSO  $PID$  yönteminde ise klasik ve PSO KD  $PID$  denetleyicilerin her ikisinde de ITSE kriteri ile en iyi sonuca ulaşılmıştır ve kesir dereceli  $PID$  denetleyicisi klasik  $PID$  denetleyicisine oranla daha iyi performans göstermiştir.

Yunuslama ve yalpalama açılarının kontrolünde ITSE kriteri kullanılmıştır ve bu kriter için maksimum aşım miktarında PSO  $PID$  denetleyicisi yükselme zamanı, kararlı hal hatası, oturma zamanı ve en iyi amaç fonksiyonu göz önüne alındığında ise PSO KD  $PID$  denetleyicisinin daha iyi sonuç verdiği gözlenmiştir.

$x$  ve  $y$  eksenlerinde ise PSO KD  $PID$  denetleyicisi maksimum aşım miktarı, yükselme zamanı, kararlı hal hatası, oturma zamanı ve en iyi amaç fonksiyonu değerlerinde GA  $PID$ , GA KD  $PID$ , PSO  $PID$  denetleyicilerine göre daha iyi performans sergilemiştir.

Son olarak sapma açısının kontrolünde de maksimum aşım miktarında GA KD  $PID$  yükselme ve oturma zamanlarında PSO KD  $PID$  denetleyicisi daha iyi performans göstermiştir. Diğer performans kriterlerinin sonuçlarının ise birbirine yakın olduğu gözlenmiştir.

Bu çalışmada ISE, ITSE, IAE, ITAE kriterleri kıyaslandığı zaman ITSE kriterlerinin diğerlerine nazaran daha iyi sonuçlar verdiği, PSO yöntemi ile elde edilen katsayılar kullanıldığında, genelde GA yönteminde bulunan katsayılara göre daha optimum sonuçlar elde edildiği görülmüştür. Klasik  $PID$  ve KD  $PID$  denetleyicileri kullanılarak elde edilen sonuçlara bakıldığında zaman KD  $PID$

denetleyicisi ile kontrol edilen quadrotorun istenen referans yörüngeyi daha iyi izlediği sonucuna varılmıştır.

Tüm sonuçlar göz önüne alındığında sezgisel optimizasyon yöntemleri olan GA ve PSO algoritmalarının  $PI^{\lambda}D^{\mu}$  ve  $PID$  denetleyicilerinin parametrelerinin belirlenmesinde etkin olarak kullanılabilceği gözlenmiştir. Belirlenen optimizasyon algoritmasına ait katsayı ve oran değerleri için PSO algoritması ile elde edilen sonuçlar GA'ya göre daha iyi sonuç vermiştir. Denetleyici parametrelerinin belirlenmesinde kullanılan amaç fonksiyonu kriterleri kıyaslandığında ITSE kriterleri diğer kriterler olan ISE, IAE ve ITAE'ye göre daha iyi çözümler sunmuştur. Denetleyiciler kıyaslandığında ise  $PI^{\lambda}D^{\mu}$  denetleyicisi klasik  $PID$  denetleyicilerine göre daha iyi performans göstermiştir.

## KAYNAKLAR

- Agrawal, O.P. (2008), "A quadratic numerical scheme for fractional optimal control problems," *ASME J Dynamic Syst, Measurement, Control*, **130** (1).
- Altuğ, E., Ostrowski, J.P. ve Mahony, R. (2002), "Control of a quadrotor helicopter using visual feedback," *IEEE International Conference on Robotics & Automation, Washington, DC*, 72-77.
- Anonim (2008), "History of helicopters", <http://helis.com>, (erişim tarihi: 3 Mart 2008).
- Aoun, M., Malti, R., Levron, F. ve Oustaloup, A. (2004), "Numerical simulations of fractional systems: an overview of existing methods and improvements," *Nonlinear Dynamics*, 117-131.
- Azfar, A.Z. ve Hazry, D. (2011), "A simple approach on implementing IMU sensor fusion in PID controller for stabilizing quadrotor flight control," *IEEE International Colloquium on Signal Processing and Its Applications, Penang, Malaysia*.
- Bagley, R.L. (1989), "Power law and fractional calculus model of viscoelasticity," *AIAA Journal*, **27** (10), 1412–1417.
- Bedford, A. ve Fowler, W. (2005), *Engineering mechanics dynamics*, Prentice Hall, New Jersey.
- Bonabeau, E., Dorigo, M. ve Theraulaz, G. (2000), "Inspiration for optimization from social behavior," *Nature*, 406.
- Bouabdallah, S. ve Siegwart R. (1999), *Modeling of the (OS4) quadrotor*, Modeling Course, EPFL.
- Bouabdallah, S. ve Siegwart, R. (2007), "Full control of a quadrotor," *IEEE International Conference on Intelligent Robots and Systems*, 153-158.
- Cao, J.Y. ve Cao, B.G. (2006a), "Design of fractional order controller based on particle swarm optimization," *International Journal of Control, Automation, and Systems*, **4** (6), 775-781.
- Cao, J.Y. ve Cao, B.G. (2006b), "Design of fractional order controllers based on particle swarm optimization," *IEEE Conference on Industrial Electronics and Applications ICIEA*.

- Cao, J.Y., Liang, J. ve Cao, B.G. (2005), "Optimization of fractional order PID controllers based on genetic algorithms," *Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou.
- Chang, W.D. (2007), "Nonlinear system identification and control using a real-coded genetic algorithm", *Applied Mathematical Modelling*, **31**, 541-550.
- Changmao, Q., Naiming, Q. ve Zhiguo, S. (2010), "Fractional PID controller design of hypersonic flight vehicle," *International Conference on Computer, Mechatronics, Control and Electronic and Engineering (CMCE)*, 466-469.
- Das, S. (2008), *Functional fractional calculus for system identification and controls*, Springer, Springer Berlin Heidelberg New York.
- Domingues, J.M.B. (2009), *Quadrotor prototype*, Portugal, 101.
- Engheta, N. (1998), "The fractional curl operator in electromagnetic," *Microwave Opt.Tech.Letter*, **17**, 86–91.
- Fan, L. ve Joo, E.M. (2009), "Design for auto-tuning PID controller based on genetic algorithms," *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 1924-1928.
- Friis, J., Nielsen, E., Andersen, R.F., Bønding, J., Jochumsen, A. ve Sørensen, A.F. (2009), *Autonomous Landing on a Moving Platform*, Control Engineering Notes 8<sup>th</sup>. Semester, 9, Aalborg University.
- Gambardella, L.M. ve Dorigo, M. (1995), "Ant-Q: A reinforcement learning approach to the traveling salesman problem," *20<sup>th</sup> International Conference on Machine Learning (ML-95)*, 252-260.
- Goldberg D.E. (1989), *Genetic algorithms in search optimization and machine learning*, Addison-Wesley, Reading, MA, 1-7.
- Grefenstette, J.J. (1986), "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-16** (1), 122-128.
- Gupta, R., Gairola, S. ve Diwiedi, S. (2014), "Fractional order system identification and controller design using PSO," *International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity (CIPECH14)*, 149-

153.

- Hamamci, S.E. (2007), "An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers," *IEEE Transactions on Automatic Control*, **52** (10), 1964-1969.
- Han, J., Di, L., Coopmans, C. ve Chen, Y.Q. (2013), "Fractional order controller for pitch loop control of a VTOL UAV," *International Conference on Unmanned Aircraft Systems (ICUAS)*, Grand Hyatt Atlanta, Atlanta, GA, 609-614.
- Hartley, T., Lorenzo, C.F. ve Qammer, H.K. (1995), "Chaos in fractional Chua's circuit," *IEEE Trans. On Circuits and Systems-I: Fundamental Theory and Applications*, **42** (8), 485-490.
- Heaviside, O. (1971), *Electromagnetic theory*, Chelsea Edition New-York.
- Henriques B. S. M. (2011), *Estimation and control of a quadrotor attitude*.
- Holland, J.H. (1975), *Adaption in natural and artificial systems*, University of Michigan Pres, Ann Arbor, MI.
- Ichise, M., Nagayanagi, Y. ve Kojima, T. (1971), "An analog simulation of non-integer order transfer function analysis for electrode process," *J. Electroanal. Chem.*, **33** (1), 253-265.
- Kennedy, J. ve Eberhart, R. (1995a), "Particle Swarm Optimization," *IEEE*, 1942-1948.
- Kennedy, J. ve Eberhart, R. (1995b) "A new optimizer using particle swarm theory," *Sixth International Symposium on Micro Machine and Human Science*, 39-43.
- Keskintürk, T. ve Söyler, H. (2006), "Global karınca kolonisi optimizasyonu," *Gazi Üniv. Müh. Mim. Fak. Dergisi*, **21** (4), 689-698.
- Li, H., Luo, Y. ve Chen, Y.Q. (2010), "A fractional order proportional and derivative (FOPD) motion controller: tuning rule and experiments," *IEEE Transactions on Control Systems Technology*, **18** (2), 516-520.
- Lu, J.G. ve Chen, G. (2009), "Robust stability and stabilization of fractional-order interval systems: an LMI approach," *IEEE Transactions on Automatic Control*, **54** (6), 1294-1299.
- Lu, J.G. ve Chen, Y.Q. (2010), "Robust stability and stabilization of fractional-



- order interval systems with the fractional order  $\alpha$  : the  $0 < \alpha < 1$  Case,” *IEEE Transactions on Automatic Control*, **55** (1), 152-158.
- Luo, Y., Chao, H., Di, L. ve Chen, Y.Q. (2011), “Lateral directional fractional  $(PI)^\alpha$  control of a small fixed-wing unmanned aerial vehicles: controller designs and flight tests,” *IET Control Theory and Applications*, 2156-2167.
- Maione, G. ve Lino, P. (2006), “New tuning rules for fractional  $PI^\lambda$  controllers,” *Nonlinear Dynamics*.
- Maiti, D., Acharya, A., Chakraborty, M., ve Konar, A. (2008), “Tuning and  $PI^\lambda D^\delta$  controllers using the integer time absolute error criterion,” *4th International Conference on Information and Automation for Sustainability (ICIAFS)*, 457-462.
- Mandelbrot, B. (1967), “Some noises with 1/f spectrum, a bridge between direct current and white noise,” *IEEE Trans. Inform. Theory*, **IT-13** (2).
- Meng, L. ve Xue, D. (2009), “Design of an optimal fractional-order PID controller using multi-objective GA optimization,” *Chinese Control and Decision Conference (CCDC)*, 3849-3853.
- Michini B., Redding J., Ure, K., Cutler, M. ve How, J.P (2011), “Design and flight testing of an autonomous variable-pitch quadrotor,” *IEEE International Conference on Robotics and Automation*.
- Monje, C., Chen, Y.Q., Vinagre, B.M., Xue, D. ve Feliu, V. (2010), *Fractional order systems and controls: fundamentals and applications (advances in industrial control)*. London: Springer-Verlag.
- Na, G., Zhi-hong, Q. ve Hai-tao, W. (2012), “Neutral speed stability control law of aircraft design based on fractional order  $PI^\lambda D^\mu$ ,” *24th Chinese Control and Decision Conference (CCDC)*, 2802-2805.
- Oldham, K. ve Spainer, J. (1974), *Fractional calculus: theory and applications of differentiation and integration to arbitrary order*, New York: Academic Press.
- Ota, T. ve Omatu, S. (1996), “Tuning of the PID control gains by GA,” *IEEE Conference on Emerging Technologies and Factory Automation*, **1**, 272-274.

- Oustaloup, A. ve Bansard, M. (1993), "First generation CRONE control," *International Conference on Systems, Man and Cybernetics*, **2**, pp. 130–135.
- Oustaloup, A. ve Commande, L. (1991), *CRONE: Commande Robuste d'Ordre Non Entier*, Hermes, Paris.
- Oustaloup, A., Lanusse, P. ve Mathieu, B. (1993a), "Second generation CRONE control," *International Conference on Systems, Man and Cybernetics*, **2**, 136–142.
- Oustaloup, A., Lanusse, P. ve Mathieu, B. (1993b), "Third generation CRONE control," *International Conference on Systems, Man and Cybernetics*, **2**, 149–155.
- Özyetkin, M.M. (2013), *Kesirli dereceli kontrol sistemlerinin dayanıklılık analizi ve tasarımı*, Doktora Tezi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Malatya.
- Podlubny, I. (1994), *Fractional-order systems and fractional-order controllers*, Slovak Academy of Sciences Institute of Experimental Physics, UEF-03-94, 1-18.
- Podlubny, I. (1999), "Fractional-order systems and  $PI^\lambda D^\mu$ -controllers," *IEEE Transactions on Automatic Control*, **44** (1), 208-214.
- Podlubny, I., Dorcak, L. ve Kostial, I. (1997), "On fractional derivatives, fractional-order dynamic systems and  $PI^\lambda D^\mu$  controllers," *36th Conference on Decision & Control San Diego, California USA*, 4985-4990.
- Podlubny, I., Dorcak, L. ve Misanek, J. (1995), "Application of fractional order derivatives to calculation of heat load intensity change in blast furnace walls," *Transactions of Technical University of Kosice*, **5** (5), 137–144.
- Portmann M.C. (1996), "Genetic algorithms and scheduling: a state of art and some propositions," *Workshop on Production Planning and Control*, Mons, Belçika.
- Priya, C. ve Lakshmi, P. (2011), "Fractional order controller design and particle swarm optimization applied to a nonlinear system," *IEEE-International Conference on Recent Trends in Information Technology (ICRTIT)*, MIT,

- Anna University, Chennai, 959-964.
- Quanser Inc. (2010), *Qball-X4 user manual*, Canada.
- Raza, S.A. ve Gueaieb, W. (2010), *Motion control*, (Edt. Federico Casolo), chapter 12, INTECH.
- Shames, I. (1997), *Engineering mechanics dynamics*, Prentice Hall, New Jersey.
- Sharma, R., Rana, K.P.S. ve Kumar, V. (2014), "Statistical analysis of GA based PID Controller optimization for robotic manipulator," *International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 713-718.
- Sun, H.H., Abdelwahab, A.A. ve Onaral, B. (1984), "Linear approximation of transfer function with a pole of fractional power," *IEEE Trans on Automatic Control*, **AC-29** (5), 441-444.
- Tang, K.S., Man, K.F., Chen, G. ve Kwong, S. (2001), "A GA-optimized fuzzy PD+I controller for nonlinear systems," *27th Annual Conference of the IEEE Industrial Electronics Society*, 718-723.
- Taylor, W.R. (1977), *Jane's book of remotely piloted vehicles*, Collier Books.
- Tepljakov, A.A., Petlenkov, E. ve Belikov, J. (2011), "FOMCON: Fractional-order modeling and control toolbox for MATLAB," *18<sup>th</sup> International Conference, Mixed Design of Integrated Circuits and Systems*, Gliwice, Poland, 684-689.
- Tsao, Y.Y., Onaral, B. ve Sun, H.H. (1971), "An algorithm for determining global parameters of minimum phase systems with fractional power spectra," *IEEE Trans. Inst. And Meas*, **38** (3), 723-729.
- Valerio, D. (2005), "Ninteger v. 2.3 fractional control toolbox for MATLAB," <http://web.ist.utl.pt/~duarte.valerio>.
- Valerio, D. ve Costa, J.S. (2006), "Tuning of fractional PID controllers with Ziegler-Nichols type rules," *Signal Processing*, **86**, 2771-2784.
- Wang, L., Ling, M., Wang, D., Chen, Y.Q. ve Wang, C. (2008), "Line-of-sight stabilization system based on fractional-order control," *IEEE 2<sup>nd</sup> International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA'2008)*, Shenzhen, 10-12.
- Waslander, S.L., Hoffmann, G., Jangand, J. ve Tomlin, C. J. (2005), "Multi-

agent X4-Flyer testbed control design: integral sliding mode vs. reinforcement learning,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Wen, X.J., Wu, Z.M. ve Lu, J.G. (2008), “Stability analysis of a class of nonlinear fractional-order systems,” *IEEE Transactions on Circuits and Systems-Express Brieds*, **55** (11), 1178-1182.

Wu, C.J., Lee, T.L., Fu, Y.Y. ve Lai, L.C. (2007), “Auto-tuning fuzzy PID control of a pendubot system,” *4th IEEE International Conference on Mechatronics, (ICM2007)*.

Xue, D. ve Chen, Y.Q. (2002), “A comparative introduction of four fractional order controllers,” *4<sup>th</sup> World Congress on Intelligent Control and Automation*, Shanghai, P.R. China, 3228-3235.

Zhao, C., Xue, D. ve Chen, Y.Q. (2005), “A fractional order PID tuning algorithm for a class of fractional order plants,” *IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, 216-221.