



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



İKİ UZUVLU BİR ROBOT KOLUNUN
POZİSYON KONTROLÜ İÇİN OPTİMİZE
EDİLMİŞ PID KONTROLÇÜ TASARIMI

Said MÜFTÜ

YÜKSEK LİSANS TEZİ

Mekatronik Mühendisliği Anabilim Dalı

Şubat-2023
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

İKİ UZUVLU BİR ROBOT KOLUNUN POZİSYON KONTROLÜ İÇİN OPTİMİZE EDİLMİŞ PID KONTROLÇÜ TASARIMI Said MÜFTÜ

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Mekatronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Barış GÖKÇE

2023, 80 Sayfa

Jüri

Doç. Dr. Barış GÖKÇE

Dr. Öğr. Üyesi Ümit ÖNEN

Dr. Öğr. Üyesi Erdem DİLMEN

Günümüz teknoloji çağında robot kolları hayatımızın her alanında etkin rol almaktadır. Gelişen endüstri ile birlikte robot kollarının önemi her geçen gün artmaktadır. Robot kollarının çalışmasındaki en zor kısımlardan biri kontrolüdür. Robot kolunu istenilen şekilde hareket yapması istenirse kontrolü büyük önem göstermektedir.

Bu tez çalışması kapsamında, mikro denetleyici ile iki uzuvlu bir robot kolunun optimize edilmiş oransal integral türevsel denetleyici (PID) ile kontrol optimizasyonu gerçekleştirilmiştir. İki uzuvlu bir robot kolunun kontrolü için gerekli olan kinematik işlemler Arduino Matlab Destek Paketi (support Package for Arduino) kullanılarak geliştirilmiştir. Bu çalışmada doğru akım (DA) fırçalı bir motor için transfer fonksiyonu motor veri sayfasında verilen değerler kullanılmıştır. Kontrol sistemi için geri bildirim alan etkili enkoder sayesinde elde edilmiştir. İki uzuvlu robot kolunu oluşturan her bir eksendeki motor için ayrı ayrı konum kontrolü gerçekleştirilmiştir. Her bir motor için gerçekleştirilen konum kontrolünde parçacık sürü, yapay arı koloni ve kaos oyun optimizasyon algoritmaları ile kontrolcü parametreleri tespit edilmiştir ve robot kolu üzerinde belirlenen referans konum değeri ile elde edilen sonuçlar karşılaştırılmıştır. Bu tez çalışmasında konum kontrolünden sonra belirlenen referans şekil olarak belirlenen kare ve daire yörüngesi takibi iki uzuvlu robot kolu ile gerçekleştirilmiştir. Kare ve daire yörünge için tasarlanan PID kontrolcünün parametreleri yine parçacık sürü, yapay arı koloni ve kaos oyun optimizasyon algoritmaları ile elde edilmiştir. Her bir optimizasyon algoritması sonucu elde edilen PID parametreleri ile kare ve daire yörüngelerinin gerçekleştirilmesi gerçek zamanlı olarak yapılmış ve başarılı sonuç elde edilmiştir. Her bir yörünge için yapılan üç adet meta sezgisel optimizasyon algoritması sonucu yörünge takibi ve elde edilen kontrolcü parametreleri kontrolcü üzerinde gerçekleştirilerek optimizasyon algoritmaları karşılaştırılmıştır.

Anahtar Kelimeler: DC motor, Optimizasyon, PID Kontrol, Robot Kolu

ABSTRACT

MS THESIS OPTIMIZED PID CONTROLLER DESIGN FOR POSITION CONTROL OF A TWO-LIMB ROBOT ARM Said MÜFTÜ

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN MECHATRONICAL ENGINEERING**

Advisor: Assoc. Prof. Dr. Barış GÖKÇE

2023, 80 Pages

**Jury
Assoc. Prof. Dr. Barış GÖKÇE
Assist. Prof. Dr. Ümit ÖNEN
Assist. Prof. Dr. Erdem DİLMEN**

In today's technology age, robot arms play an active role in every aspect of our lives. With the developing industry, the importance of robot arms is increasing day by day. One of the most difficult parts in the operation of robot arms is its control. If the robot arm is desired to move as desired, its control is of great importance.

Within the scope of this thesis, it is thought that a two-limb robot arm can be controlled with an optimized proportional integral derivative controller (PID) with a microcontroller. The kinematic operations required for the control of a two-legged robot arm are logically programmed with the Matlab support Package for Arduino. In this study, the transfer function for a DC brushed motor was found by the values determined in the catalog section. The feedback for the control system is obtained thanks to the field-effect encoder. Separate position control was carried out for the motor on each axis forming the two-legged robot arm. In the position control performed for each motor, particle swarm, artificial bee colony, chaos game optimization algorithms and controller parameters were determined and the results obtained with the reference position value determined on the robot arm were compared. In this thesis, the reference square trajectory and the circular trajectory determined after the position control were performed with a two-legged robot arm. The parameters of the PID controller designed for square and circle orbits are again obtained with particle swarm, artificial bee colony, chaos game optimization algorithms. With the PID parameters found as a result of each optimization algorithm, the robot arm is directed in the direction of the determined square and circle and the results are obtained in real time. As a result of three metaheuristic optimization algorithms for each trajectory, trajectory tracking and the obtained controller parameters were performed on the controller and the optimization algorithms were compared.

Keywords: DC motor, Optimization, PID Control, Robot arm

ÖNSÖZ

Eđitim hayatım boyunca emeđi geen tđm hocalarıma ve bir kez olsun desteđini esirgemeyen danıřmanım Sayın Do. Dr. Barıř GÖKE hocama;

Hayatım boyunca beni destekleyen anne, babam ve kardeřlerim bařta olmak üzere alıřmam boyunca yardımlarını esirgemeyen saygıdeđer hocalarım Dr. Öđr. Üyesi Eřref BOĖAR, Do. Dr. Ersin DEMİR, Do. Dr. Berkay ERGENE'ye; kıymetli alıřma arkadařlarım Arř. Gör. Elif ÖZKUL, Arř. Gör. Emirhan YELEKİN, Arř. Gör. Dr. Fatma Betül KÖŐKER'e; Yüksek lisans eđitimim boyunca desteđini esirgemeyen abim Adem TÜRKMEN'e ve sevgili öđrencilerim Evren OTUR ve Erkut AKSU'ya ok teřekkür ederim.

Said MÜFTÜ
KONYA-2023

İÇİNDEKİLER

ÖZET.....	iv
ABSTRACT	v
ÖNSÖZ.....	vi
İÇİNDEKİLER	viii
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xiii
SİMGELER VE KISALTMALAR	xiv
1. GİRİŞ	1
1.1. Çalışmanın Amaç ve Motivasyonu	3
2. KAYNAK ARAŞTIRMASI	4
2.1. Fırçalı DC Motorlar.....	8
2.2. Alan Etkili Enkoder.....	9
2.3. Robot Kontrolü ve Optimizasyonu	10
3. MATERYAL VE YÖNTEM.....	14
3.1. İki eksenli Robotun Mekanik Tasarımı.....	14
3.3. DC Motor	16
3.3.1 Faulhaber DC Motor	16
3.3.2. DC Motor Transfer Fonksiyonu.....	17
3.4. Faulhaber Enkoder	19
3.5. Redüktör.....	19
3.6. Motor Sürücüsü.....	20
3.7. Proje Bağlantı Şeması	20
3.8. İki Uzunlu Robotun Kinematik Denklemlerinin Bulunması	21
3.9. İki Uzunlu Robot Kolunun Kontrolör Tasarımı.....	25
3.10. Parçacık Sürü Optimizasyon Algoritması.....	26
3.11. Yapay Arı Koloni Optimizasyon Algoritması	28
3.12. Kaos Oyun Optimizasyon Algoritması	30
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	34

5. SONUÇLAR VE ÖNERİLER	61
5.1 Sonuçlar	61
5.2 Öneriler	61
6. KAYNAKLAR	62
EKLER.....	66

ŞEKİLLER DİZİNİ

Şekil 1.1. PID blok şeması.....	3
Şekil 2.1. DC motor yapısı.....	9
Şekil 2.2. Alan etkili enkoder	10
Şekil 2.3. Robota eksen takımının yerleştirilmesinin görseli	11
Şekil 3.1. Tasarlanan iki uzuvlu robot kolu tasarımı	15
Şekil 3. 2. Üretilen iki uzuvlu robot kolu	15
Şekil 3. 3. Arduino Mega 2560 geliştirme kartı	16
Şekil 3. 4. Faulhaber enkoderli redüktörlü DC motor	17
Şekil 3. 5. DC motor eş değer devresi	18
Şekil 3. 6. L298 motor sürücüsü	20
Şekil 3. 7. Proje bağlantı şeması	21
Şekil 3. 8. İleri ve ters kinematik blok diyagramı.....	22
Şekil 3. 9. Uzuvlu robotun eksen takımlarının yerleştirilmesi	23
Şekil 3. 10. Çalışma blok şeması	26
Şekil 3. 11. PSO akış şeması.....	28
Şekil 3. 12. YAKA akış şeması	30
Şekil 4.1. DC motor için PWM-Volt değeri	34
Şekil 4.2. $\theta 1$ için Girilen referans değerine karşılık PSO, YAKA, CGO karşılaştırması	36
Şekil 4.3. $\theta 1$ için girilen referansın PSO, YAKA, CGO karşılaştırmasının detaylı görüntüsü.....	37
Şekil 4.4. $\theta 1$ için girilen referans değerinde PSO, YAKA, CGO yakınsama işlemi	37
Şekil 4.5. $\theta 2$ için girilen referans değerine karşılık PSO, YAKA, CGO karşılaştırması	38
Şekil 4.6. $\theta 2$ için girilen referansın PSO, YAKA, CGO karşılaştırmasının detaylı görüntüsü.....	39
Şekil 4.7. $\theta 2$ için girilen referans değerinde PSO, YAKA, CGO yakınsama işlemi	39
Şekil 4.8. $\theta 1$ için CGO ile elde edilen PID parametrelerinin gerçekleştirilmesi	40
Şekil 4.9. $\theta 2$ için CGO ile elde edilen PID parametrelerinin gerçekleştirilmesi	40

Şekil 4.10. $\theta 1$ için PSO ile elde edilen PID parametrelerinin gerçekleştirilmesi	41
Şekil 4.11. $\theta 2$ için PSO ile elde edilen PID parametrelerinin gerçekleştirilmesi	41
Şekil 4.12. $\theta 1$ için YAKA ile elde edilen PID parametrelerinin gerçekleştirilmesi	42
Şekil 4.13. $\theta 2$ için YAKA ile elde edilen PID parametrelerinin gerçekleştirilmesi	42
Şekil 4.14. İki uzuvlu robot kolunun çalışma alanı	43
Şekil 4.15. Robot koluna yaptırılan kare yörünge hareketi	44
Şekil 4.16. Kare hareket için zamansal $\theta 1$ ve $\theta 2$ değişimi	45
Şekil 4.17. Kare hareket için gerçekleştirilen CGO referans takibi.....	46
Şekil 4.18. Kare hareket için gerçekleştirilen PSO referans takibi.....	46
Şekil 4.19. Kare hareket için gerçekleştirilen PSO referans takibi.....	47
Şekil 4.20. Kare hareket için CGO, YAKA karşılaştırması	48
Şekil 4.21. Kare yörünge için CGO sayesinde elde edilen PID parametrelerinin $\theta 1$ açısındaki referans takibi	49
Şekil 4.22. Kare yörünge için CGO sayesinde elde edilen PID parametrelerinin $\theta 2$ açısındaki referans takibi	49
Şekil 4.23. Kare yörünge için PSO sayesinde elde edilen PID parametrelerinin $\theta 1$ açısındaki referans takibi	50
Şekil 4.24. Kare yörünge için PSO sayesinde elde edilen PID parametrelerinin $\theta 2$ açısındaki referans takibi	50
Şekil 4.25. Kare yörünge için YAKA sayesinde elde edilen PID parametrelerinin $\theta 1$ açısındaki referans takibi	51
Şekil 4.26. Kare yörünge için YAKA sayesinde elde edilen PID parametrelerinin $\theta 2$ açısındaki referans takibi	51
Şekil 4.27. Robot koluna yaptırılan daire yörünge hareketi	52
Şekil 4.28. Kare hareket için zamansal $\theta 1$ ve $\theta 2$ değişimi	53
Şekil 4.29. Daire hareketi için gerçekleştirilen CGO referans takibi	54
Şekil 4.30. Daire hareketi için gerçekleştirilen PSO referans takibi	54
Şekil 4.31. Daire hareketi için gerçekleştirilen YAKA referans takibi	55

Şekil 4.32. Daire hareketi için CGO sayesinde elde edilen PID parametrelerinin $\theta 1$ açısındaki referans takibi	56
Şekil 4.33. Daire hareketi için CGO sayesinde elde edilen PID parametrelerinin $\theta 2$ açısındaki referans takibi	57
Şekil 4.34. Daire hareketi için PSO sayesinde elde edilen PID parametrelerinin $\theta 1$ açısındaki referans takibi	58
Şekil 4.35. Daire hareketi için PSO sayesinde elde edilen PID parametrelerinin $\theta 2$ açısındaki referans takibi	58
Şekil 4.36. Daire hareketi için YAKA sayesinde elde edilen PID parametrelerinin $\theta 1$ açısındaki referans takibi	59
Şekil 4.37. Daire hareketi için YAKA sayesinde elde edilen PID parametrelerinin $\theta 2$ açısındaki referans takibi	59

ÇİZELGELER DİZİNİ

Çizelge 1.1. Kontrol yöntemlerinin sınıflandırılması	2
Çizelge 2.1. PID kontrolcünün parametrelerinin belirlenmesi için yapılan çalışmalar	4
Çizelge 3.1. Faulhaber 1524006sr DC motor özellikleri.....	17
Çizelge 3.2. E2-16 serili FAULHABER Encoder özellikleri.....	19
Çizelge 3.3. Değişken parametrelerinin tabloya girilmesi.....	22
Çizelge 3.4. İki uzuvlu robot kolunun D – H çizelgesi.....	24
Çizelge 4.1. θ_1 değeri için PSO, YAKA ve CGO algoritmalarından bulunan PID parametreleri.....	35
Çizelge 4.2. θ_2 değeri için PSO, YAKA ve CGO algoritmalarından bulunan PID parametreleri	38
Çizelge 4.3. Kare hareketi için bulunan PID parametreleri	48
Çizelge 4.4. Daire hareketi için bulunan PID parametreleri.....	55

SİMGELER VE KISALTMALAR

Simgeler

A	: Amper
cm	: Santimetre
KB	: Kilobayt
kg	: Kilogram
mA	: Miliamper
V	: Volt

Kısaltmalar

AA	: Arı Algoritması
CGO	: Kaos Oyun Optimizasyon
D-H Metodu	: Denavirt Hartenberg Yöntemi
DC	: Doğru Akım
GA	: Genetik Algoritma
HDL	: Donanım Tanımlama Dilleri
SADC	: Seri analogdan dijitale dönüştürücü
SMC	: Kayan Mod Kontrol
SVM	: Destek vektör makine
PGA	: Programlanabilir Kazanç Amplifikatörü
PID	: Oransal İntegral Türev Denetleyici
PSO	: Parçacık Sürü Optimizasyonu
PWM	: Sinyal Genişlik Modülasyonu
USB	: Evrensel Seri Veri yolu
YAKA	: Yapay Arı Koloni Optimizasyon Algoritması
YA	: Yarasa Algoritması
ZN	: Ziegler Nichols

1. GİRİŞ

Ekonomisi gelişmekte veya gelişmiş olan ülkelerin endüstrisinde seri üretimi geliştirmek amacıyla robot kolları aktif rol almaktadır. Robot kolları, firmalar için büyük bir gider olarak görülen çalışan giderlerini azaltmaktadır. Endüstri içerisinde tehlikeli alanlarda çalışan insanların yerini robot kolları almaktadır. Robot kolları, seri üretimi geliştirmek için tutma, itme, kaynak yapma, kesme işlemi başta olmak üzere birçok alanda kullanılmaktadır. Robot kolunun kontrolü için kullanılan kısımlar olarak hata değeri hassas ve hassas olmayan olarak ayrılabilir (Çengelci ve Çimen, 2005). Bu tez çalışması kapsamında endüstri içerisinde iki eksenli olarak kullanılan robot kolunun hassas kontrol edilebilmesi için mevcut çalışmalara katkı sağlaması planlanmaktadır.

Robot kolları, motorlar ve/veya diğer tahrik elemanları tarafından oluşturulan elektromekanik bir üründür. Bu elektromekanik sistem, sensörler tarafından yönlendirilebilir ve tasarlanan bir programlama mikro denetleyici aracılığıyla kontrol edilebilir. Robot kolları bağlantı düzenlemesine ve eklem hareketine göre sınıflandırmaktadır. Bu sınıflandırma kartezyen, silindirik, küresel, SCARA ve mafsallı şeklindedir (Myint, 2015). Robot kolu, malzeme taşıma, kaynak, montaj, dağıtım, termal püskürtme boyama ve delme gibi birçok uygulamada yaygın olarak kullanılmaktadır. Otomasyon, mekanik, elektronik ve yazılım tabanlı sistemlerin kontrolüyle ilgilendiğinden otomasyon ve robotik yakından ilişkilidir. Robot kolları endüstriyel otomasyonun verimliliğini arttırdığından endüstriyel otomasyonda geniş bir kapsama sahiptir. İnsan kolunun yaptığı hareketleri büyük bir doğrulukla yapan robot kolları, endüstri içerisinde alma, yerleştirme ve yetenekli uygulamalar gibi daha birçok kısımda aktif olarak kullanılmaktadır.

Fırçalı DC motorlar yaşamımızda birçok alanda karşımıza çıkmaktadır. Endüstri, araçlar, elektrik sistemleri kısaca kontrol sisteminin olduğu birçok alanda aktif olarak bulunur. Bir kontrol sistemi tasarımında matematiksel modelin çok önemli olduğu iyi bilinmektedir. DC motor için makine davranışını iyi bir doğrulukla temsil eden birçok model vardır. Bu modeller içerisinde bulunan parametrelerin katsayıları kontrol sisteminin referans değere yakınlığı için çok önemlidir. Fırçalı DC motorlar, doğru akım güç kaynağından çalıştırılmak üzere tasarlanmış içten komütasyonlu olan elektrik motorlarıdır (Maheriya ve Parikh, 2016).

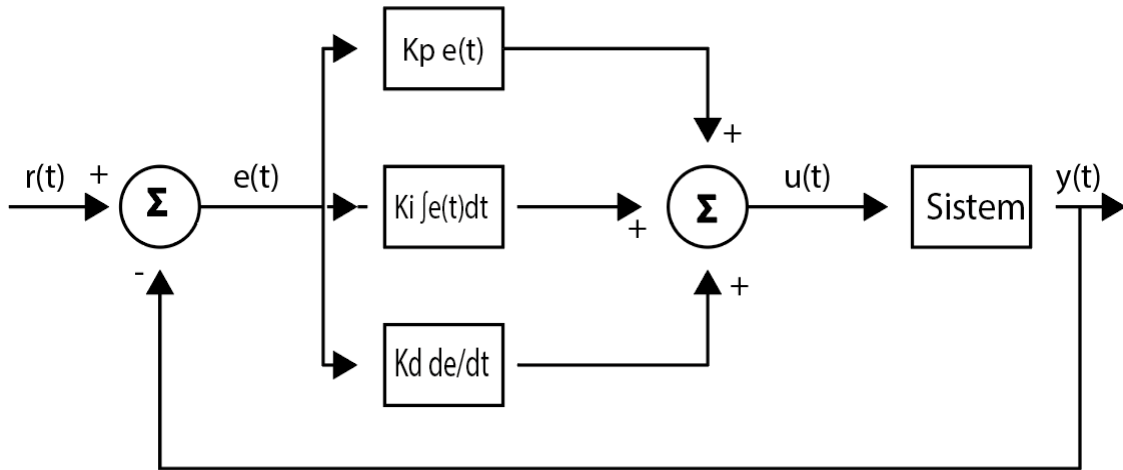
Son zamanlarda alışılmış tahrik sistemlerinin görevini, Fırçalı DC motorları almaya başlamıştır. DC motorlar, programlanabilme, hassas hareket kontrolü ve istenilen kontrolcüyü tasarlama gibi özelliklere sahiptir. Ayrıca pozisyonları kullanıcı tarafından kontrol edilebilen, redüktör sayesinde tork kapasiteli ve yüksek hassasiyetli motorlardır. Bu nedenle robotik projelerinde yaygın olarak kullanılmaktadır. Robot teknolojisinin en popüler olan alt kategorilerinden biri de robot kollarıdır.

Bir sistemi kontrol edebilmek için birçok yöntem kullanılmaktadır. Bu yöntemlerin sınıflandırılması ile ilgili bilgilendirme Çizelge 1.1’de belirtilmiştir. Mevcut çalışmalar içerisinde en çok tercih edilen yöntem olan PID Model tabanlı olup klasik kontrol yöntemi sınıfındadır.

Çizelge 1. 1. Kontrol yöntemlerinin sınıflandırılması

Model Tabanlı Kontrol	Model Tabanlı Olmayan Kontrol
Klasik Kontrol Yöntemi	Yapay Sinir Ağları Esaslı Kontrol Yöntemi
Modern Kontrol Yöntemi	Bulanık Mantık Esaslı Kontrol Yöntemi
Gürbüz Kontrol Yöntemi	

PID kontrolör kısaca oransal, integral ve türev modunun birleşimidir. PID’nin genel blok şeması Şekil 1.1’de belirtilmiştir. PID güçlü bir kontrolördür. PID kontrolör uzun bir geçmişe sahiptir. Analog çağdan dijital bilgisayar kontrol sistemi çağına kadar olan teknoloji değişikliklerinden kurtulmuştur. Endüstride var olan pazar için üretilen ilk kontrolördür. Endüstri için PID büyük bir öneme sahiptir. PID kontrolörü öne çıkaran üç ana sebep vardır. Bunlar geçmiş başarı kaydı, kullanım kolaylığı ve geniş kullanılabilirliktir. PID kontrolörün diğer bir avantajı ise parametre belirlenmesinde geniş yöntemler mevcuttur (Maheriya ve Parikh, 2016).



Şekil 1.1. PID blok şeması

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt} \quad (1.1)$$

Eşitlik 1.1’de PID kontrolörün değişkenleri belirtilmiştir. Bu ifadede K_p oransal kazanç, $e(t)$ hata değeri, K_i integral kazancı K_d türev kazancıdır.

PID kontrolörün parametrelerinin belirlenmesi aktif bir araştırma konusudur. PID kontrolcünün parametrelerinin belirlenmesi üzerine birçok çalışma yapılmıştır ve sınıflandırma gerçekleştirilmiştir. Johnson ve Moradi 2005 yılında PID ayarlama stratejilerini bir süreç modelinin ve model tipinin mevcudiyetine göre sınıflandırma gerçekleştirmiştir. Modelsiz parametrik olmayan model, derinlemesine veri, parametrik model yöntemleri olmak üzere dört ayrı kısma bölmüşlerdir (Johnson ve Moradi, 2005). Rodriguez ve arkadaşları ise beş ayrı başlık olarak sınıflandırmışlardır. Bu başlıklar analitik yöntemler, sezgisel yöntemler, frekans yanıtı yöntemleri, optimizasyon yöntemleri, uyarlanabilir ayar yöntemleri şeklindedir (Rodriguez-Molina ve ark., 2020).

1.1. Çalışmanın Amaç ve Motivasyonu

İki uzuvlu robot kolunu alan etkili enkoderli DC motor ile tasarlayıp mikro denetleyici ile kontrol etmek amaçlanmıştır. Tasarlanan PID kontrolcü sistem ile kontrolü gerçekleştirmek istenmiştir. Kontrolcü katsayılarını belirlemede daha önce kullanılmış olan Parçacık sürü, Yapay arı koloni optimizasyon algoritması ile literatürde daha önce kullanılmamış olan Kaos oyun meta sezgisel optimizasyon algoritmasını kullanıp sonuçları karşılaştırmak hedeflenmiştir.

2. KAYNAK ARAŞTIRMASI

Teknoloji, tarih boyunca sürekli bir gelişme içerisinde. Teknolojinin gelişmesi incelendiğinde bazı dönemlerde ani teknolojik gelişmeler yaşanmıştır. Bu ani gelişmeler ile paralel olarak insan hayatı da aniden etkilenmiştir. Ani teknolojik gelişmelere örnek olarak verilebilecek alanlardan birisi robot kollarındaki değişimdir. Robot kollarındaki yaşanan teknolojik gelişmeler insanlığı doğrudan etkilemektedir. Robot kollarındaki kontrol çalışmalarının gelişmesiyle birlikte endüstri içerisindeki kullanım oranı artmıştır. Her geçen gün daha fazla alanda yer bulan robot kolları seri üretimdeki verimliliği ile ülkelerin üretim kapasitesini arttırmaktadır.

Robot kelimesi 19. Yüzyılın başlarında kullanılmış gibi düşünülse de robotlara benzer ilk tasarımların MÖ 3000 yıllarına kadar uzanmaktadır. Günümüzde ise birçok kontrolcü ve kontrolcü içerisindeki değişkenleri belirlemede farklı çözüm yöntemleri ile oldukça gelişmiş durumdadır. PID kontrolcü 1890'lı yıllarda ilk kullanımı olmasına rağmen hala endüstriyel robot kollarında aktif bir şekilde rol almaktadır. PID kontrolcünün katsayılarını belirlemek için literatürde kullanılan birçok yöntem vardır. Çalışmalar içerisinde en sık rastlanan yöntem Ziegler Nichols metodudur. Bu metod John G. Ziegler ve Nathaniel B. Nichols tarafından 1942 yılında geliştirilmiştir (Bansal ve ark., 2017). PID kontrolcünün katsayılarının belirlenmesi ile ilgili literatürde yapılan bazı çalışmalar Çizelge 2.1'de belirtilmiştir.

Çizelge 2. 1. PID kontrolcünün parametrelerinin belirlenmesi için yapılan çalışmalar (Joseph ve ark., 2022)

Yıl	İsim	Çalışma ismi
2006	Kim ve Cho	GA, PSO, GA-BF, GA-PSO kullanarak AVR için PID kontrolcü
2008	Nagaraj ve Arkadaşları	Armatür kontrollü DC motoru kontrol için GA, PSO, EP, Kappa-Tav Sürekli döngü ZN yöntemleri kullanarak PID kontrolcü tasarımı
2012	Bansal ve Arkadaşları	PID kontrolcü parametrelerinin belirlenmesi için (Z-N) yöntemi, C-C yöntemi, AİS, ACO, DE, EP, ANN, SA, SVM, PSO, FL ve RSM tabanlı PİL) 14 yöntemin sınıflandırılması
2012	Glosal ve Arkadaşları	Sürü zekâsı hesaplaması için PID kontrolcü parametrelerinin ZN, BFO, PSO ve GA ile hesaplanması
2014	Kushwah ve Patra	

		DC motorun hız kontrolü için GA, PSO ve Bulanık mantık Algoritmalarıyla PID kontrolcü tasarımı
2014	Zulu ve John	Quadratör için yapay sinir ağları, bulanık mantık, güçlü, uyarlanabilir, Kayan Mod, lineer Quadratic düzenleyici kullanarak PID kontrolcü tasarımı
2014	Srinivas ve Arkadaşları	Matlab/Simulink kullanarak PID kontrolcünün optimum ayar parametrelerini Z-C yöntemi, C-C yöntemi, İSE, İAE, İATE ve GA olmak üzere altı yöntemde tahminlenmesi
2015	Emmanuel ve Inyama	PID kontrol parametrelerini, yapay sinir ağları (YSA), Genetik Algoritma (GA), Bulanık Mantık, Yazılım ayarı, Ziegler Nichols gibi yöntemlerle ayarlanması
2016	Fister ve arkadaşları	PID kontrol parametrelerinin parçacık sürü, guguk kuşu arama, yarasa, hibrit yarasa, genetik algoritma, duyarlı farklılaştırılmış evrim algoritmaları ile ayarlanması
2016	Dixit ve Pragati Joshi	PSO ve GA kullanarak otomatik voltaj regülatörü için PID kontrolör oluşturulması
2017	Gholap ve arkadaşları	GA, PSO ve SA kullanarak PID kontrolcü parametrelerini ayarlamak için zaman alanı analizi yineleme sayısı ve amaç fonksiyonunun değerinin hesaplanması
2017	Bhawana ve Neelu	PID kontrolcünün parametrelerini ayarlamak için sınıflandırma ve optimizasyon teknikleri
2018	Divya ve Nirmalkumar	PID kontrolör ayarı için sinir Ağları Ağları (SA), evrimsel programlama (EP), genetik Algoritma (GA), parçacık sürü optimizasyonu (PSO), karınca koloni optimizasyonu (ACO) ile yumuşak Hesaplama teknikleri
2019	Bharat ve arkadaşları	Bulanık mantık (BM), GA kullanarak PID kontrolör tasarlanması

Ruderman ve arkadaşları bir DC motorun sistem tanımlama ve LQR kontrol tasarımı üzerine yeni bir kontrol tasarımı yaklaşımı geliştirmiştir. LQR tasarımı, ikinci dereceden durum hatasını ve kontrol çabasını en aza indiren optimum durum geri besleme kontrolü sağlamıştır. Yaptığı deneysel sonuçlar ile yüksek hassasiyetli servo uygulamaları için kontrolör tasarımının uygulanabileceğini göstermiştir (Ruderman ve ark., 2008).

Sailan ve Kuhnert bir hidrolik pompanın valfine bağlı DC motorun açılma konumunun kontrolünü sağlamak için PID kontrolör uygulamıştır. Valf bir tam tur olmayan sınırlı bir aralıkta açılıp kapanmaktadır. Gerçek bir sistem üzerinde yapılan

deneyde sistem ve sensörlerin doğrusal olmamasından kaynaklanan salınımı bastırmak için PID kontrolörün iyi bir seçim olduğunu göstermiştir (Sailan ve Kuhnert, 2013).

Vazquez ve arkadaşları fırçalı bir DC motorda hız ve konum tahmini için, dalgalanma bileşenine dayalı destek vektör makinaları (SVM) kullanarak yeni bir yöntem sunmuşlardır. Alınan darbeler arasındaki ters mesafe kullanılarak motor hızı, tüm belirlenen darbeler sayılarak konum tahmin edilmiştir. Çalışmada fırçalı DC motor üzerinde gerçekleştirilen testler yöntemin, hızın sabit olduğu veya dinamik olarak değiştiği çok çeşitli hız ve durumlarda doğru çalıştığını göstermiştir (Vazquez-Sanchez ve ark., 2011).

Maung ve arkadaşları mikro denetleyici kartını 12V fırçalı Namiki DC motorunu kontrol etmek için kullanmışlardır. Yapılan çalışmada PWM sinyalinin yürütmek ve yön kontrolörünü sağlamak için L298N motor sürücü kartı kullanmışlardır. Adım girişi, sinüs girişi ve potansiyometre girişi test edilmiş ve istenilen konuma az derecelerdeki sapma ile ulaşılmıştır. PID denetleyici kullanılan bu çalışmada robotik kol konum kontrol sistemi ve diğer endüstriyel uygulamalar için uygulanabileceği belirtilmiştir (Maung ve ark., 2018).

Das ve Banerjee FPGA tabanlı denetleyici üzerine fırçalı DC motorlar için bir PID denetleyicinin prototiplenmesi üzerine çalışmışlardır. Giriş kısmı, referans girişini ve algılanan geri besleme sinyalini ölçeklendirmek için bir analog sinyal koşullayıcı, bir seri analogdan dijital dönüşürücü (SADC) ve bir programlanabilir kazanç amplifikatöründen (PGA) oluşturulmuştur. Çıkış kısmı ise DC servo motorun armatür devresini sürmek için bir H köprüsü güç amplifikatöründen oluşturulmuştur. Kullanılan platform, 8-bit PicoBlaze mikrodenetleyicili bir Spartan 3E FPGA'dır. Bu çalışma ile birlikte PID denetleyicilerinin FPGA tabanlı yumuşak sistemlere uygulanabileceği gösterilmiştir (Das ve Banerje, 2009).

Knezevic DC motor sürücüsünün sensörsüz konumlandırılması için basit bir yöntem belirtmiştir. Dalgalanma sinyallerini filtrelemek için DC motorun doğrusal olmayan bir matematiksel modeli analiz edilmiş ve basitleştirilmiştir. Yöntemini deneysel olarak çalışmasında sunmuştur. Düşük maliyetli, düşük çözünürlüklü yardımcı uygulamaları için doğru ve etkili bir çözüm olduğunu belirtmiştir (Knezevic, 2013).

Afjei ve arkadaşları DC motorun hız kontrolü için geri besleme mekanizmasında herhangi bir doğrudan şaft transdüseri kullanmadan kontrolü için yeni bir teknik sunmuşlardır. Bu yöntem fırçalar kollektör kanadından bir sonrakine geçerken motor ana giriş güç hatlarında üretilen akım/gerilim değişimini kullanmışlardır. Yapılan çalışma

fırçalı bir dc motor üzerinde test edilmiştir. Bu sistem 60/1 oranında dişli kutusu ile dc motorun konumunu kontrol edebilmek için kullanılmıştır (Afjei ve ark., 2022).

Berber ve arkadaşları PID kontrolü kullanarak otonom bir mobil robotun kontrolünü sağlamışlardır. Çalışmada mobil robotun kinematik modeli çıkartılmış ve PID parametreleri PSO ve Ziegler Nichols (ZN) ile belirlenip karşılaştırılmıştır. Bu çalışma ile PSO'nun ZN'a göre konum kontrolünde daha başarılı sonuç verdiği belirtilmiştir (Berber ve ark., 2016).

Ulusoy ve Güneş mobil robot kolunun kontrolü için robot kolunun konumunu belirlemede dikey ve yatay olarak jiroskop ve ivme ölçer kullanmışlardır. Geri besleme olarak alışılmış enkoder veya potansiyometre olan sistemlerden farklı olarak yapılan bu çalışmada PID parametrelerini belirlemede ZN yöntemi ve PSO algoritması kullanılmıştır. Bu çalışmada geri besleme sistemi farklı da olsa PSO algoritmasının ZN metoduna göre üstünlüğü simülasyon çıktıları ile belirtilmiştir (Ulusoy ve Güneş,2019).

Şirin Akkaya FPGA tabanlı robot kontrolü çalışmasında Deo NANO FPGA kartı ile beş eksenli bir robot kolunun pozisyon kontrolünü gerçekleştirme üzerine çalışmıştır. Robotun eklemlerini kapalı çevrim mekanizma olarak düşünülmüş ve PID kontrolcü tasarlanmıştır. Çalışmasında simülasyon çıktılarına yer vermiştir. Çalışmanın sonucunda FPGA ile birden fazla serbestlik dereceli bir robotun, kapalı sistemin eş zamanlı kontrolünü gerçekleştirmiştir (Akkaya, 2013).

Yiğit ve Çelik dört serbestlik dereceli bir robot kolunun tutucu ucun pozisyon kontrolünü gerçekleştirmiştir. 2-DOF PID kontrolör ile gerçekleştirilen bu çalışmada kinematik hesaplamalar yapılmış DC motor kullanılmıştır. Robot kolu Matlab/Sim-Mekanik ortamında transfer fonksiyonu bulunarak modellenmiştir. PID ve 2-DOF PID kontrolörünün katsayıları PSO ile optimize edilmiştir. Bu çalışmanın sonucunda PID kontrolörün 2-DOF PID kontrolörüne göre daha iyi sonuç verdiği belirtilmiştir (Yiğit ve Çelik, 2018).

Beyhan eksenli robot kolunun pozisyon kontrolü için iç döngüde yer çekimi dengeli PID kontrolü, dış döngüde ise empedans kontrol kullanmıştır. Matlab programında sistem tanımlama (system identification) eklentisiyle yer çekimi dengelemeli PID kontrolün katsayılarını hesaplamıştır. Empedans kontrolünde kütle sönüm yay modelini kullanmıştır. Bu çalışmanın sonucunda PID'ye kıyasla empedans kontrolünde yaklaşık yüzde 22'lik bir oranda azalmış şekilde denge noktasına oturduğu belirtilmiştir (Beyhan, 2021).

Meshrom ve arkadaşları FPGA kullanarak Robot kolunun pozisyonu üzerine çalışmışlardır. Spartan-2 yongasında VHDL diliyle çalışmışlardır. Çalışmada robot kolunun tahriki için step motor kullanmışlardır. FPGA çok eksenli bir çalışmada tüm motorları aynı anda işlem yapabilmesiyle büyük fayda sağlamaktadır. Bu çalışma ile literatüre FPGA ve Step motor ile fayda sağlamıştır (Meshrom ve ark., 2009)

Mayetin iki serbestlik dereceli robotun yörünge kontrolü üzerine yaptığı çalışmasında klasik, Bulanık mantık, Kayma kipli kontrol yöntemlerini kullanmış ve klasik yöntemde PID parametrelerini yalnızca PSO algoritmalarıyla gerçekleştirmiştir. İki serbestlik dereceli robot üzerinde sinüzoidal yörünge hareketini, uç konumun dairesel yörünge hareketini deneysel sonuçlarla simülasyon olarak gerçekleştirmiştir. Bu çalışmada robotik sistemlerdeki kullanılabilir kontrol yöntemleri üzerine karşılaştırma üzerine yoğunlaşmıştır (Mayetin,2013).

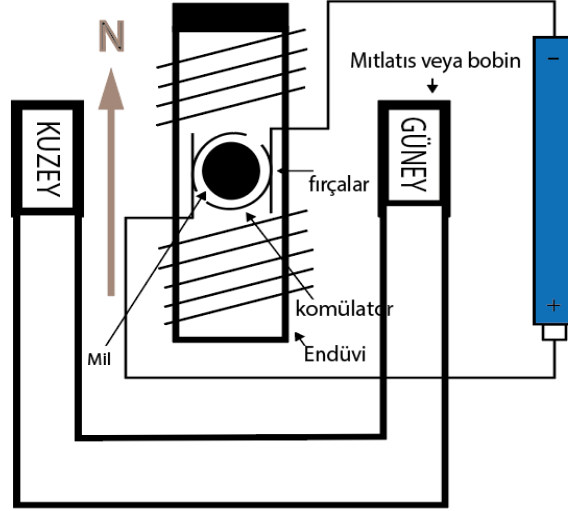
Kaplan ve arkadaşları Fırçasız DC motorun Konum kontrolünü PID ve bulanık mantık ile gerçek zamanlı olarak gerçekleştirmiştir. STM32F4 discovery geliştirme kartı ile yapılan çalışmada LMD18200 motor sürücü kullanılmıştır. Bu çalışma ile literatüre fırçasız DC motor ile yapılan kontrol çalışması eklenmiştir (Kaplan ve ark., 2020).

Soriano ve arkadaşları iki serbestlik derecesine sahip bir SCARA robotun kontrolü için kayan mod kontrolör (SMC) geliştirmiştir. Geliştirilen bu kontrolcü konum kontrolü sağlamak için optimize edilmiştir. Çalışmada enerji tasarrufu üzerine odaklanılmıştır. Optimizasyon algoritması olarak yarası algoritması (YA) kullanılan bu çalışmada benzetim çalışması üzerinde bir kontrolcü gerçekleştirilmiştir (Soriano ve ark., 2021).

İlgen ve arkadaşları iki serbestlik dereceli SCARA tip robot kolunun kontrolünü kayan mod kontrolör (SMC) yöntemi ile incelemiştir. Robotun modellenmesi MSC Adams'ta gerçekleştirilmiştir. Kontrolcünün parametreleri arı algoritması (AA) ile optimize edilmiştir. Çalışmada kontrolcünün parametreleri genetik algoritma ile de optimize edilmiş ve yapılan çalışma ile literatüre arı algoritmaları da kontrolcü parametrelerini belirlemede kullanılabilir olduğu sonucuna varılmıştır (İlgen ve ark., 2022).

2.1. Fırçalı DC Motorlar

Fırçalı DC motorlar birçok farklı uygulamada karşımıza çıkmaktadır. Bu motorlar ucuzdur. Sürmesi kolaydır ve her boyutta ve şekilde kolayca bulunur. Basit bir fırçalı DC motorun yapısı Şekil 2.1'de gösterilmiştir. Bu motorlar stator, rotor, fırçalar ve bir komütatör bileşenlerinden oluşmaktadır.



Şekil 2.1. DC motor yapısı (Condit,2004)

Stator, rotoru çevreleyen durağan bir manyetik alan oluşturur. Bu alan elektromanyetik sargılar veya kalıcı mıknatıslar tarafından üretilir fırçalı DC motorların türleri, statorun yapısı veya elektromanyetik sargıların güç kaynağına bağlanma şekline ile ayırt edilir. Armatür olarak da isimlendirilen rotor bir veya daha fazla sargıdan oluşur. Bu sargılar enerjilendiğinde manyetik bir alan oluştururlar. Bu durum rotor alanının manyetik kutupları stator tarafından üretilen zıt kutuplara çekilerek rotorun dönmesine sebep olur. Motor döndükçe sargılar sürekli olarak rotor tarafından üretilen manyetik kutupların statorda üretilen kutupları aşmaması için farklı bir sırayla enerji verilir. Alandaki bu geçiş rotor sargılarına komütasyon denir (Condit, 2004).

Fırçalı DC motorlar, Fırçasız DC, AC, endüksiyon motor türlerinden farklı olarak motor türlerindeki akımı değiştirmek için bir kontrolöre ihtiyaç duymaz. Sargılar üzerinde yapılan işlem ile yapılır. Fırçalı Doğru Akım Motorunun (BDC) aksında komütatör ismi verilen bakır levha bulunur. Motor dönme işlemi yaparken fırçalar komütatör üzerinden kayar. Segmentler farklı rotor sargılarına bağlandığından motorun fırçalarına bir voltaj uygulandığında motorun içinde dinamik bir manyetik alan oluşur. Bu sebepten fırçalı dc motorda aşınmadan dolayı fırçaların ömrü uzun değildir.

2.2. Alan Etkili Enkoder

Motor miline paralel olarak yerleştirilen alan etkili enkoder yarı iletken bir kristalden oluşur. Milin dönmesinin normal zaman türevi yerine doğrudan ölçülebilmesini sağlar. Bu sayede okuma minimum açısal hızda gerçekleştirebilmesi

sağlanır. Bu enkoder çeşidinin içerisinde İndiyum Arsenit (InAs) ve İndiyum Antimoid (InSb) kristalleri vardır. Bu kristalin genişliği ile kodlayıcı için açılal çözünürlük enkoder türüne göre değıştirilebilir (Strandt, 1963). Alan etkili enkoderin görseli Şekil 2.2’de belirtilmiştir.



Şekil 2.2. Alan etkili enkoder (Palacin ve Martinez, 2021)

Açı enkoderleri temelinde mildeki konumu dijital değere dönüştürme cihazlarıdır. Bu enkoderler Çift yönlü bir sayıcıyla birlikte açılal konum ve hızın dijitale dönüşümü sağlar. Geçmişinde fırça kodlayıcılar ve optik diskler olarak isimlendirilen bu enkoderler küçük boyutlu olması, uzun ömürlü olması projeler için büyük avantaj sağlamaktadır (Strandt, 1963).

2.3. Robot Kontrolü ve Optimizasyonu

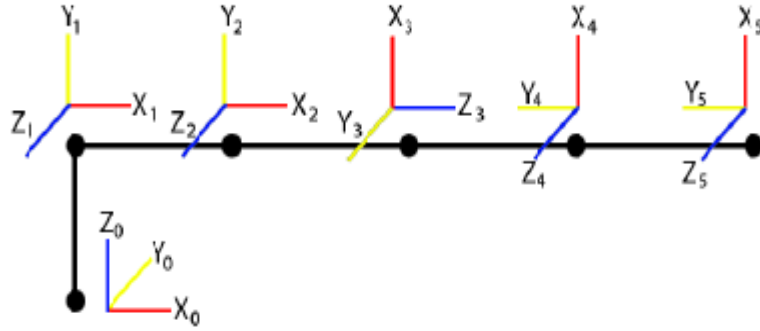
Robotik sistemler, son yılların dikkat çekici çalışma alanlarından biridir. İnsanın yaptığı işler olarak karar verme ve fiziksel aktivite gibi uygulamalarda ve aktif olarak seri üretimde kullanılmaktadır. Robot kolları her geçen gün hayatımızın daha fazla alanında karşımıza çıkmaktadır. Robot kolunu anlamak için elektrik-elektronik, makine, bilgisayar mühendisliği, matematik gibi geniş bir bilgi ağı gereklidir (Doğan, 2010).

Robotik sistemler için çalışma uzayı robotun en uç noktasının gidebileceği noktalar olarak ifade edilir. Robotik sistemlerin kontrolü için öncelikle robot kinematığını elde etmek gerekir. Robot kollarının kinematik çözümü, ileri (bazı kaynaklarda düz) ve ters kinematik olarak iki ayrı kısımda incelenir. İleri kinematik analiz, robotun uç noktasının konumunun ve oryantasyonunun bulunması olarak tanımlanmasıdır. Bu durumun tersi ise yani elde mevcut olan uç noktadan eklem konfigürasyonu elde etmek ise ters kinematiktir. Ters kinematik analizde doğrusal olmayan denklemlerle işlem yapıldığı için ileri kinematik analize göre daha zordur. Doğrusal olmayan denklemlerin

benzersiz çözümleri yoktur ve birleştirilemez. Diğer bir söyleyişle zor çözümlenen matematiksel denklemler içerir. Ters kinematik analizde birbirinden farklı yaklaşımlar vardır. Bunlar İteratif yaklaşım, optimal yaklaşım, analitik yaklaşım, bulanık mantık yaklaşımı, yapay sinir ağları yaklaşımıdır (Gül, 2021).

Robot kollarının hareketlerinin analizi için o robotun kuvvet, enerji ve atalet gibi dinamik büyüklüklerinin öğrenilmesi gerekir. Robotik projelerindeki ileri kinematik, robot eklemlerinin hızları, ivmeleri, konumları arasındaki ilişkiyle belirlenir. Robotikte dönel eklem ve prizmatik eklem olmak üzere iki çeşit eklem vardır. Dönel eklem R harfi ile prizmatik eklem ise P harfi ile genellikle gösterilir. Örneğin; üç eksenli bir robot kolunun ortasında prizmatik eklem var ise bu RPR şeklinde gösterilir.

Robotun uç noktasının konumunu bulmak için ileri kinematik analiz yapılır. İleri kinematik analiz için genellikle analitik yöntem ya da 1955 yılında Denavirt ve Hartenberg'in ortaya çıkardığı Denavirt-Hartenberg yöntemi kullanılmaktadır. Bu yöntem kısaca D-H metodu da denilmektedir. Bu metot için öncelikle x, y, z olmak üzere eksen takımları yerleştirilir (Taşer ve ark., 2019). Örnek bir eksen takımı Şekil 2.3'de verilmiştir.



Şekil 2.3. Robota eksen takımının yerleştirilmesinin görseli (Bingül ve Küçük, 2005).

Eksen takımlarının yerleştirilmesinde bazı kurallar vardır. Dönel olan eklemler için dönme yönüne prizmatik olan eklemler için prizmatik olan eklemler için öteleme yönü Z olarak belirlenir. Z eksenine dik ve kol boyunca olan eksen X eksenini olarak belirlenir. İki eksen bilindiği için üçüncü eksen sağ el kuralı ile rahatça belirlenebilir. Eksenler yerleştirildikten sonra yapılacak işlem ise D-H metodu için olan parametrelerin girilmesidir. Bu parametreler α_i (Bağ açısı), d_i (Bağ kaçıklığı), θ_i (Eklem açısı) a_i (bağ uzunluğu) dir. D-H tablosu bu parametreler ile oluşturulur.

D-H tablosu bulunduktan sonraki adım ise dönüşüm matrisini bulmaktır. Bu matris her bir eklem için ayrı ayrı hesaplanır. Dönüşüm matrisi Eşitlik 2.1’de verilmiştir.

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$T = A_1 \times A_2 \times A_3 \times A_4 \quad (2.2)$$

Düz kinematik için olan son adım ise bu matrislerin sırasıyla çarpma işlemi yapılmasıdır. Bu işlem Eşitlik 2.2’deki gibi yapılır. Matriste çarpma işleminde bazı kurallar vardır. Bu kurallara uygun işlem yapılarak transformasyon matrisi bulunabilir. Elde edilen transformasyon matrisinden her bir değişkeni ters kinematik yardımı ile belirlenebilir. (Bingül ve Küçük, 2005).

Robot kollarını oluşturan değişkenler tasarlanan kontrolcü ile referansa yaklaşılabılır. Robot kollarının kontrolü için kullanılan birçok kontrolcü ve bu kontrolcülerin katsayılarının belirlenmesi problemi için kullanılan birçok yöntem vardır. Bu tez çalışması kapsamında PID kontrolcü kullanılmıştır. PID kontrolcüyü oluşturan parametrelerin bulunması problemi için kullanılan yöntemler EK-4 kısmında belirtilmiştir. Bu tez çalışması kapsamında ise meta sezgisel optimizasyon algoritmalarından parçacık sürü, yapay arı koloni ve kaos oyun optimizasyon algoritmasıdır.

Optimizasyon en genel haliyle belirlenen amaç fonksiyonunun sınırlı veya sınırsız kısıtlar ile en iyi değerini bulma süreci olarak tanımlanabilir. Başta mühendislik sistemleri olmak üzere ekonomi, sağlık işletme gibi daha birçok alanda karşımıza çıkmaktadır. Optimizasyon problemlerini sürekli ya da ayrık dinamik veya statik kısıtlı ya da kısıtsız gibi çeşitli kategorilere ayırmaktadırlar. Optimizasyon algoritmaları, yapay zekanın alt alanlarındandır. Yapay zekanın artan popülaritesiyle birlikte optimizasyon algoritmaları her geçen gün farklı problemlerde çözüm olarak kullanılmaktadır (Tang ve ark., 2021).

Optimizasyon problemlerinin çözümü, değişkenlerin birbirine bağımlılığına, doğrusal olmayan sınır değişkenlerinden dolayı gibi sebeplerden çözümü elde etmek zor olarak görülür. Optimizasyon yöntemlerine bakıldığında problemi çözebilmek için matematiksel ve meta sezgisel yöntemler olarak iki farklı yöntem önermişlerdir. Matematiksel yaklaşımda büyük ölçekli sınırlı problemleri çözmeye her zaman verimli

değillerdir. Meta sezgisel yaklaşım ise kabul edilebilir bir zamanda sonuç türetebilen olarak geliştirilmektedirler. Bu özelliği ile her geçen gün farklı alanlarda daha geniş kullanılmaktadırlar (Boğar, 2021).

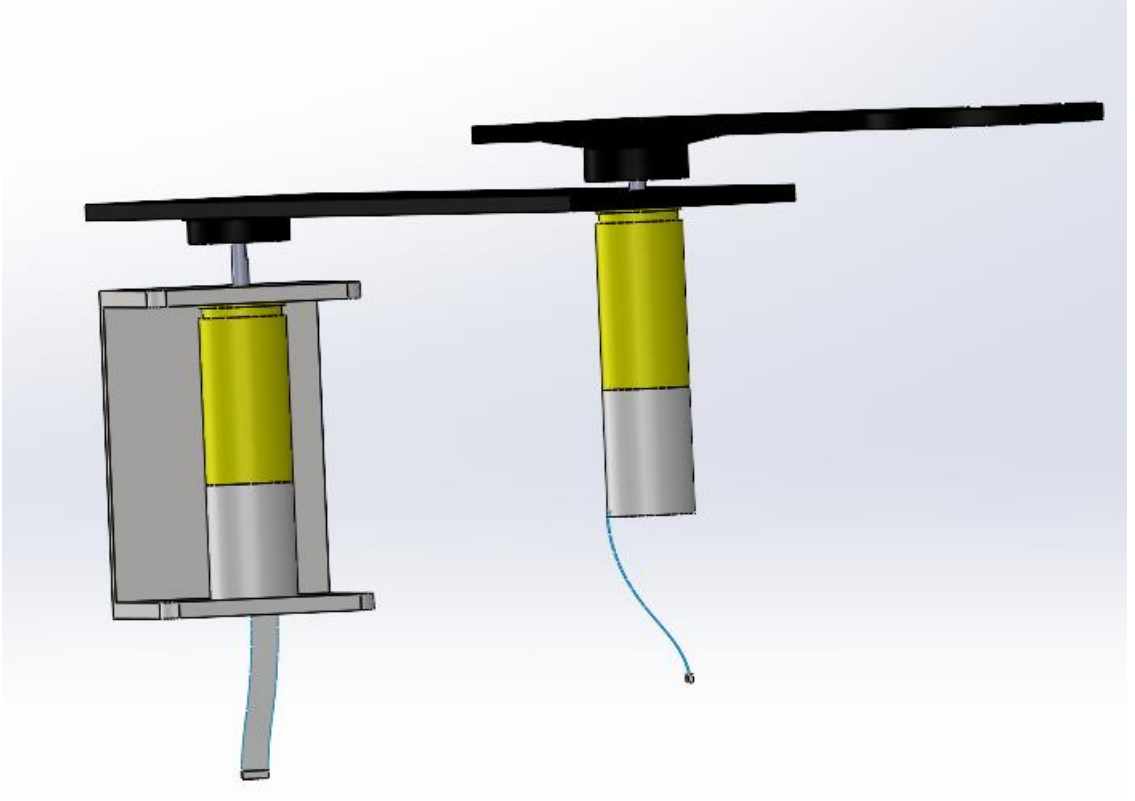
Meta sezgisel algoritmalar genellikle ilham kaynaklarına göre sınıflandırılabilir. Diğer bir sınıflandırma ise başlangıç çözüm türü yani çoklu ve tek çözüm tabanlı olarak sınıflandırılır. Çoklu çözüm tabanlılara genellikle popülasyon tabanlı, tekli tabanlıya ise yörünge tabanlı olarak isimlendirilir (Boğar, 2021). İlhan kaynağına göre meta sezgisel optimizasyon algoritmalarını sınıflandırma ise insan, evrimsel, sürü mantığı, fen bilimleri tabanlı olarak genellikle sınıflandırılır. Bu çalışma içerisinde PID kontrolcünün parametrelerinin belirlenmesi için kullanılan sürü mantığına dayanan parçacık sürü, yapay arı koloni ve kaos oyun optimizasyon algoritması yöntemleri ile ilgili detaylı bilgi materyal ve yöntem başlığı altında verilmiştir.

3. MATERYAL VE YÖNTEM

Bu kısımda çalışma içerisinde kullanılan materyal ve yöntemler hakkında bilgiler verilmiştir. Bu çalışma içerisinde iki uzuvlu bir robotun gerçek zamanlı kontrolü için mekanik tasarımı ve pozisyonu için kontrolcü tasarımı gerçekleştirilmiştir. Mikrodenetleyici kartında gerçekleştirilen bu kontrol sistemi tasarımı Matlab Simulink ortamında tasarlanmıştır. Tahrik elemanı olarak fırçalı DC motor kullanılan bu çalışmada motorun milindeki hızını düşürüp torku arttıran redüktörden bahsedilmiştir. Fırçalı DC motoru sürmek için kullanılan motor sürücüsü ve motor milinin son konumu için gerekli olan enkoder belirtilmiştir. Tasarlanan PID kontrolcü ve bu kontrolcünün parametrelerini belirleme problemini çözebilmek için kullanılan parçacık sürü optimizasyon algoritması, yapay arı koloni algoritması ve kaos oyun optimizasyon algoritması hakkında bilgiler verilmiştir.

3.1. İki eksenli Robotun Mekanik Tasarımı

DC motorların mil kısmının kayma yapmaması ve motorları sabitleyebilmek için ölçüleri uygun olacak şekilde tasarım yapılmıştır. Yapılan tasarım üç ayrı parça olarak tasarlanmıştır. İki adet robot kolu ve robot kol sistemini sabitlemek için gerekli tasarlanan parçalar 3 boyutlu yazıcıdan çıkartılıp birleştirilmiştir. Robot kollarına ait teknik resimler EK-1 kısmında Uzuv 1 için (A), Uzuv 2 için (B) olarak belirtilmiştir. Çalışmanın diğer kısımlarında kullanılan L1 ölçüsü 17,75 ve L2 ölçüsü 8,25cmdir. Tasarlanan 2 eksenli robota ait tasarım görseli Şekil 3.1’de ve bu tasarıma göre üretilen iki eksenli robot kolu ise Şekil 3.2’de verilmiştir.



Şekil 3.1.Tasarlanan iki uzuvlu robot kolu



Şekil 3.2. Üretilen iki uzuvlu robot kolu

3.2. Mikro Denetleyici Geliştirme Kartı

Kontrol yazılımının geliştirilmesi ve elektronik kontrol sistemi için mikrodenetleyici olarak Arduino Mega tercih edilmiştir. Arduino Mega maliyet olarak uygun olması ve yazılım ve uygulama kolaylığı açısından tercih edilmiştir. İçerisinde Atmega 2560 mikrodenetleyicisi barındıran bu geliştirme kartı, 54 adet dijital I/O pini, 15 adet PWM pini, 16 adet analog giriş pini, 16 MHz kristal osilatör özelliklerine sahiptir. Arduino Mega geliştirme kartına ait görsel Şekil 3.3.' de belirtilmiştir.



Şekil 3.3. Arduino Mega 2560 geliştirme kartı

3.3. DC Motor

DC motorlar, doğru akım olan elektrik gücünü mekanik güce dönüştüren elektrik makinalarıdır. DC motorların hızını sargılarındaki akımın gücü değiştirilerek veya değişken bir besleme voltajı kullanarak kontrol edilebilir. Geleneksel bir fırçalı DC motor temel olarak iki parçadan oluşur. Sabit olan kısım stator, dönen kısım ise armatör veya rotor adı verilir. Mil kısmındaki hareket rotora bağlıdır (Ramasubramanian, 2016).

3.3.1 Faulhaber DC Motor

Faulhaber firmasının 1524006SR serisinden olan redüktörlü, enkoderli DC motor 6V'luk ana gerilime sahiptir. Şekil 3.4'te gösterilen enkoderli redüktörlü fırçalı DC motor 6.68 Nm durma torkuna sahiptir. Çıkış gücü 1.7 watt'tır. Bu çalışma kapsamında kullanılan bu motor 1.5 mm mil çapına sahiptir. Mil kısmı dairesel ve küçük olduğu için

herhangi bir sisteme yerleştirilmesinde zorluklar meydana getirmektedir. Daha detaylı özellikleri Çizelge 3.1’de verilmiştir.

Çizelge 3.1. Faulhaber 1524006sr DC motor özellikleri

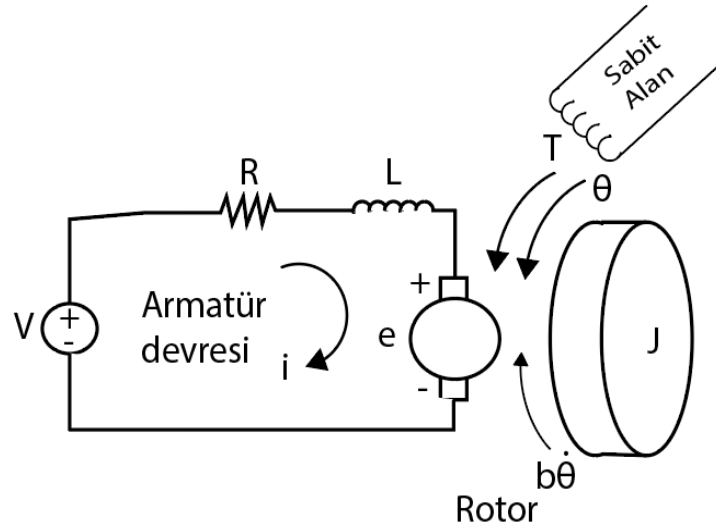
Özellik Adı	Değeri
Rotor Ataleti	0,66 gcm ²
Vizkoz sönümlenme	0
Elektriksel Sabit	0.172A/mNm
Tork Sabiti	5,8 mNm/A
Rotor Endüktansı	70 µH
Terminal Direnci	1,1Ω



Şekil 3.4. Faulhaber enkoderli redüktörlü DC motor

3.3.2. DC Motor Transfer Fonksiyonu

Kontrol sistemlerinde öncelik sistemin modelini elde etmektir. Bu çalışma kapsamında robotik sistemin tahrik elemanı olarak DC motor kullanılmıştır. Bu sebepten DC motorun transfer fonksiyonu veya durum uzay denklemi bulunmalıdır. Şekil 3.5’de DC motorun eşdeğer devresi gösterilmiştir (Maheriya ve Parikh, 2016).



Şekil 3.5. DC motor eş değer devresi

DC motorun transfer fonksiyonunu bulmak istersek devreye KVL uygularsak;

$$V = R_a * I_a * \frac{di_a}{dt} + E \quad (3.1)$$

Alan akımı sabit olduğundan ve akı sabit olacağından, armatör dönerken bir e.m.f indüktör,

$$E = \varphi * \omega \quad (3.2)$$

$$E = kb * w \quad (3.3)$$

Eşitlik 3.3'de kb geri besleme sabiti w açısal hızı ifade eder.

$$T a \varphi * i_a \quad (3.4)$$

$$T = K_t * i_a \quad (3.5)$$

Burada K motor tork sabitidir. Atalet momenti ve katsayılı dinamik tork denklemi şu şekilde verilir.

$$T = J * \frac{d^2\theta}{dt^2} + B * \frac{d\theta}{dt} \quad (3.6)$$

Eşitlik 3.1, 3.4, 3.5, 3.6 'nın laplace dönüşümünü alırsak;

$$V(s) - E(s) = I_a(s) * (R_a + s * L_a) \quad (3.7)$$

$$E(s) = K_b * s * \theta(s) \quad (3.8)$$

$$T(s) = K_t * I_a(s) \quad (3.9)$$

$$T(s) = (s^2 * J + s * B) * \theta(s) \quad (3.10)$$

$$T(s) = (s * J + B) s * \theta(s) \quad (3.11)$$

Eşitlik (3.6), (3.7), (3.8), (3.9), (3.10)'dan çıkış açısı ($\theta(s)$ 'in giriş voltajı $V(s)$ 'e oranıyla transfer fonksiyonu Eşitlik 3.12 ile bulunabilir (Maheriya ve Parikh, 2016).

$$\frac{\theta(s)}{V(s)} = \frac{K_t}{(R_a + s + La)(J * s * B)s + K_t * K_b * s} \quad (3.12)$$

Çizelge 3.1'de bulunan değerler Rotor ataleti J, vizkoz sönümlenme B, elektriksel sabit K_b , tork sabiti K_t , Rotor indüktansı L rotor direnci R Eşitlik 3.12'de bulunan transfer fonksiyonunda yerine konulup Faulhaber DC motor için transfer fonksiyonu bulunmuştur. DC motor için bulunan transfer fonksiyonu Eşitlik 3.13 de belirtilmiştir.

$$\frac{\theta(s)}{V(s)} = \frac{5,8 \times 10^{-3}}{(5,1 + 70 \times 10^{-6}s)(0,66 \times 10^{-7}s + 1)s + 5,8 \times 10^{-3}s} \quad (3.13)$$

3.4. Faulhaber Enkoder

İki kanallı olan FAULHABER Enkoder devir başına 16 çevrim ile dijital çıkış vermektedir. Şaft hızı, dönme yönü gösterilmesi ve açı kontrolü için kullanılır. 90° faz kaymalı iki kanal sağlamasına ek olarak çıkış sinyalleri için uygulanan gerilim ayrıca bağlantılı bir şerit kablo aracılığıyla görüntülenebilir. Bu tez çalışmasında kullanılan Faulhaber IE2-16 enkoderin özellikleri Çizelge 3.2'de belirtilmiştir.

Çizelge 3.2. E2-16 serili FAULHABER Encoder özellikleri

Özellik Adı	Değeri
Devir Başına Çevrim	16 N
Çıkış Sinyali, Kare Dalga	2 kanallı
Uygulanan Voltaj	4.....18 V DC
Akım Tüketimi, tipik($V_{DD}=12$ V DC)	Tipik 6, maks. 12 mA
Çıkış Akımı, (kabul edilebilir maksimum)	15 mA
Faz Kayması	90 ± 45 °e

3.5. Redüktör

Bu çalışma kapsamında kullanılan redüktör Faulhaber firmasının kendi motorları için ürettiği redüktör kullanılmıştır. Kullanılan bu redüktörün içerisindeki sistem çelik,

muhafazası ise metaldir. 76:1 redüksiyon oranına sahip bu redüktörler 3°'lik açılı kaymasıyla iyi bir performans göstermektedir.

3.6. Motor Sürücüsü

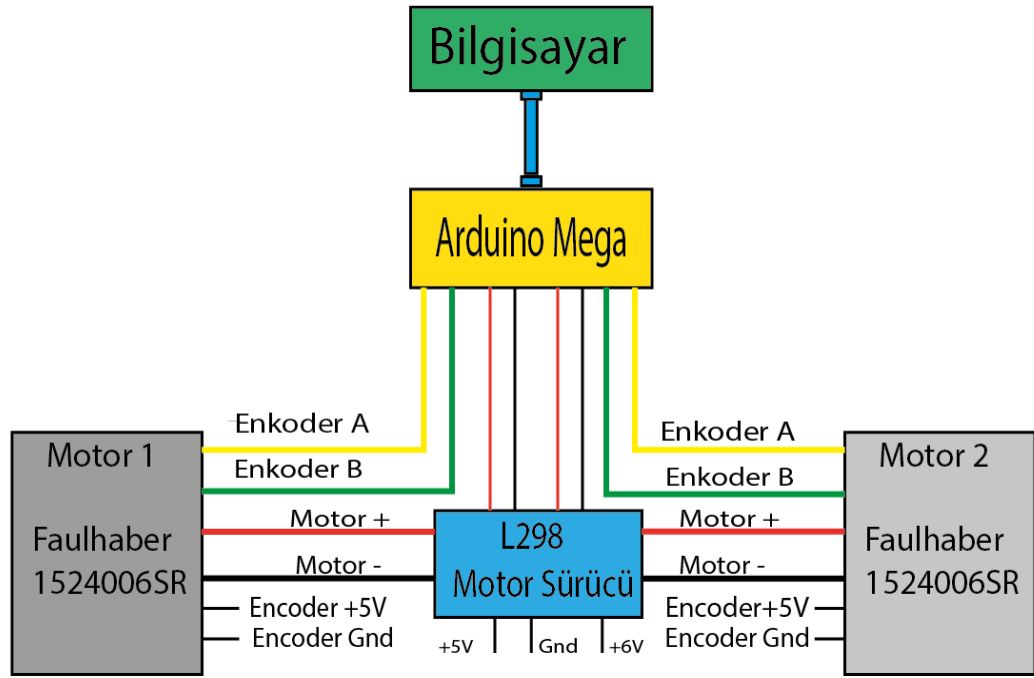
L298 motor sürücüsü röleler, bobinler, DC ve kademeli motorları sürmek için üretilmiş sahip tam köprü sürücüdür. Sürücüyü devre dışı bırakmak veya etkinleştirmek için üretilmiş iki etkinleştirme pinine sahiptir. İki ayrı motoru kontrol edebilir ve her kanala 2A akım verebilir. Sıcaklık ve devre koruması vardır. Üzerinde regülatörü vardır (Peerzada ve ark., 2021). L298 motor sürücüsünün görseli Şekil 3.6'da verilmiştir.



Şekil 3.6. L298 motor sürücüsü

3.7. Proje Bağlantı Şeması

Gerçekleştirilen robot kolu 2 uzuvlu olduğu için 2 adet redüktörlü enkoderli Faulhaber firmasının ürünü olan motor kullanılmıştır. Arduino Mega kartı ile motorlar arasında PWM sinyali için L298 motor sürücü kullanılmıştır. Sistem enkoderin hassasiyetinden dolayı güç kaynağından beslenmiştir. Çalışma içerisinde kullanılan bağlantı şeması Şekil 3.7'de detaylı olarak verilmiştir.



Şekil 3.7. Proje bağlantı şeması

3.8. İki Uzunlu Robotun Kinematik Denklemlerinin Bulunması

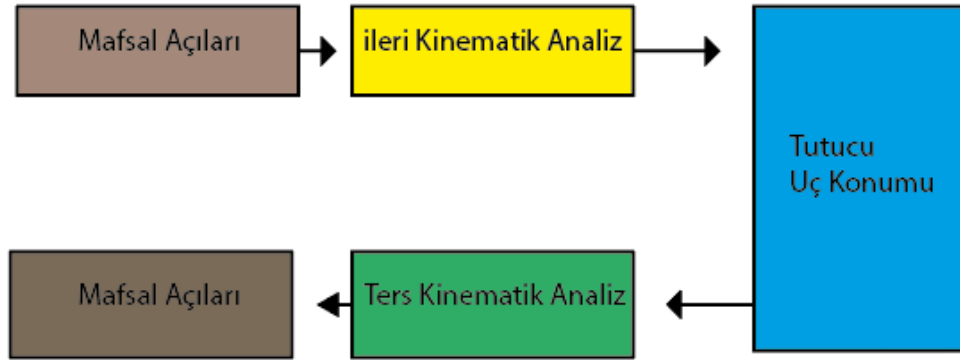
Robotik Sistemlerde robot kolunun kinematik analizinde dönele ve prizmatik olmak üzere iki hareket vardır. Bu çalışmada kontrolü düşünülen robot kolunun rotasyonel olan iki eksenine sahiptir. Robot kollarını hareket ettirebilmek için tahrik elemanı gereklidir. Bu tahrik elemanı sayısı arttıkça sistemin kontrolü daha zor hale gelmektedir. Sistemin iyi kontrol edilebilmesi için kinematik hesabın doğru yapılması gerekmektedir.

Robotik sistemlerde uç elemanın üç boyutlu uzaydaki toplam yer değiştirmesi problemi üzerine birçok yöntem üzerine çalışılmıştır. Bunlardan biri olan ve en çok tercih edilenlerden biri D-H metodudur. Denavit ve Hartenberg tarafından 1955 yılında geliştirilmiştir. Bu metod da 4x4 homojen matrisi işlemleri ile uç nokta yer değiştirilmesi hesaplanması amaçlanır (Fu ve ark., 1987).

İki serbestlik derecesine sahip bir robot kolu iki rijit uzvun uç kısmı açık mekanizma oluşturacak biçimde mafsallarla birleştirilerek tahrik elemanları ile hareketlendirilmesi şeklinde modellenilebilir. Bir robot kolu, referans koordinat takımına göre öteleme ve dönme şeklinde iki temel hareket yapabildiğinden her bir mafsal üzerinde koordinat takımının bulunduğu kabul edilir. Robot kolunun başlangıç noktası sabit kabul edilip kinematik analiz ile robot kolunun uzaysal konumu elde edilir. Kinematik analiz için iki temel yöntem vardır (Bingül ve Küçük, 2005):

1. İleri kinematik yöntem: Uç noktanın sabit eksen takımlarına göre konumu ve yönlendirilmesi.
2. Ters kinematik yöntem: Uç noktanın istenen konumu ve yönlendirilmesi.

İki eksenli bir robot kolu ele alındığında, bağımlı olmayan değişkenler mafsalsal değişkenleri olduğundan ters kinematik yöntem daha sık kullanılır (Bingül ve Küçük, 2005). Daha açıklayıcı olan bir blok diyagramı Şekil 3.8’de verilmiştir.



Şekil 3.8. İleri ve ters kinematik blok diyagramı

Denavirt-Hartenberg yönteminde robotik sistemin eklem değişkenlerini belirlemek hedeflenir. D-H metodunda kaç adet dönme ve öteleme varsa bir fazlası olacak şekilde numaralandırılır. Bu numaraların her birine koordinat eksenleri yerleştirilir ve dönme eksenine dik, öteleme eksenine paralel olacak şekilde Z eksenine yerleştirilir. Z ekseninden sonra X eksenine Z eksenine dik ve kol boyunca uzanan eksene Şekil 3.4’de olduğu gibi X eksenine yerleştirilir. Üç eksen olduğu için X ve Z eksenine sağ el kuralı uygulanarak boş eksene ise Y eksenine yerleştirilir. Eksenler yerleştirildikten sonra ise D-H yöntemindeki dört parametre belirlenir. Bu 4 parametre a_i , α_i , d_i , θ_i dir (Denavirt ve Hartenberg, 1955). Çizelge 3.3’de belirtildiği gibi robotun serbestlik derecesi satır sayısına eşittir. d_i ve θ_i parametrelerinin herhangi biri değişken iken a_i ve α_i robotun hareketleri ile değişmeyen sabitlerdir.

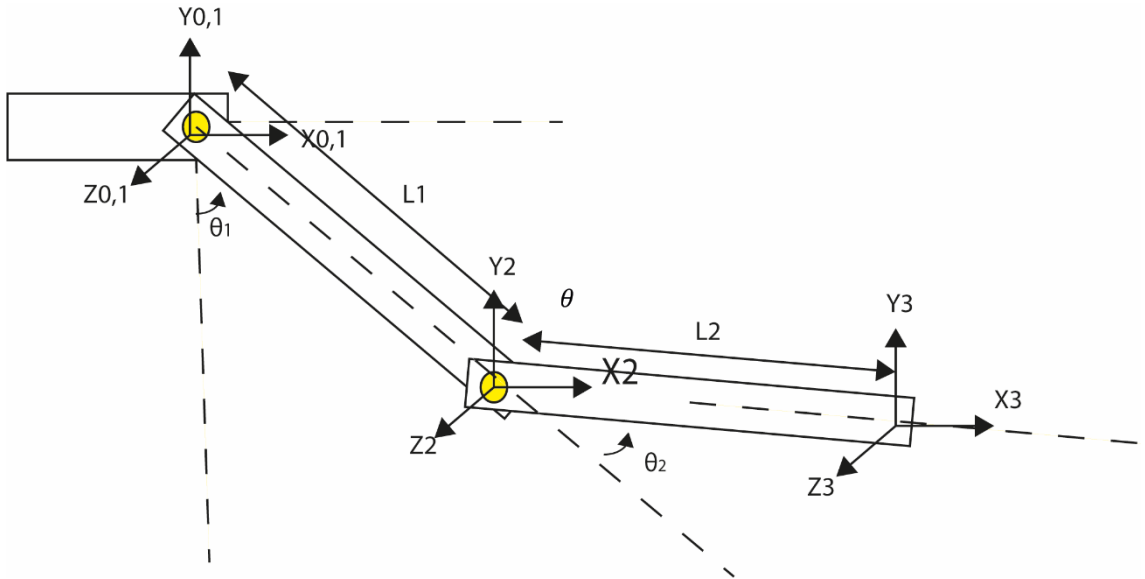
Çizelge 3.3. Değişken parametrelerinin tabloya girilmesi (Bingül ve Küçük, 2005)

Eksen numarası	Değişkenler				
i	a_i	α_i	d_i	θ_i	d_i veya θ_i
1	α_0	a_0	d_0	θ_0	d_0 veya θ_0
2	α_1	a_1	d_1	θ_1	d_1 veya θ_1

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i * \cos \alpha_i & \sin \theta_i * \sin \alpha_i & a_i * \cos \theta_i \\ \sin \theta_i & \cos \theta_i * \cos \alpha_i & -\cos \theta_i & a_i * \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

$$T = \begin{bmatrix} l & l & l & D \\ l & R & l & D \\ l & l & l & D \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

Eşitlik 3.14'deki A_i matrisinde Çizelge 3.3'deki her eksen sırası ile yazılarak dönüşüm matrisleri A_1, A_2 , sırasıyla bulunur. Her bir eksenin homojen dönüşüm matrisi çarpma işlemi yapılarak düz kinematik için gerekli olan matris (transformasyon matrisi) bulunur. Bulunan bu matris 4x4 boyutundadır. Eşitlik 3.14'deki transformasyon matrisinde R matrisinin bulunduğu 3x3'lük matris dönme 1x3'lük olan D matrisi ise konum ile ilgilidir. Eşitlik 3.15 de dönme ve öteleme ile ilgili kısım belirtilmiştir. Bu çalışma kapsamında elde edilen kinematik ifadeler için öncelikle iki eksenli robotun eksen takımlarını elde etmek gerekir. Eksen takımlarının elde edildiği şematik resim Şekil 3.9'da verilmiştir.



Şekil 3.9. Uzunlu robotun eksen takımlarının yerleştirilmesi

Kontrol edilebilmesi düşünölen iki uzuvlu robot kolu iki serbestlik derecesine sahiptir. Eklem deęişkenleri olan θ_1 ve θ_2 'de dönme hareketi mevcuttur. l_1 , l_2 uzuv boylarıdır. Belirlenen eksenlere göre D-H parametreleri Çizelge 3.4'de gösterilmiştir.

Çizelge 3.4. İki uzuvlu robot kolunun D – H çizelgesi

D-H Tablosu	θ	d	a	α
0-1	ϑ_1	0	0	0
1-2	ϑ_2	0	l_1	0
2-3	0	0	l_2	0

Elde edilen D – H tablosuna göre robotun uç elemanının tabana göre konum ve oryantasyonunu gösteren transformasyon matrisi Eşitlik 3.16'deki şekilde hesaplanabilir.

$$T = A_1 * A_2 * A_3 \quad (3.16)$$

Transformasyon matrisini bulabilmek için öncelikle A_1 , A_2 , A_3 , matrislerinin en sade halini belirleyip ardından transformasyon matrisi bulunabilir. Eşitlik 3.16'deki işlem aşağıdaki gibi elde edilir.

$$A_1 =$$

$$\begin{bmatrix} \cos(\vartheta_1) & -\sin(\vartheta_1) & 0 & 0 \\ \sin(\vartheta_1) & \cos(\vartheta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$$A_2 =$$

$$\begin{bmatrix} \cos(\vartheta_2) & -\sin(\vartheta_2) & 0 & l_1 \\ \sin(\vartheta_2) & \cos(\vartheta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

$$A_3 =$$

$$\begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

Buna göre Eşitlik 3.17, 3.18, 3.19 deki matrisler, Eşitlik 3.16'de yerine koyularak robot transformasyon matrisi Eşitlik 3.20' deki gibi bulunur.

$$T = \begin{bmatrix} \cos(\vartheta_2 + \vartheta_1) & -\sin(\vartheta_2 + \vartheta_1) & 0 & l_2 \cos(\vartheta_2 + \vartheta_1) + l_1 \cos(\vartheta_1) \\ \sin(\vartheta_2 + \vartheta_1) & \cos(\vartheta_2 + \vartheta_1) & 0 & l_2 \sin(\vartheta_2 + \vartheta_1) + l_1 \sin(\vartheta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

Kinematik daha önce belirtildiği gibi düz ve ters olmak üzere iki kısımdan oluşur. Bu hesaplamalar bu çalışma içerisinde matematiksel işlemlerle ve paket programlarla sonuçlandırılmıştır. Geometri yardımı ile yapılan hesaplama Eşitlik 3.21 ve Eşitlik 3.22'de verilmiştir.

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (3.21)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (3.22)$$

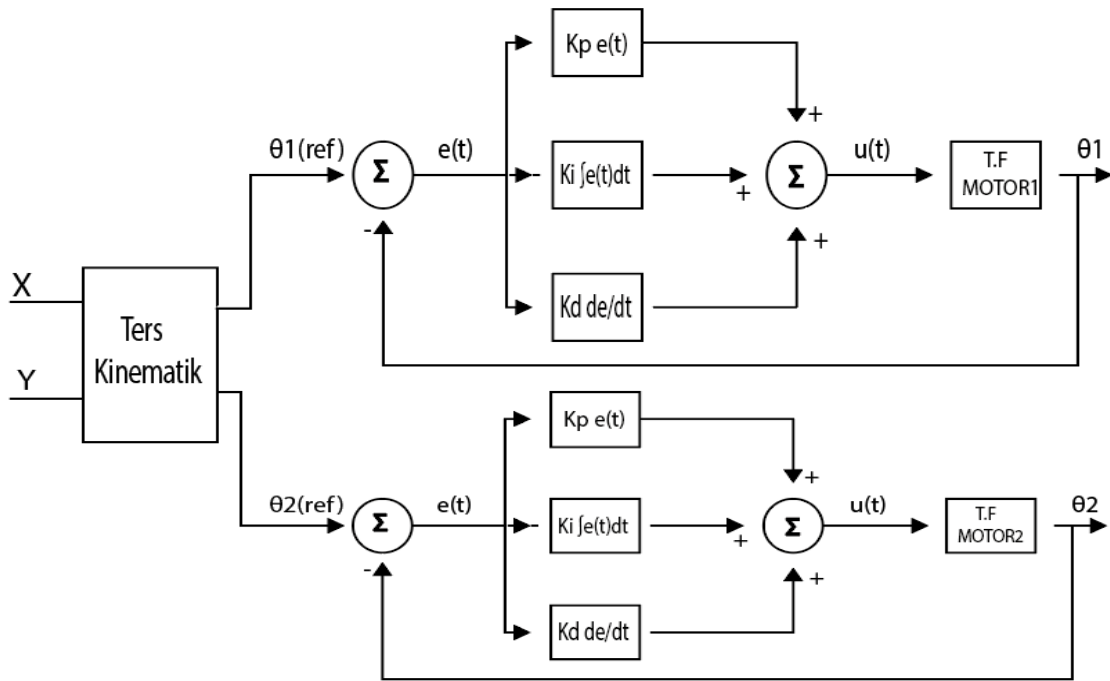
Ters kinematik işlemler sonucunda değişkenlerimiz olan θ_1 ve θ_2 'nin denklemleri Eşitlik 3.23 ve 3.24 'da elde edilmiştir (Bingül ve Küçük, 2005).

$$\theta_1 = \text{atan}\left(\frac{y}{x}\right) + \tan^{-1}\left(\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2}\right) \quad (3.23)$$

$$\theta_2 = +\cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \quad (3.24)$$

3.9. İki Uzunlu Robot Kolunun Kontrolör Tasarımı

Oransal İntegral türev denetleyici (PID) kontrol alanında kapsamlı olarak kullanılan geri besleme denetleyici yöntemidir. Sistemin mevcut çalışma durumu ile amaçlanan durum arasındaki farkı sürekli olarak hesaplar. Amacı hatayı minimuma indirmektir. Bu hatayı minimuma indirirken içerisindeki türevsel, oransal, integral katsayılarını değiştirme işlemi yapılmaktadır. Herhangi bir katsayıya sıfır verilerek PD, PI olarak da tanımlanabilir. Bu çalışma için yapılan simülasyon tasarımı Şekil 3.10'da gösterilmiştir.



Şekil 3.10. Çalışma blok şeması

PID parametrelerinin optimize edilmesi için birçok farklı yöntemler kullanılmaktadır. Bu amaç için literatüre bakıldığında birçok optimizasyon algoritması görülmektedir. Bu çalışma kapsamında parçacık sürü optimizasyon, yapay arı koloni ve de kaos oyun optimizasyon algoritmaları kullanılarak PID parametrelerinin optimizasyonu yapılmıştır.

3.10. Parçacık Sürü Optimizasyon Algoritması

Parçacık sürü optimizasyon algoritması, 1995 yılında tasarlanmış olsa da son yılların en popüler evrimsel optimizasyon yöntemlerinden biridir. Sürü mantığında hareket eden canlılardan esinlenilmiştir. Popülasyon mantığına dayandığı için grup içerisindeki üyeler, en makul performansı gösteren üyeyi izleme eğilimindedir. Diğer meta-sezgisel optimizasyon algoritmalarına göre en büyük farkı popülasyon içerisindeki tüm üyelerin geçmiş deneyimlerinden faydalanmasıdır (Karahan, 2012).

Belirgin durumlarda, mevcut seçeneklerden en yakın sonuç verme işlemine optimizasyon denir. Evrende yaşamsal faaliyetlerini tek başına yaparken zorlanan ancak sürünün bir bireyini oluşturduğunda grup içerisinde en iyi bireyin durumunda etkileyerek ilerleyen aşamada karşılaşılabilecek sorunları çözümlenmede daha iyi sonuç veren canlılar

vardır. Parçacık sürü optimizasyonu (PSO) kurt, göçmen kuşları ve arılar gibi sürü halinde canlı türlerinden esinlenilmiştir. PSO diğer optimizasyon yöntemlerine benzer fakat matematiksel olarak daha basit olması optimizasyonu algoritma olarak uygulamasını kolaylaştırmıştır. Bu kolaylıktan dolayı birçok alanda aktif olarak kullanılmaktadır (Pektemir, 2017).

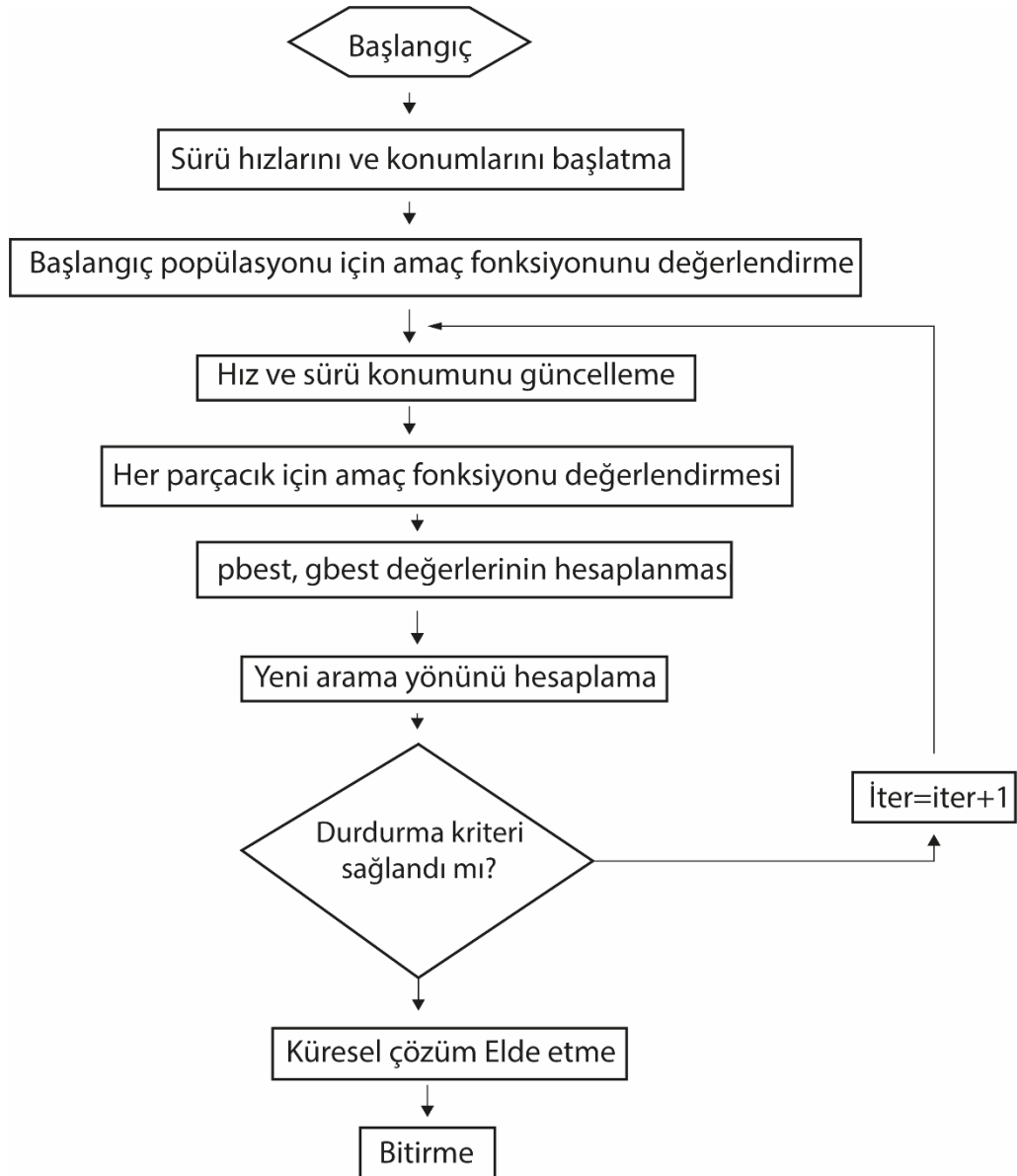
PSO'nun çalışma disiplini durum uzayında rastgele olan sonuçların mevcut sonuçlardaki en iyiye yönlendirilmesini sağlar.

Her kısım bir hız değerine sahiptir. Bu hız değeri bir sonraki konumu belirler. Her bir adımda parçacığın mevcut hızı ve önce ki adımdaki hesaplanan hızı toplama işlemi ile yeni konum bulunur. Parçacığın yeni konumunu bulması ile ilgili matematiksel ifade Eşitlik 3.25'de verilmiştir.

$$x_n = x_{n-1} + v_n \quad (3.25)$$

Eşitlik 3.25'de x_n parçacığın konumu, v_n parçacığın hızıdır. Parçacığın son adımdaki hızı ifadesi ise Eşitlik 3.26'da verilmiştir. Eşitlik 3.26'da v_{n-1} Parçacığın bulunduğu en iyi noktanın konumu P_{n-1} , tüm parçaların anlık bulunduğu en iyi noktanın konumu g_{n-1} dir (Durusu, 2022). Parçacık sürü optimizasyon algoritmasını oluştururken kullanılan algoritmanın akış şeması Şekil 3.11' de belirtilmiştir.

$$v_n = c_1 \cdot v_{n-1} + c_2 (x_{n-1} - P_{(n-1)}) + c_3 (x_{n-1} - g_{n-1}) \quad (3.26)$$



Şekil 3.11. PSO akış şeması (Azab, 2009)

3.11. Yapay Arı Koloni Optimizasyon Algoritması

Yapay arı koloni algoritması (YAKA) doğada birçok bilime inceleme konusu olan arıların besin ihtiyacını karşılamak için yaptığı davranışları modelleyerek elde edilmiş sürü mantığına dayanan algoritmadır. Arılar aralarında yaptığı iş bölümü ile kendilerini organize edip gerçekleştirir. Yaka içerisinde besin kaynağı sayısı ile arı sayısı eşittir. Bir diğer eşitlik ise işçi arı ile gözcü arı sayısıdır. Kovadaki besin bitince görevli olan arı yeni besin kaynağı arayan olmaktadır. YAKA en çok miktarda besin içeren kaynağın yerini tespit etmede çalışarak durum uzayındaki problemi maksimuma ya da minimuma veren çözümü bulmayı amaçlar (Öztürk ve Öztürk, 2016).

Yaka algoritmasında öncelikle tüm arılar en çok besin bulunan konumu bulmak için rastgele bir uzaklığa gider. İlk gidilen konumların koordinatları arıların çalışma alanı içerisinde rastgele başlatılır.

$${}^0x_j = \min_{x_j} + \lambda(\max_{x_j} - \min_{x_j}), \quad \min_{l_j} \leq l_j \leq \max_{l_j} \quad (3.27)$$

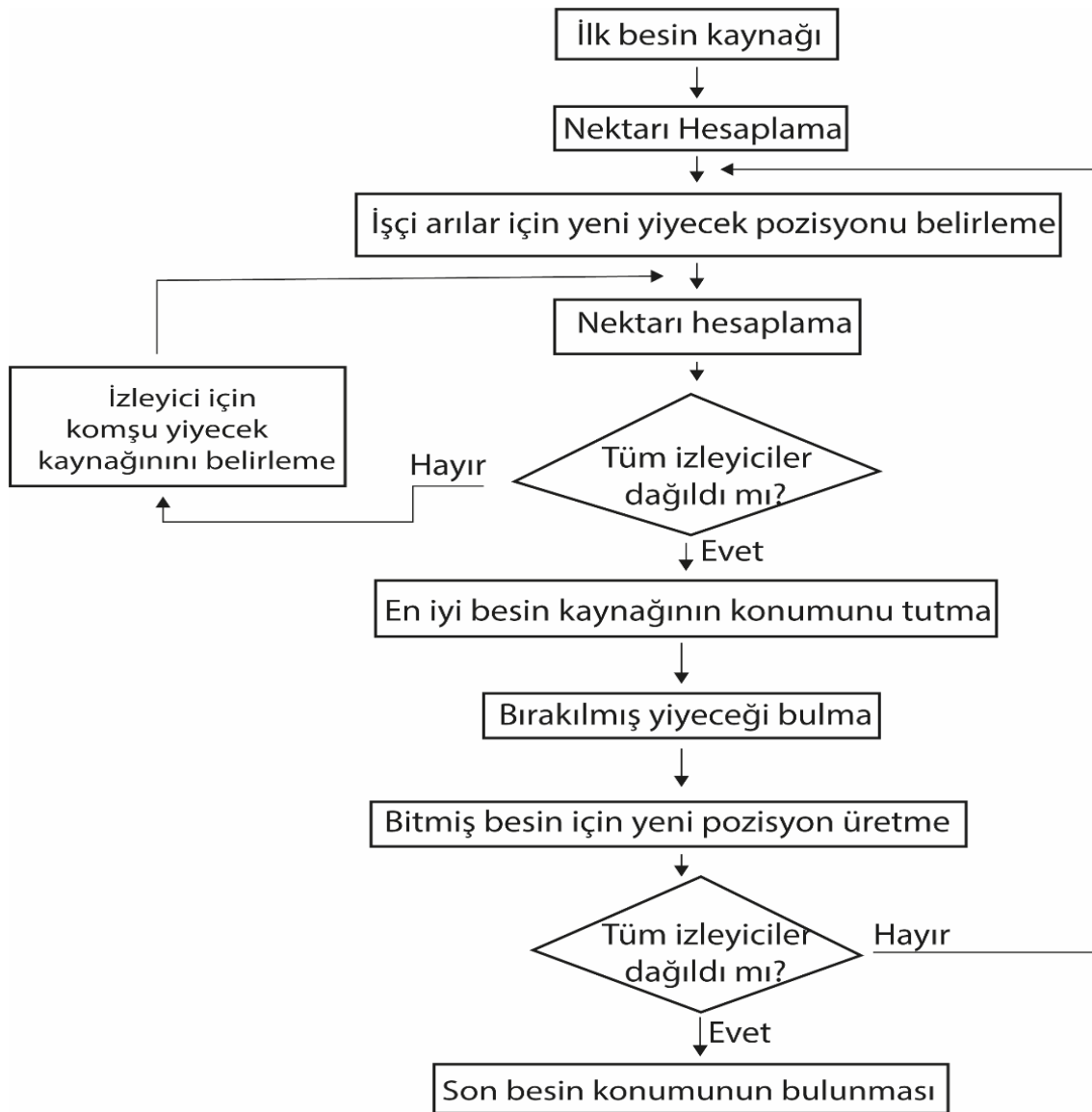
Eşitlik 3.27'deki I tasarım değişkenlerinin Çalışma uzayındaki yerini ifade etmektedir. Optimizasyon içerisinde amaç fonksiyonunun değerinin hesaplanması, tüm arılar için koordinatlar seçildikten sonra arıların son koordinatlardaki buldukları besin miktarlarını belirlemek anlamına gelir. Bu işlemden sonra en çok besin bulan arılar aynı konumlara giderken diğer arılar izleyici görevinde kalır. Çok besin bulan arılar dans ederek izleyici olan arıları yanına çekmeye çalışır. Burada en çok besini bulandan en aza doğru bir yönelim kabul edilir (Sevim ve Sönmez, 2014).

$$p_i = \frac{(if(x))^{-1}}{\sum_{m=1}^n (mf(x))^{-1}} \quad (3.28)$$

Eşitlik 3.28 ile izleyici olan arıların kaç tanesinin dans eden arıların yanına gideceği bulunur. Optimizasyon olarak bu işlemde besin kaynağı tarafındaki arıların hepsinin yeri tekrardan hesaplanır.

$$yeni {}_i l_j = isth {}_i l_j + \emptyset (isth {}_i l_j - isth {}_i l_j) \quad (3.29)$$

Eşitlik 3.29 'de \emptyset değeri 1 ile -1 arasında bir reel sayıdır. Optimizasyon içerisinde i besin kaynağı numarasını, j tasarım değişkeninin adını k ise SN ile 1 arasında bir sayıdır. K, i değerine eşit olamaz. Yapay arı koloni optimizasyon algoritmasını oluştururken kullanılan algoritmanın akış şeması Şekil 3.12' de belirtilmiştir.



Şekil 3.12.YAKA Akış şeması (Hassan ve ark., 2017)

3.12. Kaos Oyun Optimizasyon Algoritması

Kaos oyun optimizasyon algoritması Talatahari ve Azizi tarafından geliştirilen son yıllarda gelecek vaat eden popülasyon tabanlı optimizasyon algoritmasıdır. Kaos isimli oyundan esinlenerek yapılan bu optimizasyon algoritması bir çokgen üzerinde kaos oyunu metodolojisi yardımıyla fraktalların üretildiği belirli kaos teorisi ilkeleridir. Bu oyunda rastgele oluşturulmuş başlangıç noktalarından rastgele yinelemeli bir süreç aracılığıyla fraktal desenler oluşturmanın bir biçimidir. Sierpinski üçgeni olarak bilinen geometrik yapıyı oluşturma hipotezine dayalı olarak tasarlanmıştır (Boğar, 2022).

CGO yüksek hesaplama süresi verimliliği ve kolay uygulama gibi olumlu yönleriyle kısıtlı optimizasyon problemlerini çözmeye oldukça etkilidir. CGO 'nun diğer optimizasyon algoritmalarına göre en büyük farkı ise parametresiz olmasıdır. Yani popülasyon büyüklüğü, maksimum yinleme gibi parametre dışında herhangi bir parametreye ihtiyaç duymaz. Bu özelliği sayesinde zorlu problemlerin üstesinden gelir. Birçok optimizasyon algoritmasında olduğu gibi rastgele başlangıç arama adayları üreterek optimizasyona başlar. Bu başlama işlemine popülasyon büyüklüğüne, karar değişkeninin sayısına ve çözüm uzayının sınırlarına bağlı olarak gerçekleştirir (Boğar, 2022).

$$\underline{x}_i^j = \underline{x}^j + \text{rand.}(\bar{x}^j - \underline{x}^j), \quad i=1,2,3,\dots,N; \quad j=1,2,3,\dots,d; \quad (3.30)$$

Eşitlik 3.30'da bir Sierpinski üçgeni içinde \underline{x}_i^j adayının i değişkeninin ve j konumunu, N uygun birey sayısı yani (popülasyon büyüklüğünü) d bireylerin boyutu j değişkenin konumunu temsil eder. Rand rastgele sayıyı ifade eder. Tüm uygun bireylerin pozisyonlarını tutan bir matris Eşitlik 3.31'deki gibi temsil edilebilir.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_N \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \cdot & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \dots & x_N^d \end{bmatrix} \end{bmatrix}_{N \times d} \quad (3.31)$$

Başlangıç bireylerinin uygunluk değerlerini, önceden tanımlanmış bir amaç fonksiyonu kullanılarak hesaplanır.

$$F(x) = \begin{bmatrix} f_1([x_1]) \\ f_1([x_1]) \\ \vdots \\ f_N([x_N]) \end{bmatrix}_{N \times 1} \quad (3.32)$$

Eşitlik 3.32'de $F(x)$ arama adaylarının uygunluk değeridir. Uygunluk değerinden sonra yeni arama adayları üretme aşamasına geçilir. Bu aşamada Sierpinski üçgeninin tüm yapısını tamamlamak için yeni uygun bireyler seçilir. Bunun için üç birey ile bir geçici/ara üçgenin çizildiği varsayılır. (i) şimdiye kadarki en iyi uygunluk değerine sahip en iyi birey (x_{best}), (ii) popülasyon matrisinden rastgele seçilmiş bireyleri ifade eder

(xmg), (iii) i. Birey (xi). Kaos oyunu metodolojisinde, üç tohum (xbest,xmg,xi) bir kalıp kullanılarak geçici üçgende yeni bireyler oluşturulur. Matematiksel olarak Eşitlik 3.33'de ifade edilebilir (Boğar, 2022).

$$\begin{aligned}
 x_i^{new1} &= x_i + \alpha_1 \cdot (\beta_1 \cdot x_{best} - \gamma_1 \cdot x_{mg}), \\
 x_i^{new2} &= x_{best} + \alpha_2 \cdot (\beta_2 \cdot x_i - \gamma_2 \cdot x_{mg}), \\
 x_i^{new3} &= x_{mg} + \alpha_3 \cdot (\beta_3 \cdot x_i - \gamma_3 \cdot x_{best})
 \end{aligned} \tag{3.33}$$

Burada $\alpha^* \in \{1,2,3\}$ için β^* ve γ^* , bir zar atma değerine karşılık gelen 0 veya 1 değerine sahip rastgele tam sayılardır. α^* , Eşitlik 3.34'deki dört farklı mekanizma sayesinde araştırma için dengeleyici olarak kullanılır. Her bir bireyin konumu rastgele olarak bu bireylerin biri tarafından seçilir.

$$\alpha_* = \left\{ \begin{array}{l} rand \\ 2 \cdot rand \\ (\delta \cdot rand) + 1 \\ (\epsilon \cdot rand) + (\sim \epsilon) \end{array} \right\} \tag{3.34}$$

Burada ϵ ve δ sıfır ile bir arasında tam sayıdır. CGO üç yeni birey dışında çözüm çeşitliliği arttırmak için dördüncü bir yeni tohumun üretildiği bir mutasyon aşamasına sahiptir. Dördüncü yeni tohumun konumu, rastgele seçilen karar değişkenlerindeki bazı rastgele varyasyonlara dayalı olarak belirlenir. Matematiksel olarak Eşitlik 3.35'deki gibi ifade edilebilir.

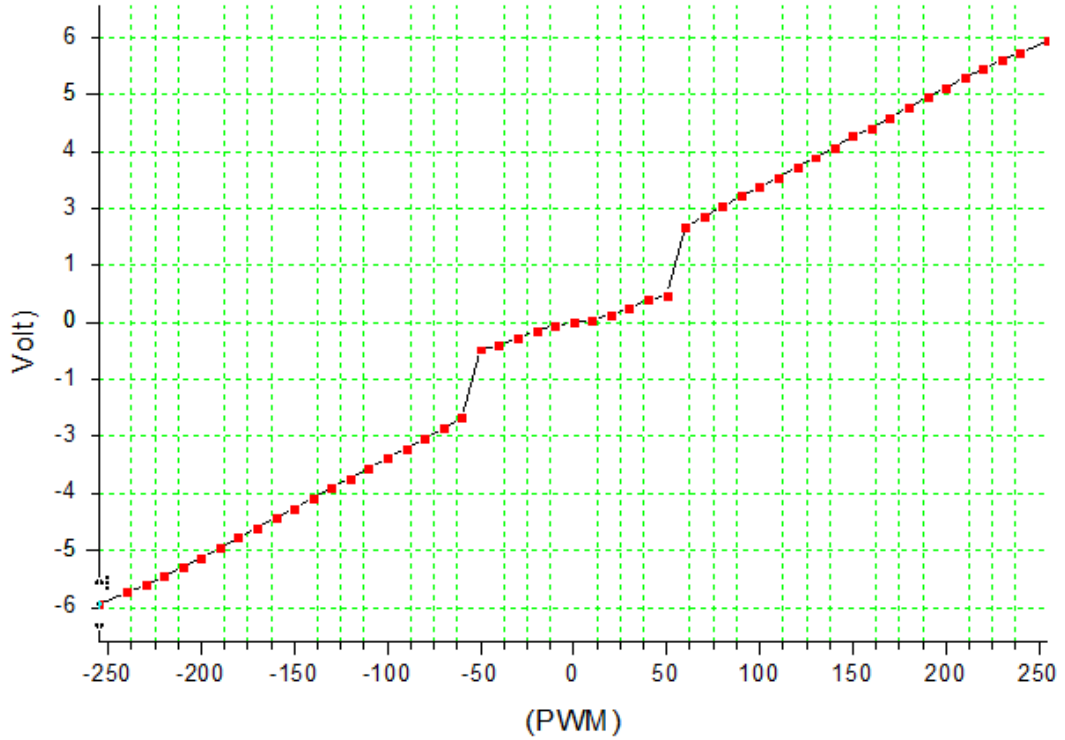
$$x_i^{new4} = x_i^k \quad (x_i^k = x_i^k + rand), \quad k=[1,2,3,\dots,d] \tag{3.35}$$

Yeni arama adayları oluşturma aşamasından sonra, yeni bireyler önceden tanımlanmış arama uzayının ötesine geçebileceğinden bir sınır kontrol yöntemi gerçekleştirilir. Bu yöntemde ihlal eden değişkenlerin her biri en yakın alt veya üst sınırına sabitlenir (Talathari ve Azizi, 2021). Ek olarak yeni bireylerin popülasyona dahil edilip edilmeyeceğine karar vermek için bir güncelleme yöntemi kullanılır. Yeni bireylerin uygunluk değerleri mevcut olanlardan daha iyi ise bu bireyler popülasyona

enjekte edilir (Boğar, 2022). Bu aşamada en kötü benzerlik değerine gelen en kötü bireyler popülasyondan atılır. Son adım ise bir durdurma adımı ile optimizasyon süreci sonlandırılır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

İki uzuvlu robot kolunu kontrol edebilmek için öncelikle tahrik elemanını kontrol edebilmek gereklidir. Bu sistemde kullanılan fırçalı DC motor, robot kolunun her bir mafsalında yer almaktadır. Herhangi bir motordaki açısal sapma robotun tutucu ucunu doğrudan etkilemektedir. Bu sebeplerden dolayı bu çalışma içerisinde öncelikle DC motor kontrolü üzerine odaklanılmıştır. DC motordaki elde edilen en uygun parametrelili PID kontrolcü üzerinden robot kolu kontrolü gerçekleştirilmiştir. Kontrol edilecek DC motorun çalışma gerilimi ile doğru orantılı PWM değerleri -255 ile +255 arasındadır. DC motor sırasıyla 255 ve -255 PWM değerlerinde istenilen çalışma performansında sırasıyla 6V ve -6V olarak çalışmaktadır. Faulhaber fırçalı DC motor 2.7V veya -2.7V değerlerinde, diğer bir deyişle 50 ile 60 arası PWM veya -50 ile -60 arası PWM değeri motor istenilen değer olmasa da harekete geçmektedir. Kullanılan motorun Voltaj ile oluşturulan PWM sinyalleri ile ilgili detaylı grafik Şekil 4.1’de verilmiştir.



Şekil 4.1. DC motor için PWM-Volt değeri

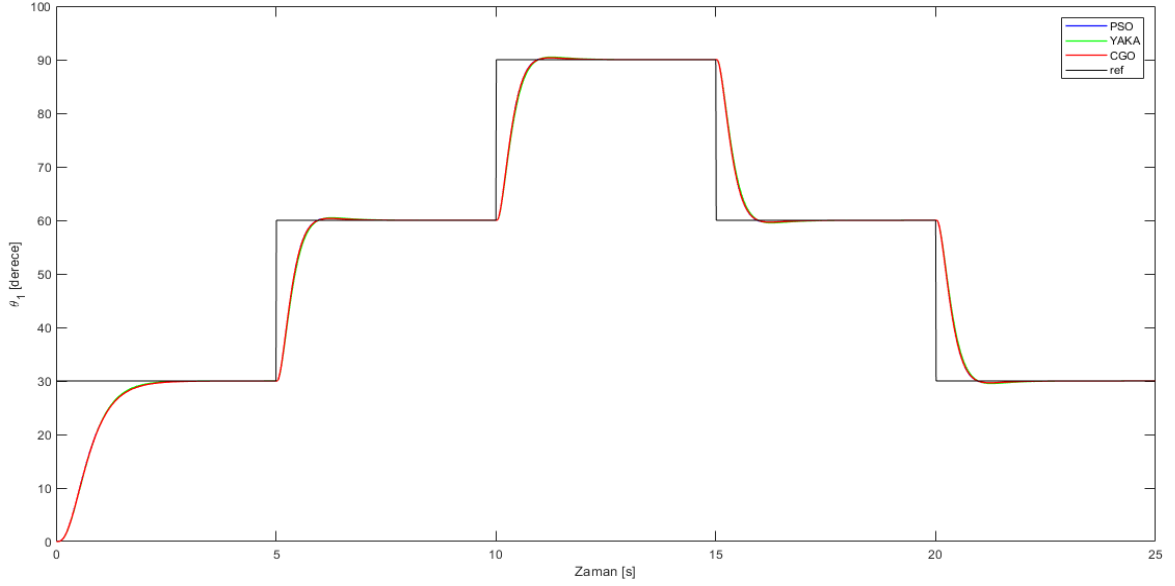
Bu çalışmada serbestlik derecesi iki olan iki uzuvlu bir robota tahrik elemanı olarak planlanan iki adet DC motor için üç ayrı PID parametrelerini optimize etmek için

üç tane optimizasyon algoritması kullanılmıştır. Bu algoritmalar parçacık sürü optimizasyon algoritması, yapay arı koloni algoritması, kaos oyun optimizasyon algoritmasıdır. Adil karşılaştırma için tüm algoritmaların popülasyon sayısı ve maksimum fonksiyon değerlendirme sayısı sırasıyla 18 ve 360 olarak ayarlanmıştır. Ayrıca PSO için bilişsel ve sosyal sabitler 2, atalet ağırlığı [0,9 0,4] bu değerler RMİT üniversitesindeki Xiaodong Li çalışmalarından alınmıştır (Li, 2006). YAKA için ise Limit 100 olarak ayarlanmıştır. Yaka için limit değeri optimizasyon algoritmasının mucidi olan Derviş Karaboğa'nın Erciyes üniversitesinde yaptığı çalışmadan alınmıştır (Karaboğa, 2005). Bu parametreler çalışma boyunca tüm hareket senaryoları için sabit tutulmuştur. Çizelge 4.1'de θ_1 kontrolü için her bir meta-sezgisel optimizasyon algoritması tarafından elde edilen DC motorun PID parametreleri k_p , k_i , k_d ayrı ayrı belirtilmiştir.

Çizelge 4.1. θ_1 değeri için PSO, YAKA ve CGO algoritmalarından bulunan PID parametreleri

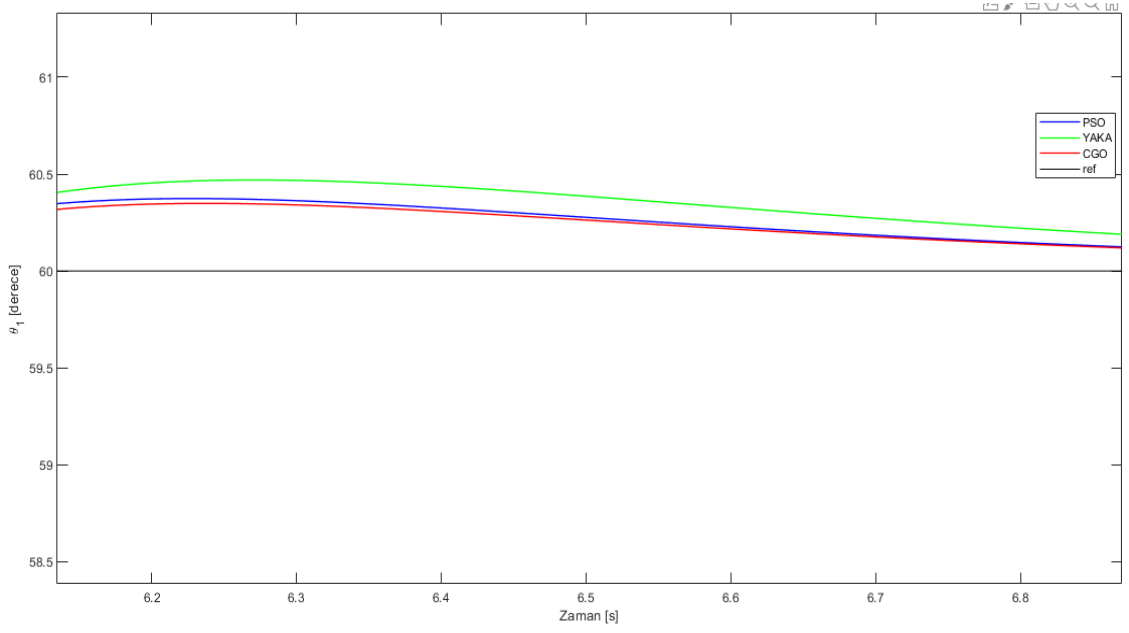
Kontrolcü parametresi	PSO	YAKA	CGO
k_p	29,40	28,69	29,27
k_i	1,24	1,24	1,23
k_d	13,49	12,8	13,48

İki uzuvlu robot kolunun θ_1 değerlerinin kontrol simülasyonu zaman ve açı (derece) grafiği Şekil 4.2’de verilmiştir.



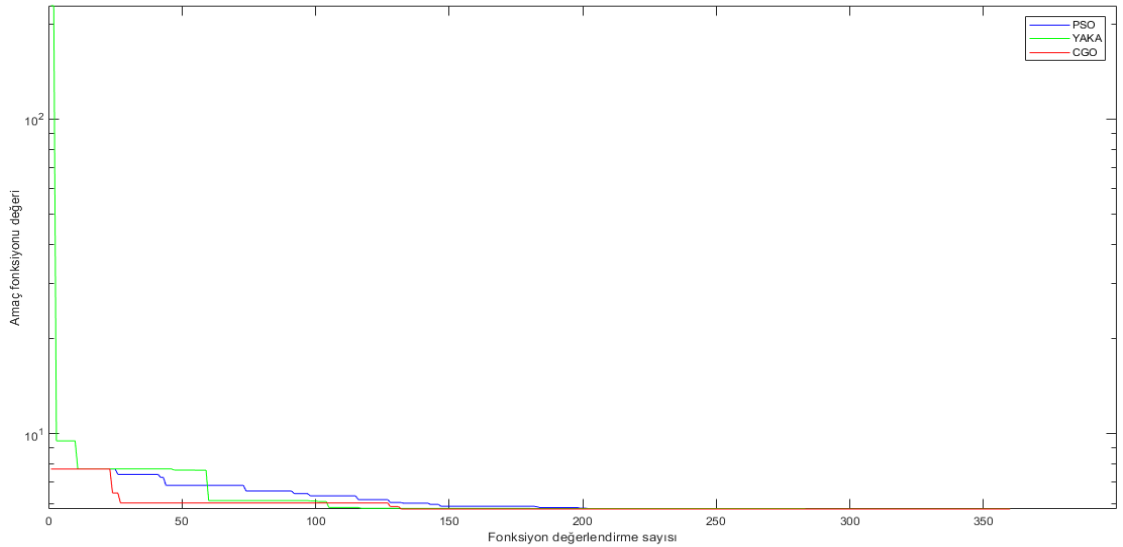
Şekil 4.2. θ_1 için Girilen referans değerine karşılık PSO, YAKA, CGO karşılaştırması

Şekil 4.2’de sisteme sırasıyla girilen 30° , 60° , 90° , 60° ve 30° açı değerleri için PSO, YAKA ve CGO algoritmaları iç içe olduğu görülmektedir. Şekil 4.3’de belirtilen şekilde iç içe olan grafiğin 60° ye oturması ile ilgilidir. Şekil 4.3’den CGO, PSO’nun biraz önünde YAKA ise bu optimizasyon algoritmalarının biraz gerisinde sonucuna ulaşılabilir.



Şekil 4.3. θ_1 için girilen referansın PSO, YAKA, CGO karşılaştırmasının detaylı görüntüsü

Şekil 4.4’de görüldüğü üzere CGO algoritmasının PSO ve YAKA algoritmalarına göre daha iyi yakınsadığı görülmektedir.



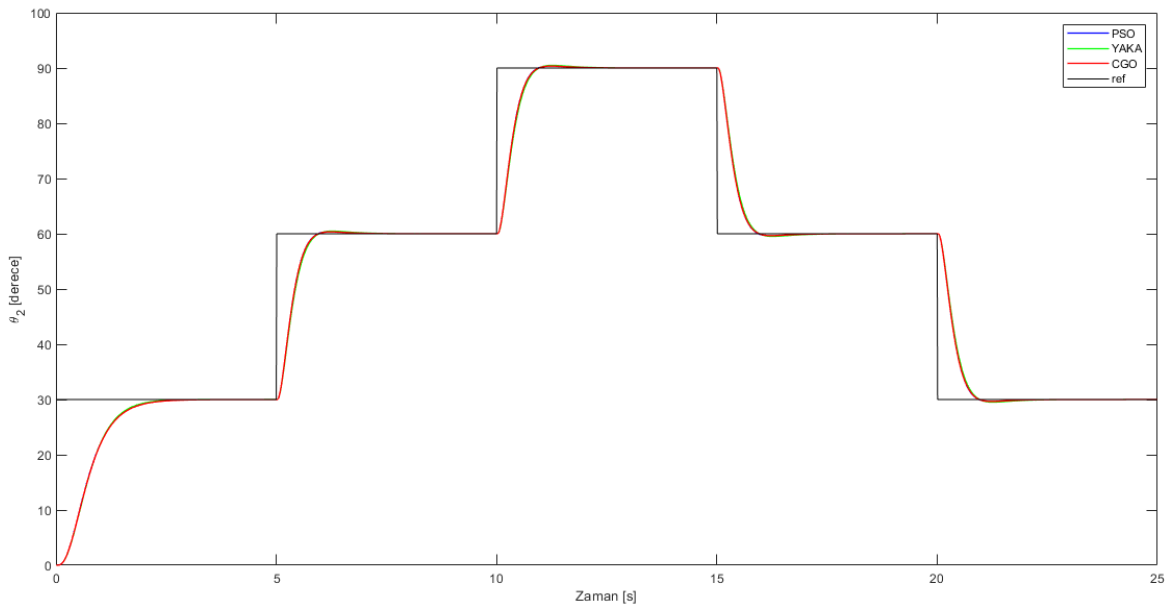
Şekil 4.4. θ_1 için girilen referans değerinde PSO, YAKA, CGO yakınsama işlemi

Çizelge 4.2’de θ_2 kontrolü için her bir meta sezgisel optimizasyon algoritması tarafından elde edilen DC motorun PID parametreleri k_p , k_i , k_d ayrı ayrı belirtilmiştir.

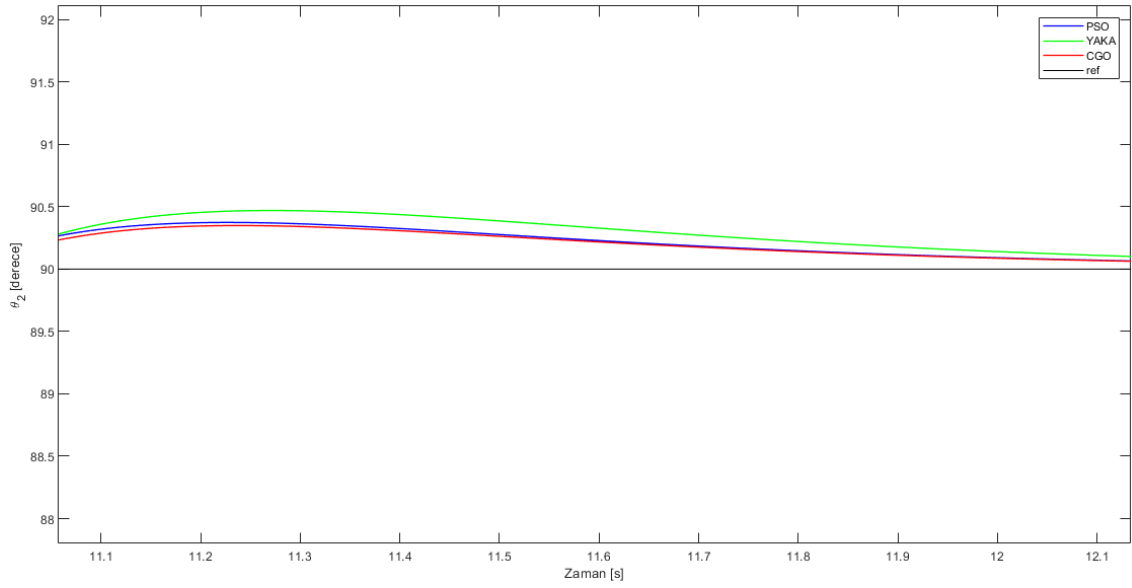
Çizelge 4.2. θ_2 değeri için PSO, YAKA ve CGO algoritmalarından bulunan PID parametreleri

Kontrolcü parametresi	PSO	YAKA	CGO
k_p	29.41	28.69	29.28
k_i	1.24	1.24	1.23
k_d	13.49	12.82	13.48

İki uzuvlu robot kolunun θ_2 değerlerinin simülasyon kontrolü zaman ve derece grafiği Şekil 4.5'te verilmiştir.

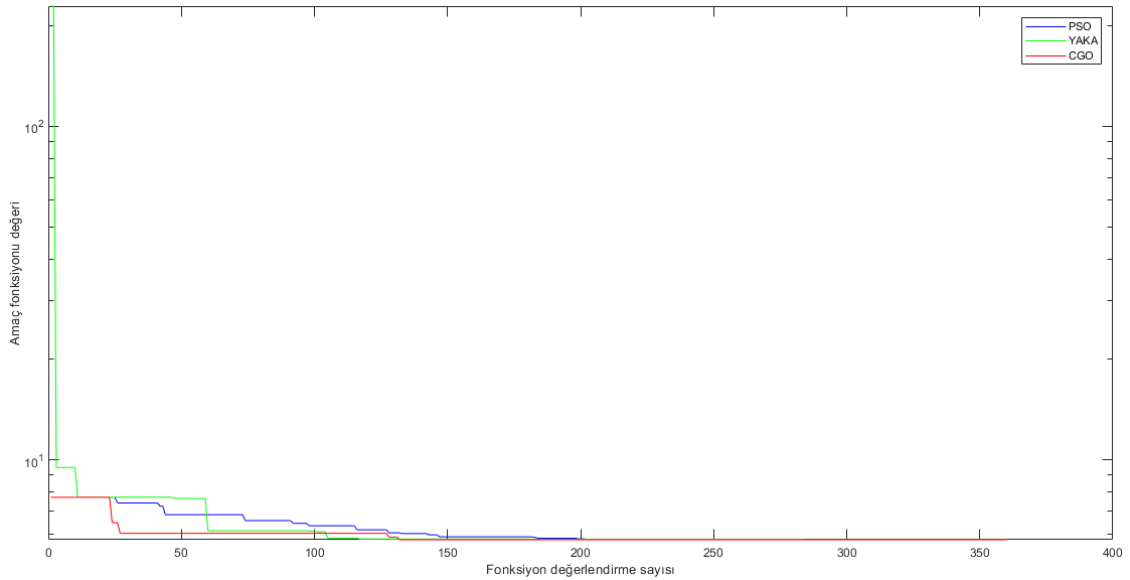
**Şekil 4.5.** θ_2 için girilen referans değerine karşılık PSO, YAKA, CGO karşılaştırması

Şekil 4.5'te θ_1 'de optimizasyon algoritmalarının iç içe olması durumunun aynı θ_2 'de de görülmüştür. Sisteme girilen sırasıyla 30° , 60° , 90° , 60° , 30° açı değerleri için PSO, YAKA ve CGO algoritmaları iç içe olduğu görülmektedir. Şekil 4.6'da her bir optimizasyon parametresinin grafikte 90° 'ye yakınsadığı görülmektedir. Şekil 4.6'da açığa ilk oturmada CGO ve PSO'nun YAKA'ya göre daha iyi olduğu, PSO ve YAKA'nın da birbirine çok yakın olduğu da görülmektedir.



Şekil 4.6. θ_2 için girilen referansın PSO, YAKA, CGO karşılaştırmasının detaylı görüntüsü

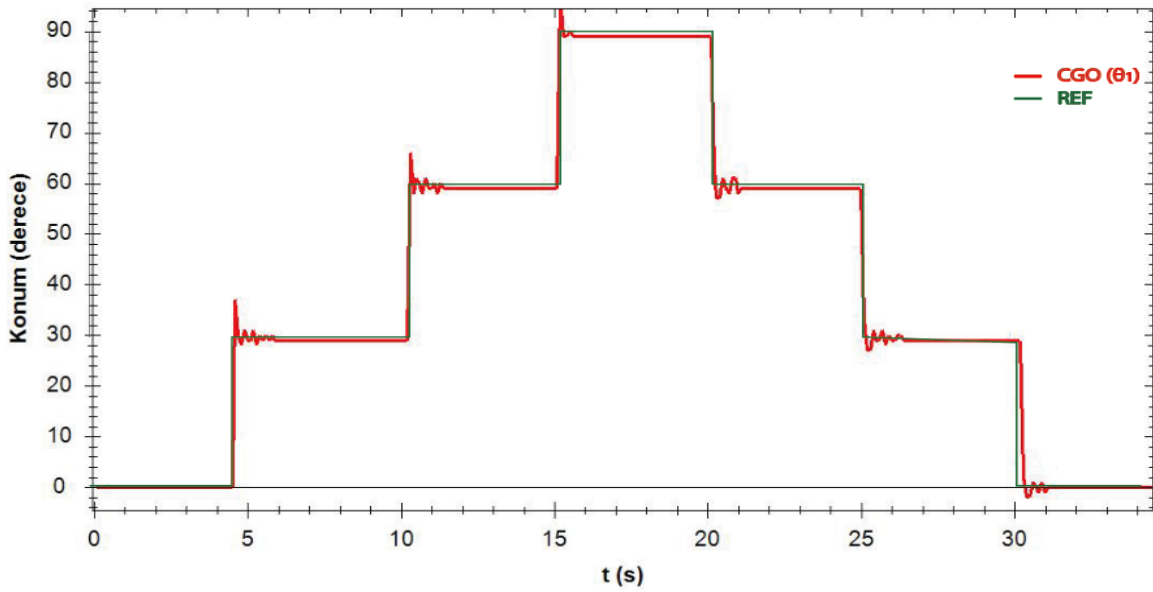
Şekil 4.7’de görüldüğü üzere CGO algoritmasının PSO ve YAKA algoritmalarına göre daha iyi yakınsadığı görülmektedir



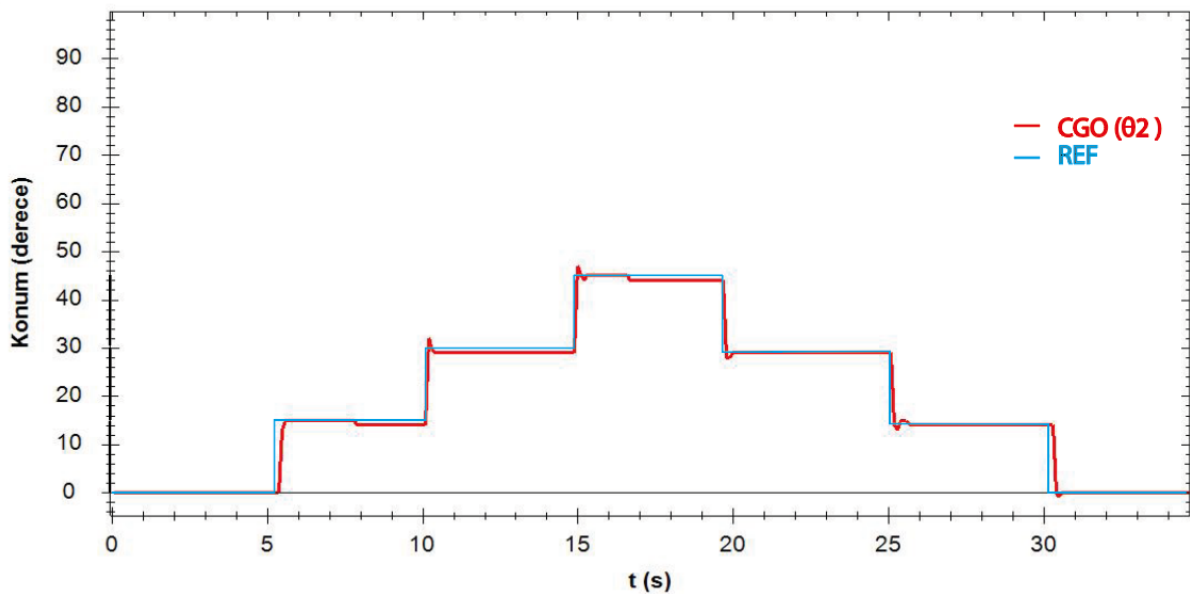
Şekil 4.7. θ_2 için girilen referans değerinde PSO, YAKA, CGO yakınsama işlemi

Yapılan optimizasyon işlemi sonucu birbirine çok yakın ancak ufak bir farkla CGO algoritmasının daha iyi olduğu söylenebilir. CGO, PSO ve YAKA metasezgisel algoritmalarından elde edilen PID parametreleri gerçek sistem üzerinde denemiştir. Mikro

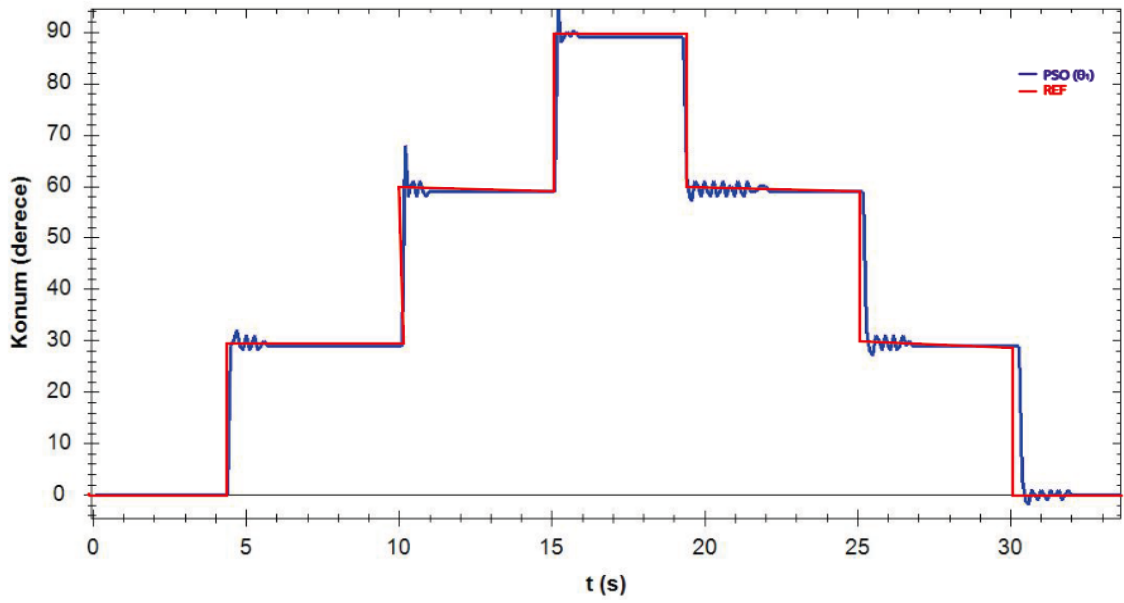
denetleyicinin seri portundan alınan enkoder verileri ile tasarlanan kontrolcü için metasezgisel optimizasyon algoritmalarından elde edilen parametreler ile belirlenen referans değerine hareket etmektedir. Belirlenen referans değeri ise her bir motor için beş saniyelik aralıklarla sırasıyla 30° , 60° , 90° , 60° , 30° değerleridir. CGO parametreleri ile bulunan θ_1 ve θ_2 'nin referans açısı takibi sırasıyla Şekil 4.8 ve Şekil 4.9'da verilmiştir. PSO parametreleri ile bulunan θ_1 ve θ_2 'nin referansı takibi sırasıyla Şekil 4.10 ve Şekil 4.11'de verilmiştir. YAKA parametreleri ile bulunan θ_1 ve θ_2 'nin referansı takibi sırasıyla Şekil 4.12 ve Şekil 4.13'de verilmiştir.



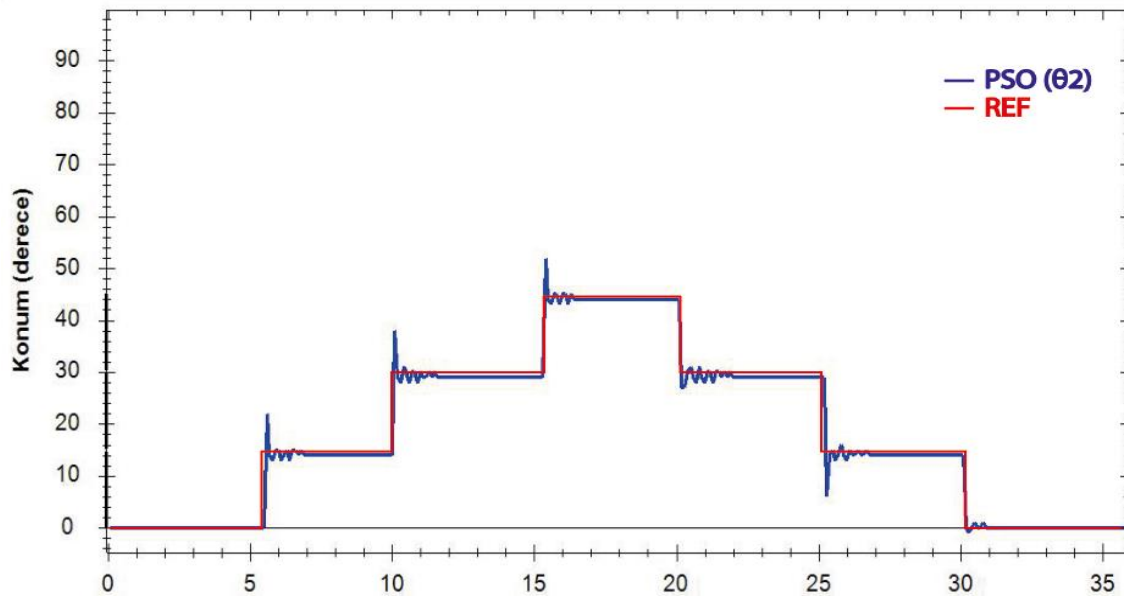
Şekil 4.8. θ_1 için CGO ile elde edilen PID parametrelerinin gerçekleştirilmesi



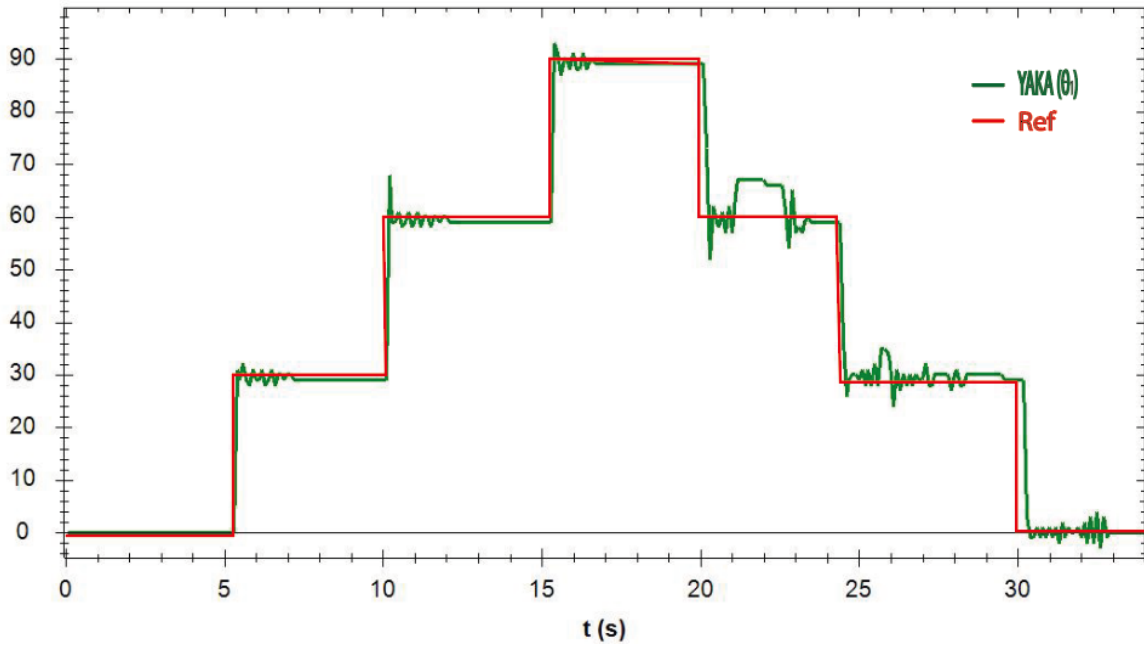
Şekil 4.9. θ_2 için CGO ile elde edilen PID parametrelerinin gerçekleştirilmesi



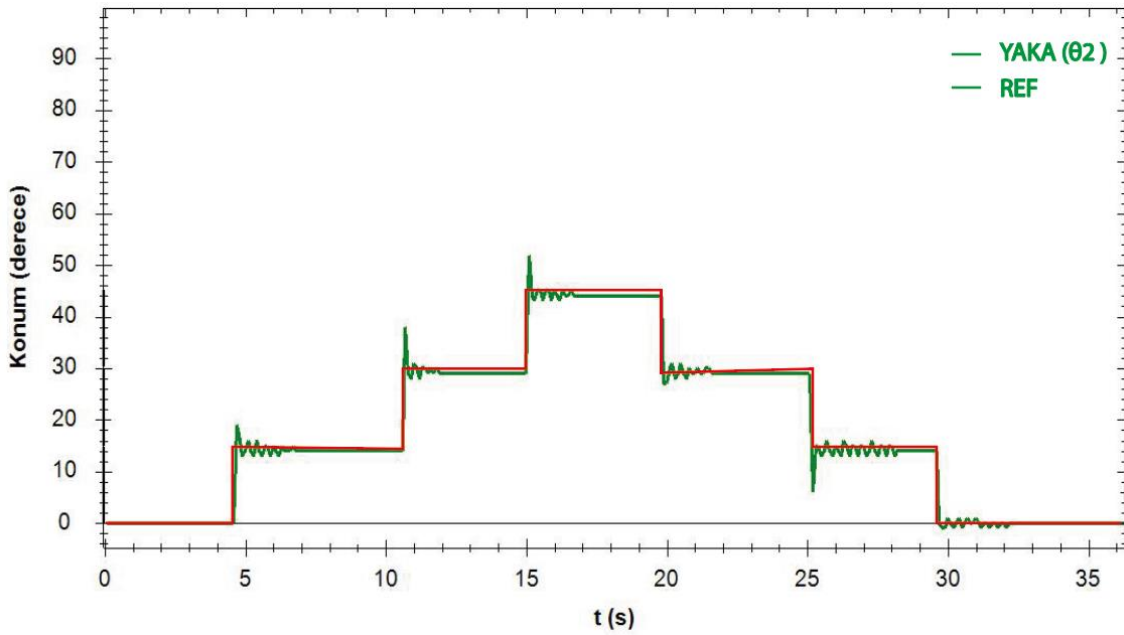
Şekil 4.10. θ_1 için PSO ile elde edilen PID parametrelerinin gerçekleştirilmesi



Şekil 4.11. θ_2 için PSO ile elde edilen PID parametrelerinin gerçekleştirilmesi



Şekil 4.12. θ_1 için YAKA ile elde edilen PID parametrelerinin gerçekleştirilmesi



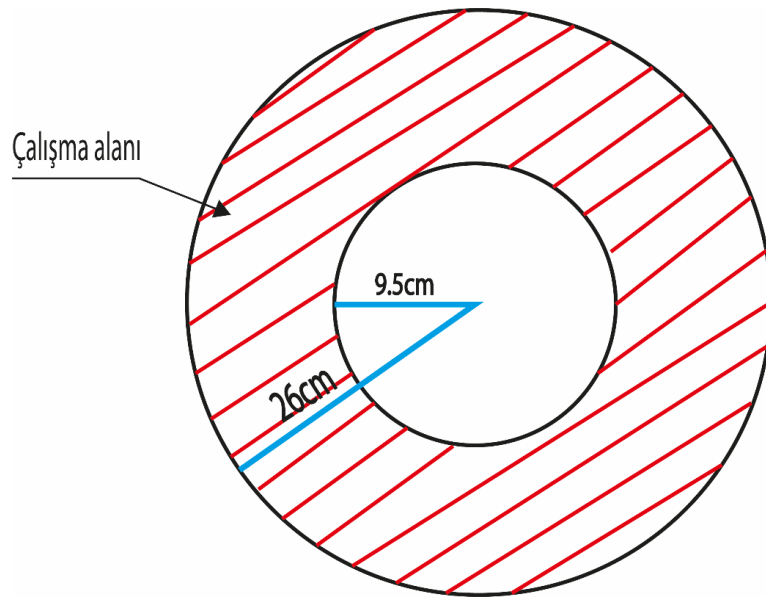
Şekil 4.13. θ_2 için YAKA ile elde edilen PID parametrelerinin gerçekleştirilmesi

Gerçek zamanlı olarak elde edilen Şekil 4.8, Şekil 4.9, Şekil 4.10, Şekil 4.11, Şekil 4.12, Şekil 4.13 şekillerinden benzetim çalışmasında bulunan sonuç ile yakın olduğu ancak farklılıklar olduğu sonucuna varılmıştır. Her bir optimizasyon algoritması için θ_2 değerinin θ_1 değerine göre benzetim çalışmasına daha yakın değer verdiği

gözlemlenmiştir. θ_2 değerleri içerisinde simülasyon çalışmasına göre ve istenilen en iyi değere CGO algoritmasından elde edilen parametreler vermiştir.

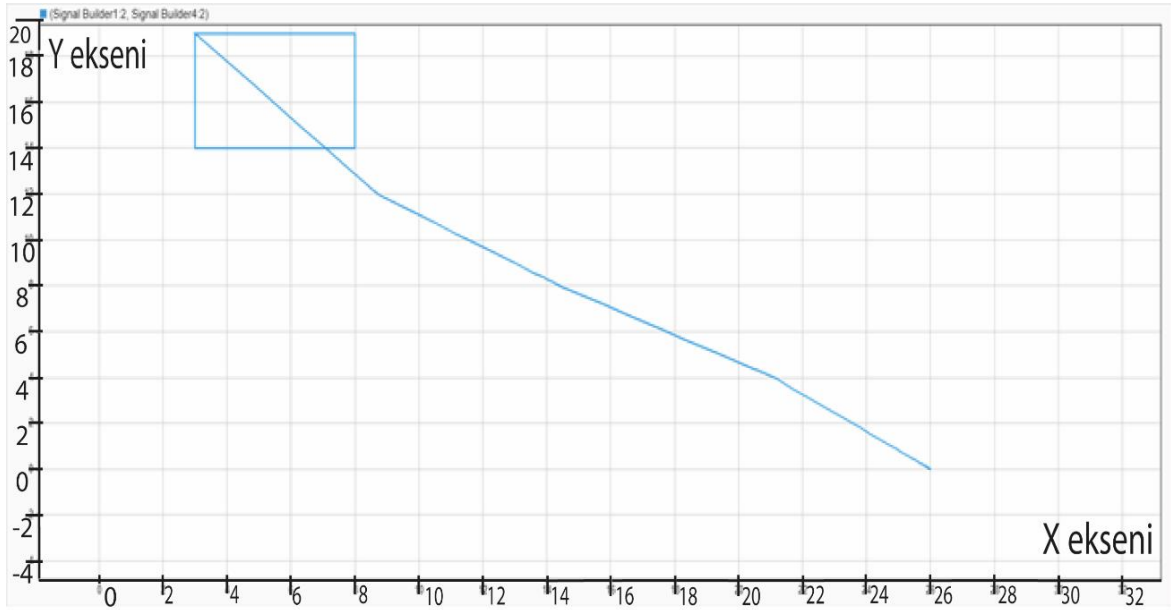
Robot kollarına belirlenen referans hareketi yaptırabilmek için tasarlanan kontrol sistemi hata oranı düşük olmalıdır. Robot kollarının doğru çalışmasını etkileyen diğer bir önemli faktör ise süre ile ilgili yapılan işlemlerdir. Robot kollarını kontrol edecek elektronik kartlar, hareket planlaması ve kontrol sistemi için gerekli matematiksel işlemlerden dolayı ek bir hata oluşturmaktadır. Bu çalışma içerisinde minimum hata ile çalışan robot kolu elde etmek hedeflendiği için bilgisayar ile Arduino Mega arasında sürekli bir iletişim kurulmuştur. EK-2 kısmında verilen blok diyagramı Matlab Simulink ortamındaki “MATLAB® Support Package for Arduino® and Simulink” ile geliştirilmiştir.

Herhangi bir dc motora bağlanmış robot kolu ve robot kolunun uç kısmına bağlanmış bir kalem ile yalnızca robot kolu uzunluğu yarıçapı olan çemberi çizebiliriz. İki uzuvlu bir robot koluna ise L_2-L_1 ve $L_2 + L_1$ yarıçaplarındaki çemberin arasında kalan herhangi bir bölge için çizim yapılabilir. Bu durum çalışma alanı ile ifade edilir. Bu tez çalışması kapsamında elde edilen robot kolu ile L_1 kol uzunluğu 17,75 cm L_2 kol uzunluğu 8,25 cm olan çalışma alanı iç çember 9,5 cm, dış çember 26 cm arasında tutucu uç hareket edebilmektedir. Gerçekleştirilen iki uzuvlu robot kolunun çalışma alanı Şekil 4.14’ de verilmiştir.



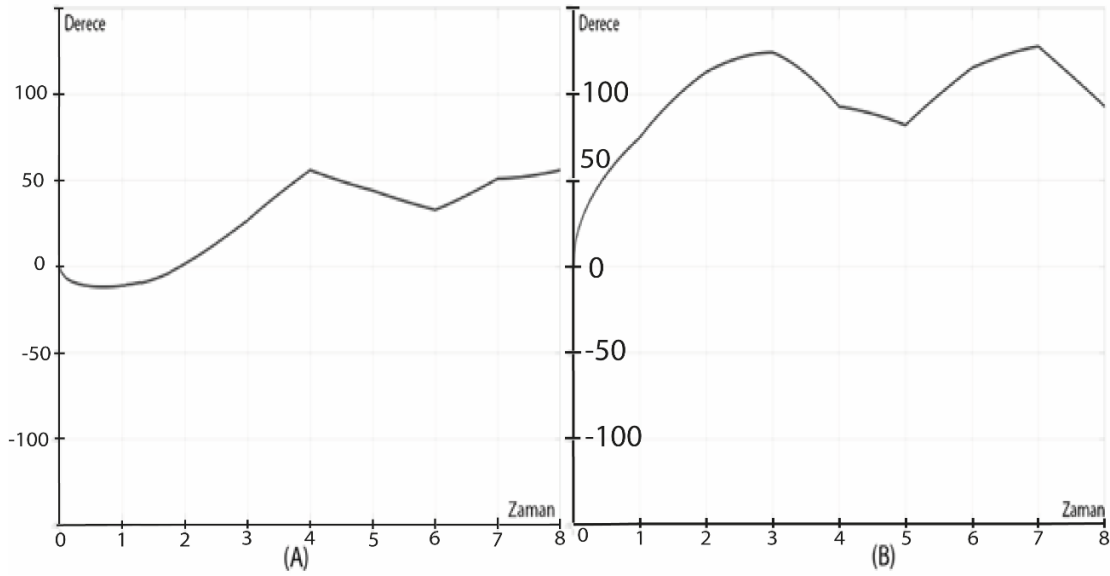
Şekil 4.14. İki uzuvlu robot kolunun çalışma alanı

Robot kollarında belirlenen çalışma alanı içerisinde kinematik işlemler ve tasarlanan kontrolcü ile hareket gerçekleştirilebilir. Bu tez çalışmasının son kısmında iki uzuvlu robot kolu ile belirlenen kare ve çember hareketi gerçekleştirilmiştir. İlk olarak kare hareket gerçekleştirilmiştir. Koordinat eksenleri üzerinde belirlenen kenar uzunluğu 5 cm olan referans çizim üzerinden takip gerçekleştirilmiştir. Şekil 4.15’ de gerçekleştirilen kare şeklindeki yörünge hareketinin referansı belirtilmiştir.



Şekil 4.15. Robot koluna yaptırılan kare yörünge hareketi

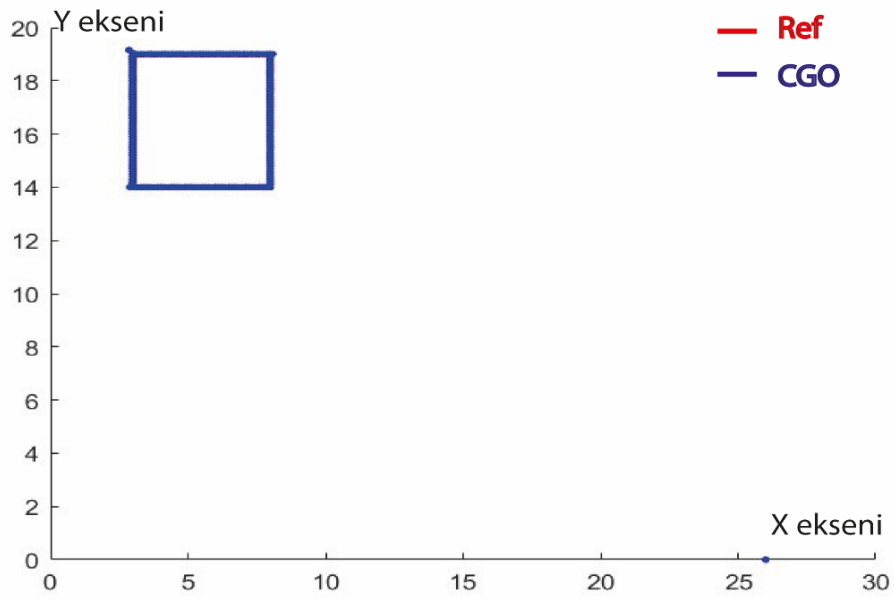
Şekil 4.15’ de elde edilen yörüngeyi gerçekleştirebilmek için her bir uzva etki eden motorların gerekli açı değişimlerine birlikte başlamaları ve yapacakları hareketleri gerçekleştirirken süre kaybı olmaması gerekmektedir. Matlab’da Simulink ortamında oluşturulan sistemde Ters kinematik bloğuna giren x ve y eksenleri konum bilgileri, açı değerine dönüşmektedir. Kare hareketi gerçekleştirebilmek için gerekli zaman sekiz saniye olarak denenmiştir. İlk dört saniye robot kolunun x ekseninde üç, y ekseninde 19 noktasına yönlendirilmesi için ayrılmıştır. Dört saniyeden sonra ise her bir saniyede koordinat eksenine noktasal olarak ifade edilirse sırasıyla (8,19) , (8,14) , (3,14) ve son olarak (3,19)’a kare için başlangıç konumuna gelip kareyi oluşturmaktadır. Robot kolunun bu kare şeklindeki yörüngeyi takip edebilmesi için gerekli olan açı değişimleri Şekil 4.16’da belirtilmiştir. Şekil 4.16’da θ_1 ‘deki değişim (A) ve uç konuma yakın olan motor için ise yani θ_2 deki değişim (B) gösterilmiştir.



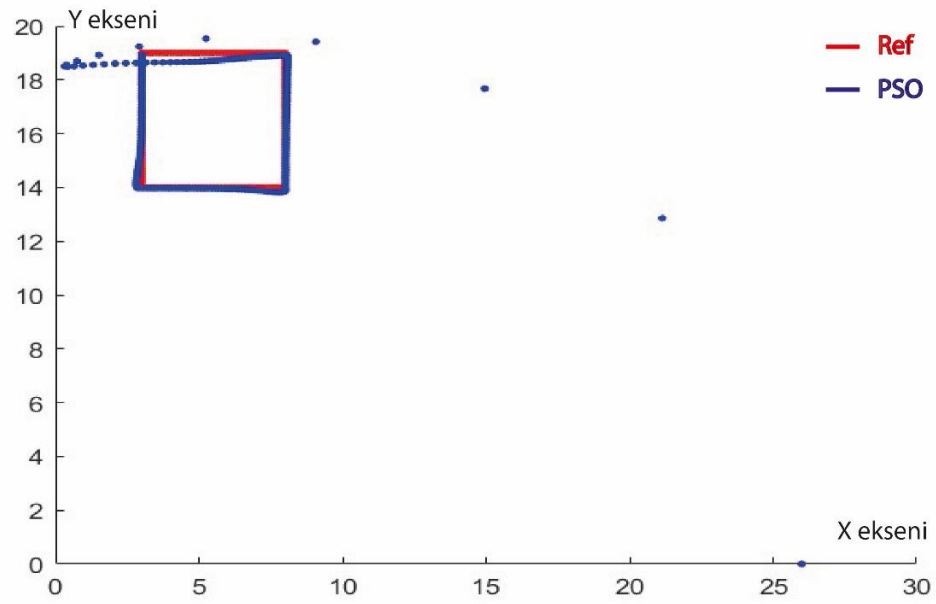
Şekil 4.16. Kare hareket için zamansal θ_1 ve θ_2 değişimi

Elde edilen referans θ_1 ve θ_2 değerleri robot koluna iletildiğinde enkoderden gelen örnekleme zamanı boyunca konum verisi elde edilmektedir. Tasarlanan PID kontrolcü ile minimum hata değerli yeni θ_1 ve θ_2 değeri üretilmektedir. PID kontrolcü içerisinde yer alan parametreler, Kaos Oyun, Parçacık Sürü ve Yapay Arı optimizasyon algoritması ile bulunmuştur.

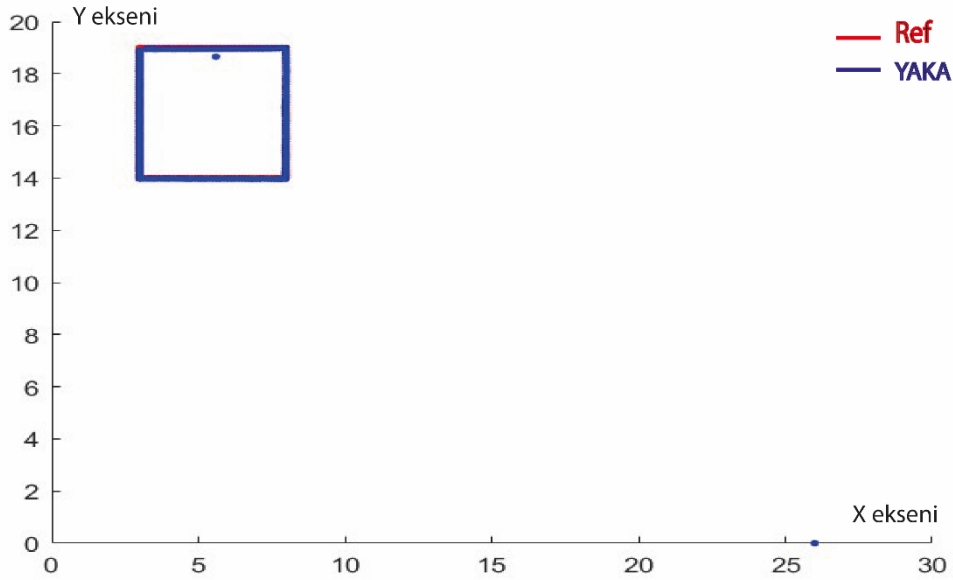
Kare hareket için yapılan bu optimizasyon işleminde maksimum fonksiyon değerlendirme sayısı 40, popülasyon sayısı 50 dir. PID parametrelerini sıraya koyarsak k_{p_1} , k_{i_1} , k_{d_1} , k_{p_2} , k_{i_2} , k_{d_2} alt limit [0 0 0 0 0 0], üst limit ise [20.000 20.000 18.000 20.000 20.000 18.000] olarak gerçekleştirilmiştir. Elde edilen sonuçlar görsel olarak CGO için Şekil 4.17, PSO için Şekil 4.18, YAKA için Şekil 4.19’ da verilmiştir.



Şekil 4.17. Kare hareket için gerçekleştirilen CGO referans takibi

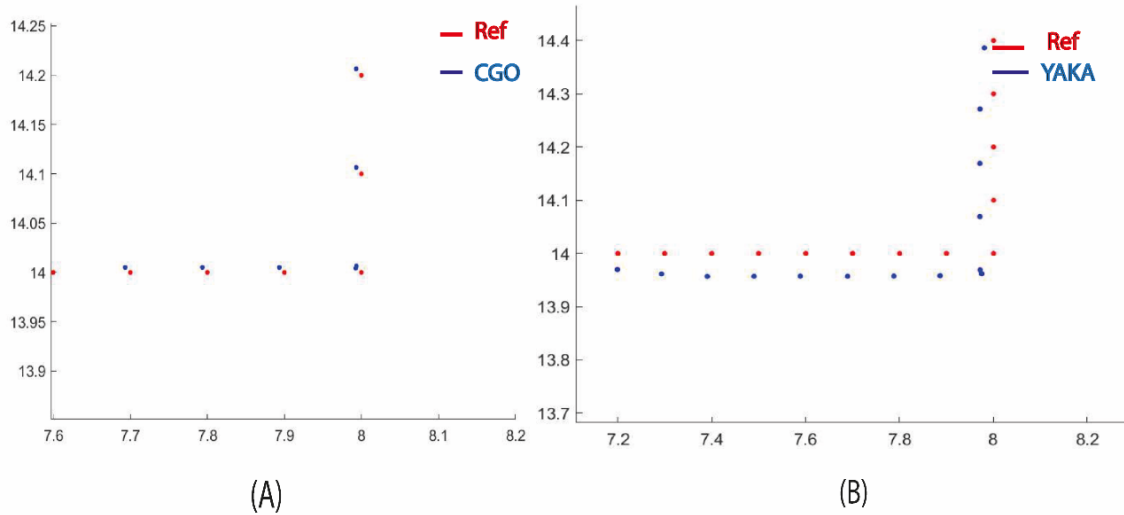


Şekil 4.18. Kare hareket için gerçekleştirilen PSO referans takibi



Şekil 4.19. Kare hareket için gerçekleştirilen PSO referans takibi

Kaos Oyun ve Yapay Arı Koloni Optimizasyon Algoritmalarının Parçacık Sürü optimizasyon algoritmalarına göre daha iyi sonuç verdiği Şekil 4.17 Şekil 4.18 Şekil 4.19 da açıkça görülmektedir. Yapay arı ve Kaos Oyun ise birbirine yakın bir sonuç vermiştir. Kaos oyun optimizasyon algoritması yapay arı koloni optimizasyon algoritmasına göre ufak bir farkla daha iyi sonuç verdiği Şekil 4.20.'de görülmektedir. Şekil 4.20.'de (A) kısmında kaos oyun optimizasyon algoritmasının (B) kısmında ise yapay arı koloni optimizasyon algoritmasının kareyi oluşturan (8,14) noktasındaki yakından görüntüsüdür. Referans değer olan kırmızı noktalara CGO, YAKA 'ya göre daha yakın konumdadır.



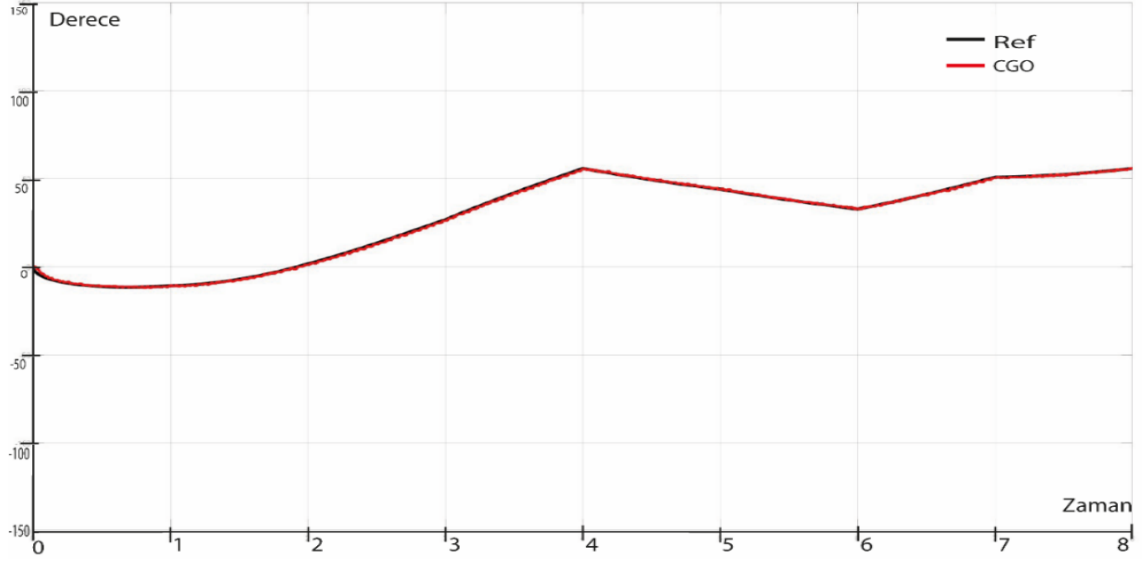
Şekil 4.20. Kare hareket için CGO, YAKA karşılaştırması

Yapılan optimizasyon işlemi sonucunda her bir optimizasyon algoritması için k_{p_1} , k_{i_1} , k_{d_1} , k_{p_2} , k_{i_2} , k_{d_2} olmak üzere altı adet optimizasyon parametreleri elde edilmektedir. Bulunan bu parametreler EK-2 kısmında bulunan PID bloklarına yazılmaktadır. Bulunan PID parametreleri Çizelge 4.3.'de verilmiştir.

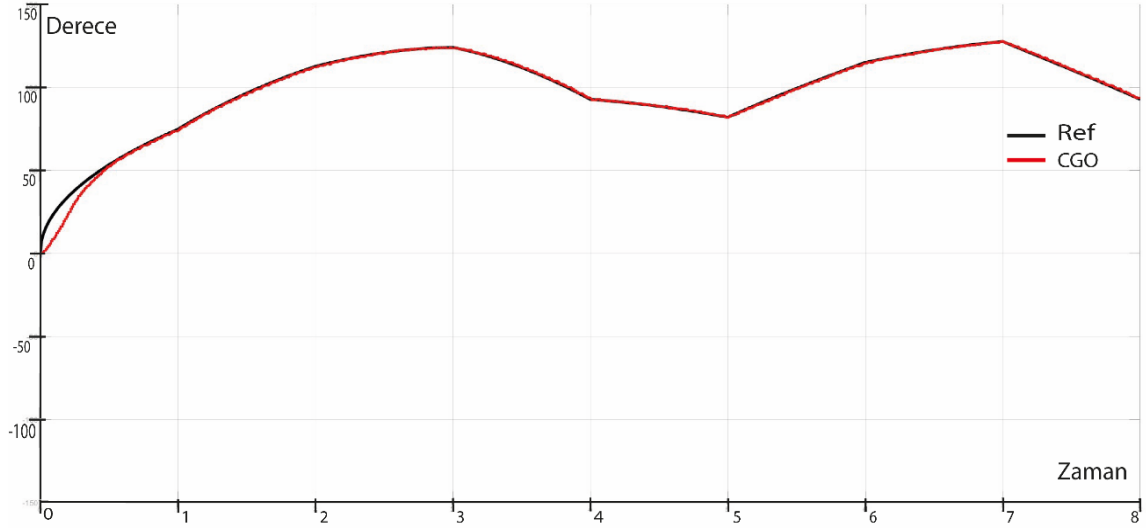
Çizelge 4.3. Kare hareketi için bulunan PID parametreleri

Kontrolcü parametresi		CGO	PSO	YAKA
PID θ_1	k_{p_1}	19853	15.016	17667
	k_{i_1}	799	17675	0
	k_{d_1}	1082	3446	0
PID θ_2	k_{p_2}	19775	19374	20000
	k_{i_2}	0	9252	1835
	k_{d_2}	1118	3776	1157

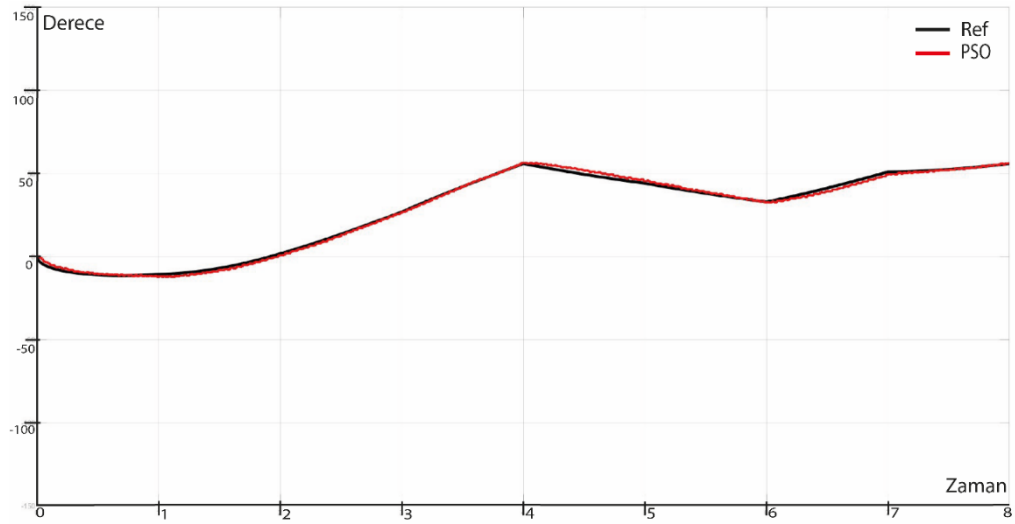
PID parametrelerini belirleyen CGO, PSO, YAKA her bir optimizasyon algoritması gerçek sistem üzerinde ayrı ayrı denenmiştir. Üç optimizasyon algoritması da referansa yakın değerler vermiştir. Kaos oyun optimizasyon algoritması için θ_1 i şekil 4.21 ve θ_2 için şekil 4.22 'de, Parçacık sürü optimizasyon algoritması için θ_1 i şekil 4.23 ve θ_2 için şekil 4.24, Yapay arı koloni algoritması için θ_1 i şekil 4.25 ve θ_2 için şekil 4.26, olarak belirtilmiştir.



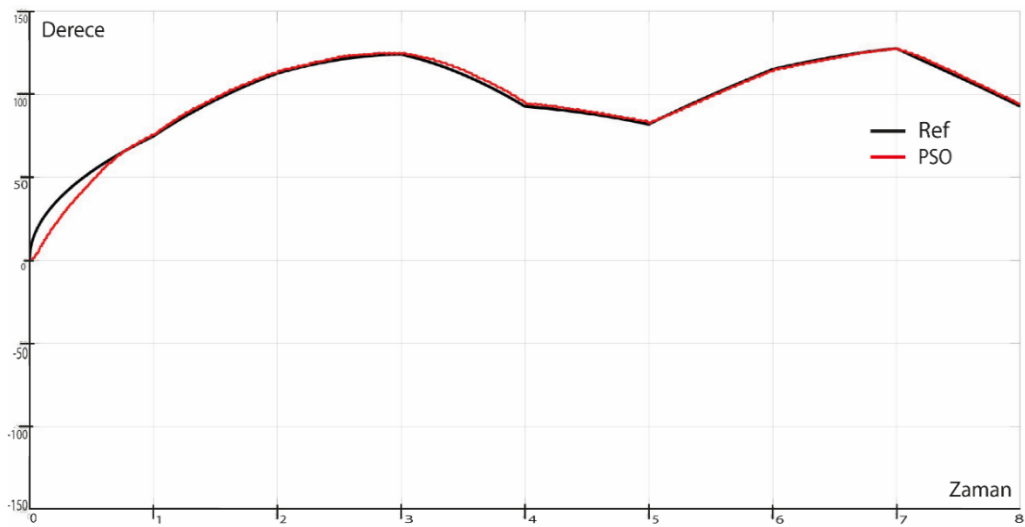
Şekil 4.21. Kare yörünge için CGO sayesinde elde edilen PID parametrelerinin θ_1 açısındaki referans takibi



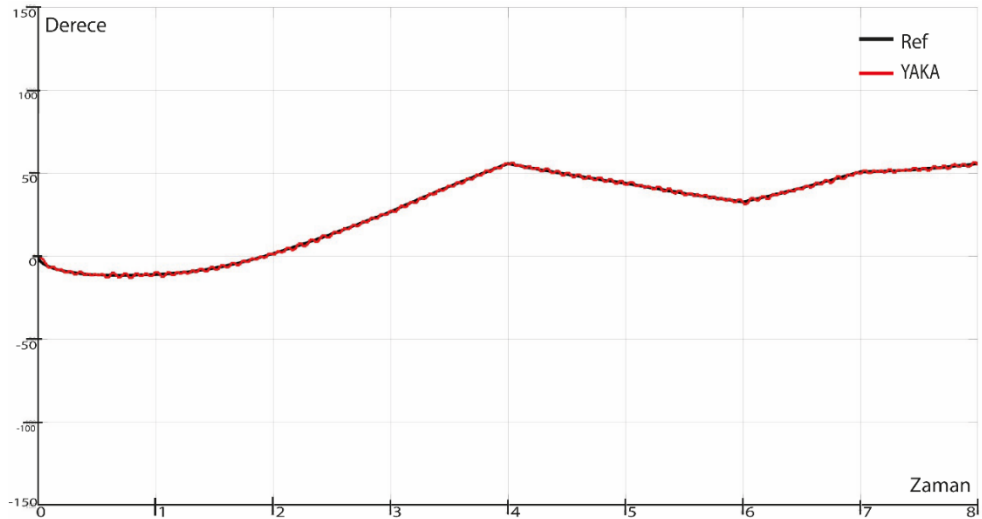
Şekil 4.22. Kare yörünge için CGO sayesinde elde edilen PID parametrelerinin θ_2 açısındaki referans takibi



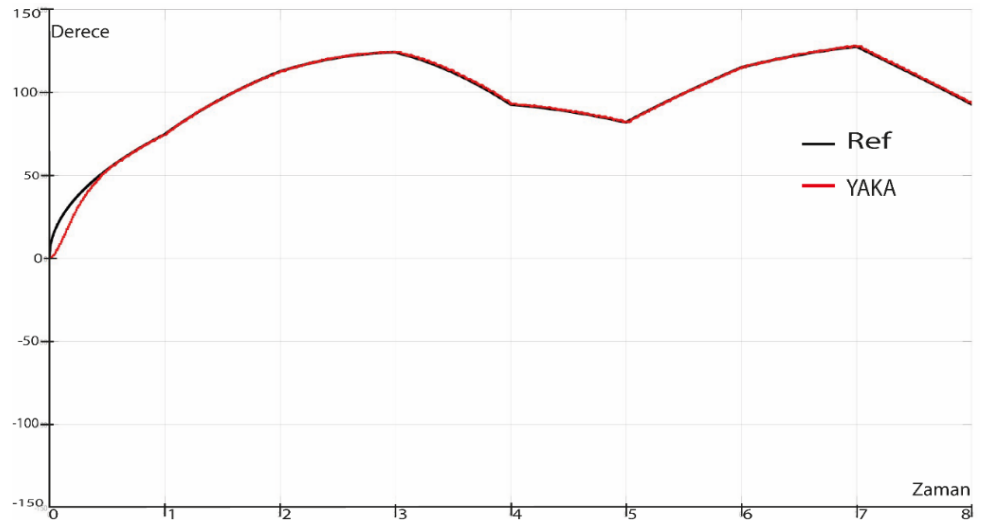
Şekil 4.23. Kare yörünge için PSO sayesinde elde edilen PID parametrelerinin θ_1 açısındaki referans takibi



Şekil 4.24. Kare yörünge için PSO sayesinde elde edilen PID parametrelerinin θ_2 açısındaki referans takibi



Şekil 4.25. Kare yörünge için YAKA sayesinde elde edilen PID parametrelerinin θ_1 açısındaki referans takibi

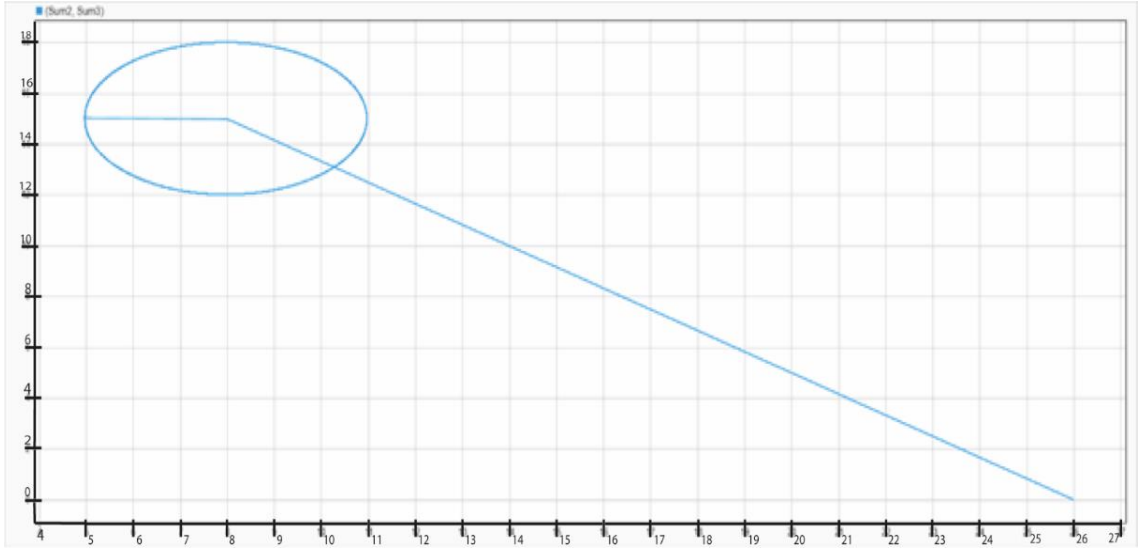


Şekil 4.26. Kare yörünge için YAKA sayesinde elde edilen PID parametrelerinin θ_2 açısındaki referans takibi

Kare hareket için gerekli kontrolcü parametreleri optimizasyon algoritmaları sayesinde bulunmuş olup optimizasyon işlemi sonucu birbirine yakın ancak ufak bir farkla CGO algoritmasının daha iyi sonuç verdiği görülmüştü. Gerçek sistem üzerinde yine her bir algoritmanın referansa yakın sonuç verdiği görülmektedir. Üç optimizasyon algoritması içinde θ_2 değerinin θ_1 'e göre özellikle ilk başlangıç anında biraz yavaş kaldığı görülmektedir. Bu durumun sebebi yazılımsal olarak olarak kesme pinlerinin öncelik sırası olduğu düşünülmektedir. Şekil 4.21, Şekil 4.22, Şekil 4.23, Şekil 4.24, Şekil 4.25, Şekil 4.26, incelendiğinde θ_1 ve θ_2 için CGO, PSO ve YAKA dan ufak farklarla daha iyi sonuç verdiği görülmektedir.

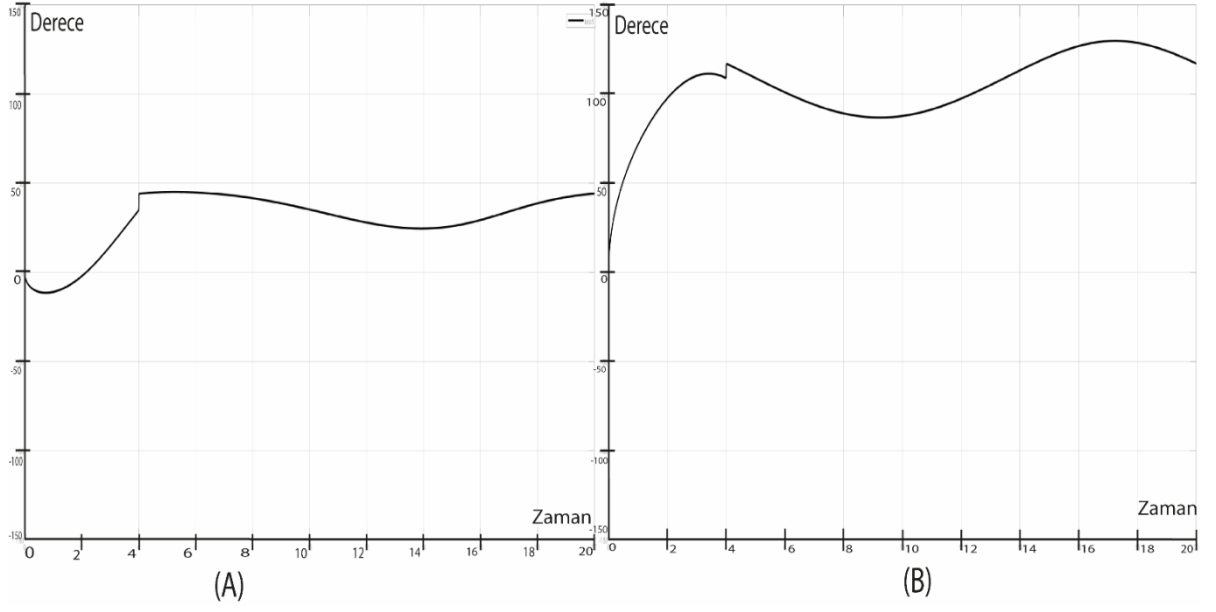
Robot kollarının uç noktasının herhangi bir geometrik şekli takip etmesini istersek en çok zorlanacağı kısım keskin köşelerin olduğu geometrik şekillerdir. Kare, üçgen, altıgen gibi geometrik şekiller keskin köşeli iken daire herhangi bir köşe noktasına sahip değildir. Robotların keskin köşede zorlanmasının sebebi ani açı değişimidir. Bu tez çalışması kapsamında yapılan diğer bir yörünge takibi ise dairesel harekettir. EK-3 kısmında belirtilen blok diyagramı sayesinde 2 uzuvlu robot koluna belirlenen referans dairesel yörünge hareketi gerçekleştirilmiştir.

Robot kolu, koordinat eksenleri olarak (26,0) konumunda iken yani θ_1 ve θ_2 açı değerlerinin 0 derece olduğu konumdan başlayıp belirlenen dairenin merkez noktasına gitmektedir. Dairenin merkez noktası (8,15) noktasındadır. Sonrasında ise daireyi oluşturan (5,15) noktasından daireyi çizmeye başlayıp (11,15) noktasından geçip yarıçapı 3 cm olan ve yine (5,15) noktasına gelerek daire hareketini gerçekleştirmektedir. Robot koluna yaptırılan ikinci yörünge hareketi Şekil 4.27.'de verilmiştir.



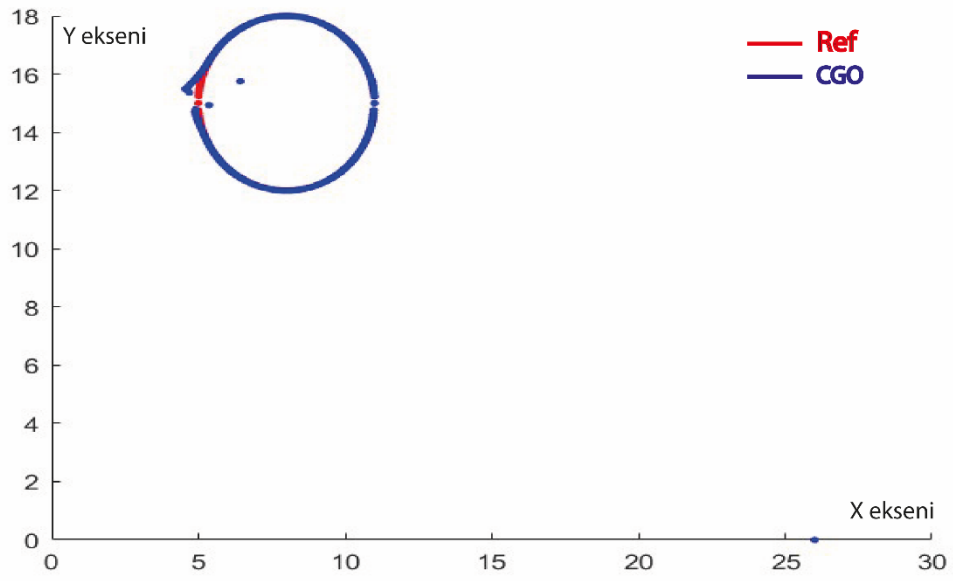
Şekil 4.27. Robot koluna yaptırılan daire yörünge hareketi

Daire hareketini gerçekleştirmek için ilk 4 saniyelik zamanda, daireyi oluşturan (5,15) noktasına gitmesi içindir. Daireyi oluşturan (5,15) noktasından başlayıp yine aynı noktaya gitmesi ise 16 saniye boyuncadır. Daire hareketi için toplamda 20 saniye robot kolu hareket etmektedir. Bu dairesel yörüngeyi gerçekleştirmek için robot kollarında oluşan θ_1 ve θ_2 değişimleri Şekil 4.28'de verilmiştir. Şekil 4.28'de (A) θ_1 'in 20 saniye boyunca olan değişimidir. Şekil 4.28' de (B) ise θ_1 'in 20 saniye boyunca olan değişimidir.

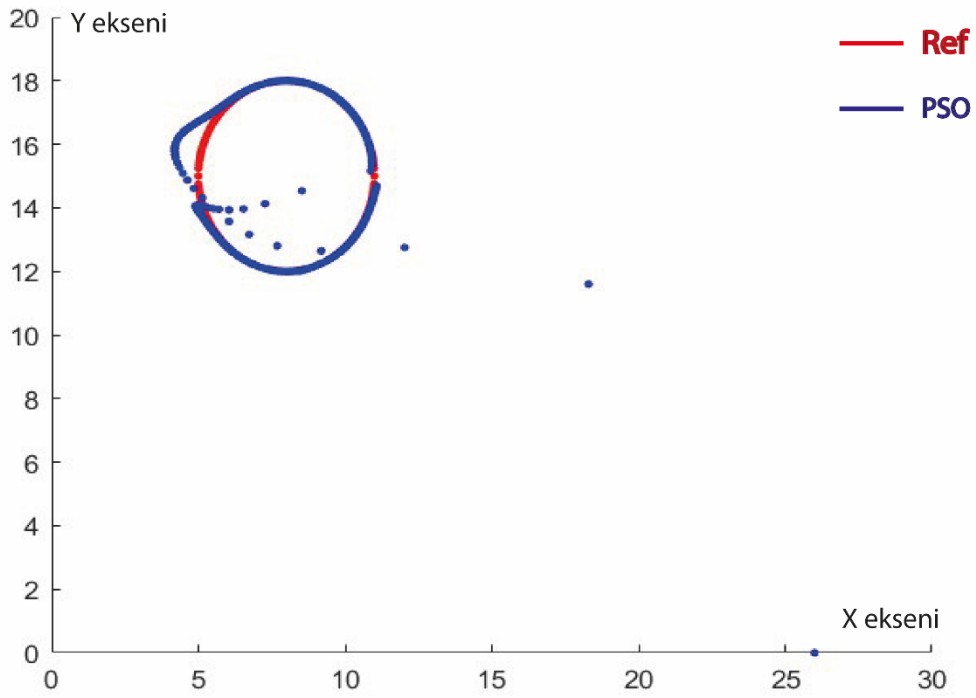


Şekil 4.28. Daire hareket için zamansal θ_1 ve θ_2 değişimi

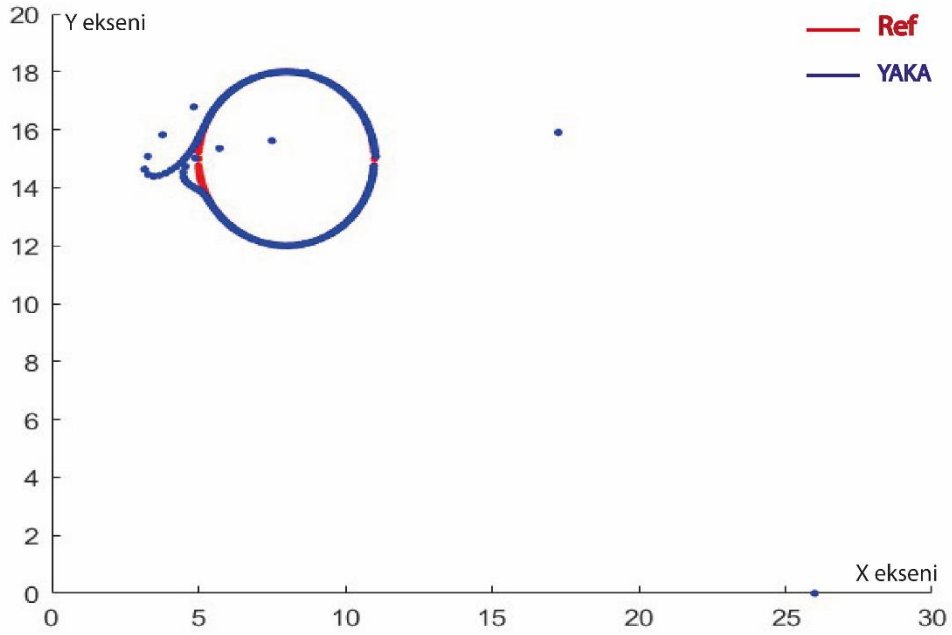
Dairesel yörünge hareketi için PID parametrelerini belirlemede yine Kaos oyun, Parçacık sürü, Yapay arı koloni algoritması kullanılmıştır. Maksimum fonksiyon değerlendirme sayısı 10, popülasyon sayısı 50'dir. PID parametrelerini sıraya koyarsak k_{p_1} , k_{i_1} , k_{d_1} , k_{p_2} , k_{i_2} , k_{d_2} için alt limit [0 0 0 0 0 0] üst sınır için ise [35.000 35.000 25.000 35.000 35.000 25.000] olarak gerçekleştirilmiştir. Elde edilen sonuçlar görsel olarak CGO için Şekil 4.29, PSO için Şekil 4.30, YAKA için Şekil 4.31' de verilmiştir.



Şekil 4.30. Daire hareketi için gerçekleştirilen CGO referans takibi



Şekil 4.31. Daire hareketi için gerçekleştirilen PSO referans takibi



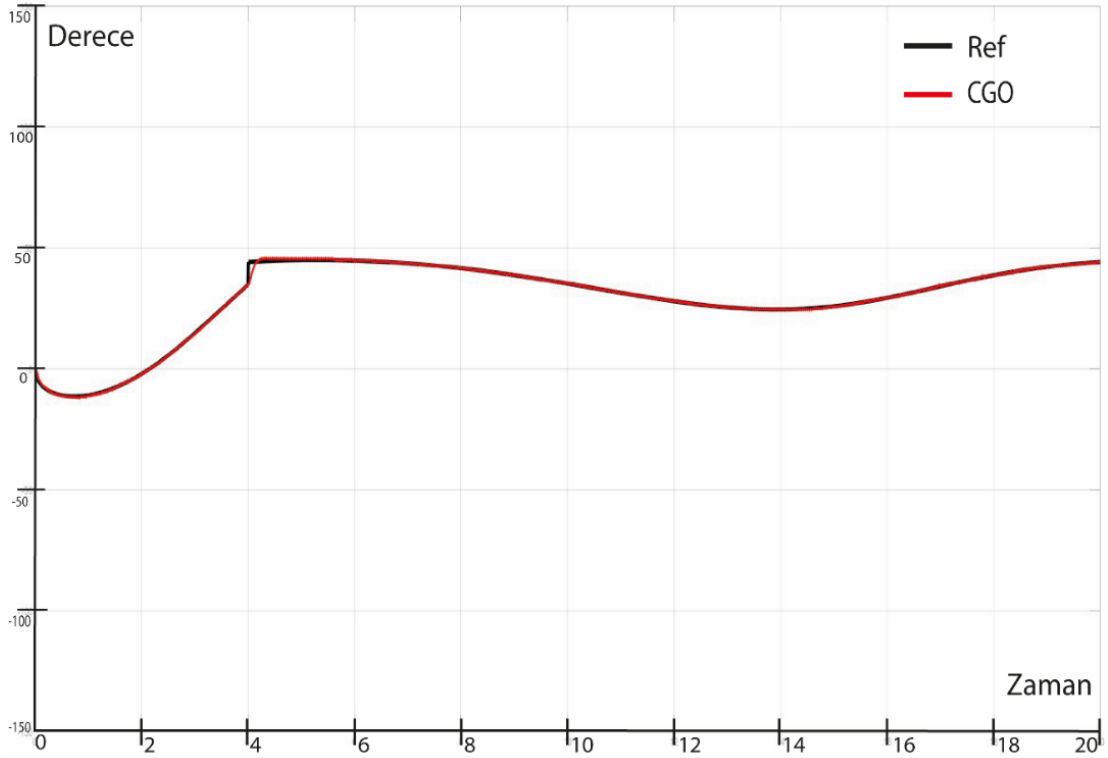
Şekil 4.32. Daire hareketi için gerçekleştirilen YAKA referans takibi

Yapılan optimizasyon işlemi sonucunda Şekil 4.29 Şekil 4.30 Şekil 4.31’de görüldüğü üzere kırmızı renkli olan referans değerini, mavi olarak renklendirilen optimizasyon işlemi sonucunu belirtmektedir. Bu şekillerden görüldüğü üzere daire hareketini her üç algoritma da gerçekleştirebilmektedir. Takip edebilme açısından en iyi olan optimizasyon algoritması Kaos oyun optimizasyon algoritması sonrasında yapay arı koloni algoritması sonrasında ise Parçacık sürü optimizasyon algoritmasıdır. Dairesel yörünge için yapılan optimizasyon işlemi sonucunda altı adet kontrolcü parametresi elde edilmiştir. Bulunan bu parametreler Çizelge 4.4.’de verilmiştir.

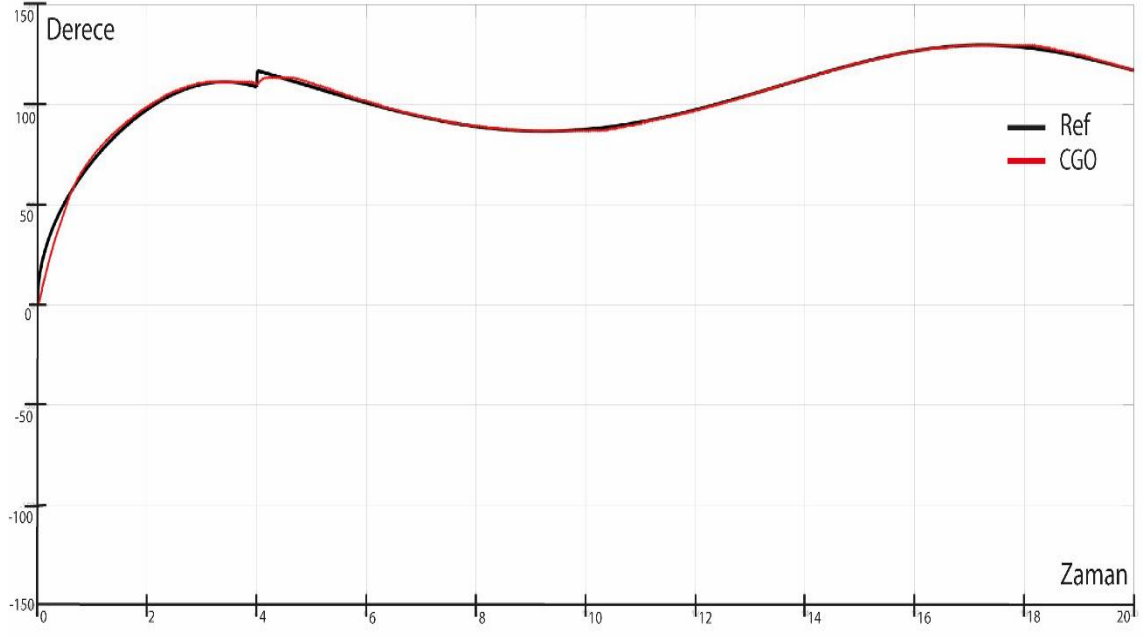
Çizelge 4.4. Daire hareketi için bulunan PID parametreleri

	Kontrolcü parametresi	CGO	PSO	YAKA
PID Teta1	k_{p1}	35000	27639	7699
	k_{i1}	35000	7556	13451
	k_{d1}	680	10080	417
PID Teta2	k_{p2}	35000	28084	33845
	k_{i2}	319	30175	28751
	k_{d2}	30	3594	6305

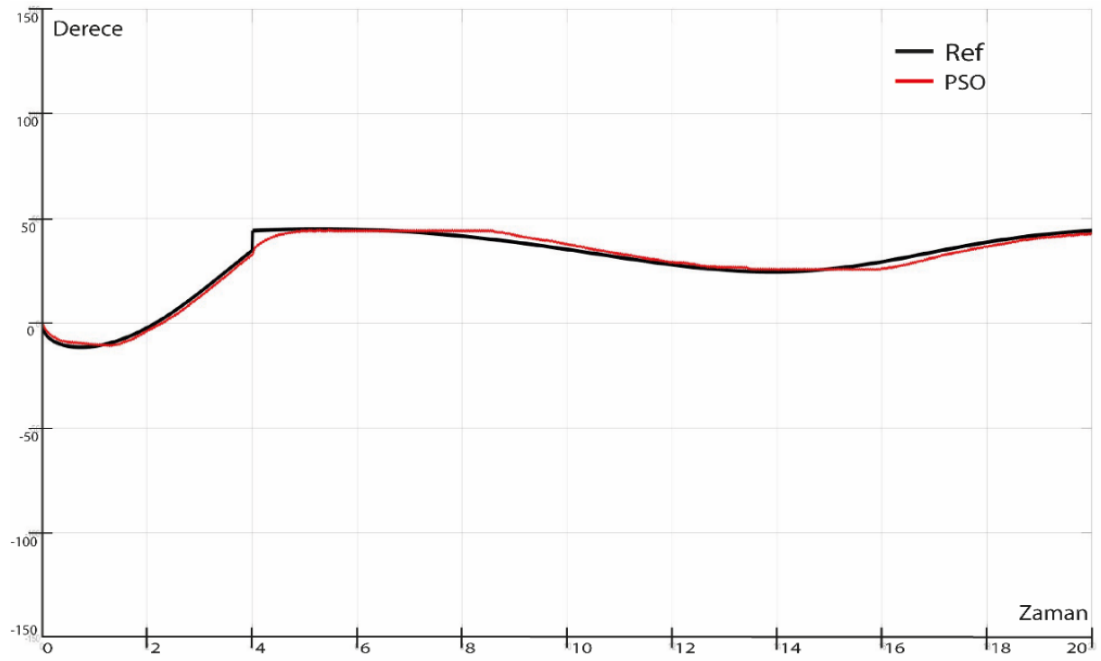
Daire hareketi için elde edilen PID parametreleri EK-3 kısmında tasarlanan blok diyagramı içerisinde kontrolcü bloğuna yerleştirilerek gerçekleştirilmiştir. CGO, PSO, YAKA algoritmaları sonucu elde edilen parametreler ayrı ayrı gerçek sistem üzerinde denenmiştir. Daire hareketi için belirlenen referans θ_1 ve θ_2 açı değerlerini CGO için elde edilen sonuçlar Şekil 4.32 Şekil 4.33 'de PSO için Şekil 4.34 Şekil 4.35, YAKA için Şekil 4.36 Şekil 4.37 de belirtilmiştir



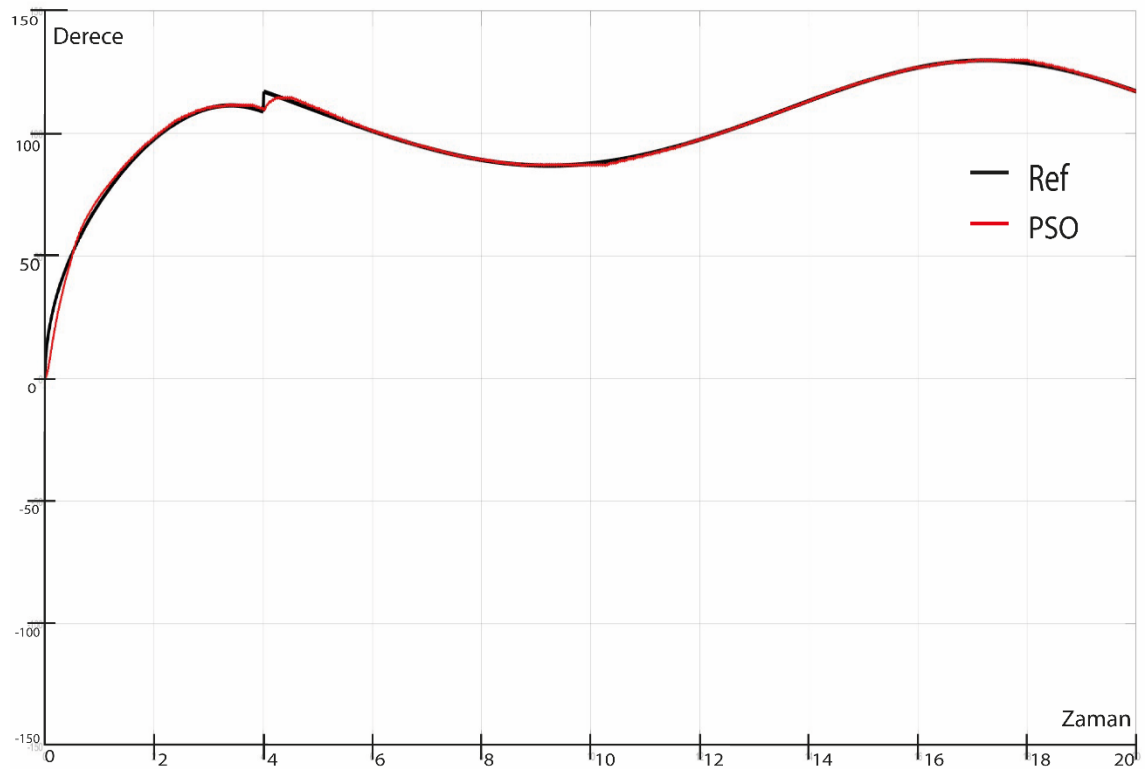
Şekil 4.33. Daire hareketi için CGO sayesinde elde edilen PID parametrelerinin θ_1 açısındaki referans takibi



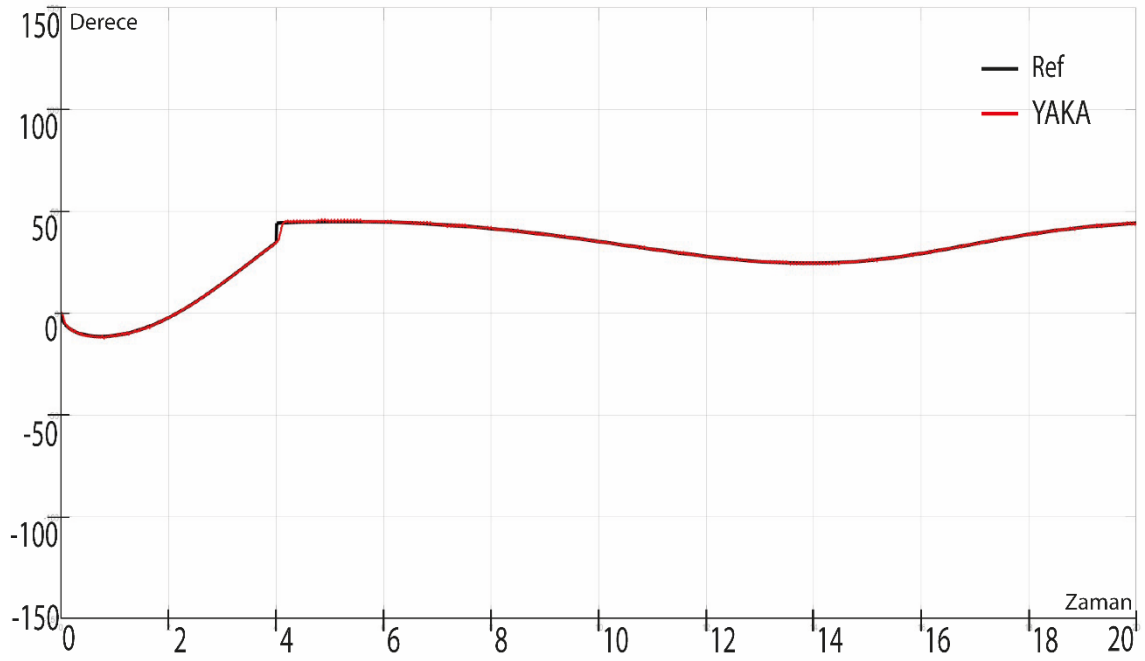
Şekil 4.34. Daire hareketi için CGO sayesinde elde edilen PID parametrelerinin θ_2 açısındaki referans takibi



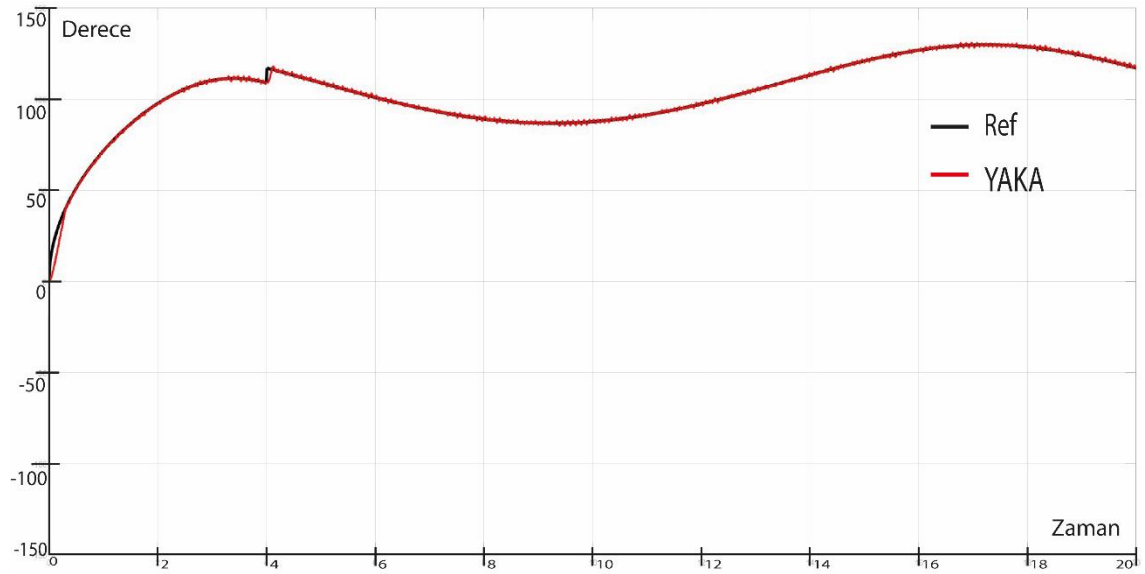
Şekil 4.35. Daire hareketi için PSO sayesinde elde edilen PID parametrelerinin θ_1 açısındaki referans takibi



Şekil 4.36. Daire hareketi için PSO sayesinde elde edilen PID parametrelerinin θ_2 açısındaki referans takibi



Şekil 4.37. Daire hareketi için YAKA sayesinde elde edilen PID parametrelerinin θ_1 açısındaki referans takibi



Şekil 4.38. Daire hareketi için YAKA sayesinde elde edilen PID parametrelerinin θ_2 açısındaki referans takibi

Bu çalışma içerisinde gerçekleştirilen iki uzuvlu robot kolu ile çalışma alanı içerisinde yer alan daire geometrik şeklini robot kolunun uç noktasıyla gerçekleştirilmiştir. Kare hareket kısmında elde edilen sonuca benzer şekilde daire hareketinde de her üç optimizasyon algoritması da takibi gerçekleştirmiştir. Aralarındaki küçük farklılıklarla sıralanacak olursa CGO, YAKA, PSO şeklinde sıralanabilir.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Robot kontrolü için yapılan literatürdeki birçok çalışma servo motor ya da step motorlar ile yapılmıştır. Bu çalışmada kullanılan dc motor oldukça başarılı bir sonuç vererek robot kontrolü için dc motorunda kullanılabilceğini göstermiştir. DC motorun transfer fonksiyonu katalog değerlerinden elde edilmiş olup kontrolcü geri bildirimini enkoder sağlamıştır.

Kontrolör olarak kullanılan PID için parametrelerini belirlemede mevcut birçok algoritma vardır. Bu çalışma ile birlikte PID parametrelerini belirlemede ilk defa CGO algoritması kullanılmış ve adil bir optimizasyonda PSO, YAKA algoritmalarına göre referansa daha yakın bir sonuç vermiştir.

Optimizasyon algoritmalarının yapısında var olan parametrelerin belirlenmesi, kullanılacak problemi çözmeye farklı sonuçlar ortaya çıkarmaktadır. Her bir problem için algoritmanın içerisindeki parametre farklı değer alabilmektedir. Bu tez çalışması içerisinde kullanılan CGO algoritması yapısında parametre bulundurmamasıyla iki uzuvlu robot kolunun hareketi için en iyi sonucu vermiştir.

5.2 Öneriler

Robotik sistemlerde yapılacak çalışmalarda kontrolcü kısmı büyük önem göstermektedir. Kontrolcü parametrelerinin belirlenmesi problemi için kullanılacak yöntem içerisinde parametre sayısı az olan yöntemler seçilmelidir.

Herhangi bir eksenin açısal olarak kaybı tutucu uca etki ettiği için kullanılacak malzemede hata oranı en küçük olan seçilmelidir. Sistem için gerekli olan tork değerine göre motor seçimi yapılmalıdır. Tahrik elemanı olarak kullanılacak olan motor DC motor ise enkoderin adım aralığı yüksek, motorun rpm değeri düşük, redüktörün çevrim oranı ise yüksek olmalıdır.

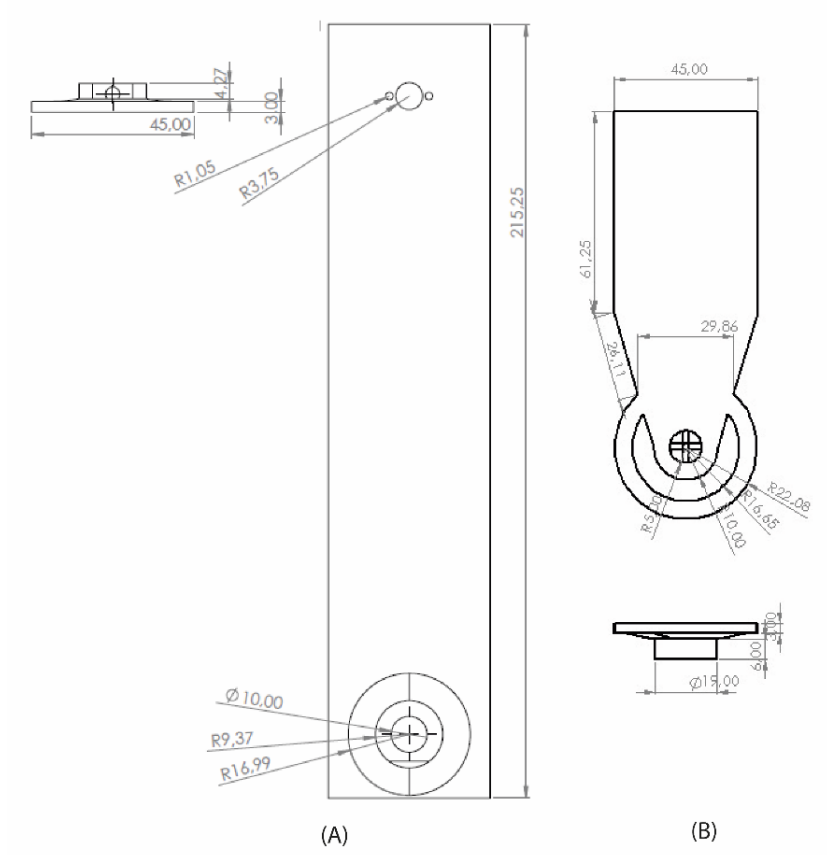
6. KAYNAKLAR

- Afjei, E., Ghomsheh, A. N., and Karami, A., 2007, Sensorless speed/position control of brushed DC motor, *International Aegean Conference on Electrical Machines and Power Electronics 2007, IEEE*, 730-732.
- Akkaya, Ş., 2013, Sahada programlanabilir kapı dizileri (FPGA) tabanlı robot kontrolü, Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.
- Azab, M., 2009, Global maximum power point tracking for partially shaded, PV arrays using particle swarm optimisation. *International Journal of Renewable Energy Technology*, 1(2), 211-235.
- Bansal, H. O., Sharma, R., and Shreeraman, P. R., 2012, PID controller tuning techniques: a review, *J. Control Eng. Technol*, 2(4), 168-176.
- Berber, Ö., Muharrem, A. T. E. Ş., ALHASSAN, H. A., ve Güneş, M., 2016, Parçacık sürü optimizasyonu ve PID ile mobil robotun optimum yörünge kontrolü, *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 19(3), 165-169.
- Beyhan, A., 2021, Üç serbestlik dereceli robotik kolun kartezyen uzayda empedans kontrolü, Yüksek Lisans Tezi, *Bursa Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Bursa.
- Bingül, Z. ve Küçük, S., 2005, Robot Tekniği I, Birsen Yayınevi, İstanbul.
- Boğar, E., 2021, Optimizasyon kuramında yeni bir metasezgisel yaklaşım: Ergen kimlik arama algoritması (AISA) ve mühendislik uygulamaları, Doktora Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli.
- Boğar, E., 2022, Chaos game optimization- least squares algorithm for photovoltaic parameter estimation, *Arabian Journal For Science and Engineering*, 5-7.
- Condit, R., 2004, Brushed DC motor fundamentals. microchip application note AN905, *Microchip Technology Inc*.
- Crowe, J., Chen, G. R., Ferdous, R., Greenwood, D. R., Grimble, M. J., Huang, H. P., ... and Zhang, Y., 2005, PID control: new identification and design methods, *Springer-Verlag*, London Limited.
- Çengelci, B., ve Çimen, H., 2005, Endüstriyel robotlar, *Makine Teknolojileri Elektronik Dergisi*, 2(2), 69-78.
- Das, A., and Banerjee, K., 2009, Fast prototyping of a digital PID controller on a FPGA based soft-core microcontroller for precision control of a brushed DC servo motor, *35th Annual Conference of IEEE Industrial Electronics 2009, IEEE*, 2825-2830.

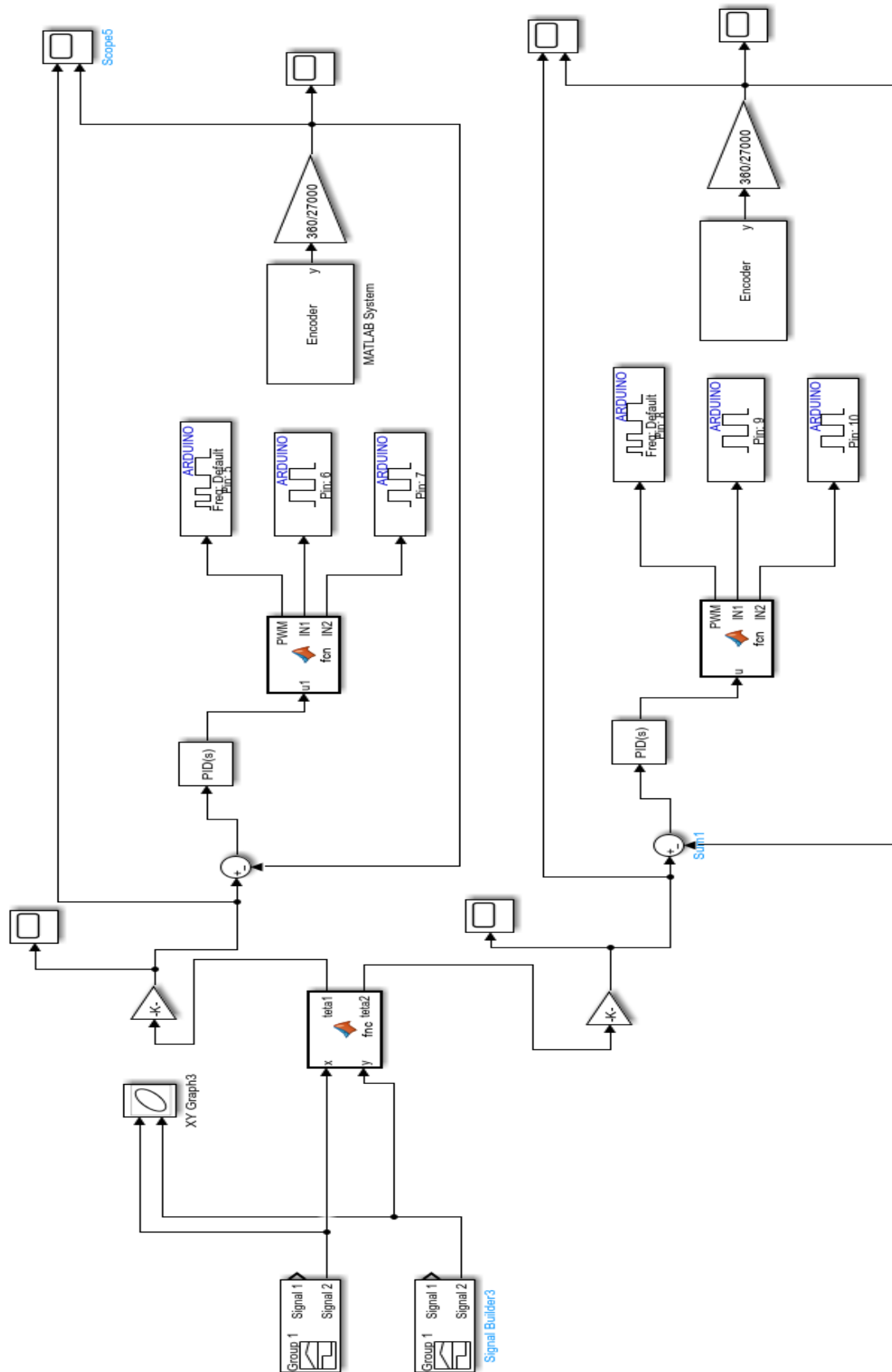
- Denavit, J. ve Hartenberg, R.S. ,1955, A kinematic notation for lower-pair mechanisms based on matrices, *ASME J. Appl. Mechanics*, 215-221.
- Doğan, E., 2010, Delta paralel robot ileri ve ters kinematik hesaplamaları, Yüksek Lisans tezi, *Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü*, Eskişehir.
- Durusu, V., 2022, Parçacık sürü optimizasyon ve Ziegler-Nichols metodunu kullanarak DC motorun hız kontrolü için PID parametrelerin belirlenmesi ve çalışılması, Yüksek Lisans Tezi, *Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü*, Afyonkarahisar.
- Fu, K. S., Gonzalez, R.C. Lee, C. S. G., 1987, Robotics: control, sensing, vision and intelligence, *McGraw-Hill Book Company*, New York.
- Gül, M.,2021, Açık kaynak medikal yardımcı robot kolun python ile ileri kinematik analizi, *Konya Mühendislik Bilimleri Dergisi*, 9(2), 395-402.
- Hassan, S., Abdelmajid, B., Mourad, Z., Aicha, S., and Abdenaceur, B., 2017, An advanced MPPT based on artificial bee colony algorithm for MPPT photovoltaic system under partial shading condition. *International Journal of Power Electronics and Drive Systems*, 8(2), 647.
- İlgen, S., Durdu, A., Gülbağçe, E., Çakan, A., ve Kalyoncu, M.,2022, The Bees Algorithm Approach to Determining SMC Controller Parameters for the Position Control of a SCARA Robot Manipulator, *Avrupa Bilim ve Teknoloji Dergisi*, (33), 267-273.
- Johnson, M. A., and Moradi, M. H., 2005, PID control, London, UK: *Springer-Verlag*, London Limited.
- Joseph, S. B., Dada, E. G., Abidemi, A., Oyewola, D. O., and Khammas, B. M., 2022, Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon*, e09399.
- Kaplan, K., Kuncan, M., Polat, H., Burak, T. E. P. E., ve Ertunç, H. M., 2020, PID ve bulanık mantık tabanlı DC motorun gerçek zamanlı konum kontrolü, *Journal of the Institute of Science and Technology*, 10(2), 900-916.
- Karaboğa, D., 2005, Kayseri, Erciyes Üniversitesi, <https://abc.erciyes.edu.tr/> [Ziyaret Tarihi: 19 Aralık 2021]
- Karahan, O., 2012, Seri ve paralel robotlarda parçacık sürü optimizasyonu ile yörünge kontrolörü tasarımı. Doktora Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli.
- Knežević, J. M., 2013, Low-cost low-resolution sensorless positioning of DC motor drives for vehicle auxiliary applications, *Transactions On Vehicular Technology*, *IEEE*, 62(9), 4328-4335.

- Maheriya, S., and Parikh, P., 2016, A review: Modelling of brushed dc motor and various type of control methods. *Journal for Research*, 1.
- Maung, M. M., Latt, M. M., and Nwe, C. M., 2018, DC motor angular position control using PID controller with friction compensation, *International Journal Of Scientific And Research Publications*, 8(11), 149.
- Mayetin, U., 2013, İki serbestlik dereceli düzlemsel robotun farklı kontrolörlerle yörünge kontrolünün gerçekleştirilmesi, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli.
- Meshram, U., Bande, P., Dwaramwar, P. A., and Harkare, R. R., 2009, Robot arm controller using FPGA, *International Multimedia 2009, Signal Processing and Communication Technologies, IEEE*, 8-11.
- Myint, W. M., 2015, Theingi., Kine-matic control of pick and place robot arm, *International Journal of Engineering and Techniques*, 1(4), 63-70.
- Öztürk, S., ve Öztürk, N., 2016, Yapay arı koloni algoritması kullanılarak görüntü iyileştirme yönteminin geliştirilmesi, *Gazi University Journal of Science Part C: Design and Technology*, 4(4), 173-183.
- Palacín, J., and Martínez, D., 2021, Improving the angular velocity measured with a low-cost magnetic rotary encoder attached to a brushed DC motor by compensating magnet and hall-effect sensor misalignments, *Sensors*, 21(14), 4763.
- Peerzada, P., Larik, W. H., and Mahar, A. A., 2021, DC motor speed control through arduino and L298N motor driver using PID controller, *International Journal of Electrical Engineering & Emerging Technology*, 4(2), 21-24.
- Pekdemir, E., 2017, Otonom bir mobil robotun parçacık sürü algoritması ile optimum yörünge kontrolü, Yüksek Lisans Tezi, *Kahraman Maraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü*, Afyonkarahisar.
- Ramasubramanian, D., 2016, Identification and control of DC motors, Master's Thesis, *Universitat Politècnica de Catalunya*, Barcelona, Spain.
- Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., and Aldape-Pérez, M., 2020, Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem. *Applied Soft Computing*, 93, 106342.
- Ruderman, M., Krettek, J., Hoffmann, F., and Bertram, T., 2008, Optimal state space control of DC motor, *IFAC Proceedings Volumes*, 41(2), 5796-5801.
- Sailan, K., and Kuhnert, K. D., 2013, DC motor angular position control using PID controller for the purpose of controlling the hydraulic pump, *In Proc. Int. Conf. Control, Eng. Inf. Technol.(CEIT)*, Vol. 1, 22-26.

- Sevim, Ö., ve Sönmez, M., 2014, Geliştirilmiş yapay arı koloni algoritması ile kafes ve düzlemsel çelik yapıların optimum tasarımı, *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 3(2), 38-51.
- Soriano, L. A., Rubio, J. D. J., Orozco, E., Cordova, D. A., Ochoa, G., Balcazar, R., and Gutierrez, G. J., 2021, Optimization of sliding mode control to save energy in a SCARA robot, *Mathematics*, 9(24), 3160.
- Strandt, E. R., 1963, Hall-effect incremental angle encoder, *IEEE Transactions On Instrumentation And Measurement*, 12(1), 22-26.
- Talatahari, S. ve Azizi, M., 2021, Chaos Game Optimization: a novel metaheuristic algorithm, *Artificial Intelligence Review*, 54(2), 917-1004.
- Tang, J., Liu, G., and Pan, Q., 2021, A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA Journal of Automatica Sinica*, 8(10), 1627-1643.
- Taşar, B., Yakut, O., Müftü, S., Yöndem, Ş. F., ve Durmuş, H., 2020, Üst ekstremite dış iskelet robot için giyilebilir kablosuz MIMU sensörler vasıtası ile insan robot etkileşim ağı tasarımı, *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 20(6), 1165-1177.
- Ulusoy, A., ve Güneş, M., 2019, mobil robot kolunun PSO ile stabilizasyonu, *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 22(4), 288-297.
- Üstün, D., 2017, Kablosuz iletişim sistemlerine yönelik anten tasarımı için melez yapay arı koloni algoritmasının geliştirilmesi ve uygulanması, Doktora Tezi, *Mersin Üniversitesi Fen Bilimleri Enstitüsü*, Mersin.
- Vazquez-Sanchez, E., Gomez-Gil, J., Gamazo-Real, J. C., and Díez-Higuera, J. F., 2011, A new method for sensorless estimation of the speed and position in brushed DC motors using support vector machines, *Transactions on Industrial Electronics, IEEE*, 59(3), 1397-1408.
- X. Hu, 2006, *Particle swarm optimization tutorial* [online], Guangdong, South China Agricultural University, <https://titan.csit.rmit.edu.au/~e46507/publications/psotutorialseal06.pdf> [Ziyaret Tarihi: 10 Eylül 2021].
- Yüksel, G., Yiğit, T., ve Çelik, H., 2018, DA motor tahrikli dört serbestlik dereceli bir robot kolun 2-DOF PID ile kontrolü, *TOK2018 Otomatik Kontrol Ulusal Toplantısı*, Kayseri, Türkiye, 22-26.

EKLER**EK-1** Robot kollarına ait teknik resim

EK-2 Kare hareket için tasarlanan blok diyagramı



EK-3 Daire hareket için tasarlanan blok diyagramı

