ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DOĞADAN ESİNLENEN OPTİMİZASYON ALGORİTMALARI KULLANARAK TOP DENGELEME DÜZENEĞİ İÇİN PID TABANLI OPTİMAL KONTROLCÜ TASARIMI

Mahmut Gökhan TURGUT

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2021

ÖZET

Yüksek Lisans Tezi

DOĞADAN ESİNLENEN OPTİMİZASYON ALGORİTMALARI KULLANILARAK TOP DENGELEME DÜZENEĞİ İÇİN PID TABANLI OPTİMAL KONTROLCÜ TASARIMI

Mahmut Gökhan TURGUT

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ökkeş Tolga ALTINÖZ

Günümüzde kullanılan sistemler genel olarak karmaşık ve doğrusal olmayan sistemlerdir. Bu sistemler üzerinde denemeler yapmak zor, tehlikeli ve maliyetli olabilmektedir. Bu tür sistemleri kontrol edebilmek hala önemli bir çalışma alanıdır. Bu sistemlerin kontrolü için benzer veya daha karmaşık sistemler kullanılmaktadır. Deneysel sistemler ile karmaşık ve doğrusal olmayan sistemler üzerinde daha güvenli ve maliyetsiz olarak çalışmalar yapılabilmektedir. Top dengeleme düzeneği de bu deneysel düzeneklerden biridir. Topun konumu iki eksende değişmektedir, bu sebeple sistem iki serbestlik dereceli (2-Dof) sistem olarak adlandırılır. İki boyutlu bir tablada serbest halde hareket edebilen bir topun pozisyonunu, istenilen pozisyona ulaştırmak için eyleyicileri (servo motorlar) kontrol etmek gerekmektedir. Topun konumu tablanın açısına göre değişir. Top dengeleme düzeneğinde önceden belirlenmiş bir sınır içerisinde topu tablada tutmak amaçlanır. Tablanın açısını ise eyleyiciler değiştirmektedir. Topu istenilen konuma getirebilmek için servo motorların kontrol edilmesi gerekmektedir. Bu sebeple, sistemin matematiksel modeli oluşturulduktan sonra kontrolcü kullanılması gerekmektedir. Bu tez kapsamında kullanılan kontrolcüler şu şekildedir, PID, PIDC, PIDA, 2-Dof PID, Bulanık-PID. Kontrolcülerin parametrelerini bulmak için optimizasyon algoritmaları kullanılmıştır. Kullanılan optimizasyon algoritmaları şu şekildedir, Ateşböceği Algoritma (FA), Parçacık Sürü Optimizasyonu (PSO), Arılar Algoritması (BA), Yabani Ot Optimizasyonu (IWO), Yapay arı kolonisi (ABC). Kontrolcüler öncelikle bilgisayar ortamında benzetim çalışmaları üzerinden tasarlanmıştır, daha sonra düzenek üzerinde uygulaması yapılmıştır.

Eylül 2021, 82 sayfa

Anahtar Kelimeler: Top dengeleme, PID, PIDA, PIDC, 2-Dof PID, PID Tabanlı Bulanık Kontrolcü, Optimizasyon, Ateşböceği Algoritma (FA), Parçacık Sürü Optimizasyonu (PSO), Arılar Algoritması (BA), Yabani Ot Optimizasyonu (IWO), Yapay arı kolonisi (ABC).

ABSTRACT

Master Thesis

PID-BASED OPTIMAL CONTROLLER DESIGN FOR BALL BALANCE USING NATURE-INSPIRED OPTIMIZATION ALGORITHMS

Mahmut Gökhan TURGUT

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Ö. Tolga ALTINÖZ

The systems used today are generally complex and nonlinear systems. Experimenting on these systems can be difficult, dangerous and costly. Being able to control such systems is still an important area of study. Similar or more complex systems are used to control these systems. With experimental systems, it is possible to work on complex and nonlinear systems more safely and cost-effectively. Ball balancing mechanism is one of these experimental setups. The position of the ball changes in two axes, so the system is called a two-degree-of-freedom (2-Dof) system. It is necessary to control the actuators (servo motors) to bring the position of a ball that can move freely on a two-dimensional table to the desired position. The position of the ball changes according to the angle of the table. In the ball balancing mechanism, it is aimed to keep the ball on the table within a predetermined limit. Actuators change the angle of the table. In order to bring the ball to the desired position, servo motors have to be controlled. For this reason, after the mathematical model of the system has been created, a controller should be used. The controllers used in this thesis are as follows: PID, PIDC, PIDA, 2-Dof PID, Fuzzy-PID. Optimization algorithms are used to finding the parameters of the controllers. The optimization algorithms used are as follows: Firefly Algorithm (FA), Particle Swarm Optimization (PSO), Bees Algorithm (BA), Weed Optimization (IWO), Artificial bee colony (ABC). Controllers were first designed through simulation studies in the computer environment, then the application was made on the mechanism.

September 2021, 82 sayfa

Key Words: Ball Balancing, PID, PIDA, PIDC, 2-Dof PID, PID Based Fuzzy Control, Optimization, Firefly Algorithm (FA), Particle Swarm Optimization (PSO), Bees Algorithm (BA), Invasive Weed Optimization (IWO), Artifical Bee Colony (ABC).

ÖNSÖZ VE TEŞEKKÜR

Çalışmalarım sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam Doç. Dr. Ökkeş Tolga ALTINÖZ'e, beni desteklemekten hiçbir zaman vazgeçmeyen sevgili eşim Merve ŞENSÖZ TURGUT'a, sevgili aileme teşekkürlerimi ve saygılarımı sunarım.

Mahmut Gökhan TURGUT
Ankara, Eylül 2021

İÇİNDEKİLER

ÖZET ..... **i**
ABSTRACT ..... **ii**
ÖNSÖZ VE TEŞEKKÜR ..... **iii**
KISALTMALAR DİZİNİ ..... **vi**
ÇİZELGELER DİZİNİ ..... **vii**
ŞEKİLLER DİZİNİ ..... **viii**
1.GİRİŞ **1**
2. KURAMSAL TEMELLER **5**
2.1 Literatür Taraması **5**
3.MATERYAL VE YÖNTEM **12**
3.1 Top Dengeleme Düzeneği **12**
3.2 Top Dengeleme Düzeneği Fiziksel Modeli **13**
3.3 Top Dengeleme Düzeneğinin Kontrolü **14**
3.3.1 PID **15**
3.3.2 PIDA **17**
3.3.3 PIDC **18**
3.3.4 2-Dof-PID **18**
3.3.5 Bulanık Mantık Denetleyicisi **19**
3.4 Optimizasyon **22**
3.4.1 Sürü zekâsı tabanlı optimizasyon algoritmaları **22**
3.4.1.1 Ateş böceği algoritması **23**
3.4.1.2 Parçacık sürüsü optimizasyonu (PSO) **24**
3.4.1.3 Yapay arı kolonisi (ABC) **26**
3.4.1.4 Arılar algoritması **28**
3.4.1.5 Yabani ot algoritması **29**
4. ARAŞTIRMA BULGULARI **31**
4.1 Sistemin Modellenmesi **31**
4.1.1 Lagrangian metodu **32**
4.1.2 Eyleyicinin modeli **35**
4.2 Optimizasyon algoritmaları **37**
4.3 Benzetim Çalışmaları **49**

4.3.1 PID kontrolcü 52
4.3.2 PIDA kontrolcü 56
4.3.3 PIDC 57
4.3.4 2-Dof-PID 59
4.3.5 Bulanık-PID 61
4.4 Laboratuvar Çalışmaları 63
5. TARTIŞMA VE SONUÇLAR 71
KAYNAKLAR 74

KISALTMALAR DİZİNİ

DOF	Serbestlik Derecesi (Degree of Freedom)
PID	Oransal-İntegral-Türev (Proportional-Integral-Derivative)
PIDA	Oransal-Integral-Türev-İvme (Proportional-Integral-Derivative Acceleration)
PIDC	Oransal-Integral-Türev Karşılaştırıcı (Proportional-Integral-Derivative Compensator)
BMD	Bulanık Mantık Denetleyicisi
ABC	Yapay Arı Kolonisi (Artifical Bees Colony)
BA	Arılar Algoritması (Bees Algorithm)
FA	Ateşböceği Algoritması (Firefly Algorithm)
IWO	Yabani Ot Optimizasyon (Invasive Weed Optimization)
PSO	Parçacık Sürüsü Optimizasyonu (Partice Swarm Optimization)
IAE	Mutlak Hatanın İntegrali (Integral Absolute Error)
ITAE	Zamanın Mutlak Hata İle Çarpılmasının İntegrali (Integral Of Time Multiplied Absolute Error)
ISE	Hatanın Karesinin İntegrali (Integral Square Error)
ITSE	Zamanın Hatanın Karesi İle Çarpılmasının İntegrali (Integral of Time Multiplied Absolute Error)
FPGA	Alanda Programlanabilir Kapı Dizileri (Field programmable gate array)
CEC	Evrimsel Hesaplama Kongresi (Congress on Evolutionary Computation)

ÇİZELGELER DİZİNİ

Çizelge 3.1 Top Dengeleme Düzeneği Semboller, Açıklamalar ve Değerler.....13
Çizelge 3.2 Ziegler Nichols 1.Metod.....16
Çizelge 3.3 Ziegler Nichols 2.Metod.....16
Çizelge 4.1 CEC17 Fonksiyonları.....38
Çizelge 4.2 ABC 2 Boyut için Sonuçlar.....39
Çizelge 4.3 BA 2 Boyut için Sonuçlar.....39
Çizelge 4.4 FA 2 Boyut için Sonuçlar.....40
Çizelge 4.5 IWO 2 Boyut için Sonuçlar.....41
Çizelge 4.6 PSO 2 Boyut için Sonuçlar.....41
Çizelge 4.7 ABC 10 Boyut için Sonuçlar.....42
Çizelge 4.8 BA 10 Boyut için Sonuçlar.....43
Çizelge 4.9 FA 10 Boyut için Sonuçlar.....43
Çizelge 4.10 IWO 10 Boyut için Sonuçlar.....44
Çizelge 4.11 PSO 10 Boyut için Sonuçlar.....45
Çizelge 4.12 ABC 30 Boyut için Sonuçlar.....46
Çizelge 4.13 BA 30 Boyut için Sonuçlar.....47
Çizelge 4.14 FA 30 Boyut için Sonuçlar.....47
Çizelge 4.15 IWO 30 Boyut için Sonuçlar.....48
Çizelge 4.16 PSO 30 Boyut için Sonuçlar.....49
Çizelge 4.17 Optimizasyon algoritmalarının bulduğu PID kontrolcüsü katsayıları.....54
Çizelge 4.18 PID kontrolcü parametrelerinin sistem cevabı sonuçları.....55
Çizelge 4.19 Optimizasyon algoritmalarının bulduğu PIDA kontrolcüsü katsayılar.....56
Çizelge 4.20 PIDA kontrolcü parametrelerinin sistem cevabı sonuçları.....56
Çizelge 4.21 Optimizasyon algoritmalarının bulduğu PIDC kontrolcüsü katsayıları.....58
Çizelge 4.22 PIDC kontrolcü parametrelerinin sistem cevabı sonuçları.....58
Çizelge 4.23 Optimizasyon algoritmalarının bulduğu 2-Dof-PID kontrolcüsü katsayıları.....60
Çizelge 4.24 2-Dof-PID kontrolcü parametrelerinin sistem cevabı sonuçları.....60
Çizelge 4.25 5x5 Bulanık Mantık Kontrolcüsü Kural Tablosu.....62
Çizelge 4.26 Optimizasyon algoritmalarının bulduğu Bulanık-PID kontrolcüsü katsayıları.....62
Çizelge 4.27 Bulanık-PID kontrolcü parametrelerinin sistem cevabı sonuçları.....63

ŞEKİLLER DİZİNİ

Şekil 3.1 Top Dengeleme Sistemi 12
Şekil 3.2 Top Dengeleme Düzeneği Kapalı Döngü Şeması 14
Şekil 3.3 İdeal PID Kontrolcüsü Genel Şeması 15
Şekil 3.4 Ölü Zaman ve Zaman Sabiti 16
Şekil 3.5 İdeal PIDA Kontrolcüsü Genel Şeması 17
Şekil 3.6 PIDC Kontrolcüsü Genel Şeması 18
Şekil 3.7 2-Dof PID Kontrolcüsü Genel Yapısı 19
Şekil 3.8 Monotonik, Üçgen, Trapezoidal Üyelik Fonksiyonları 20
Şekil 3.9 Bulanık-PID Genel Yapısı 21
Şekil 3.10 FA sözde kodu 24
Şekil 3.11 PSO sözde kodu 25
Şekil 3.12 Parçacık Hareketi 26
Şekil 3.13 ABC sözde kodu 27
Şekil 3.14 BA sözde kodu 28
Şekil 3.15 IWO sözde kodu 30
Şekil 4.1 Top Dengeleme Düzeneği Boyutları 31
Şekil 4.2 Top Dengeleme Düzeneği 32
Şekil 4.3 Motor ve Tabla Açıları 34
Şekil 4.4 İndirgemeden Sonra Sistemin Blok Diyagramı 35
Şekil 4.5 X-Yönündeki Blok Diyagram 36
Şekil 4.6 Y-Yönündeki Blok Diyagram 36
Şekil 4.7 Sistemin Genel Modellemesi 36
Şekil 4.8 Root Locus Eğrisi 50
Şekil 4.9 Bode Diyagramı 50
Şekil 4.10 Ziegler-Nichols 2. Metodu için Sistem 52
Şekil 4.11 Sistemin Tek Eksende Genel Modellemesi 53
Şekil 4.12 Ziegler Nichols Çıktısı 53
Şekil 4.13 PIDC kontrolcüsüne sistemin cevabı 59
Şekil 4.14 2-DoF-PID kontrolcüsüne sistemin cevabı 61
Şekil 4.15 ACROME Top Dengeleme Düzeneği 64
Şekil 4.16 PID Kontrolcü X yönü 65

Şekil 4.17 PID Kontrolcü Y yönü 65
Şekil 4.18 PIDA Kontrolcü X yönü 66
Şekil 4.19 PIDA Kontrolcü Y yönü 66
Şekil 4.20 PIDC Kontrolcü X yönü 67
Şekil 4.21 PIDC Kontrolcü Y yönü 68
Şekil 4.22 2-Dof PID Kontrolcü x yönü 69
Şekil 4.23 2-Dof PID Kontrolcü y yönü 69
Şekil 4.24 Bulanık-PID Kontrolcü x yönü 70
Şekil 4.25 Bulanık-PID Kontrolcü y yönü 70

1. GİRİŞ

Günümüzdeki sistemler genel olarak karmaşık ve doğrusal olmayan sistemlerdir. Bu sistemlerin üzerinde çalışmak oluşabilecek hatalar sebebiyle, göreceli olarak maliyetli ve güvenlik açısından tehlikeli olabilmektedir. Bu tür sistemleri kontrol edebilmek hala mühendisliğin önemli problemlerindendir. Bu sistemlerin kontrolünün yapılabilmesi için benzer veya daha karmaşık deneysel sistemler kullanılmaktadır. Deneysel sistemler ile daha güvenli ve maliyetsiz çalışmalar yapılabilmektedir. Dengeleme sistemleri kontrol alanında hala zorluğunu korumaktadır. Tabla üzerindeki dengeleme sistemleri, iletişim, gömülü yazılım, kontrol gibi çeşitli alanlarda teorik ve pratik bir olanak sağlamaktadır. Top dengeleme düzeneği de bu deneysel düzeneklerden biridir.

Top dengeleme düzeneğindeki amaç, iki boyutlu bir tablada serbest halde hareket edebilen bir topun pozisyonunu tablaya bağlanan eyleyiciler kullanılarak istenen pozisyona ulaşmak için eyleyicileri kontrol etmektir. Topun pozisyonu iki eksende değiştiği için bu sistem iki serbestlik dereceli sistem olarak adlandırılır. Topun konumu tablanın açısına göre değişir. Top dengeleme düzeneğinde önceden belirlenmiş bir sınır dâhilinde topu tablada tutmak amaçlanır. Topu istenilen konuma getirebilmek için servo motorların kontrol edilmesi gerekmektedir. Top dengeleme düzeneği doğrusal olmaması sebebiyle, doğrusal olmayan kontrol yöntemlerinin deneysel olarak uygulanmasına olanak sağlamaktadır. Bu sistem çok değişkenli ve açık çevrimli kararsız bir sistemdir (Itani 2017, Ker vd. 2007). Top dengeleme düzeneğinde temel olarak konum kontrolü ve yörünge takibi olmak üzere 2 kontrol problemi vardır. Konum kontrolündeki amaç, topu istenilen konuma ulaştırmak ve istenilen konumda tutmaktır. Yörünge takibindeki amaç, topu önceden belirlenmiş bir yolda ilerletmektir (Dušek vd. 2017). Bu tez kapsamında konum kontrolü üzerine çalışmalar yapılmıştır.

Modellemenin amacı sistemin girişiyle çıkışı arasındaki fiziksel bağlantıyı bulmaktır. İlk olarak sistemi tanımlayan bir matematiksel model bulunmalıdır. Modellemenin ilk adımı hareket denklemini türetmektir. Hareket denklemi, topun hareketi (x, y) ile tablanın iki

açısı (α, β) arasındaki ilişkiyi açıklar. Denklemin girişleri top dengeleme tablasının motorlarının açıları olurken çıkışları *x* ve *y* kartezyen koordinatları olacaktır (Acrome 2014, Zeeshan vd. 2012, Mujadin ve Pratama 2017). Top ve tabla sistemi için literatürde temel olarak iki modelleme yaklaşımı vardır, bunlar; Lagrangian yöntemi ve Newton hareket yasasıdır. Newton Hareket Yasasını uygulamak için, sistem *x* ve *y* yönünde iki hareket moduna ayrılmalıdır. Newton'un Hareket Yasasına göre, istenilen yönündeki hareketi bulmak için, topa etkiyen kuvvetlerin hepsi toplanmalıdır. Lagrangian yöntemi kinetik ve potansiyel enerji arasındaki farktır. Bu fark ile hareket denklemi bulunmaktadır. Lagrangian yöntemi denklem 1.1 ve 1.2'de gösterilmiştir.

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = 0 \tag{1.1}$$

$$\mathrm{L} = \mathrm{E_{kin}} - \mathrm{E_{pot}} \tag{1.2}$$

Kontrolcüler ve modellemeleri hemen hemen her alanda kullanılmaktadır. Birçok kontrolcü olsa da kontrolcülerin uygulanacağı sisteme uygun kontrolcü uygulanması önemlidir. Kontrolde kararlılık ve kararlılığın hızlı sağlanması büyük önem taşımaktadır. Bu sebeple, sistemin modellenmesi ve kontrolcü çok büyük önem teşkil eder. Açık ve kapalı döngü olmak üzere iki döngü çeşidi vardır. Açık döngü sistemlerde, çıkışın giriş üzerinde bir etkisi yoktur, bu yüzden manuel kontrol sistemi denir. Açık döngü basit yapılı, ekonomik ve istikrarlı olmasına rağmen güvenilir değildir. Kapalı döngü sistemlere otomatik kontrol sistemleri denir ve açık sistemlere göre birçok avantaja sahiptir, bu nedenle genellikle kontrol uygulamalarında kapalı döngü kontrol sistemleri tercih edilir. Bu tez kapsamında kapalı döngü kullanılacaktır.

Günümüzde sistemlerde kolay yapısı ve bilinirliği sebebiyle genellikle PID kontrolcüsü kullanılmaktadır. PID kontrolcüsünün 3 terimi vardır. Oran terimi (P), hata sinyaliyle orantılı kontrol sağlar. İntegral terimi (I), bir integratör sayesinde kararlı durum hatalarının azaltılmasını sağlar. Türev terimi (D), türev alıcı sayesinde anlık yanıtın iyileşmesini sağlar (Li vd. 2005). PID kontrolcüsünün genel formülü denklem 1.3'de gösterilmiştir.

$$u(t) = K_p * e(t) + K_i * \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (1.3)$$

Bu çalışmada PID tabanlı kontrolcüler ele alınmıştır. Bu kontrolcülerden birisi olan PIDA, PID kontrolcüsüne ivme eklenmesiyle oluşturulur. Diğer kontrolcü PIDC ise PID kontrolcüsüne seri filtre eklenerek oluşturulur. Bu tez çalışmasında eklenen filtre faz geciktirici ve ilerletici filtresidir. 2-Dof PID iki girişli tek çıkışlı bir kontrolcüdür. Ölçülen çıkış sinyaliyle referans sinyali arasındaki farka göre bir çıkış üretir. Doğrusal olmayan bir sistem olan Bulanık Mantık Denetleyicisi sistemin matematiksel modelini istememektedir. Bulanık Mantık Denetleyicisi sezgisel yöntemlere dayanmaktadır. Bulanık Mantık Denetleyicisi, PID ile kullanılabilmektedir. Tez kapsamında kullanılacak olan bir diğer kontrolcü ise Bulanık-PID kontrolcüsüdür.

Bir PID kontrolcüsünün tasarlanması kolay gözükebilir ancak performansının kontrolcünün uygulanacağı sisteme göre iyi olması için PID parametrelerinin düzgün seçilmesi gerekmektedir. Ele alınan sistemde nasıl bir kontrolcü istendiği önemlidir. Sistemin gereksimine göre yavaş, hızlı veya normal hızda kontrolcü parametreleri seçilebilir. Bu sebeple PID ve PID tabanlı kontrolcülerin parametrelerinin bulunmasında doğadan esinlenilen optimizasyon algoritmaları kullanılmıştır. Optimizasyon, problem veya problemlerde verilen koşullarda, belirli sınırlar içerisinde tüm çözümler arasında en iyi çözümü bulma işlemidir. Optimizasyon her zaman daha iyiye gitmeyi amaçlamaktadır. Doğadan esinlenilen optimizasyon algoritmaları doğadaki canlıların hareketlerinin incelenmesi sonucu ortaya çıkmıştır. Bu çalışmada kullanılan optimizasyon algoritmaları şu şekildedir, Parçacık Sürüsü Optimizasyonu (PSO), Ateşböceği Algoritması (FA), Yapay Arı Kolonisi Algoritması (ABC), Arılar Algoritması (BA), Yabani Ot Algoritması (IWO).

Ateşböceği algoritması ateşböceklerinden ilham alınarak yapılan bir algoritmadır. Algoritma ateş böceklerinin ışıklarıyla ilgilidir. Ateşböcekleri, iletişim ve avlanmak için ışıklarını ritmik olarak yakıp söndürürler. Mesafe arttıkça ışığın havada soğrulmasından dolayı belirli bir mesafede iletişim kurabilirler. Parçacık sürüsü optimizasyonu, lineer olmayan fonksiyonların optimize edilmesi için kuş sürülerinin davranışları örnek alınarak

öne sürülen bir sezgisel algoritmadır. Sürüdeki her bireye parçacık denir. Parçacık Sürü Optimizasyonu, parçacıkların lideri takip etmesine dayanmaktadır. Yapay Arı Kolonisi algoritması arılardan ilham alınarak yapılmıştır. Doğada olduğu gibi algoritmada da arılar gruplara ayrılmıştır ve her grubun görevi farklıdır. Yapay Arı Kolonisi algoritması arıların yiyecek aramasına dayanmaktadır. Arılar Algoritması da Yapay Arı Kolonisi gibi arılardan ilham alınarak yapılmıştır. İki algoritma benzerlik gösterse de çalışma şekilleri farklıdır. İstilacı Yabani Ot Algoritması, yabani ot kolonizasyonunu ve dağılım sürecini taklit eden bir algoritmadır.

2. KURAMSAL TEMELLER

Bazı kontrol problemleri gerçek hayatta deneme yapılması maliyetli ve riskli olabilmektedir. Bu sebeple deneysel düzenekler ortaya çıkmıştır. Top dengeleme düzeneği de karmaşık kontrol problemleri için önemli bir düzenektir. Bu tez kapsamında top dengeleme düzeneğine farklı kontrolcüler uygulayarak, gerçek hayatta oluşan karmaşık kontrol problemleri için kullanılabilecek kontrolcüleri ve bu kontrolcülerin parametrelerini en iyi şekilde bulabilmesi için optimizasyon algoritmaları ele alınmaktadır. Optimizasyon algoritması ile bulunan kontrolcü parametrelerinin topu tabla üzerinde istenilen konuma getirip, o konumda tutması beklenmektedir. Bu tez kapsamında endüstride en çok kullanılan kontrolcü olan PID kontrolcüsü seçilmiştir ve PID kontrolcü temelinden oluşturulan kontrolcüler kullanılmıştır. PID kontrolcüsü kolay yapısı sayesinde analog ve dijital devrelerde sıklıkla kullanılan kontrolcülerden biridir.

Çalışmada öncelikle sistem, literatürde bulunan modelleme metotlarından biriyle modellenmiştir. Sistemin doğrusal olarak modellenip daha hızlı sonuç alınması ve daha fazla kontrolcü denenmesi amaçlanmıştır. Sistemin kontrolcü parametrelerini bulmak için optimizasyon algoritmaları kullanılmıştır. Bu tez kapsamında optimizasyon algoritmaları öncelikle Evrimsel Hesaplama konferansındaki (CEC 2017) yarışmada kullanılan test problemlerine göre kıyaslanmıştır. Alınan sonuçlardan sonra kontrolcü parametrelerini bulmak için optimizasyon algoritmaları kontrolcülere uygulanmıştır. Bunun için Matlab Simulink'te simülasyon gerçekleştirilmiştir ve çıkan sonuçlar kıyaslanmıştır. Tüm kontrolcülerin tüm optimizasyon algoritmaları sonuçları performansa göre kıyaslanmıştır. Bulunan kontrolcü parametreleri laboratuvar ortamında gerçek top dengeleme sistemine uygulanmış ve sonuçları kıyaslanmıştır.

2.1 Literatür Taraması

Top dengeleme düzeneği dengeleme sistemleri, kontrol sistemleri, gerçek zamanlı işletim sistemleri vb. gibi sistemler için önemli bir deney düzeneğidir. Top dengeleme düzeneği, top ve çubuk düzeneğinin iki boyutlu halidir. Top dengeleme düzeneği ilk olarak kontrol

sistemleri uygulanması için yüksek lisansta önerilmiştir (Jorgensen 1974). Top dengeleme düzeneği, top ve çubuk sistemine göre daha karmaşıktır. Top dengeleme düzeneği, top ve çubuk düzeneğinin genişletilmiş hali olarak düşünülebilir. Top dengeleme düzeneği birçok lineer olmayan uygulamayı deneysel olarak uygulayabilmek için kullanılmaktadır (Hamdoun vd. 2021). Kesir dereceli PID kontrolcüsüyle, optimal tam sayı PID kontrolcüsü gerçek zamanlı kararlı bir sistemde karşılaştırılmak istenmiştir ve bunun için top dengeleme düzeneği seçilmiştir (Keçeci vd. 2019). Top dengeleme düzeneğinde topun pozisyonunu bulmak için dirençli dokunmatik ekran ya da kamera ile görüntü işleme yapılabilir. Ayrıca top dengeleme düzeneğinin tablası kare ya da daire olabilir.

Top dengeleme düzeneğini dengede tutmak için kontrolcü gerekmektedir. PID kontrolcüsü oran-integral-türev terimlerinden oluşmaktadır. James Watt buhar motorunu, ağırlıklı mekanik regülatör ile donatmıştır, bu ilk oransal (P) kontrollü mekanik geri beslemedir. G.R. de Prony integral kontrollü regülatör tasarlamıştır. H.N. Throop oransal ve türevli regülatör tasarlamıştır. İlk PID kontrolcüsü gemi dümen mekanizmasını otomatikleştirmek için Elmer Sperry tarafından oluşturulmuştur (O'Dwyer 2005). PID kontrolcüsü endüstrinin çeşitli alanlarında en çok kullanılan kontrolcüdür ve kontrol alanında en çok uygulama ve araştırılma yapılan kontrolcüdür (Aldemir ve Anwer 2020). PID kontrolcüsü parametrelerini bulmak için çeşitli metotlar geliştirilmiştir. Bunlardan en bilineni Ziegler-Nichols metodudur. John Ziegler ve Nathaniel Nichols tarafından 1942 yılında PID kontrolcü parametrelerinin belirlenmesi için basamak yanıtı ve frekans yanıtı olarak 2 metot ortaya koymuştur (Develi 2005). Bir diğer bilinen metot olan Cohen-Coon metodu 1953 yılında G.H. Cohen ve G.A. Coon tarafından bulunmuştur. Kablosuz sıcaklık kontrolü için PID kontrolcüsü kullanılmıştır ve kontrolcü parametrelerini bulmak için Ziegler-Nichols metodu, Cohen-Coon metodu ve deneme yanılma metodu kullanılmıştır (Aldemir ve Hapoğlu 2016). Akış kontrol sistemi işlem kontrolü için PID kontrolcüsü kullanılmıştır ve PID kontrolcü parametreleri Ziegler-Nichols metodu ve Cohen-Coon metodu ile bulunmuştur (Hambali vd. 2014). Manyetik kaldırma kontrol sistemi için PID kontrolcüsü kullanılmıştır ve PID kontrolcüsü parametreleri Ziegler-Nichols metodu ve Cohen-Coon metoduyla bulunup karşılaştırılmıştır (Isdaryani vd. 2020).

Top ve tabla sistemi kontrolcüsü için parçacık sürüsü optimizasyonu algoritması dâhil farklı algoritmalar kullanılmıştır. Parçacık sürüsü optimizasyonu ve diğer evrimsel optimizasyon algoritmaları, kısa kod satırlarıyla ifade edilebildiği için basit programlanabilir yapılarıyla mikrodenetleyicilerde kolayca kullanılabilmektedir (Gozde 2019). Parçacık sürüsü optimizasyonu, herhangi bir problemi optimize etmek için kullanılan basit bir algoritmadır (Kennedy ve Eberhart 1995). Bir su tankının seviye kontrolü için PID kontrolcüsü kullanılmıştır ve kontrolcü parametreleri bulmak için parçacık sürüsü optimizasyonuyla Ziegler-Nichols yöntemi karşılaştırılmıştır. Parçacık sürüsü optimizasyonu, Ziegler-Nichols yöntemindeki yüzde aşımını elimine etmiştir (Edaris ve Rahman 2016). Ateşböceği algoritması, ateşböceklerinin doğada ışık saçmalarından yola çıkarak oluşturulmuştur (Yang 2009). Top ve dengeleme düzeneği kontrolcüsü için ateşböceği algoritması dahil bazı optimizasyon algoritmaları kullanılmıştır (Borah vd. 2018). Arılar algoritması, 2005 yılında optimizasyon algoritması olarak sunmuştur (Pham vd. 2005). Yapay arı kolonisi algoritması, bal arı sürüsünün yiyecek arama davranışı üzerine kurulmuştur (Karaboga 2005). Yabani otlar, güçlü, istilacı büyüyen, bitkiler için ciddi tehdit oluşturan ve onları tarım için tehdit haline getiren bitkilerdir. Yabani otlar çok sağlamdır ve çevredeki değişime kolay adapte olurlar. Yabani ot algoritması, otların kolonileşmesi göz önüne alınarak ortaya konulan bir optimizasyon algoritmasıdır (Mehrabian ve Lucas 2006). Yabani ot optimizasyon algoritması top dengeleme düzeneğinde kontrolcünün parametrelerini bulmakta kullanılmıştır. Yapılan çalışmada amaç topa yörünge takibi yaptırmaktır. Yabani ot algoritmasının bulduğu sonuçlar sisteme uygulanmıştır ve düşük takip etme hatasıyla karşılaşılmıştır (Ali vd. 2019).

PID kontrolcüsünün katsayılarını bulmak için ateşböceği algoritması ve Ziegler Nichols metodu kullanılmıştır. Ateşböceği algoritması daha iyi, daha verimli olduğu gösterilmiştir (Bendjeghaba vd. 2013). PID kontrolcü parametrelerini belirlemek için arılar algoritması kullanılmıştır. Arılar algoritması, genetik algoritma ve Ziegler Nichols ile karşılaştırılıp yüzde aşım ve oturma süresinde daha iyi sonuçlar elde edilmiştir ve bu sonuçların ardından kontrolcü parametrelerini belirlemede iyi bir algoritma olacağı ileri sürülmüştür (Amirinejad vd. 2014). Dört ayaklı bir robotun bir ayağı için PID katsayılarını bulmak adına arılar algoritması uygulanmıştır ve arılar algoritmasının efektif ve güçlü olduğu

söylenmiştir (Bakırcıoğlu vd. 2016). Yapay arı kolonisi algoritmasıyla kolay, yüzde aşımı ve oturma süresi iyi olarak yüksek dereceli salınımlı sistemleri için PID kontrolcüsü katsayıları belirlenmiştir (Bagis ve Senneber 2017). PID kontrolcüsünün ve kademeli PID kontrolcüsünün parametrelerini bulmak için yabani ot algoritması uygulanmıştır (Patel vd. 2018). Fırçasız DC motor hız kontrolünde PID kontrolcüsü kullanılmıştır ve PID katsayılarını bulmak için parçacık sürüsü optimizasyonu ile ateşböceği optimizasyonu, yükselme süresi, oturma süresi, yüzde aşımı ve kalıcı durum hatası ile karşılaştırılmıştır. Ateşböceği algoritması daha iyi bulunmuştur (Merugumalla vd. 2018). Zaman gecikmeli iki dereceli bir sistem için yapay arı kolonisi algoritması ve parçacık sürüsü optimizasyon algoritması ile PID ve kesir dereceli PID katsayıları bulunmuştur. Sistem yüzde aşımı, yükselme süresi, oturma süresi ve amaç fonksiyonunun değeri üzerinden değerlendirilmiştir. ABC optimizasyon algoritması zaman gecikmeli durumda daha iyi performans vermiştir ve daha sağlamdır *(robust)* (Bingul ve Karahan 2018). Top ve çubuk sisteminde daha iyi bir kontrol sağlamak için PD+P kaskad kontrolcü kullanılmıştır ve bu kontrolcülerin parametrelerini optimize etmek için yapay arı kolonisi algoritması kullanılmıştır (Bilgiç vd. 2021).

PID kontrolcüsü yüksek dereceli sistemlerde istenilen sonuçları vermeyebilir. İstenilen performansa ulaşılması için, Seul Jung ve Richard C. Dorf 1995 yılında PIDA kontrolcüsünü ileri sürmüştür (Jung ve Dorf 1995). PIDA kontrolcüsü PID kontrolcüsüne ivme eklenilerek oluşturulmaktadır ve yüksek dereceli sistemlerde daha etkin cevaplar vermektedir. PIDA kontrolcüsünün parametreleri parçacık sürüsü optimizasyon algoritması ile bulunabilir (Donuk vd. 2018). Otomatik voltaj regülatör güç sistemi PIDA kontrolcüsüyle tasarlanmıştır. PIDA kontrolcü parametrelerini bulmak için ateşböceği algoritması kullanılmıştır (Sambariya ve Paliwal 2016). Kamyon frenleme sistemi için PID ve PIDA kontrolcüleri tasarlanmıştır. PIDA kontrolcüsünün PID kontrolcüsünden daha yumuşak, hızlı tatmin edici ve umut verici bulunmuştur (Nawikavatan ve Puangdownreong 2019). Araba süspansiyon sistemi için PIDA ve PID kontrolcüleri tasarlanmıştır. PIDA kontrolcüsü PID kontrolcüsüne göre daha hızlı ve tatminkâr sonuçlar vermiştir (Romsai ve Nawikavatan 2019). Kalp pili kontrolcüsü için PID ve PIDA kontrolcüleri kullanılmıştır, PIDA kontrolcüsünün PID kontrolcüsünden daha hızlı ve daha az salınımlı olduğu gözlenmiştir (Nawikavatan vd. 2019). Tank seviye

probleminde PID ve PIDA kontrolcüleri kullanılmıştır. PIDA kontrolcüsü PID kontrolcüsüne göre daha sağlam *(robust)* sonuçlar vermiştir (Jitwang vd. 2020).

PID kontrolcüsünde oluşan yüzde aşımını ya da geç oturma süresini azaltmak için PID kontrolcüsüne filtreler eklenebilir. Eklenen bu filtreli kontrolcüye PIDC kontrolcüsü denmektedir. P, PI, PID ve PID'ye eklenen faz ilerletici karşılaştırıcısı kontrolcülerinin kıyaslanması için bode diyagramı ve root locusları çizilmiş ve sistem kararlılığı karşılaştırılmıştır. Sisteme faz ilerletici karşılaştırıcı eklenmesiyle, bode diyagramı sonucuna göre yüksek frekansta sistemin kararlılığı artmıştır, root locusa göre ise sonuç s düzleminde sistemi sol tarafa kaydırmıştır, yani sistemi daha kararlı bölgeye itmiştir (Sreeraj 2013). PID kontrolcüsüne seri faz geciktirici ve ilerletici filtresi eklenmiştir ve PIDC kontrolcüsü oluşturulmuştur. PID ve PIDC kontrolcüleri gecikmesiz sisteme uygulanmıştır ve PIDC kontrolcüsünün PID kontrolcüsüne göre daha iyi performans gösterdiği ve daha sağlam *(robust)* olduğu sonucuna varılmıştır (Boskovic vd. 2017). Probleme göre kontrolde en büyük problemlerden biri ani yükselme olabilmektedir. Bunu önüne geçebilmek için PID kontrolcüsüne kademeli olarak faz ilerletici filtresi eklenmiştir. PIDC kontrolcüsü PID kontrolcüsünde oluşan yüzde aşımını tamamen elimine etmiştir ve daha kısa oturma süresi sağlamıştır (Kumar ve Gowthaman 2017).

Tek bir türbinle 3 eşit olmayan alan termal sistemin otomatik üretim kontrolü için 2-Dof-PID kullanılmıştır ve 2-Dof-PID katsayıları ateşböceği algoritmasıyla bulunmuştur (Dash vd. 2014). Otomatik voltaj regülatörü kontrolü için 2-Dof-PID kontrolcüsü önerilmiştir ve kontrolcü parametrelerini bulmak için parçacık sürüsü optimizasyon algoritması kullanılmıştır. 2-Dof-PID kontrolcüsü sayesinde, yük değişimindeki bozulma giderilmiştir (Indhuja ve Kamaraj 2016). Çok kaynaklı güç sisteminin yük frekansı kontrolü için gri kurt optimizasyonu kullanılarak 2-Dof-PID kontrolcüsü tasarlanmıştır. Önceden yayımlanan yayınlarla yüzde aşımı ve oturma süresiyle kıyaslanmıştır (Debnath vd. 2016). Hibrit araç tahrik sisteminin dinamik kontrolü için 2-Dof-PID ile PID kontrolcüleri karşılaştırılmıştır. Sonuca göre 2-Dof-PID kontrolcüsü PID kontrolcüsüne göre daha sağlamdır *(robust)* (Hmidi vd. 2017). Motorun faz hataları motorun hızını etkilemektedir. 2-Dof-PID kontrolcüsü bu faz hatalarını giderebilmektedir. Anahtarlamalı relüktans makinaları için 2-Dof-PID kontrolcüsü tasarlanmıştır ve PID

kontrolcüsüyle kıyaslanmıştır (dos Reis vd. 2019).

Bulanık mantık kontrolü, sistemin tam bir matematiksel modellemesini veya karmaşık hesaplamalarını gerektirmeyen, tamamen farklı bir yaklaşım sunar, aslında insanın sistemin davranışını anlama yeteneğine dayanır. Bulanık sistemde **if-then** kurallarını düzgün ayarlamak önemlidir, bulanık kontrol terimleriyle insan sezgileri benzerdir (Yaovaja 2018). İnsan bilgisine, gözlemine dayandığı için kolay ve güvenilir olmaktadır. Bulanık-PID kontrolcüsü birinci ve ikinci dereceden sistemlerin kontrolü için PLC'ye uygulanmıştır (Karasakal vd. 2005). Bulanık kontrolcü parametresi bulmak için arılar algoritması kullanılmıştır (Pham vd. 2007). Ters çevrilmiş sarkaç sistemi için Bulanık-PID kontrolcüsü tasarlanmıştır ve parametrelerini bulmak için yapay arı kolonisi ile klonal seçim algoritması hibritleştirilerek bulunmuştur (Tien ve Li 2012). 6 serbestlik dereceli insansız su altı aracının 3 açı kontrolü için 3 tane Bulanık-PID kontrolcüsü kullanılmıştır. (Mashhad ve Mashhadi 2013). Su kömür karıştırıcı gaz haline dönüştürücü bir sistemin sıcaklık ölçümünde bulanık kontrol kurallarını belirlemek için parçacık sürüsü optimizasyon algoritması kullanılmıştır. Algoritmanın optimal sonuca ulaştığı belirtilmiştir (Zhou 2013). Güneş enerjili arabalarda güneş pilinin en yüksek güç noktası kontrolünde bulanık mantık kontrolcü kullanılmış ve bulanık kontrolcünün üyelik fonksiyonlarının ayarlanmasında ateşböceği algoritması uygulanmıştır (Ajiatmo vd. 2016). Birleştirilmiş tank sisteminde için su seviyesi kontrolü için Bulanık-PID kontrolcüsü kullanılmıştır. Parçacık sürüsü optimizasyonuyla kontrolcü parametreleri bulunmuştur ancak parçacık sürüsü optimizasyon algoritması sistemi kararlı duruma getirmesi uzun denemeler almıştır (Shaikh vd. 2019). Çift yıldızlı indüksiyon motorunun tork kontrolünde bulanık PID kontrolcüsü kullanılmıştır. Parçacık sürüsü optimizasyon algoritmasıyla bulunan sonuçların tork dalgalanmalarını önemli derecede düşürdüğünü ve kalıcı durum hatasını ortadan kaldırdığını göstermiştir (Boukhalfa vd. 2019).

Mobil robotun yol planlama problemi için yabani ot algoritması kullanılmıştır (Mohanty vd. 2014). İnsansız hava muharebe aracının yörünge planı için yabani ot algoritması kullanılmıştır (Kumar vd. 2020). Yapay arı kolonisi algoritması ve Lagrange metodu ile elektrik güç sistemi yakıt maliyeti hesaplanmıştır. Yapay arı kolonisi algoritmasıyla daha optimal sonuçlar elde edilmiştir (Ozturk vd. 2011).

Optimizasyon algoritmaları hatayı minimize ve ya maksimize etmek için amaç fonksiyonu kullanmaktadırlar. Performans ölçmek için literatürde farklı amaç fonksiyonları vardır. Bunlardan en çok bilinen ve kullanılanları şunlardır, mutlak hatanın toplamı (IAE), zamanla mutlak hatanın toplamı (ITAE), hatanın karelerinin toplamı (ISE), zamanlı hatanın karelerinin toplamı (ITSE) . Bu amaç fonksiyonları meydana gelen hatayı hesaplamaktadırlar (Kılıç ve Özdemir 2018). IAE, ITAE, ISE, ITSE amaç fonksiyonlarının denklemleri sırasıyla 2.1, 2.2, 2.3 ve 2.4'de verilmiştir.

$$\int_0^\infty |e| dt \tag{2.1}$$

$$\int_0^\infty t|e| dt \tag{2.2}$$

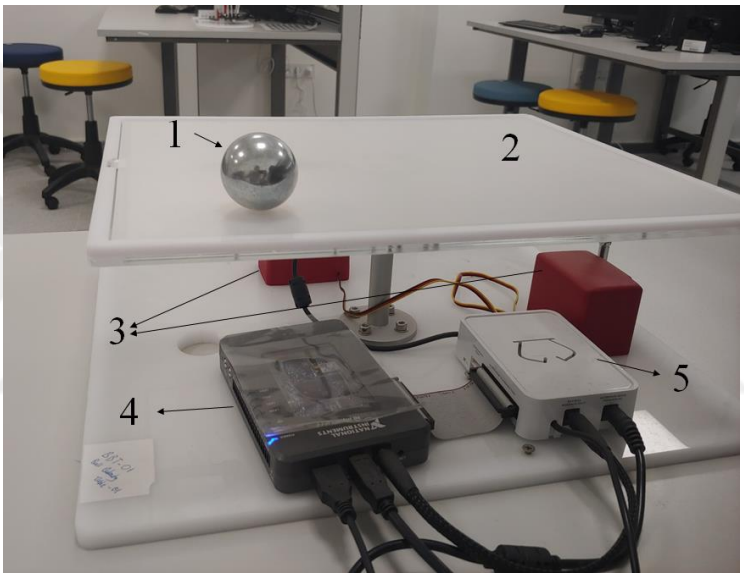
$$\int_0^\infty e^2 dt \tag{2.3}$$

$$\int_0^\infty te^2 dt \tag{2.4}$$

Amaç fonksiyonları da hata *e*, zaman *t* olarak gösterilmiştir. Buradaki hata girişte istenilen konumla çıkışta alınan konum arasındaki farktır. IAE amaç fonksiyonu olarak kullanıldığında endişe edilen şey hata oluştuğunda oluşan hatanın süresi ya da kalıcılığı değil, hatanın büyüklüğüdür. ITAE amaç fonksiyonu ilk olarak 1953 yılında Graham ve Lathrop tarafından önerilmiştir (Graham ve Lathrop 1953). ITAE amaç fonksiyonunun önerilmesinin sebebi, zaman geçtikçe oluşan hataları daha öncelikli kılmaktır. ISE amaç fonksiyonunda endişe edilen nokta IAE amaç fonksiyonuyla aynıdır. ISE hataların karelerini aldığından dolayı hatayı iyice yükseltmektedir. ITSE amaç fonksiyonu 1993 yılında Chen tarafından önerilmiştir (Chen 1993). Amaç fonksiyonu olarak istenilen fonksiyon kullanılabilir (Silva vd. 2019).

3.MATERYAL VE YÖNTEM

Dengeleme sistemleri kontrol alanındaki önemli çalışma konularından biridir. Tabla üzerindeki dengeleme sistemleri, iletişim protokolleri, gömülü uygulamalar ve kontrol uygulamaları gibi çeşitli uygulamaların teorik ve pratik yönleri göstermektedir (Braescu vd. 2012). Nesnelerin dengelenmesi özellikle robot üreticileri tarafından halen çalışma ve gelişme aşamasında olan önemli bir konudur (Itani 2017). Top dengeleme düzeneği, uygunsuz hava koşullarında, uçağın inişi sırasında yatay olarak dengelenmesi gibi kontrol problemleriyle de bağlantılıdır.



Şekil 3.1 Top Dengeleme Sistemi

3.1 Top Dengeleme Düzeneği

Bu çalışmada Acrome firması tarafından üretilen top dengeleme düzeneği kullanılacaktır. Şekil 3.1'de 1 numara çelik topu, 2 numara 2 boyutlu dokunmatik sensör ve kontrolcüyü, 3 numara RC servo motorları, 4 numara NI myRIO'yu, 5 numara güç dağıtım kutusunu göstermektedir. RC Servo Motorlar, elektrik sinyalini fiziksel bir harekete, dönme momentine dönüştüren elektromekanik cihazlardır. Kullanıldıkları alanlarda kolay ve basit çözümler sunmaktadırlar. Kullanıcıya robotik sistemleri kontrol etmelerine ve/veya tasarım yapmalarına imkân sunan taşınabilir bir gömülü cihazdır. ACROME Güç dağıtım

kutusu, myRIO'nun ve servoların bağlantıları bu güç dağıtım kutusunun üzerinde yer almaktadır. Topun tabladaki pozisyon bilgisi 17 inç dokunmatik ekran ile sağlanır. Dokunmatik ekran temel olarak, iki tabakadan oluşur. Bu tabakalar üzerinde dirençler tarafından üretilen bir voltaj oluşur. Bu tabakaları birbirinden ayıran şey hava boşluğudur. Bu iki tabaka birbirine değdiğinde, voltaj iki tabakaya ayrılır, iki tabakada oluşan bu voltajlar ekran koordinatlarına çevrilir. Bu işlem dokunmatik ekran denetleyicisi tarafından yapılır (Acrome 2014).

3.2 Top Dengeleme Düzeneği Fiziksel Modeli

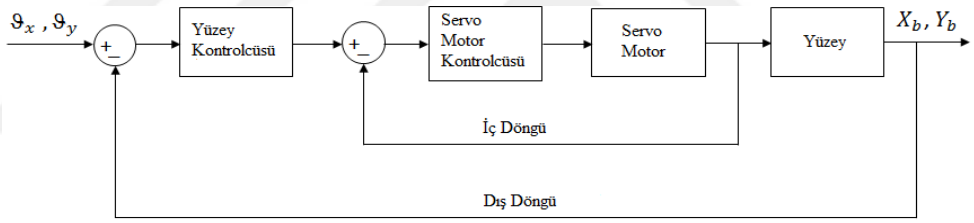
Fiziksel modelleme için Lagrangian yöntemi kullanılmıştır. Lagrangian yöntemi, genellikle karmaşık ve yüksek serbestlik dereceli sistemler için kullanılır. Aşağıdaki çizelgede top dengeleme düzeneğinin modellemesinde kullanılan parametreler gösterilmektedir.

Çizelge 3.1 Top Dengeleme Düzeneği Semboller, Açıklamalar ve Değerler

Semboller	**Açıklama**	**Değerler**
L_x	X yönünde kiriş uzunluğu	0.134
L_y	Y yönünde kiriş uzunluğu	0.168
r_m	Motor kol uzunluğu	0.0245
r_b	Topun yarıçapı	0.02
m_b	Topun kütlesi	0.26
j_b	Topun dönme ataleti	$4.16x10^{-5}$
g	Yerçekimi ivmesi	9.81
α	X yönünde tabla açısı	Değişken
β	Y yönünde tabla açısı	Değişken
ϑ_x	X yönünde motor açısı	Değişken
ϑ_y	Y yönünde motor açısı	Değişken

3.3 Top Dengeleme Düzeneğinin Kontrolü

Top ve tabla sisteminin konum kontrolü, topu levha üzerinde belirli bir pozisyonda tutmaktır (Dong vd. 2009). Topun tabla üzerinde dengelenmesi istendiğinden dolayı sürekli kontrol edilmesi gerekir (Kostamo vd. 2005). Topun dengelenmesi ve hareket ettirilmesi için kontrol sistemi gereklidir (Mujadin ve Pratama 2017, Kopichev vd. 2019). Literatür taramasında top dengeleme düzeneğinde kullanılan kontrolcüler anlatılmıştır. En çok PID kontrolcüsünün kullanılmasının sebebi, PID kontrolcüsünün kolay yapısı ve uygulanabilirliği yüzündendir. Bu tez kapsamında PID, PIDA, PIDC, 2-Dof-PID ve Bulanık-PID kontrolcüleri kullanılacaktır. Topun tabladaki konumunu korumak için iki döngülü bir otomatik kontrol sistemi temel alınmıştır. Dış döngü, topun pozisyonunu (dokunmatik dirençli panel ile elde edilen veriler) ve iç döngü tablanın pozisyonu (enkoderden elde edilen veriler) kontrol etmekten sorumludur. Şekil 3.2'de top dengeleme düzeneğinin iki döngülü şeması gösterilmektedir.

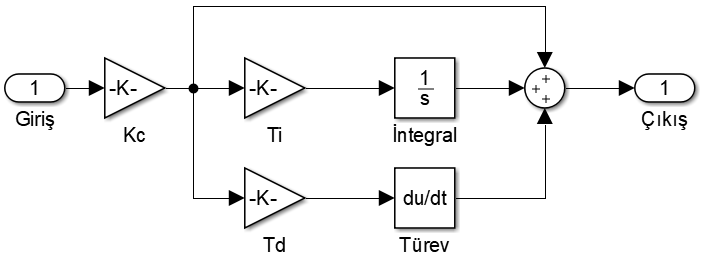


Şekil 3.2 Top Dengeleme Düzeneği Kapalı Döngü Şeması

İç döngü, geri beslemeli bir kapalı döngü sistemidir. İç döngü, servo motorların bulunduğu konumu kontrol eder. Top dengeleme düzeneğinin servo motoru, denetleyicisi sayesinde konumunu otomatik olarak kontrol eder. İç döngü anında sabit duruma gelir. Bu nedenle, iç döngü transfer fonksiyonunu bulmak kolaydır. Dış döngüyü kontrol edebilmek için kontrolcü kullanılmıştır. Bu tez kapsamında kullanılacak kontrolcüler şu şekildedir, PID, PIDA, PIDC, 2-Dof-PID, Bulanık-PID. Sistemin kontrolü, topu belirli bir konumda tutmak veya tabladaki belirli yörünge üzerinde hareket halinde tutmaktır.

3.3.1 PID

Kontrolcü parametrelerini belirlerken, ilk koşul kararlılık olmalıdır. Bir PID kontrolcüsünün tasarlanması ve ayarlanması kolay gözükebilir, ancak kısa tepki süresi ve yüksek kararlılık gibi performans hedefleri olduğunda tasarlamak zor olabilir. PID kontrolcüsü farklı şekillerde (paralel, seri, ideal) kullanılabilir. Paralel PID kontrolcüsünün genel formülü denklem 1.3'deki gibidir. Bu tez kapsamında ideal PID kontrolcüsü kullanılmıştır. Şekil 3.3'de ideal PID kontrolcüsünün genel şeması verilmiştir.

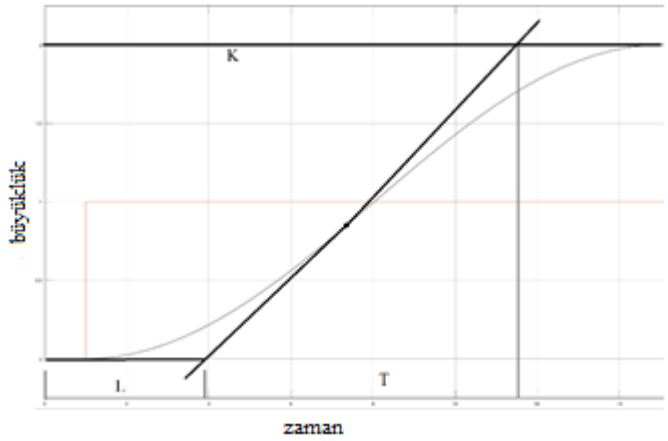


Şekil 3.3 İdeal PID Kontrolcüsü Genel Şeması

Denklem 3.1 ideal PID kontrolcü denklemini göstermektedir.

$$u(t) = K_c * \left(e(t) + \frac{1}{T_i} * \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right) \tag{3.1}$$

PID kontrolcüsünün katsayıları farklı metotlarla bulunabilir. PID kontrolcüsünün katsayıları bulunması için literatürde birçok metot önerilmiştir. Bunlardan bazıları, deneme yanılma yöntemi, Ziegler-Nichols, Cohen-Coon, Chien-Hrones-Reswick, Wang-Juang-Chan'dir (Gidemen ve Furat 2015). Bu tez kapsamında Ziegler Nichols metoduyla katsayılar bulunmuştur. Ziegler Nichols metodunun 2 yöntemi vardır. Ziegler Nichols metodunun ilk yöntemi, gecikme zamanı (ölü zaman) L ve zaman sabiti T, eğrinin yönünün değiştiği noktadan bir doğru çizilerek belirlenebilir. Şekil 3.4'de oluşan eğriden elde edilen katsayılar ise çizelge 3.2'de uygulanarak PID katsayıları bulunur.



Şekil 3.4 Ölü Zaman ve Zaman Sabiti

Çizelge 3.2 Ziegler Nichols 1.Metod

	Kp	**Ki**	**Kd**
P	$1.2\text{x}\frac{\text{T}}{\text{L}}$	∞	0
PI	$0.9\text{x}\frac{\text{T}}{\text{L}}$	$\frac{\text{L}}{0.3}$	0
PID	$1.2\text{x}\frac{\text{T}}{\text{L}}$	2xL	0.5xL

Ziegler Nichols'un ikinci metoduysa, Düşük bir kazanç (K_p) değeriyle başlayıp yavaş yavaş arttırılarak denge durumunda salınım yapana kadar arttırılır ve bu değer K_{cr} olarak alınır. Bu metoda ise değerler Çizelge 3.3'deki gibi bulunur.

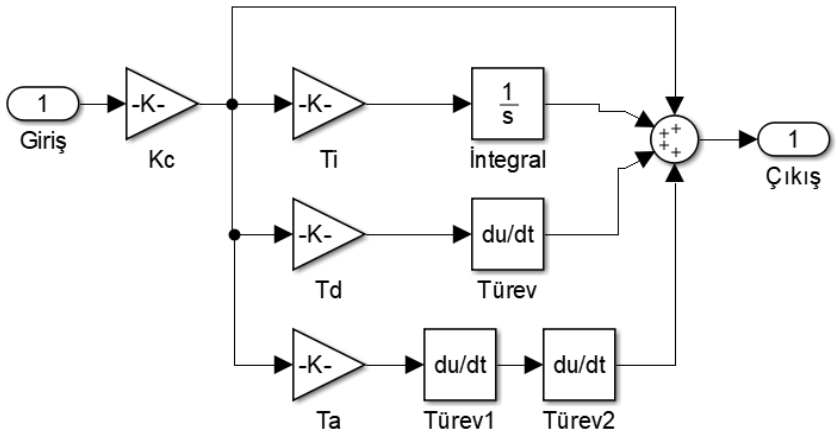
Çizelge 3.3 Ziegler Nichols 2.Metod

	Kp	**Ki**	**Kd**
P	$0.5\text{x}K_{cr}$	∞	0
PI	$0.45\text{x}K_{cr}$	$\frac{1}{1.2}\text{x}P_{cr}$	0
PID	$0.6\text{x}K_{cr}$	$0.5\text{x}P_{cr}$	$0.125\text{x}P_{cr}$

PID kontrolcüleri çok avantaja sahip olsalar da, P, I ve D parametrelerinin seçilmesi zor olabilmektedir. PID kontrolcüsünün yeterli olmadığı zamanlarda, probleme göre yüzde aşımı yüksek olduğu veya oturma süresi uzun olduğu durumlar gibi performansın iyi olmadığı durumlarda PID kontrolcüsüne filtre yada ikinci türev gibi eklemeler yapılarak, gibi daha iyi sonuçlar elde edilebilir.

3.3.2 PIDA

PID kontrolcüsü probleme bağlı olarak 1 ve 2 dereceli sistemlerde iyi çalışırken daha yüksek dereceli sistemlerde yetersiz kalabilmektedir bu sebeple yüksek dereceli sistemlerin kontrolü için Oransal-Integral-Türev-İvme (PIDA) kontrolcüsü kullanılabilir (Donuk vd. 2018). PIDA kontrolcüsü klasik PID kontrolcüsüne ivme ilave edilerek elde edilir. PIDA kontrolcüsü PID kontrolcüsüne göre daha kararlı ve daha az yüzde aşımı sağlar (Puangdownreong 2012). PID kontrolcüsüne göre bir tane sıfır eklenmiştir, bu sebeple daha kararlı çalışmaktadır. Bu tez kapsamında kontrolcü parametreleri optimizasyon algoritmasıyla bulunacak olsa da, PIDA kontrolcüsünde kazançları bulmak için farklı metotlar kullanılabilir. En çok bilinen metotlar; Jung-Dorf Yaklaşımı, Kitti's Method, Katsayılar diyagramı metodu (Sambariya ve Paliwal 2016).



Şekil 3.5 İdeal PIDA Kontrolcüsü Genel Şeması

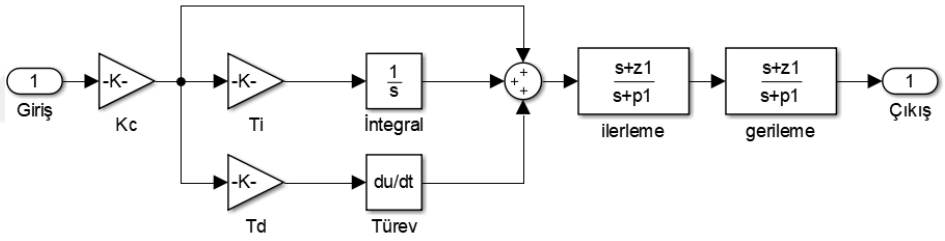
Şekil 3.5'de ideal PIDA kontrolcüsü genel şeması gösterilmektedir. Denklem 3.2'de ideal PIDA kontrolcüsünün denklemini göstermektedir.

$$u(t) = K_p * (e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt} + T_a\frac{de^2(t)}{dt^2} \quad (3.2)$$

3.3.3 PIDC

PID kontrolcüsüne seri filtre eklendiğinde mutlak hata daha fazla azaltılır. PID kontrolcülerinin sağlamlığı ve performansı PIDC kontrolcüsü kullanılarak daha da geliştirilebilir (Boskovic vd. 2016; Boskovic vd. 2017). Denklem 3.3'de PIDC kontrolcüsünün genel denklemi verilmiştir. $F_{NF}(s)$ filtreyi temsil etmektedir. Şekil 3.6'da ideal PID kontrolcüsüne faz geciktirici ve ilerletici filtresi eklenmiş PIDC kontrolcüsü gösterilmektedir.

$$C_{PIDC}(s) = \left(K_p + \frac{K_i}{s} + K_d s\right) * F_{NF}(s) \quad (3.3)$$



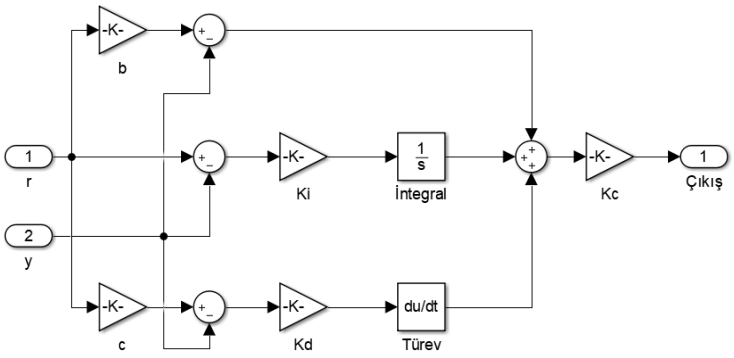
Şekil 3.6 PIDC Kontrolcüsü Genel Şeması

İstenilen filtre eklenerek kullanılabilir. Bu tez kapsamında eklenen filtre faz geciktirici ve ilerletici filtresi olmuştur.

3.3.4 2-Dof-PID

PID kontrolcüsü oran ve türev terimleri nedeniyle istenmeyen kontrol eylemlerine neden olabilir. Referansta ani değişiklik olduğunda ve kısa sürede hatanın büyük varyasyonları olduğunda, bu terimlerin eylemleri kontrol sinyalinde anlık yükselmelere yol açabilir. Bu tip kontrol sinyali davranışını önlemek için, 2-Dof-PID gibi kontrolcüler kullanılır (dos Reis 2019). 2-Dof PID kontrolcüsü, yüzde aşımı yapmadan sistemi kontrol edebilir. Bu

kontrolcü, referans sinyalindeki ani değişikliklerin kontrol sinyali üzerindeki etkisini azaltmak için de faydalıdır. 2-Dof-PID kontrolcülü sistem probleme göre, geleneksel PID kontrolcülü sistemden daha verimli olabilir, çünkü PID kontrolcüsüne göre oturma süresi daha kısa olabilmektedir. Tüm denetleyicilerin kapalı döngü performansları, dışardan harici bir giriş olmadığı durumda birbirlerine yakındır. İstenmeyen girişler olduğunda, 2-Dof PID kontrolcüsü bu istenmeyen girişleri daha hızlı elimine edecektir ve sistemi istenen hızda istikrarlı hale getirecektir (Hmidi vd. 2017). *b*, oransal olarak ağırlığı, *c* ise türev terimi üzerindeki ağırlığı temsil eder. Şekil 3.7'de ideal 2-Dof PID kontrolcüsünün içyapısı gözükmektedir.



Şekil 3.7 2-Dof PID Kontrolcüsü Genel Yapısı

Denklem 3.7'de ise ideal 2-Dof-PID kontrolcüsünün genel denklemi gösterilmektedir.

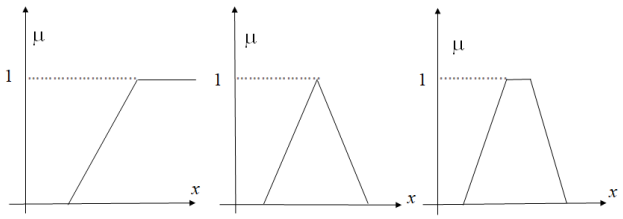
$$u(t) = K_c * \left(b + \frac{K_i}{s} + \frac{cK_d s}{T_f s + 1}\right) - K_c * \left(1 + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1}\right) \qquad (3.4)$$

3.3.5 Bulanık Mantık Denetleyicisi

Bulanık mantık kontrolü, sistemin tam bir matematiksel modellemesini veya karmaşık hesaplamalarını gerektirmeyen, tamamen farklı bir yaklaşım sunar. Bulanık sistemde **if-then** kurallarını düzgün ayarlamak önemlidir (Yaovaja 2018). Bu tür bir yaklaşım, sistemlerin temel fiziksel özelliklerine bağlı değildir ve diğer kontrol tekniklerinin başarısız olduğu çalışma koşullarında bile kontrolü sağlayabilir (Nasir vd. 2011). Bulanık

kontrolcüde kural sayısını arttırmak sonuçları daha doğru yapacaktır ve sistemin kararlı duruma ulaşması daha hızlı olacaktır (Aras vd. 2009).

Klasik küme, kesin bir sınıra sahiptir. Bir kümeye ait olmamadan, bir kümeye ait olmamaya geçiş nettir. Bir eleman kümeye aitse üyelik 1'dir, değilse üyelik 0'dir. Bulanık kümelerde ise bu işlem daha farklıdır. Üyelik değerleri 1 ve 0 ile sınırlı değildir. Üyelik değerlerini atamak için üyelik fonksiyonları gerekmektedir. Üyelik fonksiyonu, giriş uzayındaki her bir noktayı [0 1] aralığında çıkışa atayan bir eğridir. Bulanık kümede işlem yapmak için şunlar kullanılabilir, en yüksek operatör, cebirsel toplam, sınırlı toplam, kesişim. Bazı üyelik fonksiyonları şekil 3.8'de gösterilmiştir.



Şekil 3.8 Monotonik, Üçgen, Trapezoidal Üyelik Fonksiyonları

Bulanık kontrolün genel kural yapısı şöyledir;

1- **IF** ifade **THEN** durum

2- Çıkarım mekanizması, hangi kuralların mevcut durumla ilgili olduğunu belirler.

3- Bulanıklaştırma arabirimi, kural tabanlı tabloyu kullanarak bir giriş değeri için üyelik fonksiyonlarının değerlerini hesaplar.

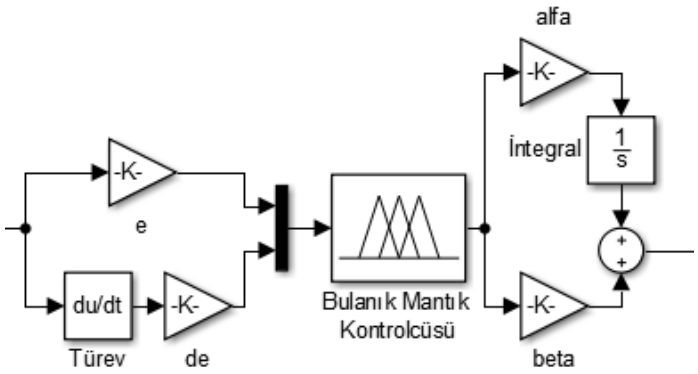
4- Bulanıklaştırma arabirimi, bulanık kümeyi en iyi temsil eden değeri belirler. Alan merkezi, maksimumdaki ortalama, maksimumdaki en büyük ve maksimumdaki en küçük gibi bulanıklaştırma yöntemleri kullanılabilir.

Literatürde en çok bilinen ve kullanılan çıkarımlar Mamdani ve Sugeno çıkarımlarıdır. Mamdani çıkarımında giriş ve çıkışlar bulanık değerlerdir. Giriş değerlerinin tetiklediği kurallara göre, üyelik değerleri hesaplanır. Daha sonra hesaplanan değerler, kuralların içerisinde geçen ve/veya mantıksal bağlaçlarına göre en yüksek ya da en düşük

operatörüne verilirler. Eğer, kural içerisinde geçen olgular birbirine ‘ve’ ile bağlıysa, hesaplanan üyelik değerleri en düşük operatörüne; ‘veya’ ile bağlıysa en yüksek operatörüne verilir. Bu operatörler, adlarından da anlaşılabildiği gibi, aldıkları birden çok değer arasından en küçüğü ya da en büyüğü döndürürler.

Sugeno çıkarımı, çıkış değerini bir fonksiyon şeklinde vermektedir. Mamdani yönteminde, giriş değerlerinin yanında çıkış değeri de bir bulanık küme olarak tanımlanır. Sugeno’da ise, girişler bulanık kümeler şeklinde olabilirken, çıkışın bir fonksiyon (genellikle polinom) şeklinde tanımlanmış olması gerekir. Eğer çıkış sabit bir sayıysa Takagi-Sugeno metodunun özel bir versiyonu olan Single Type (Singletion) kuralı olur.

Bulanık mantık kontrolcüsüyle PID kontrolcüsünün birleşmesiyle Bulanık-PID kontrolcüsü elde edilir. Bu şekilde PID kontrolcüsünün yüzde aşımı ve kalıcı durum hatası azaltılır. PID kontrolcüsü direkt olarak eklenmek yerine girişte kazanç ve türev parametreleri eklenir, bunun sebebi hatanın kendisini ve değişim hızını girdi olarak almaktır. Çıkışta ise integral ve kazanç parametrelerinin toplamı alınır. PD kontrolcüsü girişe, PI kontrolcüsü çıkışa eklenir. Bu şekilde çıkışta hız ve konum kontrol edilir. Bulanık-PID kontrolcüsünün, bulanık kontrolcü ya da PID kontrolcüsüne göre daha iyi olması beklenmektedir. Kötü yanı olarak parametre sayısının artması gösterilebilir. Şekil 3.9’da Bulanık-PID kontrolcüsünün genel şeması gösterilmektedir.



Şekil 3.9 Bulanık-PID Genel Yapısı

3.4 Optimizasyon

Optimizasyon, problem veya problemlerde verilen koşullarda, belirli sınırlar içerisinde tüm çözümler arasında en iyi çözümü bulma işlemidir. Optimizasyon her zaman daha iyiye gitmeyi amaçlamaktadır. Problemlerin çözümünde geçmişte matematiksel yöntemler yaygın olarak kullanılmaktaydı. Ancak matematiksel yöntemlerin esnek olmaması, probleme özel fonksiyon tanımlaması gereksinimleri ve gerçek hayattaki problemlerin matematiksel formülünün çok karmaşık olması gibi dezavantajlarından dolayı kullanımı giderek azalmıştır. Bu sebeple sezgisel algoritmalar geliştirilmiştir (Küçüksille ve Tokmak 2011). Sezgisel algoritmalar karışık matematiksel işlemlere tabi tutulmadan, en iyi sonuca ulaşmamızı sağlamaktadır. Sezgisel algoritmalar kesin çözümü garanti etmemektedir, çözümün yakınlarında bir çözüm sunmaktadır. Genel amaçlı sezgisel yöntemler; sürü tabanlı, biyoloji tabanlı, sosyal tabanlı, kimya tabanlı, fizik tabanlı ve müzik tabanlı olmak üzere altı farklı grupta değerlendirilmektedir, ayrıca bu algoritmaların birleşimiyle melez yöntemlerde oluşturulmaktadır (Akyol ve Alataş 2012). Sezgisel algoritmalar farklı problemlere, sistemlere kolayca uygulanıp kısa zamanda sonuç üretmesinden dolayı tercih edilirler. Sezgisel algoritmaların kesin çözüm sunmaması ve her algoritma için çok parametre istemesi dezavantajları olarak gösterilmektedir (Küçüksille ve Tokmak 2011). Bu çalışmada kontrolcü parametrelerini bulmak için 5 adet optimizasyon algoritması kullanılmıştır.

3.4.1 Sürü zekâsı tabanlı optimizasyon algoritmaları

Bir problemi çözmek için doğadan esinlenilerek oluşturulan algoritmalardır. Doğada tek başlarına iş yapamayan varlıklar, bir sürü olarak hareket ettiğinde anlamlı davranışlar oluştururlar. Sürü, birbirleriyle etkileşim halinde olan bireyler topluluğuna denir. Sürülerin yiyecek bulma, avlanma, hayatta kalma, en kısa yolu bulma gibi davranışları incelenerek algoritmalar oluşturulmuştur. Sürü zekası optimizasyon algoritmaları lineer olmayan problemlerde de kullanılabilir. Optimizasyonun yakınsama hızı, hesaplama süresi ve doğruluğu performans ölçütü olarak kabul edilir.

3.4.1.1 Ateş böceği algoritması

Ateş böceği algoritması Xin-She Yang tarafından, ateş böceklerinin davranışları incelenerek bulunmuştur (Yang 2008). Ateş böcekleri yapıları gereği ani ritmik ışıklar üretebilmektedir. Bu üretilen ışıklar çiftleşmek, avlarını cezbedebilmek veya avcılardan korunmak için olabilir (Belen vd. 2014). Yang ateş böceklerini modellerken bazı varsayımlarda bulunmuştur. Bunlar;

1- Tüm ateş böcekleri cinsiyetsizdir.

2- Çekicilik ışık parlaklığıyla orantılıdır, bu sebeple daha az parlak olan ateş böceği parlak olan ateşböceğine doğru hareket edecektir.

3- Eğer ikisi de eşit parlaktaysa rasgele hareket edeceklerdir.

Parametre olarak popülasyondaki birey sayısı, çekiciliğin ve parlaklığın başlangıç değeriyle birlikte emilim katsayısı değeri girilmelidir. Ateş böceklerinin ışık yoğunluğu (I) denklem 3.5'de gösterilmiştir. Işık yoğunluğu, başlangıç ışık yoğunluğu (I_o), uzaklık (r) ve ışığın sabit emilim katsayısına (y) bağlıdır.

$$I = I_o * e^{-y*r} \tag{3.5}$$

Işık yoğunluğu ortamda emilmesinden dolayı, ateş böceğinin çekiciliği (β) diğer ateş böceğinin uzaklığına bağlı olarak değişmektedir. Bu sebeple *i* ve *j* konumlarındaki ateş böcekleri arasındaki mesafeye (r_{ij}) göre farklılık olur. Çekicilik, başlangıç değeri (β_o), iki ataş böceği arasındaki mesafenin karesi ve sabit emilim katsayısına bağlıdır. Denklem 3.6'da ateş böceklerinin çekiciliği gösterilmektedir.

$$\beta = \beta_o * e^{(-y*r)^2} \tag{3.6}$$

İki ateş böceği arasındaki mesafe denklem 3.7'da gösterilmiştir.

$$R_{ij} = | x_i - x_j | = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{3.7}$$

İki boyutlu uzayda eşitlik denklem 3.8'de gösterilmiştir.

$$R_{ij} = \sqrt{\left(\mathrm{x}_i - \mathrm{x}_j\right)^2 + \left(\mathrm{y}_i - \mathrm{y}_j\right)^2} \quad (3.8)$$

Ateş böceklerinin hareketleri denklem 3.9'de gösterilmiştir.

$$x_i^{t+1} = x_i + \beta_o e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + \alpha\varepsilon_i^t \quad (3.9)$$

i konumdaki ateş böcekleri bu formülle daha çekici olan *j*'deki ateş böceğine ilerlemektedir. Bu eşitlikte α rastlantı parametresi ve ε ise gauss dağılımından gelen rasgele sayıdır. ε global aramayı sağlaması için eklenmiştir. β_o=1 ve α ile ε [0,1] arasında alınmalıdır (Boz ve Çimen 2017). FA algoritmasının sözde kodu şekil 3.10'da gösterilmiştir.

```
Ateşböceği popülasyonunu, ışık absorbe katsayısını, sabit emilim katsayısını tanımla.
while t=1'den tüm popülasyona kadar
  for i=1'den sürüdeki her ateşböceği için
    for i=1'den sürüdeki her ateş böceği için
      if I_i>I_j
        ateşböceği i'den j'ye doğru hareket eder
      end
      Çekicilik, uzaklık ve e^(-γ*r)^2 ile değişir
      Yeni çözümleri değerlendirilir ve ışık yoğunluğu güncellenir.
    end
  end
Ateşböcekleri sıralanır ve global en iyi bulunur.
end
```

Şekil 3.10 FA sözde kodu

3.4.1.2 Parçacık sürüsü optimizasyonu (PSO)

Parçacık sürüsü optimizasyonu sosyolog-psikolog James Kennedy ve elektrik mühendisi Russel Eberhart tarafından 1995 yılında bulunmuştur. PSO, kuş ve balık sürülerinin hareketlerinden esinlenilerek bulunmuştur. PSO'da her bir birey parçacık olarak, parçacıkların oluşturduğu topluluk ise sürü olarak adlandırılır. Sürülerdeki her birey belirli bir amaç için birlikte çalışmak zorundadır. Bireyler amaçlarına ulaşabilmek için kurallara uymaktadır ve elde ettiği bilgileri sürüdeki bireylerle paylaşmaktadır (Özsağlam

ve Çunkaş 2008). Parçacıklar ortamlarına adapte olma, yiyecek kaynağı bulma ve avcılardan kaçmak için bilgi paylaşırlar. Hız ve konumlarını en yakındaki parçacığa göre düzenlerler. PSO algoritmasında her parçacık potansiyel bir çözümdür (Asta ve Uyar 2011). PSO algoritmasının sözde kodu şekil 3.11'de gösterilmiştir. Sırasıyla hız ve pozisyon yenileme denklemleri 3.10 ve 3.11'de gösterilmiştir.

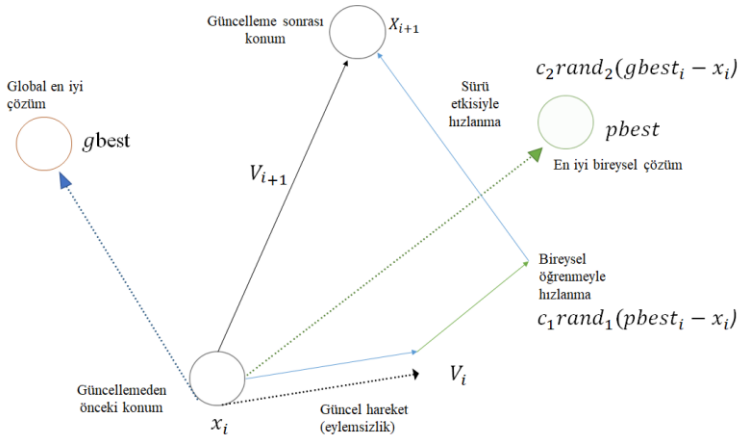
$$v_{i+1} = w * v_i + c_1 * rand_1 * (pbest_i - x_i) + c_2 * rand_2 * (gbest_i - x_i) \quad (3.10)$$

$$x_{i+1} = x_i + v_{i+1} \quad (3.11)$$

for *i*=1'den parçacık sayısına kadar
  Başlangıç durumlarını (sürüyü, sürüdeki parçacıkların hızlarını ve konumlarını) oluştur.
end
do
  for *i*=1'den parçacık sayısına kadar
    Tüm parçacıkların uygunluk değerini hesapla.
    if parçacık öncekinin en iyisinden iyiyse
      Kişisel en iyi değer
    end
  end
  for *i*=1'den parçacık sayısına kadar
    Pozisyon ve hızları yenile.
  end
while istenilen olana kadar ya da iterasyon bitene kadar

Şekil 3.11 PSO sözde kodu

c_1 *ve* c_2 öğrenme faktörüdür, c_1 bireysel öğrenmeyi gösterirken c_2 sürü etkisiyle öğrenmeyi göstermektedir. Öğrenme faktörü katsayıları düşük seçilirse hedefe yavaş ulaşılır, yüksek seçilirse global en iyiye gidememe durumu olabilir ancak hızlıca istenilene ulaşabilir. $pbest$ bireyin geçmiş konumlarından en iyi çözümü veren konum iken $gbest$ bireyler arasında en iyi konumdur. $rand$ ise (0-1) arasında düzgün dağılım ile rasgele sayı üretmektedir. Bunun sebebi algoritmanın stokastikliğini sürdürebilmesidir. w eylemsizlik ağırlığıdır, 1'den küçük seçilmeli ve her iterasyonda doğrusal olarak azaltılmalıdır (Der vd. 2008). Şekil 3.12'de parçacık sürüsündeki parçacıkların bir adım hareketi gösterilmektedir.



Şekil 3.12 Parçacık Hareketi

3.4.1.3 Yapay arı kolonisi (ABC)

Yapay arı kolonisi algoritması bal arılarının yiyecek aramasından esinlenilerek, Derviş Karaboğa tarafından 2005 yılında oluşturulmuş bir optimizasyon algoritmasıdır. Arıların yaşamlarını devam ettirebilmesi için yiyecek bulmaları gerekmektedir. Bu sebeple sürekli olarak yiyecek kaynağı arayışı içindelerdir. Kaynağı bulduklarındaysa kaynağın kovana uzaklığı, nektar miktarı gibi etkenler önemlidir. Ancak algoritmada sadece nektar miktarı ölçüt olarak alınmıştır.

Arılar görevi belirli olan işçi arılar ve belirli olmayan işçi arılar olarak iki farklı gruba ayrılır. Görevi belirli olan işçi arılar önceden belirlenmiş kaynaklara gidip kovana nektar getirmekle görevlidirler. Kaynaklarının nektar miktarı, uzaklığı, kalitesi gibi bilgileri kovanda dans ederek paylaşırlar. Görevi belirli olmayan işçi arılar ise ikiye ayrılır. Bunlar; kovanda bekleyip görevli arılardan bilgi alan arılar ve rasgele kaynak arayan kâşif arılardır. Yapay arı kolonisi algoritmasında görevli arıların sayısıyla kaynak sayısı eşittir. Kaynağındaki nektarı biten arılar gözcü arı olacaktır ve kovana gidip diğer arılardan bilgi bekleyeceklerdir. Kaynakların konumları problemlerin muhtemel sonuçları olurken, kaynakların yiyecek miktarıysa çözümün kalitesini göstermektedir.

Algoritmanın modellemesi şu şekildedir. Görevli işçi arılar kaynaklardaki nektar miktarını hesaplamaktadır. Aynı zamanda kâşif arılar başka kaynaklar aramaktadır.

İstenilen kaynak bulunana kadar en iyi kaynak hafızada tutulur. Kaynakların bulunması için alt ve üst sınırlardan yararlanılır (Küçüksille ve Tokmak 2011, Akyol ve Alataş 2012). Kaynağın yerini bulmak için denklem 3.12 kullanılır.

$$x_{ij} = x_j^{min} + rand(0,1) * \left(x_j^{max} - x_j^{min}\right) \quad (3.12)$$

x_{ij}, kaynağın yerini ifade eder. x_j^{min}, j. parametrenin alt değeri, x_j^{max} ise j. parametrenin üst değerini ifade eder. İşçi arı yeni kaynağın kalitesini değerlendirmektedir.

$$v_{ij} = x_{ij} + \Phi_y * \left(x_{ij} - x_{kj}\right) \quad (3.13)$$

Φ_y, -1 ile 1 arasında üretilen rasgele bir değerdir. Kaynağın seçilme olasılığı denklem 3.14'de verilmiştir. Denklemdeki uygunluk nektar miktarını göstermektedir (Eke vd. 2011).

$$p_i = \frac{uygunluk_i}{\sum_{i=1}^{SN} uygunluk_i} \quad (3.14)$$

Şekil 3.13'de yapay arı kolonisi algoritmasının sözde kodu verilmiştir.

```
Parametreleri ayarla (Popülasyon, konum, iterasyon)
Başlangıç popülasyonunu başlat ve uygunluğunu değerlendir, en iyi çözümü ayarla.
Do while (iterasyon<En yüksek iterasyon)
      for i=1'den tüm işçi arılar için
        Rastgele bir çözüm seç ve rastgele komşu yapısını uygula
      end
      for i=1'den tüm gözcü arılar için
        Rulet tekerleği seçimine dayalı popülasyondan çözüm seç
        Çözüm üzerinde rastgele bir komşu yapısı uygula
        if   (en iyi çözüm< bulunan yeni çözüm)
          Bulunan yeni çözüm = en iyi çözüm
        end
    end
    Şimdiye kadar bulunan en iyi çözüm
    kâşif arılar terkedilmiş besin kaynağını belirler ve yeni besin kaynağı ile değiştirir
end
```

Şekil 3.13 ABC sözde kodu

3.4.1.4 Arılar algoritması

Arılar algoritması bal arılarının yiyecek araması incelenerek, Pham tarafından 2005 yılında önerilmiştir. Arılar buldukları kaynağın bilgilerini kovana gittiklerinde dans ederek kovanda bekleyen arılara iletirler. Kaynağın kalitesi uzaklık, miktar gibi özelliklere göre belirlenir. 3 tip arı vardır. Kâşif arılar, bir bilgisi olmadan yeni kaynak arayan arılardır. Görevli arılar, yiyecek kaynağını bulmuş ve kovana dönüp bilgi veren arılardır. İzci arılar, kovanda bekleyip görevli arıların dansını izleyip bilgi edinerek kaynağa giden arılardır. Arılar algoritmasında kullanılan parametreler şunlardır, kâşif arılar (n), kâşif arıların gidip n nokta içinden seçtiği bölge sayısı (m), m bölgedeki en iyi bölge sayısı (e), en iyi e bölgeye gönderilen arı sayısı (nep), bölge boyutu (ngh), kalan ($m - e$) bölgeye gönderilen arı sayısı (nsp)'dir. Öncelikle n adet kâşif arı rasgele araştırma alanına konumlandırılmalıdır. Kâşif arılar konumların uygunluklarını değerlendirir. Uygunluğu en iyi olan arılar elit arı olarak adlandırılır. Elit arıların bölgelerine komşu olan bölgelere arılar gönderilir ve araştırma başlatılır. Daha iyi kaynağı olan bölgeler seçilir ve diğer bölgelere göre daha fazla arı gönderilir. Popülasyonu yenilemek için bölgelerdeki uygunluğu en iyi olan arı seçilir. Kalan arılar rasgele dağılarak yeni kaynaklar aramaktadır. Algoritma istenilen ölçüt oluşana kadar devam eder (Tapkan vd. 2008, Yüce vd. 2013, Zeybek 2014). Arılar algoritmasının sözde kodu şekil 3.14'de gösterilmiştir.

```
Parametreleri ayarla
Popülasyonu başlat, popülasyonun uygunluğunu değerlendir
Do
        Seçkin arıları ve komşu aramasını seçin
        Komşu araması için başka bir bölgeler seçin
        Seçilen bölgeler çevresinde arıları toplayın ve uygunluğu değerlendirin
        Her bölgeden en uygun arının bölgesini seçin
        Kalan arıları rastgele arama yapmak ve uygunluklarını değerlendirmek için
atayın
while durdurma ölçütü sağlanana kadar yap
```

Şekil 3.14 BA sözde kodu

3.4.1.5 Yabani ot algoritması

Yabani ot algoritması, doğadaki yabani otların sürü halinde işgalinden esinlenilerek, Mehrabian ve Lucas tarafından 2006 yılında önerilmiştir (Mehrabian ve Lucas 2006). Yabani otlar rasgele dağılarak arazileri istila ederler ve boş olan alanları kaplarlar. Her yabani ot bu alanda yeni bağımsız yabani otlar üretir. Üretilen bu yeni yabani otlar, üreten yabani otun uygunluğuna bağlı olarak fazla ya da az üretilir. Daha uygun olan otlar daha hızlı ve daha fazla tohum üretir. Yeni yabani otlar rasgele dağılım göstermektedir. Arazideki sınırlı kaynaktan dolayı belirli bir sayıda yabani ot üremektedir. Belirli sayıda yabani ot oluştuktan sonra sadece daha iyi uygunluk gösteren yabani otlar kalır ve yeni yabani otlar üretebilirler. Aralarındaki rekabetçi çekişme, zamanla daha iyi uyum sağlamalarına ve gelişmelerine sebep olur (Okkan vd. 2017). Algoritmalarda en iyi uygunluk değeri olan çözüm diğer çözümlere göre daha iyi olarak nitelendirilir. Bu sebeple uygunluk değeri daha kötü olan çözümlerin gelişmesine olanak sağlanmamaktadır ancak bu düşünce algoritmaların olasılık karakteristiğine terstir. Çünkü bazı durumlarda uygunluğu daha düşük olan çözümler daha fazla bilgi verebilmektedir. Yabani ot optimizasyonu algoritmasında tüm uygunluğa sahip otlar kullanıldığından bu sorunu aşmaktadır.

Algoritma için öncelikle rasgele popülasyon oluşturulmalıdır. Oluşturulan popülasyonda her yabani ot uygunluk değerine göre yeni yabani otlar üretecektir. Denklem 3.18'de oluşturulacak yeni yabani otun denklemidir.

$$s = [s_{min} + (s_{max} - s_{min}) * (\frac{u_i - u_{worst}}{u_{best} - u_{worst}})] \tag{3.15}$$

s_{max} oluşturulacak en fazla yeni yabani ot sayısı, s_{min} oluşturulacak en az yeni yabani ot sayısı, u_{worst} en kötü uygunluk değeri, u_{best} en iyi uygunluk değeri, u_i ise i. bireyin uygunluk değerini ifade etmektedir. Oluşturulan yeni yabani otlar ebeveynlerinin yakınında olmaktadır. Denklem 3.19 ile ebeveynlerinin yakınlarına yerleştirilirler.

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{iter_{max}{}^n} * \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{3.16}$$

$iter_{max}$ en fazla tekrar sayısını, $iter$ tekrar sayısını, $\sigma_{initial}$ başlangıç uzaklık değerini, σ_{final} bitiş uzaklık değerini, n doğrusal olmayan katsayıyı belirtmektedir. Yeni yabani otların ebeveynlerine olan uzaklığı iterasyon ile değişmektedir. İterasyon arttıkça ebeveynlerine olan uzaklık azalmaktadır. En yüksek popülasyon değerine ulaşılınca eliminasyon başlamaktadır. Uygunluk değeri yüksek olan yabani otlar kalırken, uygunluk değeri daha düşük olan yabani otlar popülasyondan ayrılacaktır (Yılmaz 2014). Yabani ot optimizasyonu algoritması sözde kodu şekil 3.15'de verilmiştir.

```
Parametreleri ayarla (iterasyon, popülasyon, başlangıçtaki yabani ot sayısı), rastgele
popülasyon oluştur
for it=1'den iterasyon bitene kadar
  Kolonideki maksimum ve minimum uygunluğu hesapla
  Standart sapmayı iterasyona göre hesapla
  for i=1'den popülasyondaki her birey için
    Uygunluğuna göre yabani ot sayısını hesapla
    Ebeveyn yabani ot etrafında normal dağılım ile arama alanı üzerinde oluşturulan
    tohumları rasgele dağıtın.
    Oluşturulan yabani otları çözüm kümesine ekleyin
    if(Popülasyon en yüksek popülasyon boyutundan büyükse)
      Popülasyonunu uygunluklarına göre azalan şekilde sıralayın
      Popülasyon sayısı en yüksek popülasyon sayısına eşit olana kadar fazla olan
      yabani otları uygunluk derecesine göre popülasyondan çıkar.
    end
  end
Sonraki iterasyon
end
```

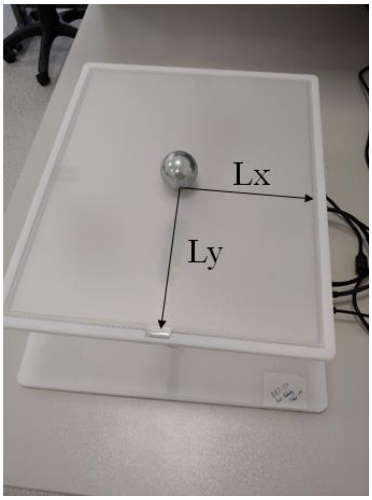
Şekil 3.15 IWO sözde kodu

4. ARAŞTIRMA BULGULARI

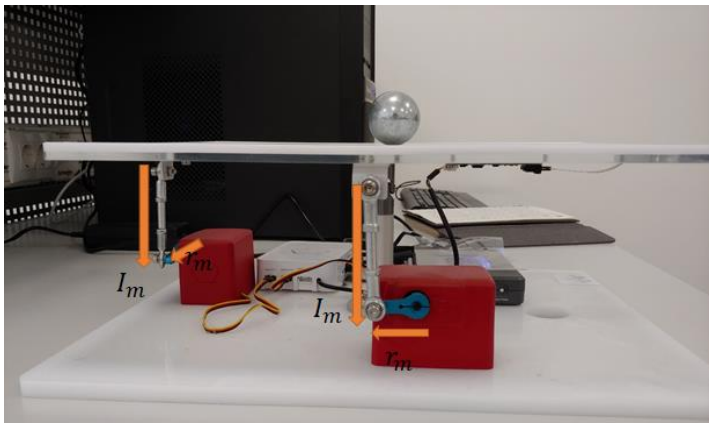
Çalışmada öncelikle Lagrangian metodu ile sistemin modellemesi yapılmıştır. Daha sonra optimizasyon algoritmaları CEC 2017 fonksiyonlarında denenmiştir. 5 farklı optimizasyon algoritması (parçacık sürüsü optimizasyon algoritması, yapay arı kolonisi optimizasyon algoritması, arılar optimizasyon algoritması, ateşböceği optimizasyon algoritması ve yabani ot optimizasyon algoritması) 9 farklı problemi, 10 kere 3 farklı popülasyon ile çözmüştür. Popülasyon problem boyutunu temsil etmektedir. Daha sonra Matlab simulink ortamında kontrolcüler (PID, PIDA, PIDC, 2-Dof-PID, Bulanık-PID) tasarlanmıştır. Tasarlanan kontrolcülerin parametrelerini bulmak için sistemle birleştirilip optimizasyon algoritmalarının hepsi 10 defa sisteme uygulanmıştır ve en iyi sonuçlar not alınmıştır. Bulunan sonuçlar ACROME top dengeleme düzeneğine Labview aracılığı ile uygulanmıştır.

4.1 Sistemin Modellenmesi

Modellemenin amacı sistemin girişiyle çıkışı arasındaki fiziksel bağlantıyı matematiksel denklemler kullanarak göstermektedir. Şekil 4.1'de top dengeleme düzeneğinin üstten görünümü, şekil 4.2'de ise servo motorlar ve servo motorların tablaya uzaklıkları gösterilmiştir.



Şekil 4.1 Top Dengeleme Düzeneği Boyutları



Şekil 4.2 Top Dengeleme Düzeneği

Model oluşturulurken bazı varsayımlar yapılmıştır, bunlar;

- Top ve tabla her zaman birbirlerine değmektedir.
- Top kaymadan tablada ilerlemektedir.
- Tüm sürtünme kuvvetleri ve ortaya çıkan torklar ihmal edilmiştir.
- Step motorda adım atlaması olmamaktadır.

İlk olarak sistemi tanımlayan bir matematiksel model bulunacaktır. Modellemede ilk önce hareket denklemini oluşturulacaktır. Hareket denklemi, topun hareketiyle *(x, y)* tablanın iki açısı (*α, β*) arasındaki ilişkiyi vermektedir. Denklemin girişleri top dengeleme tablasının motorlarının *x* ve *y* açıları olurken çıkışlarıysa *x* ve *y* kartezyen koordinatları olacaktır (Acrome 2014, Zeeshan vd. 2012, Mujadin ve Pratama 2017).

4.1.1 Lagrangian metodu

Lagrangian metodu, genellikle birden daha fazla serbestlik derecesine sahip olan göreceli olarak karmaşık sistemlerin matematiksel modelinin elde edilmesi için kullanılan bir yöntemdir. Lagrangian metodunun denklemi, denklem 1.1 ve 1.2'de gösterilmiştir. Toplam kinetik enerji ise denklem 4.1 ile hesaplanabilir.

$$E_{kin} = E_{kin,T} + E_{kin,R} = \frac{1}{2} m_b {v_b}^2 + \frac{1}{2} j_b {w_b}^2 \qquad (4.1)$$

$E_{kin,T}$ çeviri kinetik enerjisiyken, $E_{kin,R}$ dönme kinetik enerjisidir. Çeviri kinetik enerjinin denklemi 4.2'de verilirken, dönme kinetik enerji denklemi 4.3'de verilmiştir. Bu kinetik enerjilerin toplamı toplam kinetik enerjiyi denklem 4.4'de gösterildiği şekilde verir.

$$E_{kin,T} = \frac{1}{2} m_b {v_b}^2 = \frac{1}{2} m_b ({x_b}^2 + {y_b}^2) \tag{4.2}$$

$$E_{kin,R} = \frac{1}{2} j_b {w_b}^2 = \frac{1}{2} j_b \frac{{v_b}^2}{{r_b}^2} = \frac{1}{2} j_b \frac{({\dot{x}_b}^2 + {\dot{y}_b}^2)}{{r_b}^2} \tag{4.3}$$

$$E_{kin} = \frac{1}{2} m_b ({\dot{x}_b}^2 + {\dot{y}_b}^2) + \frac{1}{2} j_b \frac{({\dot{x}_b}^2 + {\dot{y}_b}^2)}{{r_b}^2} = \frac{1}{2} (m_b + \frac{j_b}{{r_b}^2})({\dot{x}_b}^2 + {\dot{y}_b}^2) \tag{4.4}$$

Potansiyel enerji denklemi ise 4.5'de verilmiştir.

$$E_{pot} = m_b g h_b = -m_b g x_b \sin(\alpha) - m_b g y_b \sin(\beta) \tag{4.5}$$

Denklem 1.2'yi bu denklemlerden bulabiliriz.

$$L_{kin} = \frac{1}{2} (m_b + \frac{j_b}{{r_b}^2})({\dot{x}_b}^2 + {\dot{y}_b}^2) + m_b g x_b \sin(\alpha) + m_b g y_b \sin(\beta) \tag{4.6}$$

Bulunan denklemin türevi alındığında istenilen yöne hareketin formülü bulunabilir.

$$\ddot{x_b} = \frac{m_b g {r_b}^2}{m_b {r_b}^2 + j_b} \sin(\alpha) \tag{4.7}$$

$$\ddot{y_b} = \frac{m_b g {r_b}^2}{m_b {r_b}^2 + j_b} \sin(\beta) \tag{4.8}$$

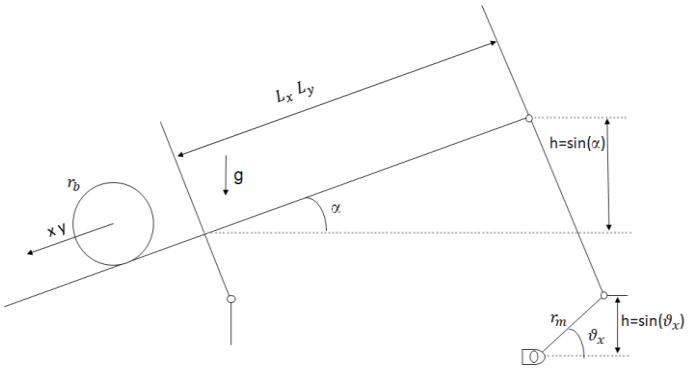
Sistem değerleri çizelge 1'de verilmiştir. Denklem 4.7'de değerler yerine konduğunda denklem 4.9 oluşur.

$$\ddot{x_b} = \frac{5 * g}{7} * \sin(\alpha) \tag{4.9}$$

Denklem 4.8'de değerleri yerine koyduğumuzda ise denklem 4.10 oluşur.

$$\ddot{y_b} = \frac{5 * g}{7} * \sin(\beta) \tag{4.10}$$

Sistemin çıkışları topun *x* ve *y* yönünde koordinatlarıyken (x_b, y_b), sistemin girişleri motor açılarıdır (ϑx, ϑy).



Şekil 4.3 Motor ve Tabla Açıları

Motor ve tabla açıları arasındaki ilişkiyi şekil 4.3'e bakarak yazabiliriz. X yönündeki hareket için bu ilişkiye bakacak olursak,

$$\sin(\vartheta_x)\,\mathrm{r_m} = \sin(\alpha)\,\mathrm{L_x} = h \tag{4.11}$$

$$\sin(\alpha) = \frac{\mathrm{r_m}}{\mathrm{L_x}}\sin(\vartheta_x) \tag{4.12}$$

4.9 denkleminde $\sin(\alpha)$ yerine bulunan sonucu yazarsak denklen 4.14'ü elde ederiz. Aynı şekilde *y* yönü içinde tekrarlayarak denklem 4.16 elde ederiz.

$$\ddot{\mathrm{x}}_\mathrm{b} = \frac{5 * \mathrm{g} * \mathrm{r_m}}{7 * \mathrm{L_x}} * \sin(\vartheta_x) \tag{4.13}$$

$$\ddot{\mathrm{x}}_\mathrm{b} = 1.28 * \sin(\vartheta_x) \tag{4.14}$$

$$\ddot{y}_\mathrm{b} = \frac{5 * \mathrm{g} * \mathrm{r_m}}{7 * \mathrm{L}_y} * \sin(\vartheta_y) \tag{4.15}$$

$$\ddot{\mathrm{y}}_\mathrm{b} = 1.022 * \sin(\vartheta_y) \tag{4.16}$$

Hız ve türevin sıfır olduğu x=0, y=0 noktasında, sistemi doğrusallaştırabilmek için açı değişiminin küçük olduğu kabul edilir. Küçük açılar için $\sin(\vartheta x) \approx \vartheta x$, $\sin(\vartheta y) \approx \vartheta y$ kabul edilir, denklem 4.17 ve 4.18 elde edilir.

$$\ddot{\mathrm{x}}_\mathrm{b} = \frac{\mathrm{m_b} g \mathrm{r_b}^2 \mathrm{r_m}}{(\mathrm{m_b}\mathrm{r_b}^2 + \mathrm{j_b})\mathrm{L_x}}\vartheta_x = K_{BBT,X}\vartheta_x \tag{4.17}$$

$$\ddot{\mathrm{y}}_\mathrm{b} = \frac{\mathrm{m_b} g \mathrm{r_b}^2 \mathrm{r_m}}{(\mathrm{m_b}\mathrm{r_b}^2 + \mathrm{j_b})\mathrm{L_y}}\vartheta_y = K_{BBT,Y}\vartheta_y \tag{4.18}$$

Denklemlerin Laplace dönüşümü alınırsa denklem 4.21 ve 4.22 ortaya çıkar.

$$s^2 \mathrm{X_b}(s) = K_{BBT,X} \vartheta_x(s) \tag{4.19}$$

$$s^2 X_\mathrm{b}(s) = K_{BBT,Y} \vartheta_y(s) \tag{4.20}$$

$$G_{BBT,X}(\mathrm{s}) = \frac{\mathrm{X_b}(s)}{\vartheta_\mathrm{x}(s)} = \frac{K_{BBT,X}}{s^2} \tag{4.21}$$

$$G_{BBT,Y}(\mathrm{s}) = \frac{\mathrm{X_b}(s)}{\vartheta_\mathrm{y}(s)} = \frac{K_{BBT,Y}}{s^2} \tag{4.22}$$

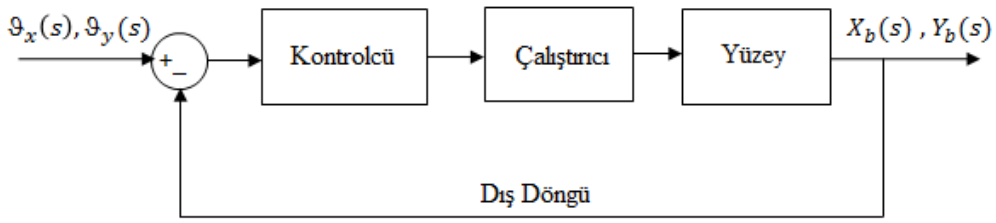
4.1.2 Eyleyicinin modeli

Top dengeleme düzeneğinin eyleyicisi, eyleyicisi servo motorlardır. Servo motorun transfer fonksiyonu birinci dereceden bir fonksiyon olarak düşünülebilir.

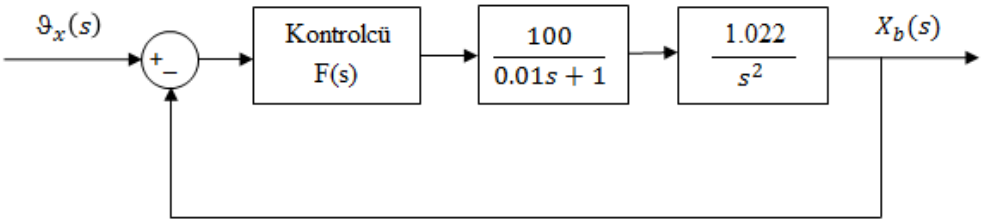
$$G_m(\mathrm{s}) = \frac{\vartheta(s)}{\mathrm{V}_m(s)} = \frac{K_M}{\tau \mathrm{s} + 1} \tag{4.23}$$

K_M kazanç ve τ çalıştırıcı zaman sabitidir. Top dengeleme düzeneği kapalı bir sistem olduğu için teknik belgelerde, K_M 100, τ ise 0.01 olarak verilmiştir (Acrome 2014). Denklem 4.24'de servo motorun birinci dereceden transfer fonksiyonu vardır.

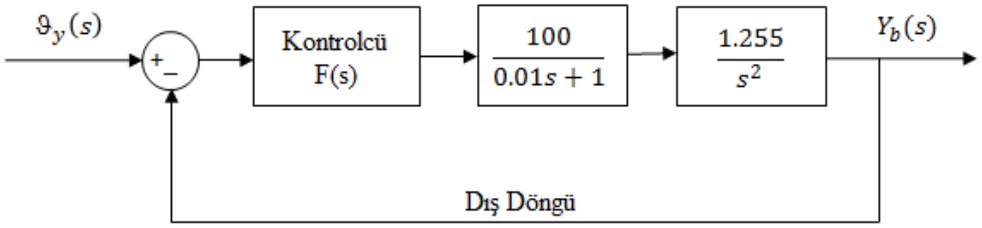
$$G_m(\mathrm{s}) = \frac{100}{0.01s + 1} \tag{4.24}$$



Şekil 4.4 İndirgemeden Sonra Sistemin Blok Diyagramı

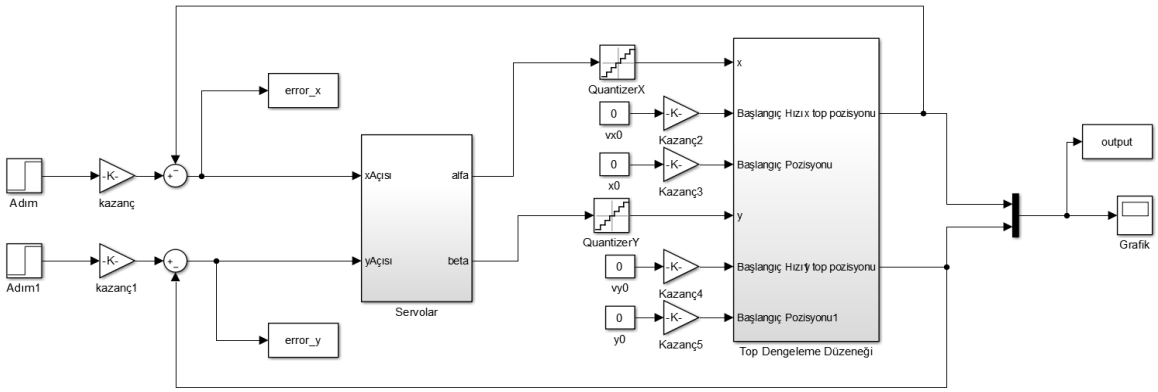


Şekil 4.5 X-Yönündeki Blok Diyagram



Şekil 4.6 Y-Yönündeki Blok Diyagram

Sistemin genel blok diyagramı şekil 4.4'de verilmiştir. İki yön içinde ayrı ayrı olarak şekil 4.5 ve 4.6'da gösterilmiştir. Sistem modellenip transfer fonksiyonu bulunmuştur. Sistemin kontrolcüsüz modellenmiş hali şekil 4.7'deki gibidir.



Şekil 4.7 Sistemin Genel Modellemesi

Sistemde *x* ve *y* yönündeki transfer fonksiyonları birbirine yakın olduğundan transfer fonksiyonu olarak $\frac{1}{s^2}$ kullanılmıştır ve tek yönde denemeler yapılmıştır. Bulunan sonuçlar iki yön içinde uygulanmıştır. Top dengeleme düzeneğinde bulunan dokunmatik ekranın tepkime süresi en düşük dokunmatik ekranın tepkime süresi olarak alınmıştır. En düşük

dokunmatik ekran tepkime süresi 80 milisaniye olarak kabul edilmiştir (Anderson vd. 2011). Sistemin geri beslemesine gecikme olarak 80 milisaniye eklenmiştir.

4.2 Optimizasyon algoritmaları

Sistem modellemesinden sonra kullanılacak kontrolcülerin parametrelerini bulmak için 5 ayrı optimizasyon algoritması kullanılmıştır. Bu optimizasyon algoritmalarının çalışma doğruluğunu ispatlamak ve kendi aralarında kıyaslamak için çizelge 4'de verilen CEC 17'deki 9 fonksiyonun en küçük değerlerinin elde edilmesi için uygulanmış ve karşılaştırılmıştır. Algoritmalar CEC 17'ye göre 3 farklı popülasyonda, iterasyonda ve boyutta denenmiştir. Deneme boyutları 2,10 ve 30'dur. Popülasyon sayısı boyutun 20 katıdır. Sırasıyla 2,10 ve 30 boyutları için popülasyon sayısı 40, 200, 600'dür. İterasyon sayısı CEC 20'ye göre seçilmiştir. Sırasıyla 2, 10 ve 30 için iterasyon sayısı 5000, 10000 ve 25000'dir. Tüm optimizasyon algoritmalarının arama uzayı [-100,100] arasındadır. Optimizasyon algoritmalarının tüm sonuçları aşağıdaki çizelgelerde (4.2-4.16) verilmiştir. Optimizasyon algoritmalarının bu fonksiyonlarda denenmesinin sebebi, farklı popülasyonlarda farklı problemlerde nasıl sonuçlar vereceğini görmek içindir. Kontrolcü parametresi fazla olan uygulamalarda optimizasyon algoritmalarının hepsinin sonucu bulabileceği gösterilmiştir. Aşağıdaki çizelgelerdeki sonuçlara göre optimizasyon algoritmalarının hangisinin kontrolcü parametrelerini bulurken daha doğru sonuca gittiği gözlemlenecektir.

Çizelge 4.1 CEC17 Fonksiyonları

Fonksiyon Numarası	**Fonksiyon**	**Optimum Değer**
1	$f(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	100
2	$f(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$	200
3	$f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	300
4	$f(x) = \sum_{i=1}^{D} (x_i^2 - 10\,cos(2\pi x_i) + 10)$	400
5	$g(x,y) = 0.5 + \frac{(sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$ $f(x) = g(x_1,x_2) + g(x_2,x_3) + \cdots + g(x_{D-1},x_D) + g(x_D,x_1)$	500
6	$f(x) = min\left(\sum_{i=1}^{D}(x_i - \mu_0)^2, dD + s\sum_{i=1}^{D}(x_i - \mu_1)^2\right) + 10(D - \sum_{i=1}^{D} cos(2\pi(x_i - \mu_0)))$ $\mu_0 = 2.5,\ \mu_1 = -\sqrt{\frac{{\mu_0}^2 - d}{s}}$ $s = 1 - \frac{1}{2\sqrt{D+20} - 8.2}, d = 1$	600
7	$f(x) = \sum_{i=1}^{D}\left(z_i^2 - 10\,cos(2\pi z_i) + 10\right)$ $z_i = \begin{cases} x_i & \lvert x_i\rvert < 0.5 \\ \frac{round(2x_i)}{2} & \lvert x_i\rvert > 0.5 \end{cases}$	700
8	$f(x) = sin^2(\pi w_1) + \sum_{i=1}^{D}(w_i - 1)^2\,[1 + 10sin^2(\pi w_i + 1)] + (w_D - 1)^2[1 + sin^2(2\pi w_D)]$ $w_i = 1 + \frac{x_{i-1}}{4}$	800
9	$f(x) = 418.9829 * D - \sum_{i=1}^{D} g(z_i)$ $g(z_i) =$ $\begin{cases} z_i \sin(\lvert z_i\rvert^{0.5}) & \lvert z_i\rvert \le 500 \\ \left(500 - mod(z_i,500)\right)\sin\left(\sqrt{\lvert 500 - mod(z_i,500)\rvert}\right) & \\ \quad - \frac{(z_i - 500)^2}{10000D} & z_i > 500 \\ (mod(\lvert z_i\rvert,500) - 500)\sin\left(\sqrt{\lvert mod(\lvert z_i\rvert,500) - 500\rvert}\right) - & \\ \quad - \frac{(z_i + 500)^2}{10000D} & z_i < -500 \end{cases}$	900

Çizelge 4.2 ABC 2 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	105,7759	3,9882	111,3275	100,4071
2	200,0001	$7{,}1458\text{x}10^{-5}$	200,0003	200,0000
3	300,0000	$5{,}7673\text{x}10^{-7}$	300,0000	300,0000
4	400,0000	$1{,}3966\text{x}10^{-5}$	400,0000	400,0000
5	500,0002	$2{,}0177\text{x}10^{-5}$	500,0006	500,0000
6	600,0023	$1{,}1611\text{x}10^{-5}$	600,0033	600,0003
7	700,6335	0,44537	701,3772	700,0309
8	800,0008	$8{,}0282\text{x}10^{-4}$	800,0021	800,0001
9	900,0000	$3{,}8449\text{x}10^{-8}$	900,0000	900,0000

Çizelge 4.2'nin sonucuna göre ABC algoritması 2 boyutta tüm problemler için sonuçları bulmuştur. İlk fonksiyon hariç tüm fonksiyonlarda standart sapma düşük çıkmıştır. İlk fonksiyonda ise yüksek olmasa da diğer fonksiyonlara göre yüksek çıkmıştır.

Çizelge 4.3 BA 2 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	124,5247	26,1554	180,0705	100,4342
2	200,0006	$8{,}3521\text{x}10^{-4}$	200,0026	200,0000
3	300,0000	$1{,}1\text{x}10^{-13}$	300,0000	300,0000
4	400,0001	$1{,}4755\text{x}10^{-4}$	400,0004	400,0000
5	500,0995	0,31463	500,995	500,0000
6	600,0001	$9{,}446\text{x}10^{-5}$	600,0002	600,0000
7	700,5587	0,68174	701,9899	700,0000
8	800,0223	0,059294	800,1882	800,0000
9	900,0000	$5{,}3592\text{x}10^{-14}$	900,0000	900,0000

Çizelge 4.3'ün sonucuna göre BA algoritması 2 boyutta tüm problemler için sonuçları

bulmuştur. İlk fonksiyon hariç tüm fonksiyonlarda standart sapma düşük çıkmıştır.

Çizelge 4.4 FA 2 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	100,0000	$3{,}4088 \times 10^{-9}$	100,0000	100,0000
2	200,0000	$1{,}9417 \times 10^{-9}$	200,0000	200,0000
3	300,0000	$3{,}2132 \times 10^{-9}$	300,0000	300,0000
4	400,0000	$2{,}6377 \times 10^{-9}$	400,0000	400,0000
5	500,0000	$2{,}8889 \times 10^{-9}$	500,0000	500,0000
6	600,0000	$2{,}4217 \times 10^{-9}$	600,0000	600,0000
7	701,2457	0,84555	702,2238	700,0000
8	800,0000	$2{,}7036 \times 10^{-9}$	800,0000	800,0000
9	900,0000	$1{,}5075 \times 10^{-9}$	900,0000	900,0000

Çizelge 4.4'ün sonucuna göre FA algoritması 2 boyutta tüm problemler için sonuçları bulmuştur. Diğer optimizasyon algoritmalarının aksine tüm problemleri çözmüştür ve standart sapma hepsinde düşüktür.

Çizelge 4.5 IWO 2 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	281,8957	251,3523	836,4378	100,0000
2	200,0000	$1{,}2985 \times 10^{-6}$	200,0000	200,0000
3	300,0000	$6{,}8696 \times 10^{-10}$	300,0000	300,0000
4	400,0000	$2{,}6663 \times 10^{-12}$	400,0000	400,0000
5	502,4874	2,2124	504,9748	500,0000
6	613,0255	12,9915	638,3127	600,0000
7	703,2786	0,14808	704,9748	700,0000
8	803,4824	3,8176	812,9344	800,0000
9	912,9473	19,6599	963,2338	900,0000

Çizelge 4.5'in sonucuna göre IWO algoritması 2 boyutta tüm problemler için sonuçları bulmuştur. Diğer optimizasyon algoritmalarının aksine standart sapmalar yüksek çıkmıştır. Özellikle 1. Problem için standart sapma yüksek çıkmıştır.

Çizelge 4.6 PSO 2 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	136,2379	75,2528	333,2305	100,0185
2	200,0000	$2{,}4268 \times 10^{-6}$	200,0000	200,0000
3	300,0000	$2{,}7161 \times 10^{-6}$	300,0000	300,0000
4	400,0000	$2{,}3457 \times 10^{-6}$	400,0000	400,0000
5	500,3980	0,51379	500,9950	500,0000
6	600,0000	$2{,}5887 \times 10^{-6}$	600,0000	600,0000
7	701,8200	0,6397	702,0694	700,0000
8	800,5970	0,51379	800,9950	800,0000
9	900,0000	$2{,}3815 \times 10^{-6}$	900,0000	900,0000

Çizelge 4.6'nın sonucuna göre PSO algoritması 2 boyutta tüm problemler için sonuçları

bulmuştur. PSO algoritmasında ilk problemde sorun yaşasa da diğer fonksiyonları rahat çözmüştür.

2 boyutta optimizasyon algoritmaları genel olarak 1. problemde standart sapmayı yüksek bulmuştur. Ancak FA algoritması 1. problem dâhil tüm problemleri tüm denemelerde çözmüştür. Tüm optimizasyon algoritmalarında gözüktüğü gibi 10 iterasyonda tüm fonksiyonlar için tüm algoritmalar doğru çözümü bulmuştur, yakınsamıştır.

Çizelge 4.7 ABC 10 Boyut için Sonuçlar

Fonksiyon Numarası	**Ortalama Değer**	**Standart Sapma**	**En Yüksek Değer**	**En Düşük Değer**
1	248,4965	150,4439	563,2609	100,1156
2	200,0006	$1{,}9814 \times 10^{-4}$	200,0009	200,0000
3	541,4888	96,8303	747,8356	402,366
4	400,0645	$8{,}9038 \times 10^{-4}$	400,0660	400,063
5	514,3738	2,1603	518,8469	511,3351
6	600,0091	$5{,}5408 \times 10^{-4}$	600,0099	600,0082
7	725,4075	1,4778	728,1394	723,1447
8	815,2523	1,8125	817,6595	811,4679
9	900,0080	0,0017407	900,0096	900,0045

Çizelge 4.7'nin sonucuna göre ABC algoritması 10 boyutta 3. fonksiyonun sonucuna tam yakınsayamamıştır. 1. fonksiyonun ve 3. fonksiyonun ortalama ve standart sapma değerleri yükselmiştir.

Çizelge 4.8 BA 10 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	105,7469	9,7278	130,1763	100,0093
2	200,0079	0,0018155	200,0097	200,0037
3	585,7256	441,1695	1767,8935	300,2603
4	400,6036	0,64381	401,8504	400,009
5	520,2971	3,9017	525,8689	515,9193
6	600,0092	$8{,}3711\text{x}10^{-5}$	600,0099	600,0073
7	739,3071	10,1803	762,7000	728,7335
8	823,1825	6,0802	835,8184	813,9294
9	1118,5806	100,2035	1239,5924	956,0388

Çizelge 4.8'in sonucuna göre BA algoritması 10 boyutta tüm sonuçlara yakınsamıştır. Ancak 3. fonksiyonda standart sapma ve ortalama değeri yükselmiştir. Bunun anlamı algoritma çoğu denemede 3. fonksiyonu çözememiştir. Ancak en düşük değerden anlaşılacağı gibi en az bir kere sonuca ulaşmıştır.

Çizelge 4.9 FA 10 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	100,0000	$5{,}8926\text{x}10^{-5}$	100,0001	100,0000
2	200,0000	$7{,}9374\text{x}10^{-7}$	200,0000	200,0000
3	300,0000	$8{,}394\text{x}10^{-5}$	300,0000	300,0000
4	400,0000	$3{,}1043\text{x}10^{-10}$	400,0000	400,0000
5	505,9430	3,1042	509,8156	501,9899
6	600,0000	$1{,}5671\text{x}10^{-6}$	600,0000	600,0000
7	714,3468	1,918	716,6446	711,7937
8	804,5768	1,8074	806,9647	802,9849
9	900,0000	$1{,}8176\text{x}10^{-6}$	900,0000	900,0000

Çizelge 4.9'un sonucuna göre FA algoritması 10 boyutta tüm sonuçları çözmüştür, yakınsamıştır. Tüm standart sapma ve ortalama değerleri düşük çıkmıştır. FA algoritması tüm denemelerde sonuca ulaşmıştır.

Çizelge 4.10 IWO 10 Boyut için Sonuçlar

Fonksiyon Numarası	**Ortalama Değer**	**Standart Sapma**	**En Yüksek Değer**	**En Düşük Değer**
1	269,7529	349,384647	1245,514	102,472
2	200,0060	0,0015123	200,0088	200,004
3	300,0079	0,001711	300,0100	300,0053
4	400,0093	$5{,}523x10^{-5}$	400,0100	400,0081
5	550,2452	25,4971	601,4848	519,8991
6	640,4268	8,4475	653,3801	628,331
7	937,3854	68,4351	1025,9789	811,5169
8	847,5588	15,9730	865,6669	817,9092
9	1008,4856	242,5609	1442,3917	900,0085

Çizelge 4.10'un sonucuna göre IWO algoritması 10 boyutta tüm sonuçlara yakınsamıştır. Ancak IWO algoritması 1. fonksiyon, 7. fonksiyon ve 9. fonksiyonda standart sapma ve ortalama değerleri yüksek çıkmıştır. Bu fonksiyonlar için algoritma 10 denemeden bazılarında sonuca yakınsamıştır. Bu denemelerin, 1. fonksiyona için %40'ında, 7. fonksiyona %20'sinde ve 9. fonksiyona %30'unda sonuca yakınsamıştır.

Çizelge 4.11 PSO 10 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	237,2283	154,4964	572,9833	106,7718
2	200,0008	$1{,}4603\text{x}10^{-4}$	200,0009	200,0005
3	300,0008	$1{,}378\text{x}10^{-5}$	300,0010	300,0005
4	401,0220	0,61848	402,0738	400,3614
5	518,2077	7,863	533,8285	507,9597
6	600,1098	0,34462	601,0906	600,0004
7	715,0453	5,2603	719,8333	701,9899
8	812,3375	5,0993	817,9092	803,9798
9	900,0097	0,028066	900,0895	900,0006

Çizelge 4.11'in sonucuna göre PSO algoritması 10 boyutta tüm sonuçlara yakınsamıştır. Standart sapma ve ortalama değerleri 1.fonksiyoda yüksek çıkmıştır. 1.fonksiyonda denemelerin %30'unda sonuca yakınsamıştır.

Boyut arttıkça FA algoritması hariç diğer algoritmalarda bazı fonksiyonları çözememe durumu başlamıştır. 2 boyutta genel olarak sorun yaratan fonksiyon olan 1. fonksiyonla beraber 10 boyutta 3. fonksiyonda sorun yaratmaya başlamıştır. Optimizasyon algoritmaları 3. fonksiyona yakınsamamaya başlamışlardır.

Çizelge 4.12 ABC 30 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	171,6019	107,792198	445,5040	100,4720
2	216,6641	17,5772	236,9481	205,9072
3	490,0419	117,423044	721,9730	390,1120
4	416,7101	0,069997	416,8243	416,5940
5	664,1389	4,863	671,9765	657,7656
6	600,0000	$8{,}2368 \times 10^{-7}$	600,0000	600,0000
7	889,4849	6,296	902,8778	882,4411
8	962,1523	10,0733	971,953	939,7822
9	900,0000	$6{,}4271 \times 10^{-7}$	900,0000	900,0000

Çizelge 4.12'nin sonucuna göre ABC algoritması 30 boyutta 3. fonksiyonun, 5. fonksiyonun, 7. fonksiyonun ve 8. fonksiyonun optimum sonuçlarına ulaşılamamıştır. Ortalama değerleri yüksek çıkmıştır. 5. fonksiyonda, 7.fonksiyonda ve 8. fonksiyonda standart sapma değerleri düşük çıkmıştır. Bunun anlamı, sonuç genel olarak bu aralığa yakınsamaktadır. 1. fonksiyonu çözse de standart sapma ve ortalama değer yüksek çıkmıştır.

Çizelge 4.13 BA 30 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	100,1623	0,14225	100,2853	100,0065
2	220,9957	28,4658476	292,0932	200,0008
3	416,8941	35,0174517	488,7410	361,7778
4	594,3108	20,7618	612,9949	561,2323
5	693,7388	3,9443	698,6018	688,0460
6	625,5516	2,5277	627,9578	621,3548
7	872,7901	77,7658899	1034,9870	787,2140
8	981,6784	16,5235	1002,9706	965,1623
9	1283,9736	136,4441	1547,1247	1097,1240

Çizelge 4.13'ün sonucuna göre BA algoritması 3. fonksiyon, 4. fonksiyon, 5. fonksiyon, 7. fonksiyon, 8. fonksiyon ve 9. fonksiyonu çözememiştir. Çözemediği 4. fonksiyon, 5. fonksiyon, 8. fonksiyon ve 9. fonksiyonda standart sapma değerleri de küçük çıkmıştır. Bunun anlamı, sonuç genel olarak bu aralığa yakınsamaktadır.

Çizelge 4.14 FA 30 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	100,0000	$1{,}9435\text{x}10^{-6}$	100,0000	100,0000
2	200,0000	$1{,}4894\text{x}10^{-6}$	200,0000	200,0000
3	300,0000	$3{,}4527\text{x}10^{-8}$	300,0000	300,0000
4	446,8492	26,1895	458,5616	400,0000
5	536,7325	11,5159	552,1427	524,499
6	600,0000	$5{,}6843\text{x}10^{-14}$	600,0000	600,0000
7	753,0860	5,8961	761,5032	747,7323
8	844,8243	11,1829	862,9683	834,8235
9	900,0000	$1{,}2711\text{x}10^{-13}$	900,0000	900,0000

Çizelge 4.14'ün sonucuna göre FA algoritması tüm fonksiyonları istenilen şekilde çözmüştür, yakınsamıştır. Standart sapma ve ortalama değerleri diğer algoritmalara göre küçüktür. 5. fonksiyon, 7. fonksiyon ve 8. fonksiyonda tam olarak sonucu bulamasa da yakınsamıştır.

Çizelge 4.15 IWO 30 Boyut için Sonuçlar

Fonksiyon Numarası	**Ortalama Değer**	**Standart Sapma**	**En Yüksek Değer**	**En Düşük Değer**
1	163,2845	37,787699	211,9140	117,5140
2	200,0067	0,0025081	200,0096	200,0010
3	300,0093	0,0010151	300,0100	300,0067
4	432,5434	27,1635	469,7068	400,0099
5	819,5268	29,5782	861,1627	785,5509
6	659,7069	6,5618	670,0702	647,0612
7	1014,0453	104,464251	1109,1280	829,9510
8	1070,9915	44,6692	1136,2921	995,0108
9	1544,0838	91,53055	1617,7640	1412,9830

Çizelge 4.15'in sonucuna göre IWO algoritması 5. fonksiyon, 7. fonksiyon, 8. fonksiyon ve 9. fonksiyonu çözememiştir. Çözemediği 5. fonksiyon ve 8. fonksiyonda standart sapma değerleri de küçük çıkmıştır. Bunun anlamı, sonuç genel olarak bu aralığa yakınsamaktadır.

Çizelge 4.16 PSO 30 Boyut için Sonuçlar

Fonksiyon Numarası	Ortalama Değer	Standart Sapma	En Yüksek Değer	En Düşük Değer
1	144,9231	40,04544	208,9150	104,0012
2	205,8316	18,4379	258,3066	200,0008
3	301,7620	2,2216	306,2134	300,2220
4	452,4185	34,4562	496,7418	403,9905
5	611,1365	20,4230	655,2128	580,5915
6	601,8533	3,2505	610,2368	600,0010
7	801,6939	24,9237	851,0467	776,8940
8	887,9541	17,0547	923,3742	871,6369
9	1498,5402	804,4944	3091,1522	900,9086

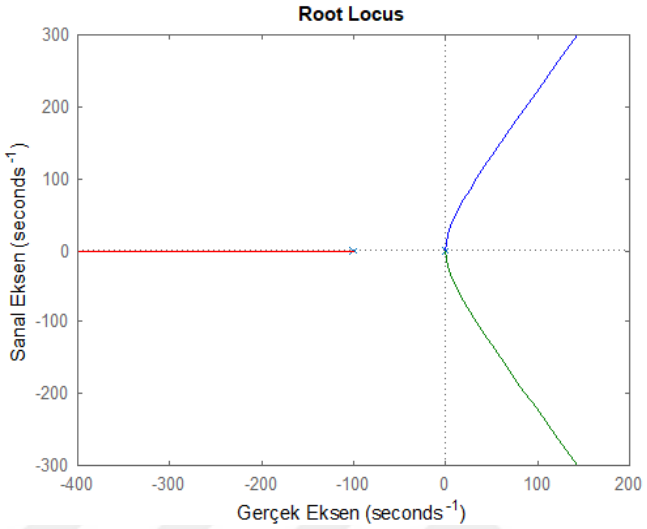
Çizelge 4.16'nın sonucuna göre PSO algoritması 5. fonksiyon, 7. fonksiyon ve 8. fonksiyonu çözememiştir. Çözemediği fonksiyonlarda aslında diğer algoritmalar kadar sonuca uzak değildir. Çözdüğü 9. fonksiyonda ise standart sapma ve ortalama çok yüksektir. 9. fonksiyon için 10 denemenin 5'inde sonuca yakınsamıştır.

Alınan sonuçlara göre optimizasyon algoritmaları 2 boyutta genel olarak sonuçlara gitmektedir. 10 boyutta optimum değerlere genel olarak gidilse de 3. fonksiyonu ve 1. fonksiyonun optimum değerine bazı algoritmalar gidememiştir. Boyutu 30'a çıkardığımızda ise çözülemeyen fonksiyonlar artmaktadır. Ancak 2, 10 ve 30 boyutta FA algoritması tüm fonksiyonları çözmüştür. FA algoritması en kısa iterasyonda yakınsamıştır ve tüm denemelerde sonuca ulaşmıştır. Yapılan denemeler sonucunda en iyi algoritmanın FA algoritması olduğu sonucuna varılmıştır.

4.3 Benzetim Çalışmaları

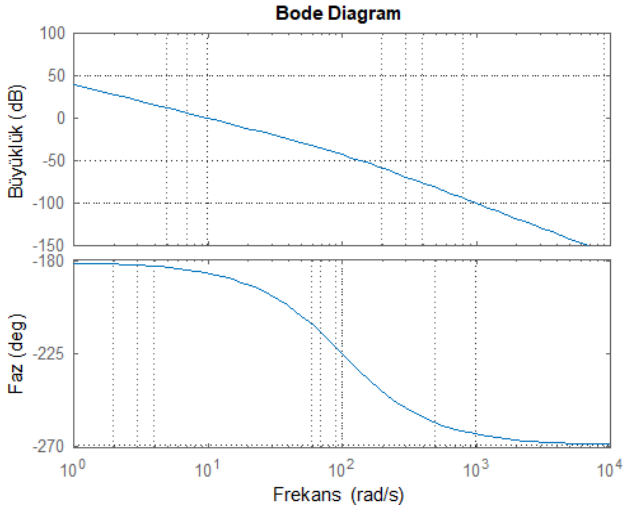
Kararsız sistemleri kararlı hale getirebilmek, kontrol edebilmek için kontrolcü algoritmaları kullanılır. Sistemin kararlı olup olmaması Root Locus yöntemi ile bulunabilir. Root Locus eğrisi çizdirildiğinde kutuplar Reel eksenin negatif bölgesindeyse

sistem kararlı, pozitif taraftaysa sistem kararsızdır. Top dengeleme düzeneğinin Root Locus eğrisi şekil 4.8'de gösterilmiştir.



Şekil 4.8 Root Locus Eğrisi

Sistemde 3 tane kutup vardır. Kutuplardan biri gerçek kutup, diğer ikisi ise kompleks eşlenik kutuptur. Kompleks eşlenik kutuplara herhangi bir K değeri verdiğimizde pozitif olmaktadırlar bu sebeple kapalı döngülü sistem kararlı değildir.



Şekil 4.9 Bode Diyagramı

Sistemin kararlı olup olmamasına Bode diyagramından da bakılabilir. Bode diyagramında frekans yanıtı, açı ve büyüklük olarak grafiklerde gösterilir. Biriminin dB olmasının nedeni daha geniş aralıklarda inceleyebilmektir. Diyagramda faz açısının -180 derece olduğu noktada büyüklüğe bakılır, eğer büyüklük pozitifse sistem kararsız, negatifse sistem kararlıdır. Şekil 4.9'da verilen Bode diyagramında faz açısının -180 derece olduğu noktada büyüklük pozitif olduğundan sistem kararsızdır.

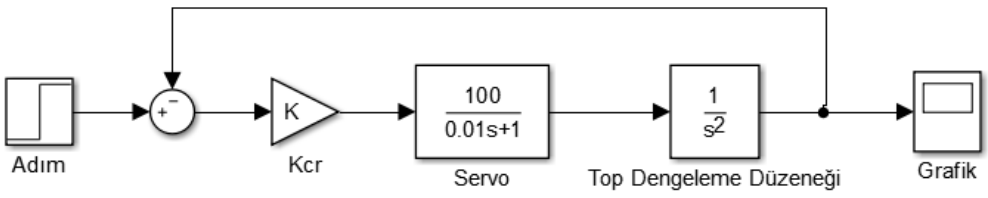
Sistemi kontrol edebilmek için kontrolcü eklenmelidir. Sistemin kontrolü için 5 farklı kontrolcü kullanılmıştır. Seçilen kontrolcüler PID tabanlı kontrolcülerdir. PID kontrolcüsü en çok kullanılan kontrolcü olmasına rağmen, üzerinde çalışılan probleme göre istenilen kısa oturma süresi, düşük kalıcı durum hatası, aşım yüzdesi olmaması gibi durumlarda yetersiz kalabilmektedir. PID kontrolcüsünün yeterli olmadığı zamanlarda, yüzde aşımının olduğu yerlerde PIDA kontrolcüsü kullanılabilir. PIDA kontrolcüsünün avantajı yüzde aşımını önlemektir. Sisteme ivme eklendiğinden yapılacak olan hareketi daha az salınım yapmaktadır. Dezavantajı olarak PID kontrolcüsüne fazladan bir parametre eklenmesi gösterilebilir. PID kontrolcüsünün yetersiz kaldığı yüzde aşımı ya da kalıcı durum hatası PID kontrolcüsüne seri filtre eklenerek azaltılabilir. Eklenecek filtrede kutup ve sıfır olacağından sistemin parametre sayısı artmaktadır bu yüzden bu bir dezavantajdır. Ancak sisteme eklenen filtreler kalıcı durum hatasını ve yüzde aşımını önlemektedir. PIDA ve PIDC kontrolcüleri yüzde aşımını önlemektedir. PIDA kontrolcüsünün parametre sayısı PIDC kontrolcüsünden daha azdır. Bu sebeple PIDA kontrolcüsü seçilebilir. Ancak sistemde kalıcı durum hatası oluşabilirse PIDC kontrolcüsü kullanılıp hem kalıcı durum hatası hem de yüzde aşımı önlenebilir. 2-Dof-PID sistemde oluşan ani değişimleri hızlıca elimine etmek için kullanılır. 2-Dof-PID kontrolcüsü referans ve çıkış sinyali girdi olarak verildiği için referanstaki ani değişiklikler tolere edilebilir. 2-Dof-PID kontrolcüsünde PID kontrolcüsüne ek olarak 2 tane parametre vardır, bu parametreler dezavantaj olarak gösterilebilir. Bu parametreler kazanç ve türev terimleri üzerinde etkilidir. Kazançlar ve çıkış sinyalini de giriş olarak almasından dolayı 2-Dof-PID kontrolcüsü PIDA ve PIDC kontrolcüsünden daha hızlı yüzde aşımını elimine etmektedir. Bulanık-PID kontrolcüsü matematiksel olarak modelleme istemediğinden diğer kontrolcülerden ayrılmaktadır. 4 parametresi olduğundan ve bulanık mantık tablosu oluşturulması gerektiğinden biraz uğraştırıcı

olabilmektedir. Bulanık-PID kontrolcüsü yüzde aşımını ve kalıcı durum hatasını azaltır. Bu sebeple diğer kontrolcülerden daha az parametresi olması ve matematiksel modelleme gerektirmemesi avantajlarındandır.

Kontrolcü parametrelerini bulmak için optimizasyon algoritmaları uygulanmıştır. Kontrolcü parametrelerini bulurken, optimizasyon algoritmalarının performansını ölçmek için kullanılan amaç fonksiyonu ISE olarak seçilmiştir. Tüm optimizasyon algoritmaları tüm kontrolcüler ile 10'ar defa birbirlerinden bağımsız bir şekilde çalıştırılmıştır ve en iyi sonuçlar çizelgede verilmiştir. Alınan sonuçları kıyaslamak için kontrolcünün yükselme süresi, kalıcı durum hatası, yüzde aşımı ve oturma süresi çizelgede gösterilmiştir. Optimizasyon algoritmalarıyla kontrolcü parametresi bulunurken, boyut kontrolcü parametresi olarak girilmiştir. İterasyon sayısı 200 olarak seçilmiştir. Arama uzayı ise [10^{-3} ,5] seçilmiştir. Arama uzayı sisteme Ziegler-Nichols yöntemi uygulandıktan sonra bulunan sonuçların değiştirilmesiyle seçilmiştir.

4.3.1 PID kontrolcü

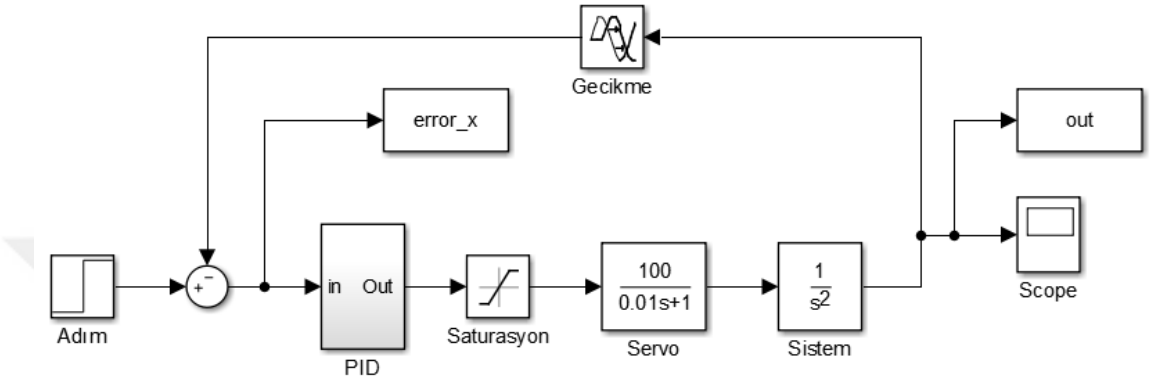
PID kontrolcüsü endüstride en çok kullanılan ve bilinen kontrolcü olduğu için parametrelerini bulmak için literatürde birçok metot önerilmiştir. Bunlardan en bilineni olan Ziegler-Nichols yöntemi ile kontrolcü parametreleri bulunmuştur. Sisteme Ziegler-Nichols'ün ikinci metodu uygulanmıştır. Şekil 4.10'da Ziegler-Nichols metodu uygulanması için oluşturulan kapalı döngü gösterilmektedir.



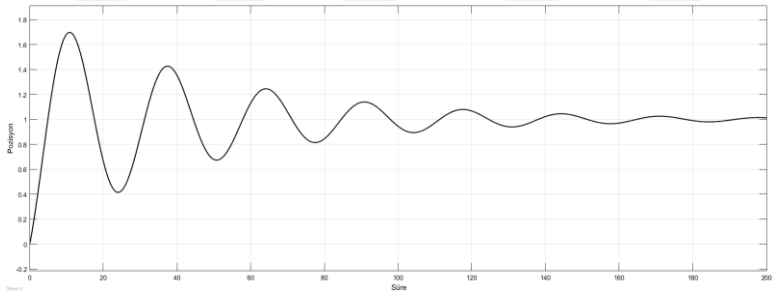
Şekil 4.10 Ziegler-Nichols 2. Metodu için Sistem

Sistemin denenmesi sonucunda K_{cr} değeri 0.001 olarak bulunurken P_{cr} değeri 20 saniye olarak bulunmuştur. Bu durumda PID katsayıları K_c=6x10^{-4}, T_i=10 ve T_d =2.5 şeklinde

bulunmuştur. Ziegler-Nichols yöntemiyle bulunan sonuçlar PID katsayıları olarak sisteme uygulandığında sistem istenilen cevabı vermemiştir, sistemin istenilen konuma gidebilmesi için 150 saniyeden fazla bir süre geçmesi gerekmektedir. Sistem çok yavaş çalışmaktadır.



Şekil 4.11 Sistemin Tek Eksende Genel Modellemesi



Şekil 4.12 Ziegler Nichols Çıktısı

Ziegler-Nichols yöntemiyle istenilen sistem kontrolü sağlanamadığı için optimizasyon algoritmaları denenmiştir. Optimizasyon algoritmalarının hepsi PID kontrolcüsünde 10'ar kere denenmiştir. Değişken sayısı az olduğundan sisteme uygulanan optimizasyon algoritmalarının tüm denemelerinde sonuçlar bulunmuştur. Bulunan sonuçlardan hatası en düşük olanlar seçilmiş ve çizelge 4.17'de verilmiştir. Sistemin genel modellemesi Şekil 4.11'de verilmiştir. Diğer kontrolcüler sisteme uygulanırken sadece kontrolcü değiştirilmiş geri kalan kısımlar aynı kalmıştır.

Topun konumunu dokunmatik ekran sayesinde okurken, ölçüm gürültüsü olabilir. Doğrusal olmayan etkiler ya da yüksek frekanslı sesler ölçümü etkileyebilir. Bu sebeple gerçek konum ile ölçülen konum arasında farklar olabilir. Bu nedenle sistemin cevabını yavaşlatmak için sistemin türev terimine dâhili bir filtre eklenmiştir. Acrome bu filtreyi deneme yanılma yoluyla 30 olarak bulmuştur (Acrome 2014). Bu tez kapsamında da 30 kullanılmıştır.

Çizelge 5 Optimizasyon algoritmalarının bulduğu PID kontrolcüsü katsayıları

Optimizasyon Algoritması	**Kc**	**Ti**	**Td**
ABC	0,05376	3,3726	1,136
BA	0,057505	3,3202	1,0834
FA	0,054758	3,3459	1,1259
IWO	0,054723	3,35	1,126
PSO	0,054734	3,3459	1,1263

Çizelge 4.17'de görüldüğü gibi algoritma sonuçları tüm optimizasyon algoritmalarında benzer çıkmıştır. Parametre sayısı az olduğu için tüm algoritmalar sonuca hızlı bir şekilde yakınsamıştır. Algoritma sonuçlarını karşılaştırmak için tüm optimizasyon algoritmalarının bulduğu PID kontrolcü parametreleri sisteme uygulanmıştır. Sonuçlar sisteme uygulanıp grafik olarak çıktı alındığında yanıtlar birbirine çok yakın olduğundan, sonuçlar çok belirgin olmamaktadır. Bu sebeple sisteme uygulanan parametrelerin sonuçları kıyaslanması için çizelge 4.18'de verilmiştir.

Çizelge 6 PID kontrolcü parametrelerinin sistem cevabı sonuçları

Optimizasyon Algoritması	Yükselme Süresi	Oturma Süresi	Yüzde Aşımı	Kalıcı Durum Hatası
ABC	0,37	5,9371	4,4454	0,0025
BA	0,3650	5,8268	4,5239	0,0023
FA	0,3710	5,9464	4,4998	0,0024
IWO	0,3708	5,9437	4,4920	0,0024
PSO	0,3710	5,9476	4,5	0,0024

Sisteme uygulanan parametreler benzer olduğundan çıktılarda benzer olmuştur. Sistemin verdiği yükselme süresi, oturma süresi, yüzde aşımı tepkisi ve kalıcı durum hatası değerleri tüm optimizasyon algoritmalarında benzer çıkmıştır. PID kontrolcüsü parametrelerini bulurken en iyi sonucu veren (en düşük hata çıktısı) FA algoritması, PID kontrolcüsü parametreleri uygulandığında kıyaslanan alanların hiç birinde en iyi sonucu vermemiştir. Yükselme süresi, oturma süresi ve kalıcı durum hatası en iyi olan optimizasyon algoritması BA algoritmasıdır. Yüzde aşımı en küçük olan optimizasyon algoritması ise ABC algoritmasıdır. Simülasyon sonuçlarına bakarak PID katsayıları sisteme uygulandığında en iyi algoritma BA algoritması olarak görülmektedir. Sistemde yüzde aşımı tüm algoritmalarda %5 civarında, yükselme süresi 0,37 civarında, oturma süresi yaklaşık 6 saniye ve kalıcı durum hatası küçük olsa da 0.0024 civarındadır. Yüzde aşımının yüksek olması sisteme anlık olarak hızlı, düşük olması sisteme anlık olarak daha yavaş bir kontrol işlemi uygulandığını gösterir. Buna bağlı olarak yükselme süresi düşük ya da yüksek olur. Oturma süresinin kısa olması sistemin hızlıca istenilen yere ulaştığını, yüksek olması daha uzun sürede ulaştığını göstermektedir. Kalıcı durum hatası istenilen yerden ne kadar sapma olduğunu göstermektedir. Diğer kontrolcüler kullanılarak yükselme süresi, oturma süresi, yüzde aşımı ve kalıcı durum hatası düşürülmesi beklenmektedir. PIDA kontrolcüsü uygulanarak yüzde aşımının düşürülmesi beklenmektedir.

4.3.2 PIDA kontrolcü

PID kontrolcüsüne ivme eklenerek oluşturulan kontrolcüdür. Konumun türevi hızı, hızın türevi ivmeyi verdiği için 2 tane türev eklenerek oluşturulur. PIDA kontrolcüsünün değişken sayısı PID kontrolcüsünden 1 fazladır. PID kontrolcüsünde olduğu gibi tüm algoritmalar tüm denemelerde sonuca ulaşmıştır. PIDA kontrolcüsü ivme eklendiği için PID kontrolcüsüne göre daha kararlı olması ve daha az yüzde aşımı sağlaması beklenmektedir.

Çizelge 4.19 Optimizasyon algoritmalarının bulduğu PIDA kontrolcüsü katsayıları

Optimizasyon Algoritması	Kc	Ti	Td	Ta
ABC	0,055322	3,4043	1,1152	1e-05
BA	0,11491	3,9331	0,61954	0,0051425
FA	0,068752	3,4972	0,92667	0,0020547
IWO	0,057299	3,3832	1,078	0,00061612
PSO	0,054171	3,3763	1,1396	$1{,}0024 \times 10^{-5}$

Çizelge 4.19'da bulunan sonuçlar sisteme uygulanmıştır. PID kontrolcüsünde olduğu gibi PIDA kontrolcüsünde de değerler yakın olduğu için grafik çıktısı alınmamıştır ve çizelge 4.20'de sonuçlar verilmiştir.

Çizelge 7 PIDA kontrolcü parametrelerinin sistem cevabı sonuçları

Optimizasyon Algoritması	Yükselme Süresi	Oturma Süresi	Yüzde Aşımı	Kalıcı Durum Hatası
ABC	0,3698	5,9016	4,4006	0,0027
BA	0,4167	4,8130	4,2748	0,0050
FA	0,3852	5,5218	4,4674	0,0033
IWO	0,5564	6,5112	5,5184	0,0038
PSO	0,3753	6,0051	4,4675	0,0026

Yapılan literatür taramasından sonra PIDA kontrolcüsünün PID kontrolcüsünden daha iyi sonuçlar vermesi beklenmektedir. Ancak bulunan kontrolcü parametre değerlerinde ivme çok düşük olduğundan ve bulunan sonuçları sisteme uygulandıktan sonra alınan sonuçlarda da gözüktüğü gibi PID ile PIDA kontrolcüleri arasında çok fazla fark oluşmamıştır. Yüzde aşımının düşmesi beklenirken, bu olay gerçekleşmemiştir ve IWO algoritmasında yükselmiştir. Yükselme süresi PID kontrolcüsüne göre BA ve IWO algoritmalarında artmış, oturma süresi BA ve FA algoritmasında düşmüş, yüzde aşımı IWO algoritmasında yükselmiş ve kalıcı durum hatası BA ve IWO algoritmalarında artmıştır. Optimizasyon algoritmalarının bulduğu sonuçlar sisteme uygulandığında en iyi yükselme süresinin ABC algoritmasında, en kısa zamanda oturma süresi ve en az yüzde aşımı BA algoritmasında, ve en az kalıcı durum hatasının PSO algoritmasında olduğu gözlenmiştir.

4.3.3 PIDC

Bu çalışmada PID kontrolcüsüne faz geciktirici ve ilerletici filtresi eklenmiştir. Eklenen filtrelerin kutupları ve sıfırları olduğundan dolayı PID kontrolcüsünden 4 fazla parametreye sahiptir. Bazı çalışmalarda PIDA kontrolcüsüne seri filtreler eklenilerek PIDC kontrolcüsü de yapılmıştır. PIDC kontrolcüsündeki amaç yüzde aşımını düşürmek ve kalıcı durum hatasını azaltmaktır. PID ve PIDA kontrolcüsünde olduğu gibi tüm algoritmalar tüm denemelerde sonuca ulaşamamıştır. Parametre sayısı diğer algoritmalara göre daha fazla olduğu için bazı algoritmalar bazı denemelerde tam sonuca ulaşamamıştır. PIDC algoritmasındaki filtre sıfırları ve kutupları karmaşık da olabilmektedir ancak bu tez kapsamında sadece gerçek sayılara bakılmıştır.

Çizelge 8 Optimizasyon algoritmalarının bulduğu PIDC kontrolcüsü katsayıları

Optimizasyon Algoritması	Kc	Ti	Td	P1	Z1	P2	Z2
ABC	0.48609	0.46126	0.14054	2.9189	5	1e-05	4.3354
BA	0.47624	0.20686	0.1851	0.87112	3.1268	0.13551	4.5706
FA	0.66055	0.18487	0.13361	0.56452	5	0.5658	4.9747
IWO	0.65722	0.18786	0.13317	0.79181	4.9902	0.37103	4.9889
PSO	0.20516	2.1168	0.32655	0.31313	0.87386	2.8488	5

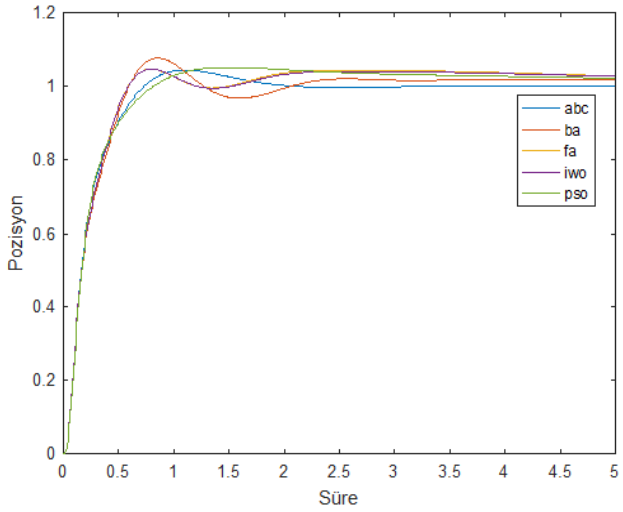
Optimizasyon algoritmalarında farklı sonuçlar çıkmıştır. Bu parametre sayısının artmasıyla ilgilidir. PIDC kontrolcüsünün sonuçları sisteme uygulanmış ve hem grafik çıktısı hem de çizelge olarak sistemin cevabının çıktısı alınmıştır.

Çizelge 9 PIDC kontrolcü parametrelerinin sistem cevabı sonuçları

Optimizasyon Algoritması	Yükselme Süresi	Oturma Süresi	Yüzde Aşımı	Kalıcı Durum Hatası
ABC	0,4321	1,5977	4,4102	$1{,}8883\text{x}10^{-6}$
BA	0,4190	1,9459	6,5775	0,0104
FA	0,3937	5,4924	4,3239	0,0038
IWO	0,3962	5,2660	4,1595	0,0051
PSO	0,4491	4,2722	4,5580	0,0057

PIDC kontrolcüsünün PID kontrolcüsünden daha iyi sonuçlar vermesi beklenmektedir. Yükselme süresi biraz artmış olsa da çok fazla bir değişiklik yoktur. Oturma süresi genel olarak azalmıştır ve ABC ile BA optimizasyon algoritmasında ciddi şekilde gözükmektedir. Yüzde aşımı genel olarak aynı kalmıştır ancak BA optimizasyon algoritmasında yükselmiştir. Kalıcı durum hatası genel olarak artsa da ABC algoritmasında ciddi oradan düşmüştür. Optimizasyon algoritmalarının bulduğu sonuçlar sisteme uygulandığında en iyi yükselme süresinin ve en az yüzde aşımı FA algoritmasında, en kısa zamanda oturma süresinin ve en az kalıcı durum hatasının ABC algoritmasında olduğu gözlenmiştir. Kalıcı durum hatası ABC algoritmasında çok düşük

gelmiştir. Bu durum ABC algoritmasının sonuca çok iyi gittiğini göstermektedir. Algoritma sonuçları genel olarak PID ve PIDA kontrolcüsüyle kıyaslandığında yükselme süresi artmış, oturma süresi ABC ve BA algoritmaları için ciddi oranda azalmış, yüzde aşımı genel olarak aynı kalmış ve kalıcı durum hatası ABC algoritmasında çok ciddi oranda düşmesine rağmen diğer algoritmalarda çok değişiklik göstermemiştir. Şekil 4.13'de PIDC kontrolcüsünün tüm algoritmaların verdiği sonuçlar sisteme uygulandığında verdiği cevap gösterilmektedir.



Şekil 4.13 PIDC kontrolcüsüne sistemin cevabı

4.3.4 2-Dof-PID

PID kontrolcüsüne 2 tane kazanç ekleyerek oluşturulur. PID kontrolcüsüne göre 2 fazla parametresi bulunmaktadır. Kontrolcünün 2 girdisi 1 çıktısı vardır. Referans ve hata kontrolcünün girdisidir. Referans ile hata kontrolcüye girip kontrolcünün içinde PID katsayılarından önce ve kazançlardan sonra hata referans sinyalinden çıkartılır. Bu işlem sonucunda PID parametrelerinin girdisi oluşur. PID ve PIDA kontrolcüsünde olduğu gibi tüm algoritmalar tüm denemelerde sonuca ulaşmıştır. PID kontrolcüsüne göre anlık yüzde aşımı ve kalıcı durum hatalarının daha aza indirgenmesi beklenmektedir.

Çizelge 10 Optimizasyon algoritmalarının bulduğu 2-Dof-PID kontrolcüsü katsayıları

Optimizasyon Algoritması	Kc	Ti	Td	B	C
ABC	0.0053636	1e-05	1.2803	50	4.543
BA	0.0050497	1e-05	1.2956	50	5
FA	0.0049574	1e-05	1.3208	50	5
IWO	0.0092293	1e-05	0.85789	49.998	4.6195
PSO	0.0049563	1e-05	1.3211	50	5

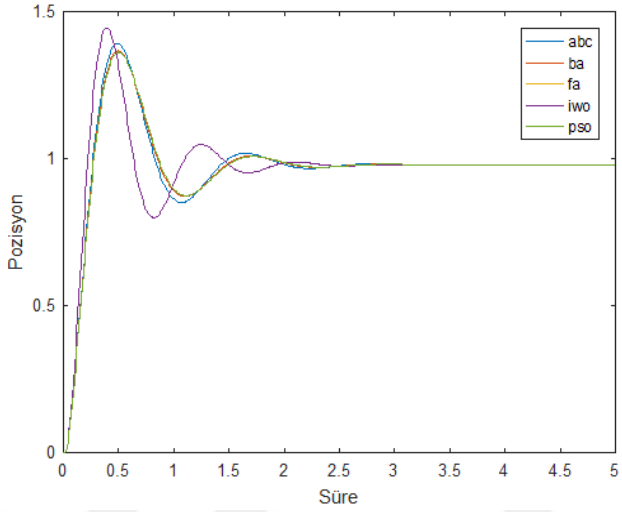
2-Dof-PID kontrolcüsünün sonuçları sisteme uygulanmış ve hem grafik çıktısı hem de çizelge olarak sistemin cevabının çıktısı alınmıştır. IWO optimizasyon algoritması diğer algoritmalara göre biraz daha farklı tepki vermiştir. Oransal terimi yüksek bulurken türev terimini düşük bulmuştur. İntegral terimi çok düşük bulunmuştur, bunun nedeni sistem anlık olarak istenilen yere hızlıca gitmektedir ve hata oluşturmamaktadır. Beklenen sistemin kalıcı durum hatası oluşturmadan hızlıca oturmasıdır.

Çizelge 11 2-Dof-PID kontrolcü parametrelerinin sistem cevabı sonuçları

Optimizasyon Algoritması	Yükselme Süresi	Oturma Süresi	Yüzde Aşımı	Kalıcı Durum Hatası
ABC	0,1781	1,8584	41,9995	0,0196
BA	0,1821	1,8715	39,4612	0,0201
FA	0,1836	1,8797	39,0184	0,0195
IWO	0,1411	1,776	47,2561	0,0197
PSO	0,1836	1,8798	39,0132	0,0196

2-Dof-PID kontrolcüsünün PID kontrolcüsünden daha iyi sonuçlar vermesi beklenmektedir. Oturma süresi ve yükselme süresi daha kısa olsa da sistemin yüzde aşımı çok fazladır. Kalıcı durum hatası da artmıştır. Beklendiği gibi sistem hızlıca oturmuştur ancak yüzde aşımı vardır ve kalıcı durum hatası biraz artmıştır. Optimizasyon algoritmalarının bulduğu sonuçlar sisteme uygulandığında en iyi yükselme süresinin ve en kısa zamanda oturma süresinin IWO algoritmasında, en az yüzde aşımı PSO

algoritmasında ve en az kalıcı durum hatasının FA algoritmasında olduğu gözlenmiştir. Algoritma sonuçları genel olarak diğer kontrolcülerle kıyaslandığında yükselme süresi kısalmış, oturma süresi ciddi oranda azalmış, yüzde aşımı ciddi oranda artmış ve kalıcı durum hatası algoritmalarda çok değişiklik göstermemiştir. Şekil 4.14'de 2-Dof-PID kontrolcüsünün tüm algoritmaların verdiği sonuçlar sisteme uygulandığında verdiği cevap gösterilmektedir.



Şekil 4.14 2-DoF-PID kontrolcüsüne sistemin cevabı

2-Dof-PID kontrolcüsünün PID kontrolcüsüne kıyasla beklenilen yüzde aşımının azalması ve kalıcı durum hatası azalması beklenildiği gibi sonuç vermemiştir. Beklenilenin aksine yüzde aşımı artmıştır ve kalıcı durum hatası çok değişiklik göstermemiştir ancak oturma süresi ciddi anlamda azalmıştır. 2-Dof PID kontrolcüsü aşırı yükselerek sistemi hızlı çalışıp, istenilen konuma hızlı gitmeye çalışarak oturma süresini kısa tutmuştur

4.3.5 Bulanık-PID

Bulanık mantık kontrolcüsü zeki kontrol metotlarından birisidir. Bulanık mantık kontrolcüsünün girişine kazanç ve türev parametreleri eklenir, bu sayede hata ve hatanın değişim hızı girdi olarak alınır. Çıkışta ise integral ve kazanç parametrelerinin toplamı

alınır. Bulanık-PID kontrolcüsünün 2 girdisi 1 çıktısı vardır. PID kontrolcüsüne göre 1 tane fazla parametresi vardır. Optimizasyon algoritmalarının hepsi Bulanık-PID kontrolcüsünde 10'ar kere denenmiştir ve en iyi olan sonuçlar alınmıştır. PID, PIDA ve 2-Dof-PID kontrolcülerinde olduğu gibi tüm algoritmalar tüm denemelerde sonuca ulaşmıştır. Bulanık mantık kontrolcüsü için Mamadani tipi bulanık kural tabanı kullanılmıştır. Denetleyicide bulanık kural tabanı, üyelik fonksiyonları, bulanık küme sayısı gibi etkenler performansı etkilemektedir. 5x5'lik bir bulanık küme oluşturulmuştur. Denetleyici performansında değişikliğe neden olan diğer bir faktör üyelik fonksiyonunun şeklidir. Üyelik fonksiyonu için üçgen ve trapez seçilmiştir. Kural tablosu aşağıdaki gibidir.

Çizelge 12 5x5 Bulanık Mantık Kontrolcüsü Kural Tablosu

	NH	NL	Z	PL	PH
NH	NH	NH	NH	NL	Z
NL	NH	NH	NL	Z	PL
Z	NH	NL	Z	PL	PH
PL	NL	Z	PL	PH	PH
PH	Z	PL	PH	PH	PH

Optimizasyon algoritmalarının bulduğu sonuçlar aşağıda verilmiştir.

Çizelge 13 Optimizasyon algoritmalarının bulduğu Bulanık-PID kontrolcüsü katsayıları

Optimizasyon Algoritması	**Error**	**Derror**	**Beta**	**Alfa**
ABC	0.021105	0.049953	4.4823	12.357
BA	0.019902	0.049748	4.4973	12.192
FA	0.019796	0.05	4.4698	12.124
IWO	0.015338	0.045524	6.7861	13.427
PSO	0.020164	0.05	4.3284	12.194

PID ve PIDA kontrolcülerinde olduğu gibi Bulanık-PID kontrolcüsünde de değerler yakın olduğu için grafik çıktısı alınmamıştır ve çizelgede sonuçlar verilmiştir. Sistemin yüzde aşımının düşmesi ve kalıcı durum hatasının azalması beklenmektedir.

Çizelge 14 Bulanık-PID kontrolcü parametrelerinin sistem cevabı sonuçları

Optimizasyon Algoritması	Yükselme Süresi	Oturma Süresi	Yüzde Aşımı	Kalıcı Durum Hatası
ABC	2,2179	3,4521	1,4139	0,0209
BA	2,2915	3,5843	1,2628	0,0213
FA	2,3127	3,6226	1,2189	0,0214
IWO	2,1629	3,4424	1,1782	0,0148
PSO	2,3239	3,6367	1,2166	0,0222

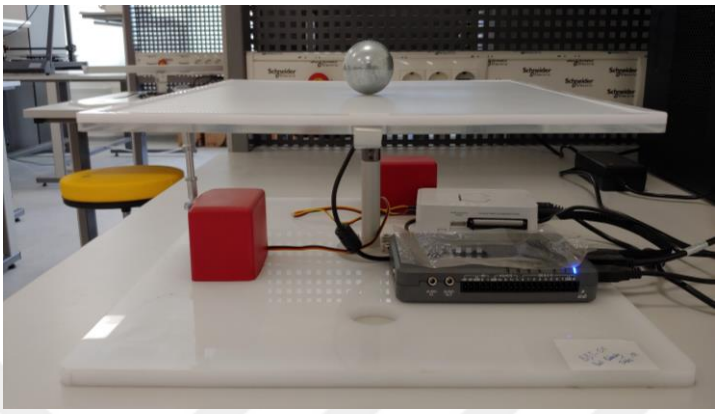
Sistemin verdiği cevaplarda genel olarak yükselme süresi artmış, oturma süresi PID kontrolcüsüne göre kısalmış ancak 2-Dof-PID kontrolcüsüne göre yavaşlamış, yüzde aşımı çok büyük oranda düşmüş, kalıcı durum hatası ise 2-Dof-PID kontrolcüleriyle benzerdir. Optimizasyon algoritmalarının bulduğu sonuçlar sisteme uygulandığında en iyi sonuçları IWO algoritması vermiştir.

4.4 Laboratuvar Çalışmaları

Sistemin modellenmesi ve kontrolcülerin hepsinin parametreleri optimizasyon algoritmalarıyla simülasyon ile Matlab Simulink ile bulunduktan sonra gerçek sistemin üzerinde denemeler yapılmıştır. Acrome'un top dengeleme düzeneği kullanmıştır. Kontrolcü ve kontrolcü parametreleri LabView aracılığıyla sisteme uygulanmıştır. Sisteme tüm kontrolcüler uygulandıktan sonra, *x* ve *y* eksenlerindeki konumlar kaydedilmiş ve Matlab ortamında grafikleri elde edilmiştir. Bu çıktılar Matlab'da çizdirilmiş ve yorumlanmıştır.

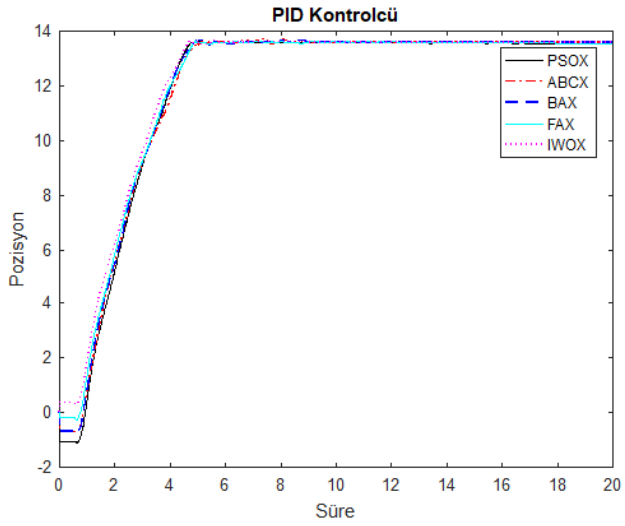
Top dengeleme düzeneğinde top tablanın tam ortasına (0,0 noktasına, orijine) konmuştur. Top koordinat sisteminin orijin noktasına konumlandırıldıktan sonra topun orijinden

10,10 konumuna gitmesi ve o pozisyonda kalması istenmektedir. Tüm kontrolcüler LabView ortamında hazırlanmıştır. Şekil 4.15'de top dengeleme düzeneği gösterilmektedir.

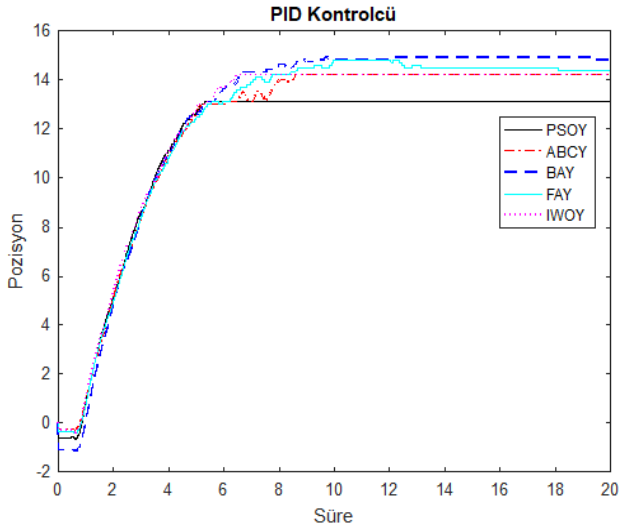


Şekil 4.15 ACROME Top Dengeleme Düzeneği

Matlab Simulink ortamında optimizasyon algoritmalarının bulduğu kontrolcü parametrelerinin hepsi sisteme uygulanmıştır. Sisteme uygulanan ilk kontrolcü PID kontrolcüsüdür. Tüm optimizasyon algoritmalarının bulduğu sonuçlar doğrultusunda *x* yönünde optimizasyon algoritmaları benzer kalıcı durum hatasıyla sonuç verirken *y* yönünde bazı optimizasyon algoritmalarının bulduğu sonuçlar değişiklik göstermiştir ve kalıcı durum hatasını azaltmıştır. Tüm algoritmalarda iki yönde de kalıcı durum hatası oluşurken *x* yönünde ki kalıcı durum hataları çok yakın olup, *y* yönündeki kalıcı durum hataları farklılık göstermektedir. Kontrolcü parametrelerinde en az kalıcı durum hatasına sahip optimizasyon algoritması PSO, en çok kalıcı durum hatasına sahip optimizasyon algoritması ise BA olarak tespit edilmiştir. Optimizasyon algoritmalarının oturma süreleri, yüzde aşımı, yükselme süreleri *x* yönü için birbirleriyle benzerdir. Ancak *y* yönü için oturma süresi, kalıcı durum hatası gibi farklılıklar gözlemlenmektedir.



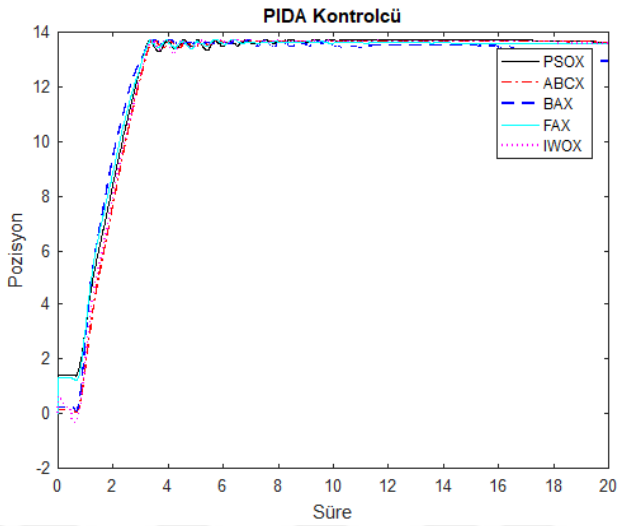
Şekil 4.16 PID Kontrolcü X yönü



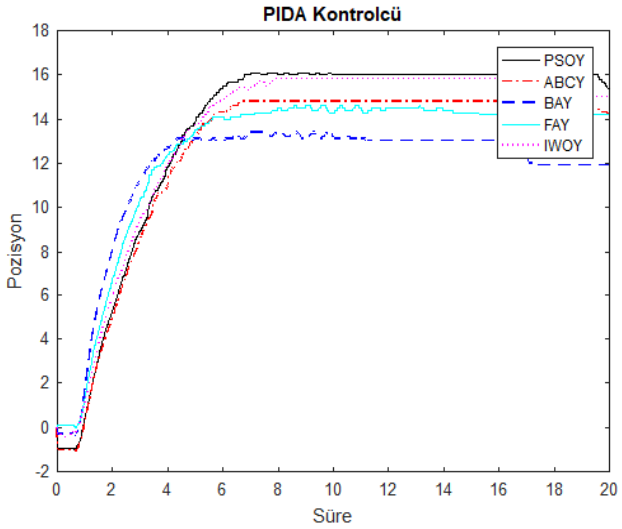
Şekil 4.17 PID Kontrolcü Y yönü

Sisteme uygulanan ikinci kontrolcü PIDA kontrolcüsüdür. Tüm optimizasyon algoritmalarının bulduğu sonuçlar doğrultusunda *x* yönünde kalıcı durum hatasında, oturma süresinde, yüzde aşımında farklılıklar olmazken, *y* yönünde bazı optimizasyon algoritmalarının bulduğu sonuçlar kalıcı durum hatasını azaltmıştır. Tüm algoritmalarda iki yönde de kalıcı durum hatası oluşurken *x* yönünde ki kalıcı durum hataları çok yakın olmasına rağmen en az kalıcı durum hatası BA optimizasyon algoritmasındadır. Optimizasyon algoritmalarının *y* yönündeki oturma süreleri ve kalıcı durum hataları birbirinden farklıdır. En az kalıcı durum hatasına sahip kontrolcü parametrelerini bulan optimizasyon algoritması PID kontrolcüsünün tam tersi olarak, BA optimizasyon

algoritması olduğu gözlemlenmiştir. En çok kalıcı durum hatası olan kontrolcü parametrelerini bulan optimizasyon algoritması ise yine PID kontrolcüsünün tam tersi olarak PSO optimizasyon algoritması olduğu gözlemlenmiştir.



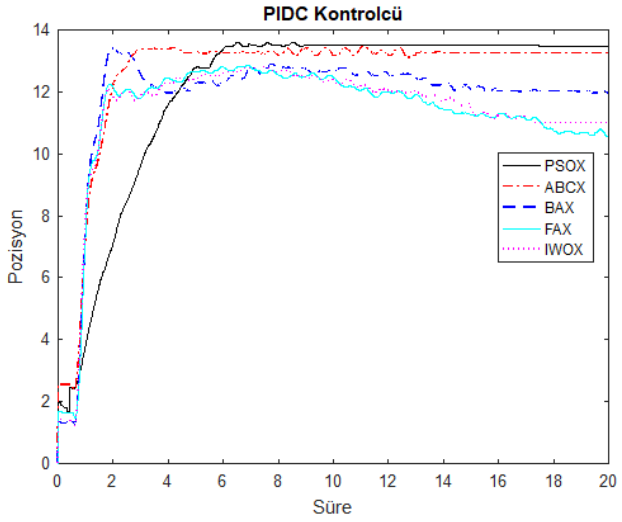
Şekil 4.18 PIDA Kontrolcü X yönü



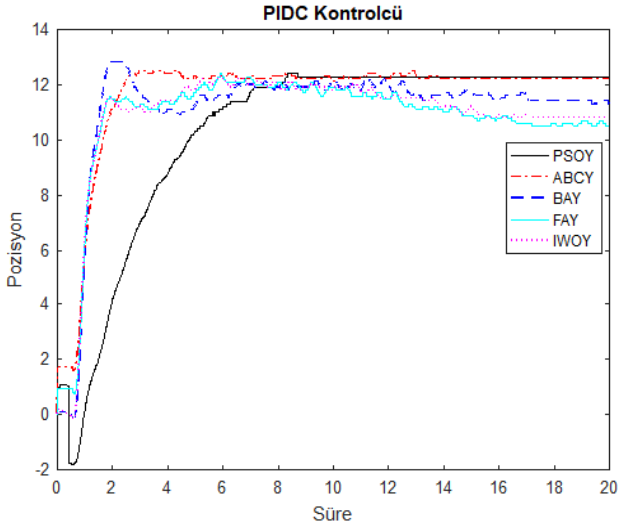
Şekil 4.19 PIDA Kontrolcü Y yönü

Sisteme PIDA kontrolcüsünden sonra uygulanan kontrolcü PIDC kontrolcüsüdür. Tüm optimizasyon algoritmalarının bulduğu sonuçlar doğrultusunda *x* ve *y* yönünde bazı optimizasyon algoritmalarının bulduğu sonuçlar oturma süresi ve kalıcı durum hatasında değişiklik göstermiştir ve kalıcı durum hatasını azaltmıştır. Tüm algoritmalarda iki yönde

de kalıcı durum hatası oluşurken *x* ve *y* yönünde ki en az kalıcı durum hatasına sahip kontrolcü parametrelerini bulan optimizasyon algoritması FA optimizasyon algoritmasıdır. FA optimizasyon algoritması en az kalıcı durum hatasına sahip olmasına rağmen osilasyona sahiptir. Topu istenen konumda tutmak için sürekli olarak tablayı hareket ettirmektedir. FA optimizasyon algoritmasından sonra en az kalıcı durum hatasına sahip olan IWO ve BA algoritmasıda FA optimizasyon algoritması gibi osile etmektedir ancak FA optimizasyon algoritmana göre daha az osile etmektedir. ABC ve PSO optimizasyon algoritmaları osile etmemektedir ancak kalıcı durum hataları daha yüksek çıkmıştır. PSO optimizasyon algoritmasının oturma süreside yüksek çıkmıştır. BA optimizasyon algoritmasının yüzde aşımı yüksektir.

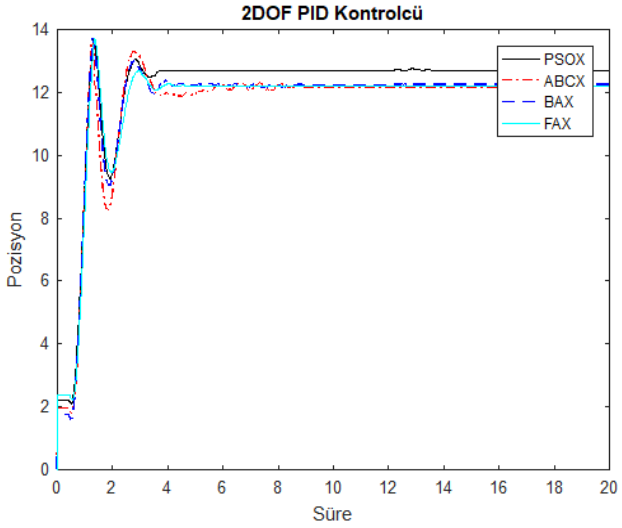


Şekil 4.20 PIDC Kontrolcü X yönü

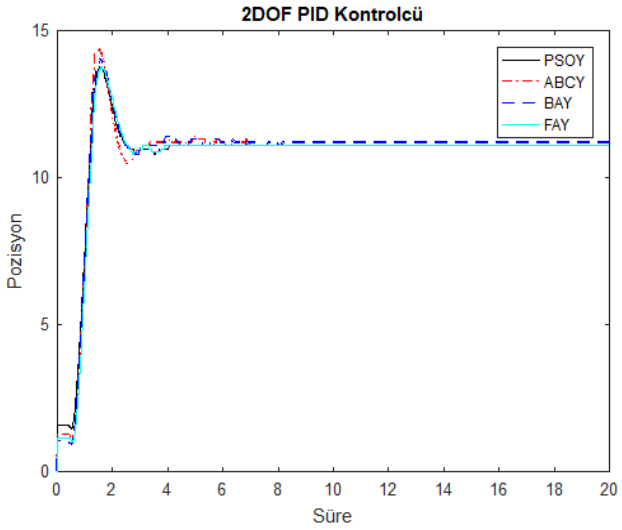


Şekil 4.21 PIDC Kontrolcü Y yönü

Sisteme PIDC kontrolcüsünden sonra uygulanan kontrolcü 2-Dof-PID kontrolcüsüdür. Tüm optimizasyon algoritmalarının bulduğu kontrolcü parametreleri sisteme uygulanmıştır. IWO kontrolcüsünün bulduğu kontrolcü parametreleri sisteme uygulandığında top tabladam fırlamıştır. Tekrar denemeler yapılmıştır, başlangıç konumu tam 0,0 olarak ayarlanıp denenmiştir ancak top sürekli tabladan fırlamıştır. Bu sebeple IWO optimizasyon algoritmasının parametrelerinin sonucu bulunmamaktadır. Geriye kalan tüm optimizasyon algoritmalarının bulduğu kontrolcü parametreleri doğrultusunda *x* ve *y* yönündeki sonuçlar çok fazla değişiklik göstermemiştir. Tüm algoritmalarda iki yönde de kalıcı durum hatası oluşmuştur. Sonuçların hepsi birbirine yakın çıkmıştır. Sadece *x* yönündeki PSO algoritmasının bulduğu sonuçta diğer sonuçlara göre %7 daha fazla kalıcı durum hatası oluşmuştur.

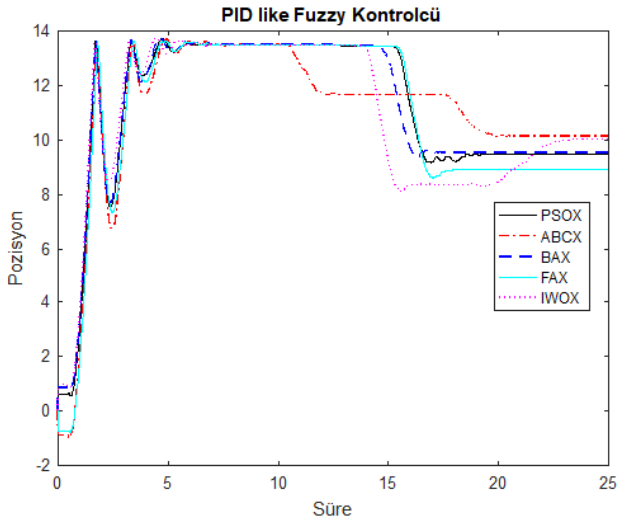


Şekil 4.22 2-Dof PID Kontrolcü x yönü

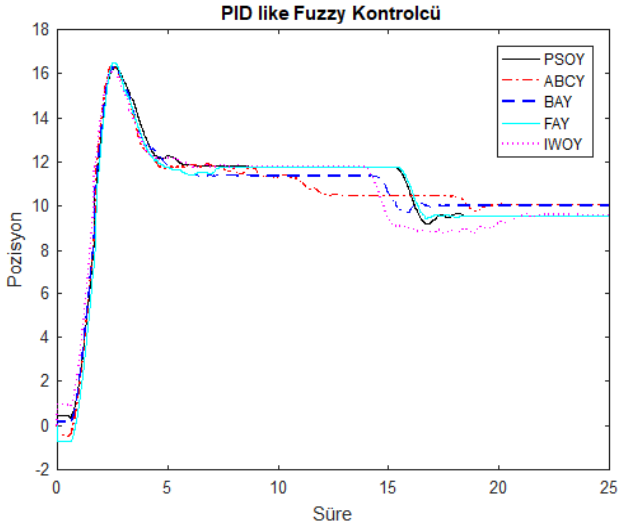


Şekil 4.23 2-Dof PID Kontrolcü y yönü

Son olarak sisteme Bulanık-PID kontrolcüsü uygulanmıştır. Tüm optimizasyon algoritmalarının bulduğu sonuçlar doğrultusunda *x* ve *y* yönünde bazı optimizasyon algoritmalarının bulduğu sonuçlar değişiklik göstermiştir, kalıcı durum hatasını azalmış ve bazı sonuçlarda gözle görülür şekilde küçülmüştür. Bu kontrolcünün diğer kontrolcülerden farklı olarak 25 saniye alınmasının nedeni IWO algoritmasının bulduğu sonuçların 20 saniyeden sonra değişiklik göstermesidir. Bulanık-PID kontrolcüsünde bulunan sonuçlar diğer kontrolcülere göre her iki yönde de daha az kalıcı durum hatası vermektedir. Ancak *x* yönündeki yüzde aşımı diğer kontrolcülere göre fazladır.



Şekil 4.24 Bulanık-PID Kontrolcü x yönü



Şekil 4.25 Bulanık-PID Kontrolcü y yönü

5. TARTIŞMA VE SONUÇLAR

Kontrol algoritmaları hayatımızın hemen hemen her yerinde kullanılmakta ve kullanımı giderek artmaktadır. Sistemlerin hızlı ve istikrarlı olması büyük önem taşımaktadır. Bunun içinde kullanılan kontrolcünün ve kontrolcü parametrelerinin iyi seçilip uygulanması önem taşımaktadır. Bu tez kapsamında yapılan çalışmada öncelikle, deney düzeneği Lagrangian metodu ile modellenmiştir. Sisteme uygulanacak kontrolcülerin parametrelerini bulmak için yapay arı kolonisi algoritması, arılar algoritması, ateşböceği algoritması, yabani ot optimizasyonu algoritması ve parçacık sürüsü optimizasyonu algoritmaları seçilmiştir. Bu optimizasyon algoritmaları kendi aralarında kıyaslanmak için CEC 17'nin ilk 9 fonksiyonunda 10'ar defa 2,10 ve 30 popülasyonda sırasıyla 5000, 10000 ve 25000 iterasyonda denenmiştir. Denemelerin sonunda sonuç olarak bulunan en yüksek değer, en küçük değer, sonuçların ortalaması ve standart sapması bulunmuştur. Daha sonra gerçek sisteme uygulanacak olan PID, PIDA, PIDC, 2-Dof PID ve Bulanık-PID algoritmaları Matlab Simulink ortamında simüle edilmiştir. Kontrolcülerin hepsine 10'ar defa optimizasyon algoritmalarının hepsi uygulanmış ve en iyi sonuçlar gerçek sisteme uygulanmak üzere not edilmiştir. Alınan sonuçlar Acrome top dengeleme düzeneğinde LabView aracılığıyla denenmiştir.

Top dengeleme düzeneği iki serbestlik derecesine sahip, kararsız bir sistemdir. Sistemin modellenebilmesi için literatürde olan Lagrangian ve Newton Hareket metotlarından, Lagrangian metodu seçilmiştir. Lagrangian metoduyla *x* ve *y* yönü için farklı 2 transfer fonksiyonu bulunmuştur. Ancak hızlı sonuca ulaşmak için ortak bir transfer fonksiyonu belirlenmiştir ve tüm kontrolcü algoritmalarının parametreleri bu şekilde bulunmuştur.

Optimizasyon algoritmaları CEC 17 fonksiyonlarında denendiğinde en iyi algoritma ateşböceği algoritması olarak gözükmektedir. Ateş böceği algoritması tüm denemelerde en hızlı şekilde ve en küçük hatayla sonuca yakınsamıştır. Optimizasyon algoritmalarında boyut arttıkça lokal minimumlar arttığından dolayı istenilen sonuca ulaşmak zorlaşmıştır. Ancak ateşböceği algoritmasında bu durum daha az yaşanmıştır. Diğer algoritmalar problemleri çözmüş olsa da bazen yakınsamakta problem yaşamışlardır. Ateşböceği

algoritması için iterasyon sayısı düşük tutulabilir ancak diğer algoritmalar için aynısı söz konusu değildir.

Yapılan çalışmada ele alınan kontrolcülerin çıkış noktası PID kontrolcüsüdür. PID kontrolcüsünün yetersiz kaldığı yerlerde diğer kontrolcüler kullanılmaktadır. Uygulanan tüm algoritmaların PID kontrolcüsünden daha iyi yanıt vermesi beklenmiştir. Simülasyon sonuçlarına bakıldığında tüm optimizasyon algoritmaları için PID ve PIDA kontrolcüleri arasında fark gözükmemektedir. PIDC kontrolcüsünde ise yapay arı kolonisi algoritması haricinde diğer optimizasyon algoritmalarının bulduğu sonuçlarda farklılık yoktur. Yapay arı kolonisi algoritması kalıcı durum hatasını neredeyse ortadan kaldırmıştır ve oturma süresini ciddi ölçüde düşürmüştür. 2-Dof PID kontrolcüsünde ise yüzde aşımı yaparak sistem hızlı çalışıp, oturma süresini kısa tutmuştur. Tüm algoritma sonuçları birbirleriyle benzerlik göstermektedir. Bulanık-PID kontrolcüsünde ise tüm algoritmalar için yükselme süresi yavaş, yüzde aşımı düşüktür.

Alınan sonuçlar *x* ve *y* yönü kontrolü için gerçek sistemde kontrolcülere uygulanmıştır. PID kontrolcüsü için tüm optimizasyon algoritmaları *x* yönünde aynı şekilde kontrol etmiştir ancak *y* yönünde parçacık sürüsü optimizasyonu daha az kalıcı durum hatasıyla kontrol etmiştir, arılar algoritması ise en kötü kontrol eden optimizasyon algoritması olmuştur. PIDA kontrolcüsünde ise PID kontrolcüsünde ki gibi *x* yönünde tüm algoritmalar aynı şekilde kontrol ederken y yönünde arılar algoritması en iyi algoritma olurken parçacık sürüsü optimizasyonu en kötü algoritma olarak gözlemlenmektedir. Ateşböceği algoritmasının, yabani ot optimizasyonunun ve arılar algoritmasının bulduğu PIDC kontrolcüsü parametreleri sisteme uygulandığında sistem osile etmiştir. En az kalıcı durum hatası ateşböceği algoritmasında olsa da sistem kararlı duruma ulaşmamıştır. 2-Dof PID kontrolcüsü yabani ot algoritmasında topu tabla üzerinde tutamamıştır. Diğer optimizasyon algoritmaları sonuçları da yüzde aşımı yapsa da topu tabla üzerinde tutmuşlardır ve sistemi kararlı hale belirli bir kalıcı durum hatasıyla getirmişlerdir. *y* yönündeki kalıcı durum hatası ciddi anlamda diğer kontrolcülere göre düşmüştür. Bulanık-PID kontrolcüsü belirli bir süre sonra en az kalıcı durum hatasıyla kontrol eden kontrolcü olmuştur.

CEC 17 sonuçlarından sonra en iyi algoritma olarak gözüken ateşböceği algoritması simülasyon sonucunda da gerçek sisteme uygulandığında da kendisini diğer algoritmalardan ayıracak bir sonuç vermemiştir. Yabani ot algoritmasının bulduğu 2-Dof PID kontrolcüsü parametreleri gerçek sisteme uygulandığında sistem topu tablada tutamamıştır bu sebeple yabani ot algoritması en kötü algoritma olarak seçilmiştir. Diğer optimizasyon algoritmalarının birbirlerine kesin bir üstünlükleri görülmemiştir ancak PIDC kontrolcüsünde yabani ot optimizasyonu, arılar algoritması ve ateşböceği algoritması sonuçları osile etmiştir. Bu sebeple parçacık sürü optimizasyonu ve yapay arı kolonisi diğer algoritmalardan daha iyi olarak yorumlanmaktadır. Kontrolcü olarak PID ve PIDA kontrolcüleri arasında gerçek sistemde ve simülasyon ortamında bir fark görülmemiştir. PIDC kontrolcüsünde simülasyonda yapay arı kolonisinin bulduğu parametre değerleri PID kontrolcüsüne göre daha iyi gözükürken gerçek sistemde fark göstermemiştir. 2-Dof-PID kontrolcüsü PID kontrolcüsüne göre daha hızlı ve kararlı bir görüntü çizmiştir. 2-Dof-PID kontrolcüsü PID kontrolcüsünden daha iyi kontrol sağlamıştır. Bulanık-PID kontrolcü en az kalıcı durum hatasına sahip kontrolcü olmuştur. PID kontrolcüsünden daha iyi bir kontrol sağlamıştır. Belirli bir süre geçtikten sonra 2-DoF-PID kontrolcüsünden daha az kalıcı durum hatasıyla sistemi kontrol etmiştir.

Simülasyon ile gerçek uygulama arasında farklılıklar oluşmuştur. Sistem gerçek hayatta farklı sonuçlar vermiştir. PID kontrolcüsünü iyileştirmek için ivme parametresi eklenerek oluşturulan PIDA kontrolcüsü ve kutup ile sıfır eklenerek oluşturulan PIDC kontrolcüsü PID kontrolcüsünden farklı çalışmamıştır. Yani PID kontrolcüsüne parametre eklemek her zaman iyi bir sonuç vermemiştir aksine çalışma süresini arttırmıştır.

Yapılan çalışmada sezgisel optimizasyon algoritmaları içinde en iyi sonuçları yapay arı kolonisi algoritması ve parçacık sürüsü optimizasyon algoritmasının bulduğu gözlenmiştir. Kontrolcü olarak PID, PIDC ve PIDA kontrolcülerinin farklarının olmadığı ancak 2-Dof PID ve Bulanık-PID kontrolcülerinin daha iyi olduğu gözlemlenmiştir.

KAYNAKLAR

Acrome Ltd. 2014. Ball Balancing Table User Manual. Acrome Ltd, 1-64 p., İstanbul.

Ajiatmo, D., Robandi, I. 2016. A Hybrid Fuzzy Logic Controller-Firefly Algorithm (FLC-FA) Based for MPPT Photovoltaic (PV) System in Solar Car. 2016 IEEE International Conference on Power and Renewable Energy (ICPRE), 21-23 Oct. 2016, IEEE, 606-610, Shanghai, China.

Akyol, S., Alataş B. 2012. Güncel Sürü Zekâsı Optimizasyon Algoritmaları. Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi. 1: 36-50.

Aldemir, A., Anwer, M., S. 2020. Determination of optimal PID control parameters by response surface methodology. Advanced Researches and Engineering Journal. 04(02), 142-153.

Aldemir, A., Hapoğlu, H. 2016. Kablosuz Sıcaklık Kontrolü İçin PID Ayarlama Yöntemlerinin Karşılaştırılması. Journal of Polytechnic. 19(1), 9-19.

Ali, H., Jassim, H., M., Hasan, A., F. 2019. Optimal Nonlinear Model Reference Controller Design for Ball and Plate System. Springer. 44, 6757–6768.

Amirinejad, M., Eslami, M., Noori, A. 2014. Automatic PID Controller Parameter Tuning Using Bees Algorithm. International Journal of Scientific & Engineering Research. 5(8), 24-28.

Anderson, G., Doherty, R., Ganapathy, S. 2011. User Perception of Touch Screen Latency. Springer. 6769, 195-202.

Aras, M., S., M., Salim, S., N., B., S., Hoo, E., C., S., Razak, I., A., B., W., A., Hairi, M., H., B. 2009. Comparison of Fuzzy Control Rules Using Matlab Toolbox and Simulink for DC Induction Motor-Speed Control. International Conference of Soft Computing and Pattern Recognition. 4-7 Dec. 2009, Malacca, Malaysia.

Asta, S., Uyar, S. 2011. A novel particle swarm optimization algorithm. 10th international conference on Artificial Evolution, Angers, France.

Bagis, A., Senberber, H. 2017. ABC Algorithm Based PID Controller Design for Higher Order Oscillatory Systems. Elektronika ır Elektronıka. 23(6), 3-9.

Bakırcıoğlu, V., Sen, M., A., Kalyoncu, M. 2016. Optimization of PID controller based on The Bees Algorithm for one leg of a quadruped robot. MATEC Web of Conferences, 17 February 2016, EDP Sciences, vol 42.

Belen, M., A., Alıcı, M., Çor, A., Güneş, F. 2014. Ateşböceği Algoritması ile Mikrodalga Transistör Performans Karakterizasyonu. Eleco 2014 Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu, 27-29 Kasım, Bursa.

Bendjeghaba, O., Ishak Boushaki, S., Zemmour, N. 2013. Firefly Algorithm for optimal tuning of PID controller parameters. 4th International Conference on Power Engineering, Energy and Electrical Drives, 13-17 May 2013, IEEE, 1293-1296, İstanbul, Turkey.

Bilgiç, H., H., Tutumlu, S., Conker, Ç. (2021). Top ve çubuk sistemi için kaskad denetleyici parametrelerinin meta-sezgisel algoritmalarla optimizasyonu. DEÜ FMD. 23(67), 157-167.

Bingul, Z., Karahan, O. 2018. Comparison of PID and FOPID controllers tuned by PSO and ABC algorithms for unstable and integrating systems with time delay. Optim Control Appl Meth. 39, 1431-1450.

Borah, M., Roy, P., Roy, B., K. 2018. Enhanced Performance in Trajectory Tracking of a Ball and Plate System using Fractional Order Controller. IETE Journal of Research. 64(1), 76-86.

Boskovic, M., C., Sekara, T., B., Radulovic, M., Cvetkovic, B. 2016. A Novel Method for Optimization of PID/PIDC Controller Under Constraints on Phase Margin and Sensitivity to Measurement Noise Based on Non-symmetrical Optimum Method. ETF Journal of Electrical Engineering. 22(1), 1-10.

Boskovic, M., Rapaic, M., R., Sekara, T., B., Mandic, P., D., Lazarevic, M., P. 2017. Pole placement based design of PIDC controller under constraint on robustness. Infoteh-Jahorına. 16, 664-68.

Boukhalfa, G., Belkacem, S., Chikhi, A., Benaggoune, S. 2019. Genetic algorithm and particle swarm optimization tuned fuzzy PID controller on direct torque control of dual star induction motor. Springer Nature. 29, 1886-96.

Boz, A., F., Çimen, M., E. 2017. Geliştirilmiş Ateşböceği Algoritması ile PID Denetleyicisi Tasarımı. 8th International Advanced Technologies Symposium (IATS'17), 19-22 Ekim, Elazığ.

Braescu, F., C., Ferariu, L., Gilca, R., Bordianu, V. 2012. Ball on Plate Balancing System for Multi-Discipline Educational Purposes. 6th International Conference on System Theory, Control and Computing (ICSTCC). 13 December 2012, Sinaia, Romania.

Chen, C., T. 1993. Analog and digital control system design: transfer function, state-space and algebraic methods. Philadelphia: Saunders College Publishing, 600, USA.

Dash, P., Saika, L., C., Sinha, N. 2014. AGC of a Multi-area Interconnected System with FACTS and Firefly Optimized 2-Dof PID Controller. 2014 International Conference on Control, Instrumentation, Energy & Communication, 31 Jan.-2 Feb. 2014, IEEE, 397-401, Calcutta, India.

Debnath, M.,K., Singh, M.,B., Mallick, R.,K. 2016. Design of optimal 2-DOF PID controller using GWO technique for automatic generation control of a multisource power system. 2016 IEEE Uttar Pradesh Section International Conference on

Electrical, Computer and Electronics Engineering (UPCON) (pp. 531-536), IEEE.

Der, O., Vural, R., A., Yıldırım, T. 2008. Parçacık Sürü Optimizasyonu Tabanlı Evirici Tasarımı. Elektronik ve Haberleşme Müh. Bölümü,Yıldız Teknik Üniversitesi.

Develi, H. 2005. Ziegler Nichols Yöntemi ve Migo Yaklaşımı. Yüksek Lisans Tezi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Elektronik Fakültesi. İstanbul.

Dong, X., Zhang, Z., Chen, C. 2009. Applying Genetic Algoritm to On-line Updated PID Neural Network Controllers for Ball and Plate System. Fourth International Conference on Innovative Computing, Information and Control. 7-9 Dec. 2009, IEEE, 751-755, Kaohsiung, Taiwan.

Donuk, K., Özbey, N., İnan, M., Yeroğlu, C. 2018. Investigation of PIDA Controller Parameters via PSO Algorithm. International Conference on Artificial Intellifence and Data Processing. 28-30 Sept, 2018, IEEE, 1-6, Malatya, Turkey.

Dos Reis, L., L., N., Bezzera, J., P., De Mesquita, V., A., Silva, W., A., Souza, D., A., De O., J., R., Correia, W., B. 2019. A 2DOF PID for Dynamic Control of Drive System Switched Reluctance Motor. 45th Annual Conference of the IEEE Industrial Electronics Society. 14-17 Oct. 2019, IEEE, 823-827, Lisbon, Portugal.

Dušek, F., Honc, D., Sharma R. 2017. Modelling of Ball and Plate System Based on First Principle Model and Optimal Control. 21st International Conference on Process Control (PC). 6-9 June 2017, IEEE, 216-221, Strbske Pleso, Slovakia.

Edaris, Z., L., Abdul Rahman, S. 2016. Performance Comparıson of Pid Tuning by Using Ziegler-Nichols and Partıcle Swarm Optimizatıon Approaches in a Water Control System. Journal of ICT. 15(1), 203-224.

Eke, İ., Taplamacıoğlu, M., C., Kocaarslan, İ. 2011. Yapay Arı Kolonisi Algoritması Tabanlı Kararlı Güç Sistemi Dengeleyicisi Tasarımı. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi. 26(3), 683-690.

Gidemen, G., Furat, M. 2015. PID Parametrelerinin Ayarlama Yöntemleri: 2. Derece Sistem Modeline Uygulanması ve Karşılaştırmalı Olarak Değerlendirilmesi. Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 30(2), 355- 362.

Gozde, H. 2019. Evolutionary Computation Based Control for Ball and Plate Stabilization System. Balkan Journal of Electrical & Computer Engineering. 7(1), 44-55.

Graham, D., Lathrop, R., C. 1953. The synthesis of "optimum" transient response: criteria and standard forms. Trans Amer Inst Elect Eng. 72(5), 273-288.

Hambali N., Masngut, A., Ishak, A., A., Janin, Z. 2014. Process Controllability for Flow Control System Using Ziegler-Nichols (ZN), Cohen-Coon (CC) and Chien-Hrones-Reswick (CHR) Tuning Methods. 2014 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), 25-25 Nov. 2014, IEEE, 1-6, Kuala Lumpur, Malaysia.

Hamdoun, M., Abdallah, M., B., Ayadi, M., Rotella, F., Zambettakis, I. 2021. Functional observer-based feedback controller for ball balancing table. SN Applied Sciences. 3(6): 1-9.

Hmidi, M., E., Salem I., B., Aamraoui, L., E. 2017. Design and Modeling of PID Controller with 2DOF: Application to Thermal Phase of an Hybrid Vehicle. 14th International Multi-Conference on Systems, Signals & Devices(SSD). 28-31 March 2017, IEEE, 419-424, Marrakech, Morocco.

Hmidi, M., E., Salem, I., B., Amraouı, L., E. 2017. 2-Dof PID for Dynamic Control of drive system Hybrid Vehicle electrical. 2017 International Conference on Green Energy Conversion Systems (GECS), 23-25 March 2017, IEEE, 1-5, Hammamet, Tunisia.

Indhuja, R., Kamaraj, N. 2016. Implementation of PSO tuned robust 2-Dof PID controller for AVR system. 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 16-17 Dec. 2016, IEEE, 159-163, Kumaracoil, India.

Isdaryani, F., Feriyonika, F., Ferdiansyah, R. 2020. Comparison of Ziegler-Nichols and Cohen Coon tuning method for magnetic levitation control system. Journal of Physics: Conference Series, 24–25 October 2019, Vol. 1450, 012033, Bali, Indonesia.

Itani, A. 2017. Ball Plate Balancing System Using Image Processing. the Degree of Master. Near East University, Mechatronic Engineering, Nicosia.

Jitwang, T., Hlangnamthip, S., Puangdownreong, D. 2020. Robust PIDA Controller Design by Cuckoo Search for Liquid-Level Control System. 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), 11-14 March 2020, IEEE, 226-229, Pattaya, Thailand.

Jorgensen, V. 1974. Ball-Balancing System For Demonstration Of Basic Concepts In The State-Space Control Theory. International Journal of Electrical Engineering Education. 11(4), 367-376.

Jung, S., Dorf, R., C. 1995. Novel Analytic Technique for PID and PIDA Controller Design. Elsevier. 29(1), 1146-51.

Karaboga, D. An idea based on honey bee swarm for numerical optimization. 2005.Technical ReportTR06, Erciyes University, Engineering Faculty, Computer Engineering Department,

Karasakal, O., Yeşil, E., Güzelkaya, M., & Eksin, İ. 2005. Implementation of a new self-tuning fuzzy PID controller on PLC. Turkish Journal of Electrical Engineering & Computer Sciences, 13(2), 277-286.

Keçeci, E., Yumuk, E., Guzelkaya, M., Eksın, I. 2019. Comparison of Optimal Integer

and Fractional Order PID Controllers on a Stabilized Real-Time System. 2019 11th International Conference on Electrical and Electronics Engineering (ELECO), 28-30 Nov. 2019, IEEE, 754-758, Bursa, Turkey.

Kennedy, J., Eberhart, R. 1995. Particle Swarm Optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 27 Nov.-1 Dec. 1995, IEEE, Vol. 4, 1942-1948, Perth, WA, Australia.

Ker, C., C., Lin, C., E., Wanj, R., T., T. 2007. Tracking and Balance Control of Ball and Plate System. Journal of Chinese Institute of Engineers. 30(3), 459-470.

Kılıç, E., & Özdemir, M. T. 2019. Güç sistemlerindeki optimum otomatik gerilim regülasyonu için çoklu amaç fonksiyonunun belirlenmesi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(1), 1-12.

Kopichev, M., M., Putov, A., V., Pashenko, A., N. 2019. Ball on the Plate Balancing Control System. IOP Conference Series Materials Science and Engineering, IOP Publishing, 26–28 July 2019, Vol. 638, 012004, Prague, Czech Republic.

Kostamo, J., Hyötyniemi, H., Kuosmanen, P. 2005. Ball Balancing System: An Educational Device for Demonstrating Optimal Control. International Symposium on Computational Intelligence in Robotics and Automation. 27-30 June 2005, IEEE, 379-384, Espoo, Finland.

Kumar, E. G., & Gowthaman, E. (2017). Cascade PID-Lead Compensator Controller for Non-overshoot Time Responses of unstable system. Energy Procedia, 117, 708-715.

Kumar, M., Sharma, P., Kumar, P. 2020. Trajectory Planning of Unmanned Aerial Vehicle using Invasive Weed Optimization. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2-4 July 2020, IEEE, 467-472, Coimbatore, India.

Küçüksille, E., U., Tokmak, M. 2011. Yapay Arı Kolonisi Algoritması Kullanarak Otomatik Ders Çizelgeleme. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 15(3), 203-210.

Li, Y., Ang, K., H., Chong, G. 2005. PID Control System Analysis, Design, and Technology. IEEE Transactions on Control Systems Technology. 13(4), 559 – 576.

Mashhadi, S. K. M. (2013, August). PID like fuzzy logic control of an unmanned underwater vehicle. In 2013 13th Iranian Conference on Fuzzy Systems (IFSC), 27-29 Aug. 2013, IEEE, pp. 1-5, Qazvin, Iran.

Mehrabian, A., R:, Lucas, C. 2006. A novel numerical optimization algorithm inspired from weed colonization. Elsevier. 1(2006), 355–366.

Merugumalla, M., K., Navuri, P., K. 2018. PSO and Firefly Algorithms based control of BLDC motor drive. 2018 2nd International Conference on Inventive Systems and

Control (ICISC), 19-20 Jan. 2018, IEEE, 994-999, Coimbatore, India..

Mohanty, P., K., Parhi, D., R. 2014. A new efficient optimal path planner for mobile robot based on Invasive Weed Optimization algorithm. Springer. 9(4), 317–330.

Mujadin, A., Pratama, A., A. 2017. High Precession Resistive Touch Screen Driver Circuitry for Ball on Plate Balancing Systems. International Symposium on Electronics and Smart Devices (ISESD), 17-19 Oct. 2017, IEEE, 242-246, Yogyakarta, Indonesia.

Nasir, A., N., K., Ahmad, M., A., Ghazali, R., Pakheri, N., S. 2011. Performance Comparison Between Fuzzy Logic Conroller (FLC) and PID Controller for Highly Nonlinear Two-wheels Balancing Robot. First International Conference on Informatics and Computational Intelligence. 12-14 Dec. 2011, IEEE, 176-181, Bandung, Indonesia.

Nawikavatan, A., Puandownreong, D. 2019. Optimal PIDA Controller Design for Truck Breaking System using Flower Pollination Algorithm. The 2019 International Electrical Engineering Congress (iEECON2019), 6-8 March 2019, IEEE, 1-4, Hua Hin, Thailand.

Nawikavatan, A., Thammarat, C., Puandownreong, D. 2019. Application of Spiritual Search to Optimal PIDA Controller Design for Cardiac Pacemaker. The 4th International Conference on Digital Arts, Media and Technology and 2nd ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, 30 Jan.-2 Feb. 2019, IEEE, 286-289, Nan, Thailand.

O'Dwyer, A. 2005. PID control: the early years. Presentation – Control in the IT Sector Seminar, May 3, 2005.

Okkan, U., Köse, O., Özsoy, M., Uysal, H. 2017. Yabani Ot ve Diferansiyel Evrim Algoritmalarının Aylık Kavramsal Bir Yağış-Akış Modeli Kalibrasyonu Üzerinden Performanslarının İrdelenmesi. DSİ Teknik Bülteni, 127.

Ozturk, A., Cobanlı, S., Duman, S., Tosun, S., Dosoglu, K. Yapay Arı Kolonisi Algoritması İle Elektrik Güç Sistemi Optimal Yakıt Maliyetinin Belirlenmesi. *6th* International Advanced Technologies Symposium (IATS'11), 16-18 May 2011, Vol. 16, 18, Elazığ, Turkey.

Özsağlam, M., Y., Çunkaş, M. 2008. Optimizasyon Problemlerşnşn Çözümü için Parçacık Sürü Optimizasyonu Algoritması. Politeknik Dergisi. 11(4), 299-305.

Patel, N., C., Debnath, M., K., Sahu, B., K., Dash, S., S., Bayindir, R. 2018. Multi-Staged PID Controller Tuned by Invasive Weed Optimization Algorithm for LFC Issues. 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA), 14-17 Oct. 2018, IEEE, 1358-1362, Paris, France.

Pham, D., T., Darwish, A., H., Eldukhri, E., E., Otri, S. 2007. Using the Bees Algorithm to tune a fuzzy logic controller for a robot gymnast. I*PROMS 2007 Innovative

Production Machines and Systems, online, 1-14 July 2007.

Pham, D., T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M. 2005. Bee Algorithm A Novel Approach to Function Optimisation. Technical Note: MEC 0501 Dec 2005.

Puangdownreong, D. 2012. Application of Current Search to Optimum PIDA Controller Design. Intelligent Control and Automation. 3(4), 303-312.

Romsai W., Nawikavatan, A. 2019. Optimal PIDA Controller Design for Quarter Car Suspension System by Intensified Current Search. 2019 Seventeenth International Conference on ICT and Knowledge Engineering, 20-22 Nov. 2019, IEEE, 1-5, Bangkok, Thailand.

Sambariya, D., K., Paliwal, D. 2016. Optimal Design of PIDA Controller Using Firefly Algorithm for AVR Power System. International Conference on Computing, Communication and Automation (ICCCA2016), 29-30 April 2016, IEEE, 987-992, Greater Noida, India.

Sambariya, D., K., Paliwal, D. 2016. Comparative Design and Analysis of PIDA Controller Using Kitti's and Jung-dorf Approach for Third Order Practical Systems. Journal of Advances in Mathematics and Computer Science, pp.1-16, Greater Noida, India.

Shaikh, H., M., Kulkarni, N., R. 2019. Perception on PSO, Fuzzy and Fuzzy-PID for Water Level Control of Coupled Tank System. 2019 IEEE Pune Section International Conference (PuneCon), 18-20 Dec. 2019, IEEE, 1-5, Pune, India.

Silva, F., O., Leite, W., D., C., Brito, A., G., Silva, A., G., D. 2019. Tuning Techniques Evaluation for Satellite Launch Vehicle Attitude Controllers. Journal of Aerospace Technology and Management, 11, 1-14.

Sreeraj, P., V. 2013. Design and implementation of PID controller with lead compensator for thermal process. International Journal of Computer Applications, 67(1), 26-31.

Tapkan, P., Özbakır, L., Baykasoğlu, A. 2008. Arı Algoritması ve Genelleştirilmiş Atama Problemi: Farklı Komşuluk Yapılarının Karşılaştırılması. Endüstri Mühendisliği Dergisi. 21(2), 2-13.

Tien, J. P., Li, T. H. S. 2012. Hybrid clonal selection algorithm and the artificial bee colony algorithm for a variable PID-like fuzzy controller design. In 2012 International conference on Fuzzy Theory and Its Applications (iFUZZY2012), 16-18 Nov. 2012, IEEE, 87-94, Taichung, Taiwan.

Yang, X. S. 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press.

Yang, X. S. 2009. Firefly Algorithms for Multimodal Optimization. Springer. 5792, 169-178.

Yaovaja, K. 2018. Ball Balancing on a Stewart Platform using Fuzzy Supervisory PID Visual Servo Control. 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA). 14-17 Aug. 2018, IEEE; 170-175, Krabi, Thailand.

Yılmaz, S. 2014. Yarasa Algoritmasının Unimodal, Multimodal ve Kaydırılmış Sayısal Optimizasyon Problemleri (CEC05) Üzerinde Geliştirilmesi. Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, 125, Isparta.

Yüce, B., Pham, D., T., Packianather, M., S., Mastocinque, E., Lambiase, A. 2013. Honey Bees Inspired Optimization Method: The Bees Algorithm. Insects. 4, 646-662.

Zeeshan, N., Nauman, M., Khan, J. 2012. Design, Control and Implementation of a Ball on Plate Balancing System. Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST), 03 April 2012, IEEE, 22-26, Islamabad, Pakistan.

Zeybek, S. 2014. A Study of Vantage Point Neighbourgood Search In The Bees Algorithm For Combinatorial Optimization Problems. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Mühendisliği Anabilim Dalı, 98, İstanbul.

Zhou, T. 2013. Temperature Control Method for Water-Coal-Mixture Gasifier System Based on Fuzzy Control Rules Optimized by PSO Algorithm. 2013 2nd International Conference on Measurement, Information and Control, 16-18 Aug. 2013, IEEE, Vol. 2, 793-796, Harbin, China.