
Araştırma Makalesi / Research Article

DC Motorun Hız Kontrolü İçin Meta-Sezgisel Algoritma Tabanlı PID Denetleyici Tasarımı

Ali Kıvanç ŞAHİN¹, Ömür AKYAZI^{2*}, Erdiñ ŞAHİN², Oğuzhan ÇAKIR³

¹Erzurum Teknik Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Elektrik Elektronik Mühendisliği, Erzurum

²Karadeniz Teknik Üniversitesi, Of Teknoloji Fakültesi, Enerji Sistemleri Mühendisliği Bölümü, Trabzon

³Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Trabzon

(ORCID: 0000-0003-4539-6419)(ORCID: 0000-0001-6266-2323)

(ORCID: 0000-0002-9740-599X)(ORCID: 0000-0002-6576-8710)

Öz

Bu makalede DC motor hız kontrolü için balina optimizasyonu (WOA) ve güve-alevi optimizasyonu (MFO) algoritmaları kullanılarak en uygun PID denetleyici parametreleri (k_p , k_i ve k_d) belirlenmiştir. PID denetleyici parametre değerleri belirlenirken her iki algoritma için de hata tabanlı bir amaç fonksiyonu olan hatanın mutlak değerinin zaman ağırlıklı integrali (ITAE) seçilmiştir. İki farklı yaklaşımla tasarlanan PID denetleyicilerin yakınsama profili analizi, geçici tepki analizi, frekans uzayı analizi (bode analizi), hata tabanlı performans ölçütlerinin analizi ve bozucu yük cevabı analizi MATLAB programı aracılığıyla gerçekleştirilmiş ve performansları karşılaştırılmıştır. Bulunan sonuçlara göre WOA tabanlı PID denetleyicinin MFO tabanlı PID denetleyiciye göre daha iyi sonuç verdiği gözlemlenmiştir. Ayrıca balina optimizasyonu algoritmasıyla ayarlanan PID denetleyicinin atom arama optimizasyonu (ASO), gri kurt optimizasyonu (GWO), yabani ot optimizasyonu (IWO) ve stokastik fraktal arama (SFS) algoritmaları kullanılarak ayarlanmış PID denetleyicilerine göre geçici tepki karakteristiğini iyileştirdiği, dayanıklılığını arttırdığı ve sistemde yük momentinden kaynaklanan bir değişiklik meydana geldiğinde bu bozulmayı daha kısa sürede toparlamayı başardığı görülmüştür.

Anahtar kelimeler: Balina optimizasyonu algoritması, DC motor hız kontrolü, Güve-alevi optimizasyonu algoritması, PID denetleyici

Meta-Heuristic Algorithm Based PID Controller Design for Speed Control of DC Motor

Abstract

In this article, the optimal parameters of PID controller (k_p , k_i and k_d) are determined by using whale optimization (WOA) and moth-flame optimization (MFO) algorithms for speed control of DC motor. While determining the PID controller parameter values, the time-weighted integral of the absolute value of the error (ITAE), which is an error-based objective function, are chosen for both algorithms. Convergence profile analysis, transient response analysis, frequency-domain analysis, error-based performance metrics analysis and load disturbance response analysis of PID controllers designed with two different approaches are carried out by MATLAB program and performance are compared. According to the obtained results, it has been observed that the WOA based PID controller gives better results than the MFO based PID controller does. In addition, the PID controller tuned with the whale optimization algorithm improves the transient response characteristic compared to PID controllers tuned using atom search optimization (ASO), grey wolf optimization (GWO), invasive weed optimization (IWO) and stochastic fractal search (SFS) algorithms, improves its robustness. It has been observed that when a change occurs due to the load moment, it manages to recover this deterioration in a shorter time.

Keywords: Whale optimization algorithm, DC motor speed control, Moth-flame optimization algorithm, PID controller

*Sorumlu yazar: oakyazi@ktu.edu.tr

Geliş Tarihi: 15.01.2021, Kabul Tarihi: 06.03.2021

1. Giriş

DC motorlar yüksek kalkış momenti, yüksek hızlı tepkisi, taşınabilir olması, birçok kontrol ayarlama yöntemine uyum sağlaması ve ucuz olmaları sebebiyle hız kontrolüne yönelik endüstriyel uygulamalarda sıklıkla tercih edilmektedir. Araştırmacılar DC motor hız kontrolünde daha iyi bir sistem cevabı elde etmek için birçok kontrol yöntemi önermiştir. Oransal-integral-türevsel (PID) denetleyici [1-4], kesir dereceli oransal-integral-türevsel (FOPID) denetleyici [3-8], bulanık (Fuzzy) denetleyici [9], bulanık-oransal-integral-türevsel (Fuzzy-PID) denetleyici [10], adaptif ağ tabanlı bulanık çıkarım sistemi (ANFIS) [11], kayan kipli denetleyici (SMC) [12] önerilen kontrol yöntemlerinden bazılarıdır.

Günümüzde araştırmacılar tarafından yeni denetleyiciler önerilmiş olursa da PID denetleyicisi endüstriyel uygulamalarda hala yaygın olarak kullanılmaktadır. Yaygın olarak kullanılmasının ana sebebi basit ve kolay uygulanabilir bir yapıya sahip olmasıyla birlikte geniş bir çalışma aralığında sağlam bir performans sunmasıdır. PID denetleyicisi oransal kazanç (k_p), integral kazanç (k_i) ve türevsel kazanç (k_d) olarak adlandırılan üç kontrol parametresi değişkenine sahiptir. PID kazanç değerlerini el ile ayarlamak oldukça zaman alan bir işlem olduğundan bu kazanç değerlerini ayarlamak için Ziegler-Nichols, Cohen-Coon ve kazanç-faz marjı gibi klasik ayarlama yöntemleri literatürde sıklıkla tercih edilmektedir. Ancak bu yöntemler en uygun sonucu elde etmede başarılı olamadıklarından [13,14] araştırmacılar son zamanlarda evrimsel algoritmalar kullanarak PID denetleyicisinin parametre değerlerini ayarlamıştır. Atom arama optimizasyonu (ASO) [3], kaotik atom arama optimizasyonu (ChASO) [3], böbrek ilhamlı (KA) [15], diferansiyel evrim (DE) [16], genetik (GA) [17], parçacık sürüsü optimizasyonu (PSO) [5], kısıtlı parçacık sürüsü optimizasyonu (CPSO) [7], sinüs kosinüs (SCA) [18], geliştirilmiş sinüs-kosinüs (ISCA) [19], gri kurt optimizasyonu (GWO) [4, 20], harris şahinleri optimizasyonu (HHO) [8, 21], henry gaz çözünürlüğü optimizasyonu (HGSO) [22], karşıt-tabanlı henry gaz çözünürlüğü optimizasyonu (OBL-HGSO) [22], jaya optimizasyonu (JOA) [23], salp sürüsü (SSA) [24], stokastik fraktal arama (SFS) [1, 25], yapay arı kolonisi (ABC) [26], yerçekimsel arama (GSA) [27] ve yabani ot optimizasyonu (IWO) [2] algoritmaları DC motor hız kontrolünde PID denetleyici parametrelerini en uygun değere ayarlamak için kullanılmıştır.

Yukarıda belirtilen literatürdeki çalışmalarda, araştırmacılar DC motor hız tepkisini, aşım yüzdesini, oturma zamanını, yükselme zamanını, tepe değeri ve kararlı durum hatası gibi geçici karakteristik değerlerini iyileştirebilmek için farklı yaklaşımlı PID denetleyiciler önermiştir. Ayrıca hiçbir algoritma DC motorun hız kontrolü için tasarlanan denetleyicinin en iyi parametre değerlerini bulacağını garanti edemeyeceğinden araştırmacılar farklı yaklaşımlı algoritmalar tercih etmiştir. Ek olarak araştırmacılar DC motorun hız tepkisini iyileştirebilmek için farklı türdeki amaç fonksiyonları kullanıp PID denetleyicinin en uygun parametre değerlerini bulmayı hedeflemiştir. DC motor hızını kontrol etmek amacıyla tasarlanan PID denetleyicisi için hatanın mutlak değerinin zaman ağırlıklı integrali (ITAE) [1, 3, 4] ve hatanın karesinin zaman ağırlıklı integrali (ITSE) [24] literatürde sıkça kullanılan amaç fonksiyonlarıdır.

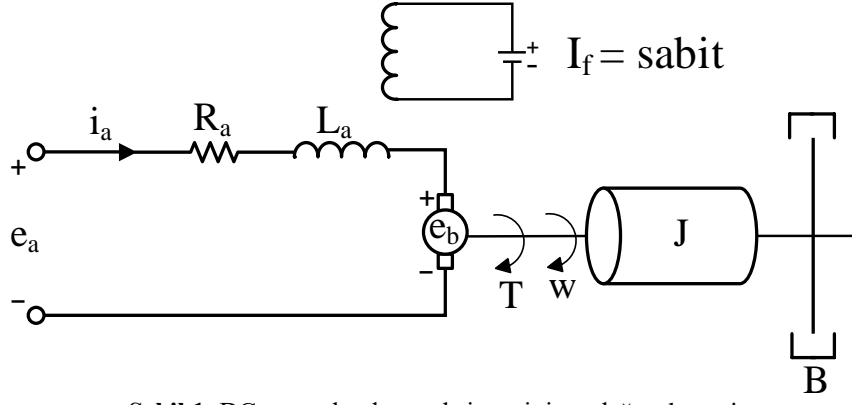
Bu çalışmada ise evrimsel algoritma olarak balina optimizasyonu ve güve-alevi optimizasyonu kullanılarak PID denetleyicisinin parametre değerleri belirlenmiş olup yakınsama profili analizi, zaman alanı analizi (birim basamak cevabı), hata tabanlı performans ölçütlerinin analizi, frekans uzayı analizi (bode analizi) ve son olarak da bozucu yük analizi yapılmıştır. Ayrıca önerilen WOA-PID ve MFO-PID denetleyicilerin literatürdeki diğer yaklaşımlı PID denetleyicilere göre daha üstün bir performans gösterdikleri gözlemlenmiştir.

Makalenin diğer bölümleri ise şu şekilde düzenlenmiştir. Bölüm 2’de DC motor sisteminin modellenmesinden, DC motorun hız kontrolü için PID denetleyicisi tasarımından ve kullanılan optimizasyon algoritmalarından bahsedilmiştir. Bölüm 3’te ise simülasyon sonuçları ve karşılaştırmalar sunulmuştur. Son bölümde ise sonuç ve öneriler verilmiştir.

2. Materyal ve Metot

2.1. DC Motor Sisteminin Modellenmesi

Bu makalede, harici uyarımlı bir DC motorun armatür gerilimi üzerinden hız kontrolü yapılmış olup Şekil 1’de ise eşdeğer devresi verilmiştir [28].



Şekil 1. DC motor hız kontrol sisteminin eşdeğer devresi

Şekil 1’de verilen DC motorun eşdeğer devresinden de görüleceği üzere e_a uygulanan armatür gerilimini (V), i_a armatür akımını (A), R_a armatür direncini (Ω), L_a armatür endüktansını (H), i_f alan akımını (A), e_b ters elektromotor kuvvetini (V), T motor momentini (N.m), w motor mili açısal hızını (rad/sn), J motorun eylemsizlik momentini (kg.m^2), K motor momenti sabitini (N.m/A), K_b elektromotor kuvveti sabitini (V.s/rad), B motor sürtünme sabitini (N.m.s/rad) ve T_L yük momentini (N.m) temsil etmektedir.

$\frac{d\theta}{dt} = w(t)$ olmak üzere sabit bir akıda indüklenen gerilim ile açısal hız doğru orantılıdır.

$$e_b = K_b \frac{d\theta}{dt} = K_b w(t) \quad (1)$$

DC motor hız kontrol sisteminin armatür devresi için diferansiyel denklemi eşitlik (2)’de verilmiştir.

$$e_a(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + e_b(t) \quad (2)$$

Yük momenti sıfır olarak alındığında atalet ve sürtünme momentlerinin toplamına karşılık gelen moment (T) armatür akımıyla doğru orantılıdır.

$$T(t) = J \frac{dw(t)}{dt} + Bw(t) = K i_a(t) \quad (3)$$

Sırasıyla eşitlik (1), (2) ve (3)’teki tüm başlangıç şartları sıfır olarak alınıp Laplace dönüşümü uygulandığında yük momentini ($T_L(s)$) sıfır için denetimsiz DC motorun transfer fonksiyonu eşitlik (7)’deki gibi elde edilir.

$$E_b(s) = K_b w(s) \quad (4)$$

$$E_a(s) = (L_a s + R_a) I_a(s) + E_b(s) \quad (5)$$

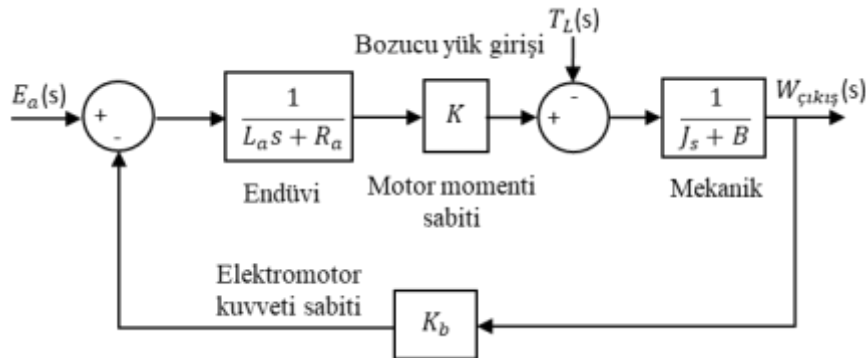
$$T(s) = (Js + B)w(s) = K I_a(s) \quad (6)$$

$$\frac{w(s)}{E_a(s)} = \frac{K}{(L_a s + R_a)(Js + B) + K_b K} \quad (7)$$

Uygulanan armatür gerilimi sıfır olarak alındığında motor hızı ile yük momenti arasındaki ilişki eşitlik (8)'deki gibidir.

$$G(s) = \frac{w(s)}{T_L(s)} = -\frac{(L_a s + R_a)}{(L_a s + R_a)(Js + B) + K_b K} \quad (8)$$

Denetimsiz DC motor hız kontrolünün açık çevrim sistemi Şekil 2'de verilmiştir. Şekil 2'den de görüleceği üzere DC motorun bileşenleri endüvi kısmı, motor momenti sabiti, mekanik kısmı ve elektromotor kuvveti sabitinden oluşmaktadır. PID denetleyicisi ile tasarlanmış olan DC motor hız kontrolünün blok diyagramı Şekil 3'te gösterilmiştir.



Şekil 2. Denetimsiz DC motorun blok diyagramı

Tablo 1. DC motor parametre değerleri

Parametre	Değer
R_a (Armatür direnci)	0.4 Ω
L_a (Armatür endüktansı)	2.7 H
J (Motorun eylemsizlik momenti)	0.0004 kg.m ²
B (Motor sürtünme sabiti)	0.0022 N.m.s/rad
K (Motor momenti sabiti)	0.015 N.m/A
K_b (Elektromotor kuvveti sabiti)	0.05 V.s

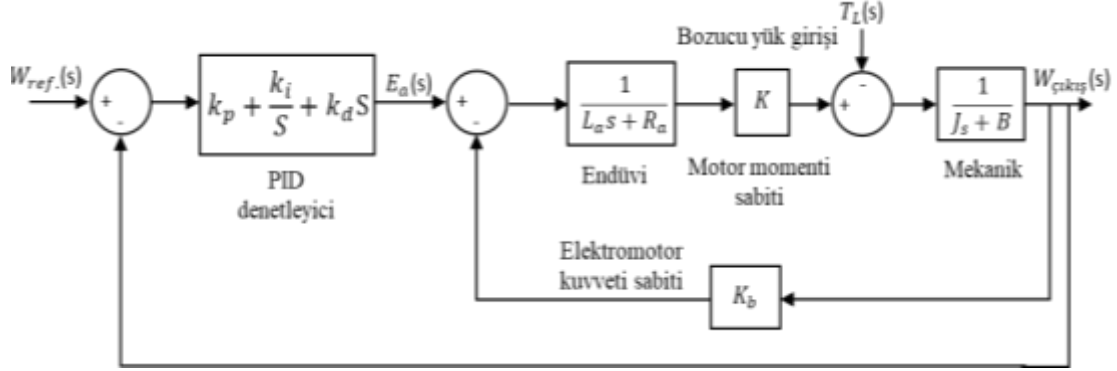
Bu çalışmada kullanılacak olan DC motorun parametre değerleri Tablo 1'de gösterilmiştir [1-4]. Tablo 1'de verilen DC motor parametre değerleri eşitlik (7) ve (8)'e yazıldığında DC motorun açık çevrim transfer fonksiyonları eşitlik (9)'daki gibi elde edilir.

$$G_{açık-çevrim}(s) = \begin{cases} \frac{w(s)}{E_a(s)} = \frac{0.015}{0.00108s^2 + 0.0061s + 0.00163} \\ \frac{w(s)}{T_L(s)} = -\frac{2.7s + 0.4}{0.00108s^2 + 0.0061s + 0.00163} \end{cases} \quad (9)$$

2.2. DC Motorun Hız Kontrolü için PID Denetleyicisi Tasarımı

DC motorun hız kontrolünün performansını iyileştirmek için PID denetleyicisi kullanılmıştır. Tasarlanan PID denetleyicisinin transfer fonksiyonu eşitlik (10)'da verilmiştir.

$$G_{PID}(s) = k_p + \frac{k_i}{s} + k_d s \quad (10)$$



Şekil 3. PID denetleyicisi ile tasarlanan DC motorun blok diyagramı

Şekil 3'te verilen PID denetleyicisi ile tasarlanan DC motorun kapalı çevrim transfer fonksiyonu eşitlik (11)'deki gibidir.

$$G_{kapalı\text{-}çevrim}(s) = \begin{cases} \frac{w_{çikis}(s)}{w_{referans}(s)} = \frac{0.015(k_d s^2 + k_p s + k_i)}{0.00108s^3 + 0.0061s^2 + 0.00163s + 0.015(k_d s^2 + k_p s + k_i)}, T_L = 0 \text{ için} \\ \frac{w_{çikis}(s)}{T_L(s)} = -\frac{(2.7s + 0.4)s}{0.00108s^3 + 0.0061s^2 + 0.00163s + 0.015(k_d s^2 + k_p s + k_i)}, E_a = 0 \text{ için} \end{cases} \quad (11)$$

2.3. Balina Optimizasyonu Algoritması

2016 yılında Mirjalili ve Lewis tarafından önerilen bu algoritma kambur balinaların kendine özgü avlanma yönteminden esinlenerek oluşturulmuştur [29]. Kambur balinalar oluşturduğu geniş ve birbirleriyle bağlantılı kabarcık-ağlar ile avlarını bir araya toplar. Kambur balinalar kabarcık-ağları içerisinde su yüzeyine yükseldikçe yine soluk alıp-vererek avına yaklaşır. Avına yaklaştıkça kabarcık-ağ çemberini gitgide daraltıp hedefini küçültür. Bu yöntemi avını bulmak, avını hareketsiz bırakmak veya kendini avdan gizlemek için kullandığı bilinmektedir. Balina optimizasyon algoritmasında bu avlanma yöntemine göre matematiksel denklemler oluşturulmuştur. Eşitlik (12)-(15)'te ilgili denklemler verilmiştir.

$$D = |CX^*(t) - X(t)| \quad (12)$$

$$X(t+1) = X^*(t) - AD \quad (13)$$

Eşitlik (12) ve (13)'teki denklemlere bakıldığında A ve C sabit vektör, X^* her iterasyon sonunda yinelenen en iyi çözümü ve t iterasyon sayısını belirtmektedir. A ve C sabit vektörlerinin hesaplanmaları sırasıyla eşitlik (14) ve (15)'te verilmiştir.

$$A = 2ar - a \quad (14)$$

$$C = 2r \quad (15)$$

Burada a iterasyon boyunca 2 'den 0 'a doğrusal olarak azalan bir vektördür ve r ise $[0,1]$ arasında rastgele bir vektördür. Balina optimizasyonu algoritmasının uygulanması aşağıdaki gibi açıklanmaktadır.

Adım 1: Popülasyon sayısını, alt sınırı, üst sınırı, iterasyon sayısını ve değişken sayısını tanımladıktan sonra algoritmadaki kambur balinaların pozisyonlarını rastgele olarak başlat.

Adım 2: Önceden tanımlanan amaç fonksiyonu kullanılarak her bir kambur balinası için amaç fonksiyonunu hesaplat.

Adım 3: En iyi kambur balınaya ait amaç fonksiyonunu bul.

Adım 4: Kambur balinaların pozisyonlarını güncelle.

Adım 5: İterasyon sayısını arttır ve durdurma kriteri sağlanana kadar Adım 2'ye geri dönerek adımları tekrar et.

Adım 6: En iyi amaç fonksiyonuna karşılık gelen en iyi kambur balina pozisyonunu geri getir.

2.4. Güve-alevi Optimizasyonu (MFO)

2015 yılında Mirjalili tarafından önerilen güve-alevi optimizasyonu algoritması (MFO), doğadaki güvelerin enine yönlendirme olarak adlandırılan navigasyon yönteminden yararlanılarak oluşturulan bir algoritmadır [30]. Güve-alevi optimizasyonu algoritmasında bu navigasyon yöntemine göre oluşturulan matematiksel denklemler eşitlik (16)-(20)'de verilmiştir. Güvelerin sayısı, d boyutlu bir uzayda uçan arama ajanları olarak kabul edilir. Bu güveler rastgele bir pozisyonda başlatılır. Güvelerin her boyuttaki konumu M matrisinde saklanır. Eşitlik (16)'dan da görüleceği üzere n güvelerin sayısını d ise boyut sayısını yani değişkenlerin sayısını belirtmektedir.

$$M = \begin{pmatrix} m_{1,1} & \cdots & m_{1,d} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \cdots & m_{n,d} \end{pmatrix} \quad (16)$$

Her bir güvenin amaç fonksiyonu eşitlik (17)'de belirtilen OM dizisinde saklanmaktadır. Burada n güvelerin sayısıdır.

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \quad (17)$$

Eşitlik (18)'de belirtilen F matrisi, alevleri temsil eder ve şimdiye kadar her bir güvenin elde ettiği en iyi konumu saklar. Ayrıca güve daha iyi bir çözüm bulduğunda alevler güncellenir. Her güve, alevlerden yalnızca birini kullanarak konumunu güncellemek durumundadır. Eşitlik (16) ve (18)'e bakıldığında F matrisi ile M matrisinin boyut olarak eşit olduğu görülmektedir. Ayrıca eşitlik (18)'deki n güvelerin sayısı d ise değişkenlerin sayısıdır.

$$F = \begin{pmatrix} F_{1,1} & \cdots & F_{1,d} \\ \vdots & \ddots & \vdots \\ F_{n,1} & \cdots & F_{n,d} \end{pmatrix} \quad (18)$$

OF dizisi alev matrisinde saklanan çözümlerin amaç fonksiyonu değerlerini alır. Burada n güvelerin sayısıdır.

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \quad (19)$$

Eşitlik (20)'de verilen denklemde güvelerin pozisyonları logaritmik sarmal işlevi temelinde arama uzayında güncellenir.

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \quad (20)$$

Eşitlik (20)'de belirtilen D_i 'inci güve ile j 'inci alev arasındaki mesafe, t [-1,1] arasında seçilen rastgele bir sayıyı, b ise logaritmik sarmalın şeklini belirten bir sabit sayıyı temsil etmektedir. Güve-alevi optimizasyonunun uygulanması aşağıdaki gibi açıklanmaktadır.

Adım 1: Popülasyon sayısını, alt sınırı, üst sınırı, iterasyon sayısını ve değişken sayısını tanımladıktan sonra algoritmadaki güvelerin pozisyonlarını rastgele olarak başlat.

Adım 2: Önceden tanımlanan amaç fonksiyonu kullanılarak her bir güve için amaç fonksiyonunu hesaplat.

Adım 3: En iyi güve amaç fonksiyonunu bul.

Adım 4: Güvelerin pozisyonlarını güncelle.

Adım 5: İterasyon sayısını arttır ve durdurma kriteri sağlanana kadar Adım 2'ye geri dönerek adımları tekrar et.

Adım 6: En iyi amaç fonksiyonuna karşılık gelen en iyi güve pozisyonunu geri getir.

Bu çalışmada hem balina optimizasyonu hem de güve-alevi optimizasyonu algoritması için k_p , k_i ve k_d alt sınır değerleri 0,001 olarak belirlenirken üst sınır değerleri ise 20 olarak belirlenmiştir. Her iki algoritma içinde popülasyon sayısı 50 alınmış olup iterasyon sayısı ise 30 alınmıştır.

3. Bulgular ve Tartışma

Bu çalışmada amaç fonksiyonu olarak ITAE tercih edilmiştir. Çünkü ITAE amaç fonksiyonu diğer hata tabanlı amaç fonksiyonlarına göre daha az salınım ve aşım gibi avantajları vardır. ITAE amaç fonksiyonun denklemi ise eşitlik (21)'de verilmiştir.

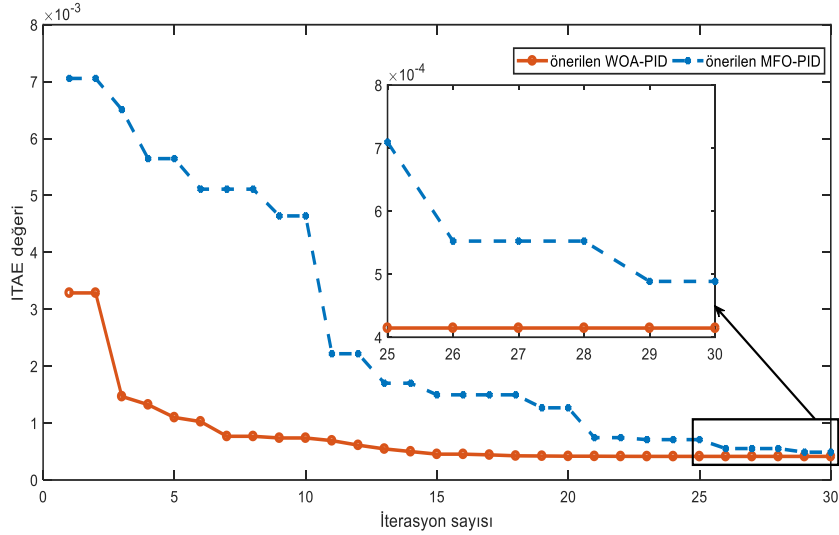
$$ITAE = \int_0^{t_{sim}} t |r(t) - y(t)| = \int_0^{t_{sim}} t |e(t)| \quad (21)$$

Eşitlik (21)'de görüleceği üzere t_{sim} simülasyon süresini, $r(t)$ referans açısal hızını, $y(t)$ gerçek açısal hızını, $e(t)$ ise referans açısal hızı ile gerçek açısal hızı arasındaki farkı temsil etmektedir. Bu çalışmada tüm simülasyonlar için geçen süre 2 saniye örnekleme zamanı ise 0,1 milisaniye olarak alınmıştır. Bu çalışmada yapılan analizler şunlardır:

- Yakınsama profili
- Zaman alanı analizi (Birim basamak cevabı)
- Hata tabanlı performans ölçütlerinin karşılaştırılması
- Frekans uzayı analizi (Bode analizi)
- Sağlık analizi (DC motor parametre değişikliklerine karşı)
- Bozucu yük cevabı analizi

3.1. Yakınsama Profili

DC motorun hız kontrolü için önerilen WOA tabanlı PID ve MFO tabanlı PID denetleyicilerin yakınsama eğrileri Şekil 4'te gösterilmiştir. WOA algoritmasının MFO algoritmasına göre daha düşük ITAE değerine sahip olduğu Şekil 4'te görülmektedir.



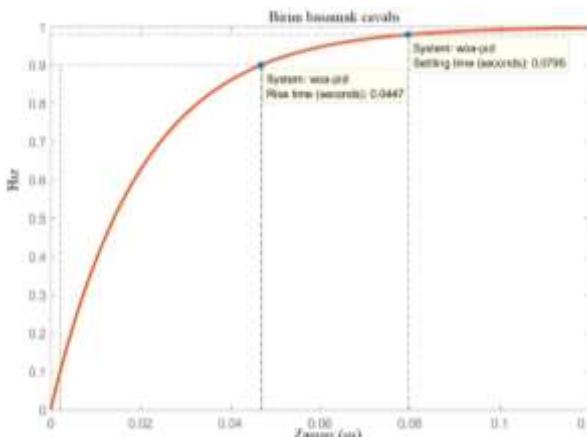
Şekil 4. WOA-PID ve MFO-PID için yakınsama eğrileri

ITAE amaç fonksiyonuna göre en uygun hale getirilen sistemin optimizasyon işlemi sonucunda elde edilen WOA-PID denetleyicisinin parametre değerleri $k_p = 20$, $k_i = 5.3442$, $k_d = 3.5419$ ve MFO-PID denetleyicisinin parametre değerleri $k_p = 19.5309$, $k_i = 5.2011$, $k_d = 3.4195$ olarak bulunmuştur. WOA tabanlı PID ile MFO tabanlı PID değerlerine göre elde edilen DC motorun kapalı çevrim transfer fonksiyonları eşitlik (22) ve (23)'te sırasıyla sunulmuştur.

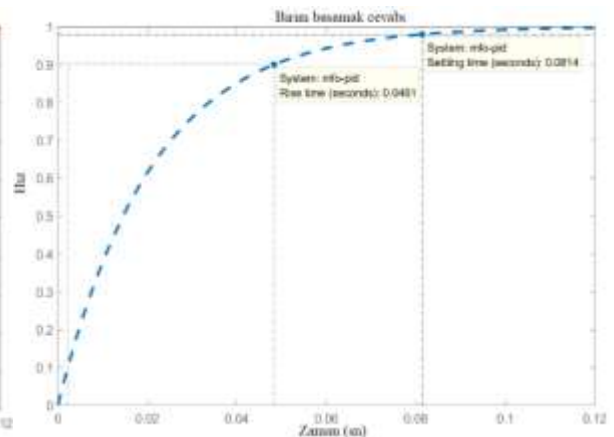
$$T_{WOA-PID} = \frac{0.05313s^2 + 0.3s + 0.08016}{0.00108s^3 + 0.05923s^2 + 0.3016s + 0.08016} \quad (22)$$

$$T_{MFO-PID} = \frac{0.05129s^2 + 0.293s + 0.07802}{0.00108s^3 + 0.05739s^2 + 0.2946s + 0.07802} \quad (23)$$

WOA-PID ve MFO-PID denetleyicileri için DC motor hızının birim basamak cevapları Şekil 5 ve Şekil 6'da sırasıyla gösterilmiştir. Şekil 5 ve Şekil 6'dan da görüleceği üzere WOA-PID denetleyicisinin yerleşme zamanı (t_s) ve yükselme zamanı (t_r) MFO-PID denetleyicisine göre daha üstün bir performans sergilemiştir.



Şekil 5. WOA tabanlı PID denetleyici



Şekil 6. MFO tabanlı PID denetleyici

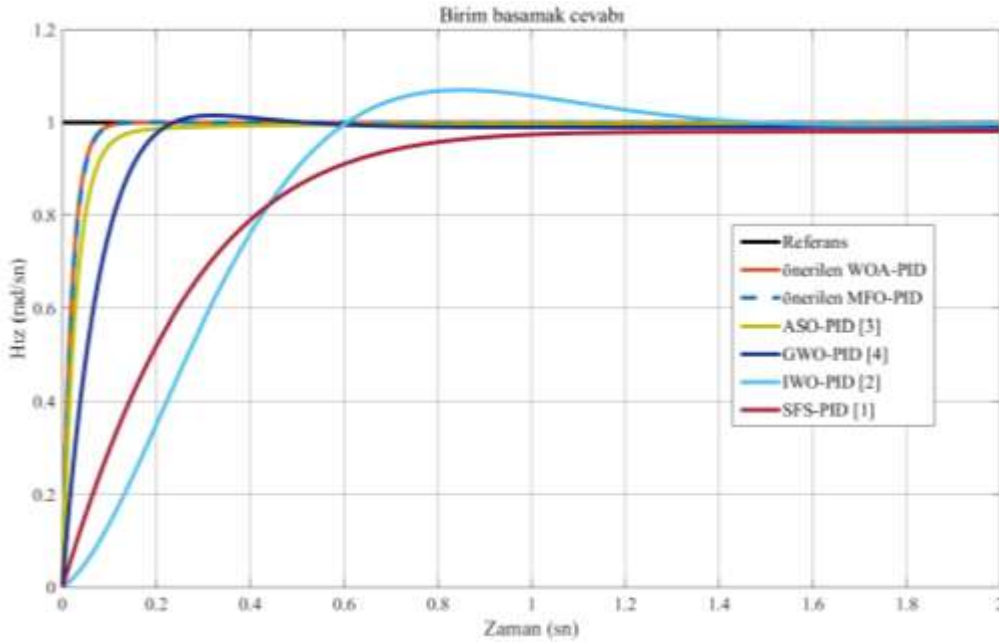
3.2. Zaman Alanı Analizi

Önerilen WOA tabanlı PID ve MFO tabanlı PID denetleyiciler ile birlikte literatürde aynı DC motor parametre değerlerini kullanan denetleyiciler ayrı ayrı kullanılmış olup birim basamak cevapları Şekil

7'de gösterilmiştir. Her bir denetleyici için elde edilen aşım yüzdesi (M_p), yerleşme zamanı (t_s) ve yükselme zamanı (t_r) değerleri Tablo 3'te verilmiştir.

Tablo 2. DC motor hız kontrolü için farklı evrimsel algoritmalar kullanılarak elde edilen PID parametre değerleri

Denetleyici	k_p	k_i	k_d
WOA-PID (önerilen)	20	5.3442	3.5419
MFO-PID (önerilen)	19.5309	5.2011	3.4195
ASO-PID [3]	11.9437	2.0521	2.4358
GWO-PID [4]	6.8984	0.5626	0.9293
IWO-PID [2]	1.5782	0.4372	0.0481
SFS-PID [1]	1.6315	0.2798	0.2395



Şekil 7. Farklı yaklaşımli PID denetleyicilerin birim basamak hız cevaplarının karşılaştırılması

Tablo 3. Farklı yaklaşımli PID denetleyicilerin geçici karakteristik değerlerinin karşılaştırılması

Denetleyici	M_p (%)	t_s (sn) ($\pm 2\%$)	t_r (sn) (0.10→0.90)
WOA-PID (önerilen)	0	0.0795	0.0447
MFO-PID (önerilen)	0	0.0814	0.0461
ASO-PID [3]	0	0.1535	0.0692
GWO-PID [4]	1.5062	0.2052	0.1388
IWO-PID [2]	6.9759	1.2533	0.4189
SFS-PID [1]	0	1.4475	0.5436

3.3. Hata Tabanlı Performans Ölçütlerinin Karşılaştırılması

Hatanın mutlak değerinin zaman ağırlıklı integrali (ITAE), hatanın karesinin zaman ağırlıklı integrali (ITSE), hatanın karesinin integrali (ISE) ve hatanın mutlak değerinin integrali (IAE) literatürde yaygın olarak kullanılan hata tabanlı performans ölçütleri olması sebebiyle karşılaştırma için seçilmiştir. ITAE, ITSE, ISE ve IAE denklemleri eşitlik (21), (24)-(26)'da sırasıyla ifade edilmiştir. WOA tabanlı PID denetleyicinin ITAE, ITSE, ISE ve IAE değerleri diğer yaklaşımli PID denetleyicilerine göre en düşük değerlere sahip olduğu Tablo 4'te görülmektedir.

$$ITSE = \int_0^{t_{sim}} t (r(t) - y(t))^2 = \int_0^{t_{sim}} t (e(t))^2 \quad (24)$$

$$ISE = \int_0^{t_{sim}} (r(t) - y(t))^2 = \int_0^{t_{sim}} (e(t))^2 \quad (25)$$

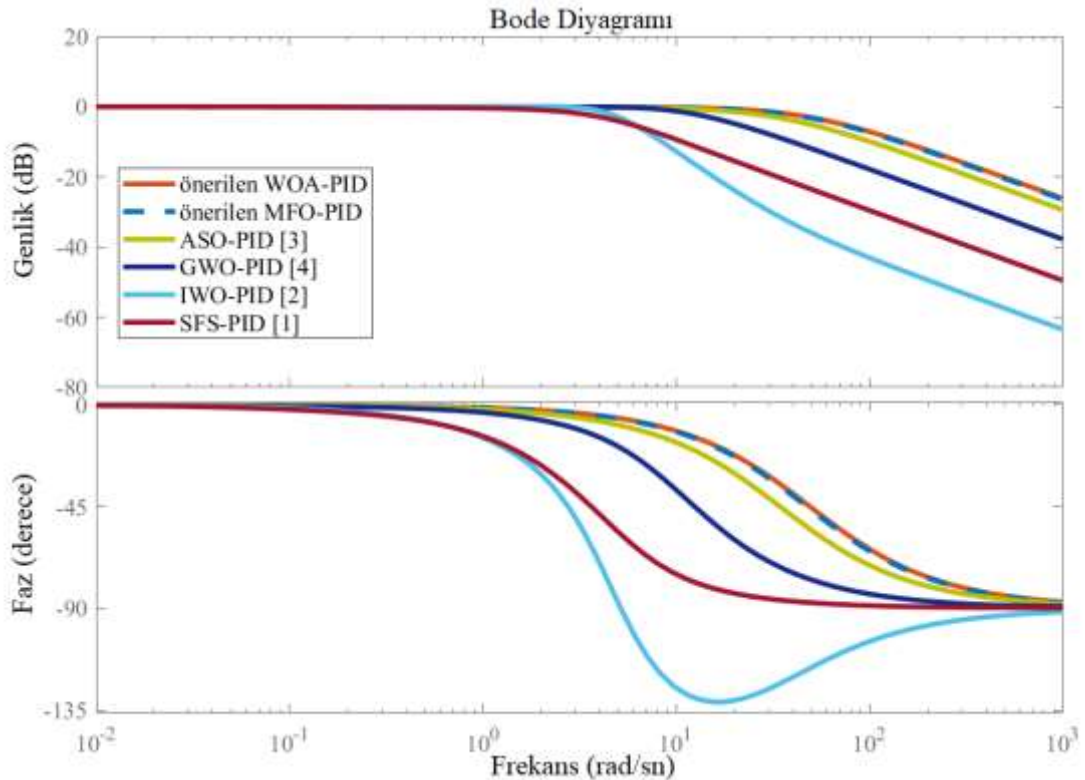
$$IAE = \int_0^{t_{sim}} |r(t) - y(t)| = \int_0^{t_{sim}} |e(t)| \quad (26)$$

Tablo 4. Farklı yaklaşımlı PID denetleyicilerin hata tabanlı performans ölçütlerinin karşılaştırılması

Denetleyici	ITAE değeri	ITSE değeri	ISE değeri	IAE değeri
WOA-PID (önerilen)	4.1448E-04	1.0332E-04	0.0102	0.0203
MFO-PID (önerilen)	4.8855E-04	1.1035E-04	0.0105	0.0211
ASO-PID [3]	0.0075	2.6210E-04	0.0151	0.0391
GWO-PID [4]	0.0223	0.0014	0.0360	0.0816
IWO-PID [2]	0.0853	0.0267	0.1927	0.3138
SFS-PID [1]	0.0900	0.0179	0.1358	0.2770

3.4. Bode Analizi

Farklı yaklaşımlar kullanılarak tasarlanan PID denetleyicilerin DC motor hızının kontrolündeki kararlı performanslarını gösterebilmek için frekans uzayı analizinden yararlanılmıştır. Frekans uzay analizi için Bode diyagramı kullanılmıştır. Şekil 8’de PID denetleyicilerin Bode diyagramları verilmektedir. Tablo 5’te ise Bode diyagramları kullanılarak elde edilen kazanç marjini, faz marjini (derece) ve bant değerleri (rad/sn) gösterilmektedir. WOA tabanlı PID denetleyicinin diğer yaklaşımlara göre tasarlanan PID denetleyicilerden kazanç marjini, faz marjini ve bant genişliği olarak en iyi değerlere sahip olduğu Tablo 5’te görülmektedir.



Şekil 8. Farklı yaklaşımlı PID denetleyicilerin Bode çizimleri

Tablo 5. Farklı yaklaşımli PID denetleyicilerin Bode analizi

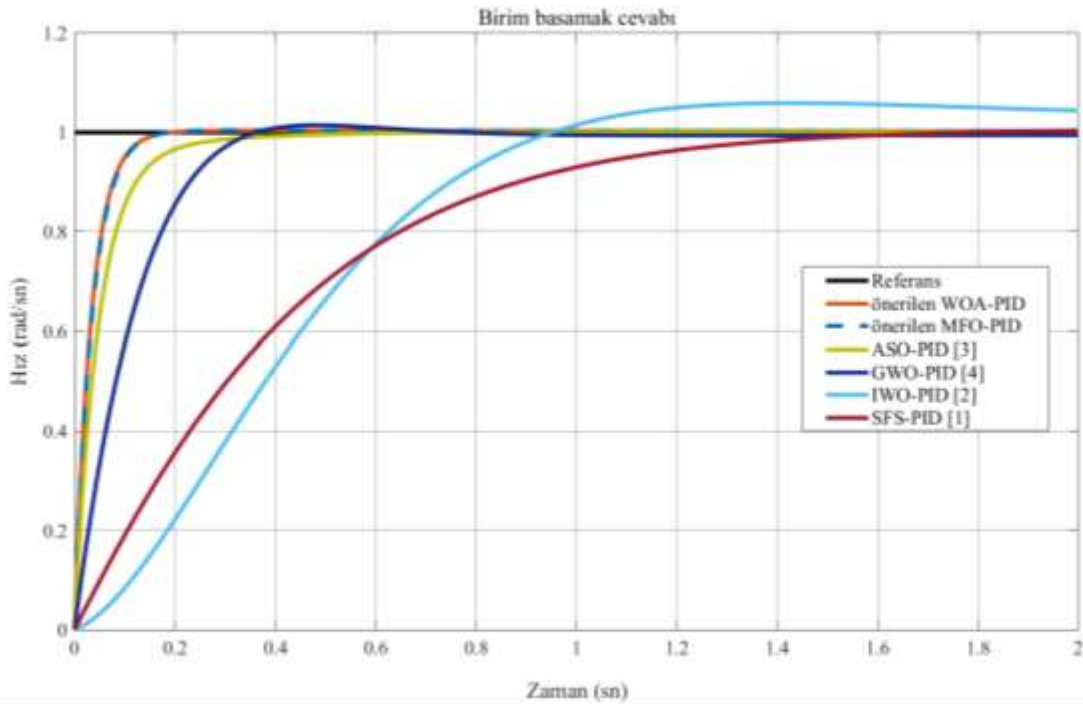
Denetleyici	Kazanç marjini (dB)	Faz marjini (açı)	Bant genişliği (rad/sn)
WOA-PID (önerilen)	Sonsuz	179.9913	49.0748
MFO-PID (önerilen)	Sonsuz	180	47.4504
ASO-PID [3]	Sonsuz	180	32.9113
GWO-PID [4]	Sonsuz	180	14.9018
IWO-PID [2]	Sonsuz	135.9149	5.0987
SFS-PID [1]	Sonsuz	180	4.1183

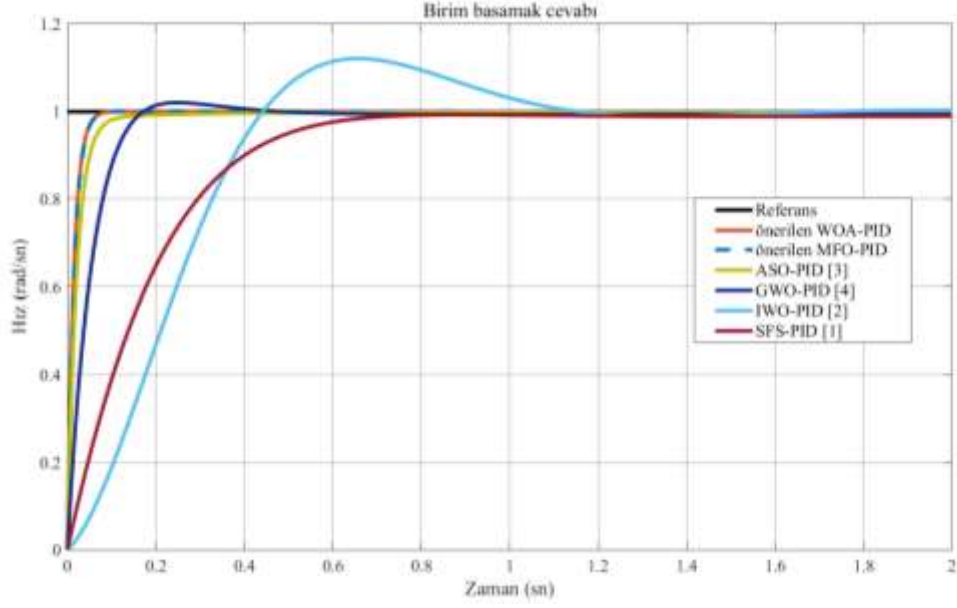
3.5. Dayanıklılık Analizi

Önerilen WOA tabanlı PID ve MFO tabanlı PID denetleyicilerle birlikte farklı yaklaşımli PID denetleyicilerin DC motordaki parametre değişikliğine karşı dayanıklılık analizi yapılmıştır. Dayanıklılık analizi için DC motordaki armatür direnci (R_a) ve motor momenti sabiti (K) \pm %50 ve \pm %40 olarak sırasıyla ayrı ayrı değiştirilmiş olup Tablo 6'da verilen dört olası çalışma noktası oluşturulmuştur. Oluşturulan tüm çalışma noktalarına göre DC motor hızının birim basamak cevaplarının karşılaştırmalı simülasyon sonuçları Şekil 9-12'de sırasıyla gösterilmiştir. Bununla birlikte geçici karakteristik değerleri Tablo 7-10'da sırasıyla verilmektedir. Tablo 7-10'dan görüleceği üzere WOA tabanlı PID denetleyicinin DC motordaki armatür direnci ve motor momenti sabitinin değişikliklerine rağmen diğer yaklaşımli PID denetleyicilere göre en düşük yükselme zamanı, yerleşme zamanı ve durum 1 hariç tüm durumlarda en düşük maksimum aşımaya sahip olduğu gözlemlenmiştir. Elde edilen sonuçlar önerilen WOA tabanlı PID denetleyicinin sağlamlığını ortaya koymaktadır.

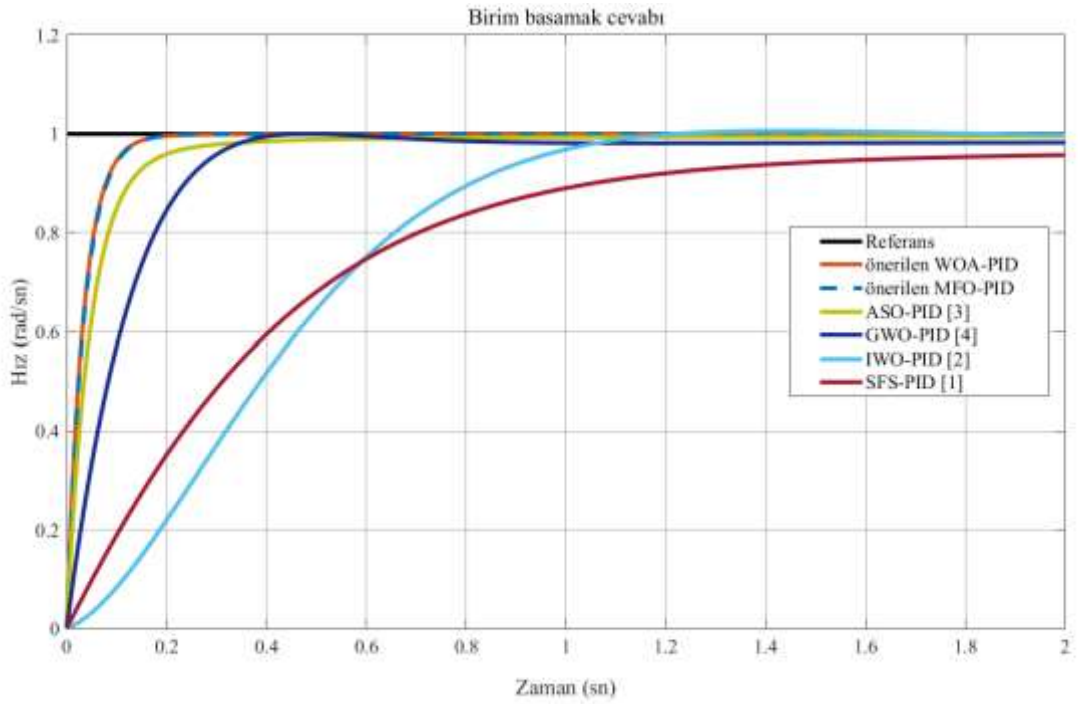
Tablo 6. DC motorun çalışma noktaları

Durum numaraları	R_a	K
(I)	0.20	0.009
(II)	0.20	0.021
(III)	0.60	0.009
(IV)	0.60	0.021

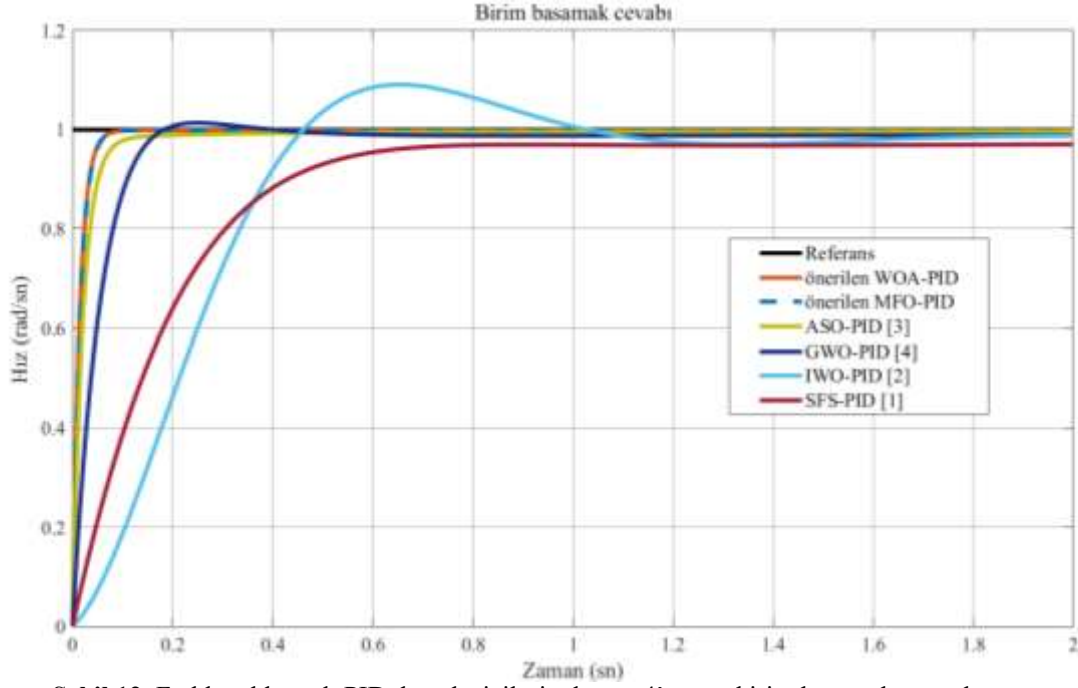
**Şekil 9.** Farklı yaklaşımli PID denetleyicilerin durum 1'e göre birim basamak cevapları



Şekil 10. Farklı yaklaşımlı PID denetleyicilerin durum 2'ye göre birim basamak cevapları



Şekil 11. Farklı yaklaşımlı PID denetleyicilerin durum 3'e göre birim basamak cevapları



Şekil 12. Farklı yaklaşımlı PID denetleyicilerin durum 4'e göre birim basamak cevapları

Tablo 7. Farklı yaklaşımlı PID denetleyicilerin durum 1'e göre geçici karakteristik değerleri

Denetleyici	M_p (%)	t_s (sn) ($\pm 2\%$)	t_r (sn) (0.10→0.90)
WOA-PID (önerilen)	0.2346	0.1283	0.0738
MFO-PID (önerilen)	0.3383	0.1307	0.0760
ASO-PID [3]	0.0133	0.2548	0.1176
GWO-PID [4]	1.4423	0.3154	0.2157
IWO-PID [2]	5.9002	4.1872	0.6356
SFS-PID [1]	0.6306	1.3557	0.8340

Tablo 8. Farklı yaklaşımlı PID denetleyicilerin durum 2'ye göre geçici karakteristik değerleri

Denetleyici	M_p (%)	t_s (sn) ($\pm 2\%$)	t_r (sn) (0.10→0.90)
WOA-PID (önerilen)	0	0.0562	0.0318
MFO-PID (önerilen)	0.0443	0.0577	0.0329
ASO-PID [3]	0	0.0982	0.0483
GWO-PID [4]	2.0515	0.2672	0.1018
IWO-PID [2]	12.1203	1.0453	0.3148
SFS-PID [1]	0	0.6247	0.3781

Tablo 9. Farklı yaklaşımlı PID denetleyicilerin durum 3'e göre geçici karakteristik değerleri

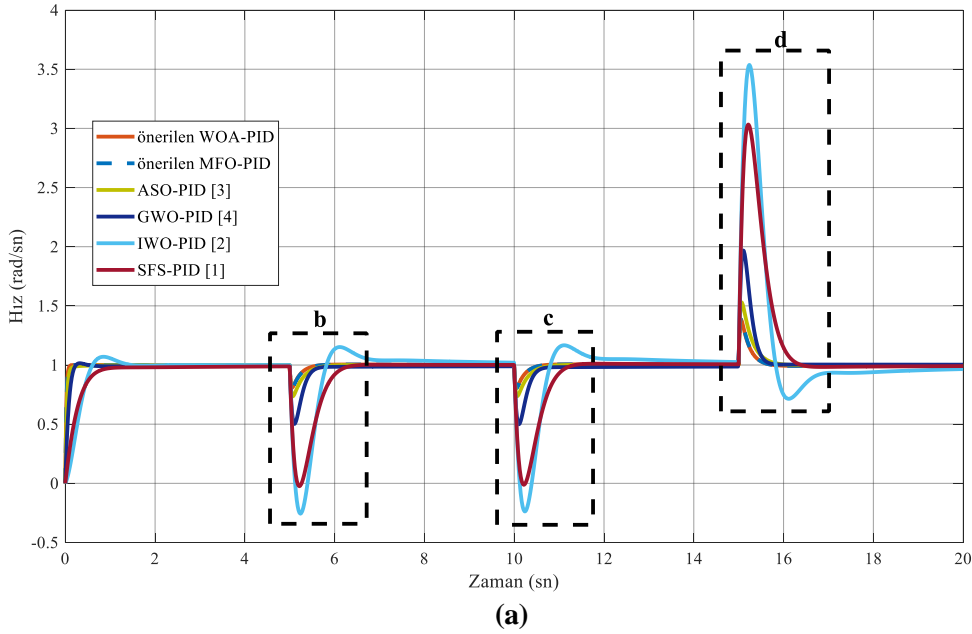
Denetleyici	M_p (%)	t_s (sn) ($\pm 2\%$)	t_r (sn) (0.10→0.90)
WOA-PID (önerilen)	0	0.1356	0.0750
MFO-PID (önerilen)	0	0.1381	0.0772
ASO-PID [3]	0	0.3177	0.1209
GWO-PID [4]	0	0.3436	0.2236
IWO-PID [2]	0.6250	1.0551	0.6968
SFS-PID [1]	0	6.1575	1.0030

Tablo 10. Farklı yaklaşımli PID denetleyicilerin durum 4'e göre geçici karakteristik değerleri

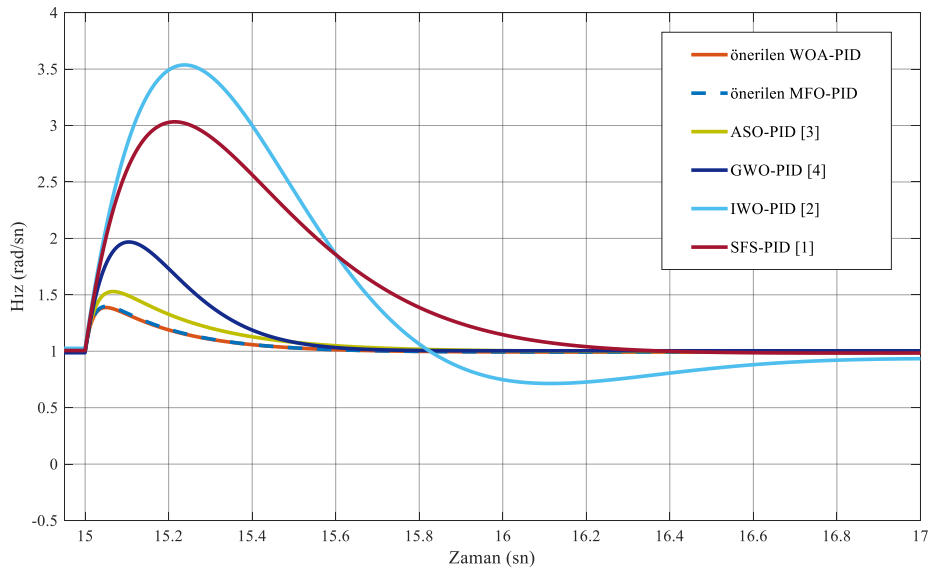
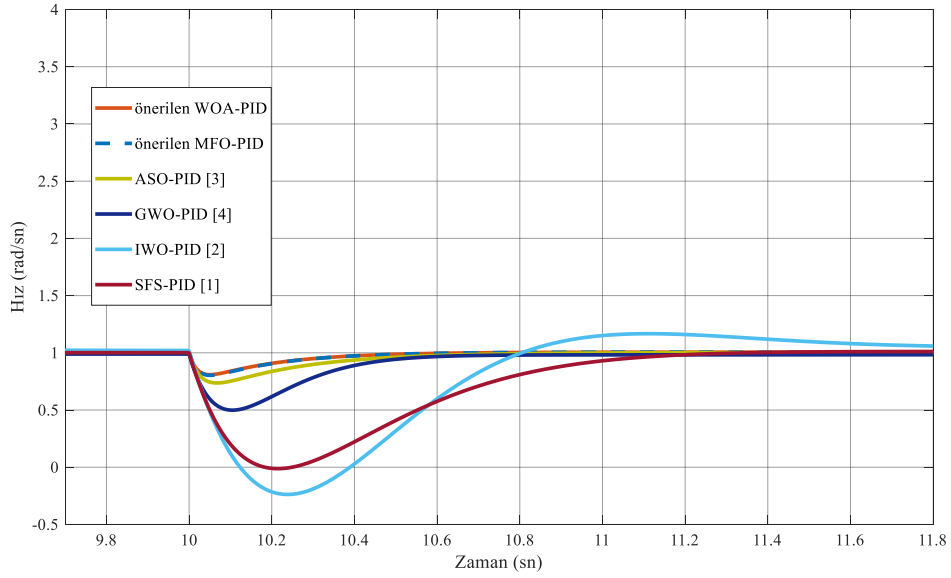
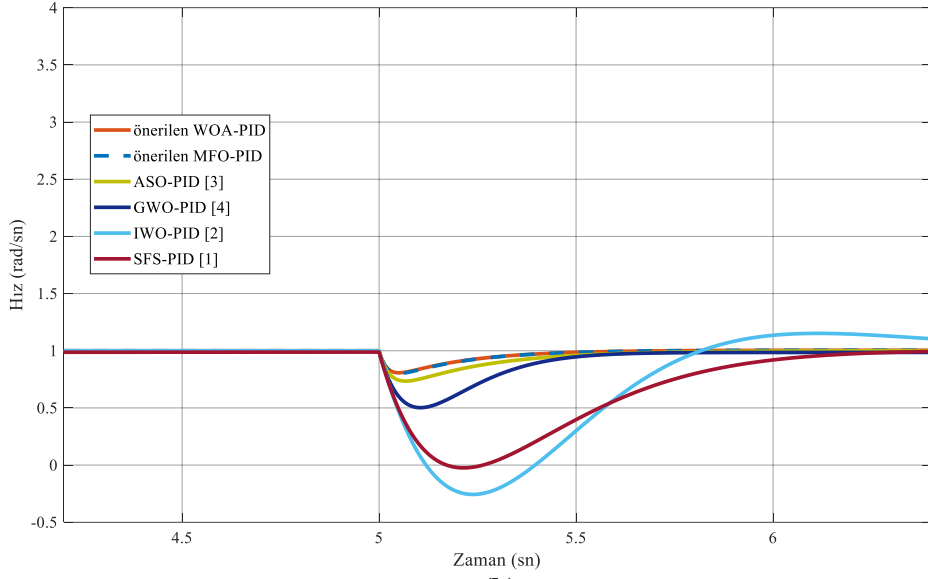
Denetleyici	M_p (%)	t_s (sn) ($\pm 2\%$)	t_r (sn) (0.10→0.90)
WOA-PID (önerilen)	0	0.0576	0.0320
MFO-PID (önerilen)	0	0.0591	0.0331
ASO-PID [3]	0	0.1058	0.0489
GWO-PID [4]	1.3669	0.1558	0.1036
IWO-PID [2]	9.0808	1.6260	0.3250
SFS-PID [1]	0	4.2741	0.4071

3.6. Bozucu yük cevabı analizi

Son olarak DC motor hız kontrol sisteminde bozucu yük momentini girişi verilmiş olup sistemin çıkış hızı tepkisindeki değişiklik durumu incelenmiştir. Şekil 13'te farklı yaklaşımli PID denetleyicilerin bozucu yük momentini girişi altında birim basamak cevapları verilmiştir. Yük momentini (T_L) $t \in [5 \text{ sn}-10 \text{ sn}]$ zaman aralığında 0,005 N.m, $t \in [10 \text{ sn}-15 \text{ sn}]$ zaman aralığında 0,01 N.m uygulanmış olup $t \in [0 \text{ sn}-5 \text{ sn}]$ ve $t \in [15 \text{ sn}-20 \text{ sn}]$ aralığında ise yük momentini (T_L) uygulanmamıştır. Farklı yaklaşımli PID denetleyiciler arasında daha net bir karşılaştırma yapabilmek için Şekil 13'te yakınlaştırılmış görüntüleri verilmiştir. Şekil 13(b), Şekil 13(c) ve Şekil 13(d)'te görüleceği üzere önerilen WOA tabanlı PID denetleyicinin diğer yaklaşımli PID denetleyicilere göre sistemdeki yük bozukluğunu bastırmada daha başarılı olduğu görülmüştür.



Şekil 13. Farklı yaklaşımli PID denetleyicilerin bozucu yük momentini girişi altında birim basamak cevapları; (b-c-d) (a)'da gösterilen siyah dikdörtgen alanların yakınlaştırılmış görüntüleri



Şekil 13. (devamı)

4. Sonuç ve Öneriler

Bu çalışmada DC motorun hız kontrolünü iyileştirebilmek için WOA tabanlı PID ve MFO tabanlı PID denetleyiciler tasarlanmıştır. Önerilen WOA-PID ve MFO-PID denetleyicilerin etkinliklerini gösterebilmek için literatürde diğer yaklaşımlarla önerilen PID denetleyicilerle karşılaştırmaları yapılmıştır. Ayrıca her bir denetleyicinin birim basamak cevabı, bode analizi, geçici tepki analizi ve bozucu yük cevabı analizi MATLAB ortamında gerçekleştirilip performans analizleri karşılaştırılmıştır. Önerilen WOA tabanlı PID denetleyicinin önerilen MFO tabanlı PID ve literatürde önerilen diğer yaklaşımlı PID denetleyicilere göre DC motorun hız kontrolünün denetiminde daha başarılı olduğu gözlemlenmiştir. Ayrıca WOA tabanlı PID denetleyicinin DC motordaki armatür direnci ve motor momenti parametrelerindeki yüksek oran değişikliklerine bile daha dayanıklı olduğu ve bozucu yük momenti girişinden dolayı oluşan sistemdeki ani değişikliğe karşı sistemi bastırmada daha etkili olduğu gözlemlenmiştir. Bunlara ek olarak gelecekteki çalışmalarda balina optimizasyonu, güve-alevi optimizasyonu, atom arama optimizasyonu, gri kurt optimizasyonu, yabani ot optimizasyonu ve stokastik fraktal arama algoritmaları kullanılarak elde edilen PID parametre değerleri gerçek zamanlı olarak DC motor hız kontrolünde denenebilir ve sonuçları karşılaştırılabilir.

Yazarların Katkısı

Çalışmada tüm katkı tarafımıza aittir.

Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

Kaynaklar

- [1] Khanam I., Parmar G. 2017. Application of SFS algorithm in control of DC motor and comparative analysis. In 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), October, 256-261.
- [2] Khalilpour R., Razmjooy A., Hosseini H., Moallem P. 2011. Optimal control of DC motor using invasive weed optimization (IWO) algorithm. In Majlesi Conference on Electrical Engineering, Majlesi New Town, Isfahan, August, 1-6.
- [3] Hekimoğlu B. 2019. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. IEEE Access, 7: 38100-38114.
- [4] Agarwal J., Parmar G., Gupta R., Sikander A. 2018. Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor. Microsystem Technologies, 24 (12): 4997-5006.
- [5] Jain R.V., Aware M.V., Junghare A.S. 2016. Tuning of fractional order PID controller using particle swarm optimization technique for DC motor speed control. In 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), July, 1-4.
- [6] Saini R., Parmar G., Gupta R. 2020. SFS based Fractional Order PID Controller (FOPID) for Speed Control of DC Motor. International Journal, 9 (4).
- [7] Roy A., Srivastava S. 2016. Design of optimal PI λ D δ controller for speed control of DC motor using constrained particle swarm optimization. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), March, 1-6.
- [8] Munagala V.K., Jatoh R.K. 2021. Design of Fractional-Order PID/PID Controller for Speed Control of DC Motor Using Harris Hawks Optimization. In Intelligent Algorithms for Analysis and Control of Dynamical Systems, 103-113.
- [9] Azman M.A.H., Aris J.M., Hussain Z., Samat A.A.A., Nazelan A.M. 2017. A comparative study of fuzzy logic controller and artificial neural network in speed control of separately excited DC

- motor. In 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), November, 336-341.
- [10] Varshney A., Gupta D., Dwivedi B. 2017. Speed response of brushless DC motor using fuzzy PID controller under varying load condition. *Journal of Electrical Systems and Information Technology*, 4 (2): 310-321.
- [11] Wang M.S., Chen S.C., Shih C.H. 2018. Speed control of brushless DC motor by adaptive network-based fuzzy inference. *Microsystem Technologies*, 24 (1): 33-39.
- [12] Dursun E.H., Durdu A. 2016. Speed control of a DC motor with variable load using sliding mode control. *International Journal of Computer and Electrical Engineering*, 8 (3): 219-226.
- [13] Bingul Z., Karahan O. 2018. A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. *Journal of the Franklin Institute*, 355 (13): 5534-5559.
- [14] Ekinci S., Hekimoğlu B. 2019. Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system. *IEEE Access*, 7: 39935-39947.
- [15] Hekimoğlu B. 2019. Böbrek-ilhamlı Algoritma ile Ayarlanan PID Kontrolör Kullanarak DC Motor Hız Kontrolü. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 8 (2): 652-663.
- [16] Syafaah L., Pakaya I., Suhardi D., Irfan M. 2017. PID designs using DE and PSO algorithms for damping oscillations in a DC motor speed. In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), September, 1-5.
- [17] El-Deen A.T., Mahmoud A.H., El-Sawi A.R. 2015. Optimal PID tuning for DC motor speed controller based on genetic algorithm. *Int. Rev. Autom. Control*, 8 (1): 80-85.
- [18] Agarwal J., Parmar G., Gupta R. 2017. Application of sine cosine algorithm in optimal control of DC motor and robustness analysis. *Wulfenia Journal*, 24 (11): 77-95.
- [19] Ekinci S., Hekimoğlu B., Demirören A., Eker E. 2019. Speed Control of DC Motor Using Improved Sine Cosine Algorithm Based PID Controller. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), October, 1-7.
- [20] Das K.R., Das D., Das J. 2015. Optimal tuning of PID controller using GWO algorithm for speed control in DC motor. In 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), October, 108-112.
- [21] Ekinci S., Izci D., Hekimoğlu B. 2020. PID Speed Control of DC Motor Using Harris Hawks Optimization Algorithm. In 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), June, 1-6.
- [22] Ekinci S., Hekimoğlu B., Izci D. 2020. Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. *Engineering Science and Technology, an International Journal*. 1-12.
- [23] Achanta R.K., Pamula V.K. 2017. DC motor speed control using PID controller tuned by jaya optimization algorithm. In 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), September, 983-987.
- [24] Hekimoğlu B., Ekinci S., Demiray V., Doguruci R., Yıldırım A. 2018. Speed control of DC motor using PID controller tuned by salp swarm algorithm. In *Proc. IENSC*, 1878-1889.
- [25] Bhatt R., Parmar G., Gupta R., Sikander A. 2019. Application of stochastic fractal search in approximation and control of LTI systems. *Microsystem Technologies*, 25 (1): 105-114.
- [26] Mishra A.K., Tiwari V.K., Kumar R., Verma T. 2013. Speed control of DC motor using artificial bee colony optimization technique. In 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE), December, 1-6.
- [27] Duman S., Maden D., Güvenç U. 2011. Determination of the PID controller parameters for speed and position control of DC motor using gravitational search algorithm. In 2011 7th International Conference on Electrical and Electronics Engineering (ELECO), December, 225-229.
- [28] Ogata K. 2004. *System Dynamics*. 4th Ed., Pearson Prentice Hall, Upper Saddle River, NJ, 1-759.
- [29] Mirjalili S., Lewis A. 2016. The whale optimization algorithm. *Advances in engineering software*, 95: 51-67.
- [30] Mirjalili S. 2015. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89: 228-249.