

Otomatik Gerilim Regülatörü Sistemi Denetleyici Tasarımı için Meta-Sezgisel Algoritmaların Performansı

Ömer ÖZTÜRK^{1*} , Bora ÇAVDAR² 

Öz

Senkron generatörlerin terminal gerilimlerinin ayarlanması otomatik gerilim regülatörleri (AVR) tarafından sağlanır. Terminal geriliminin istenilen diğer bir deyişle referans gerilimde tutulması için sistemdeki değişikliklere hızlıca tepki verebilen bir denetleyici tarafından terminal gerilimin kontrol edilmesi gereklidir. Kullanılacak denetleyicinin seçimi önemli olduğu kadar parametrelerin ayarı da önemlidir. Bu sebeple çalışmada AVR sistemi için farklı denetleyici tiplerinin parametreleri farklı optimizasyon algoritmaları ve farklı amaç fonksiyonları kullanılarak optimize edilmiştir. Bu sayede optimizasyon algoritmalarının aynı koşullarda farklı durumlar altında performansları ortaya koyulmuştur. AVR sisteminde kullanılan denetleyiciler oransal-integral-türevsel denetleyici (PID), kesir dereceli PID (FOPID) ve FOPID denetleyicisine ek ikinci türev operatörü içeren versiyonu FOPIDD kullanılmaktadır. Bu denetleyicilerin parametreleri zebra optimizasyon algoritması (ZOA), karahindiba optimizasyon algoritması (DO) ve çiçek tozlaşma optimizasyon algoritması (FPA) ile optimize edilmiştir. Optimizasyon sürecinde ise zaman ağırlıklı mutlak hatanın integrali (ITAE) ve hata tabanlı yaklaşıma karşı olarak oluşturulan Zve-Lee Gaing (ZLG) amaç fonksiyonları kullanılmıştır. Elde edilen denetleyici-amaç fonksiyonu-algoritma performansları zaman bölge analizi, yakınsama eğrisi, kutu grafikleri ve diğer istatistiksel yöntemler ile karşılaştırılmıştır.

Anahtar Kelimeler: Otomatik gerilim regülatörü, Denetleyici tasarımı, Meta-sezgisel algoritma, ITAE, ZLG.

Performance of Meta-Heuristic Algorithms for Automatic Voltage Regulator System Controller Design

Abstract

The terminal voltage of synchronous generators is regulated by automatic voltage regulators (AVR). In order to maintain the terminal voltage at the desired, i.e. reference voltage, it is necessary to control the terminal voltage, it is necessary to control the terminal voltage by a controller that can react quickly to changes in the system. The choice of the controller to be used is as important as the setting of the parameters. For this reason, the parameters of different controller types for the AVR system are optimized using different optimization algorithms and different objective functions. In this way, the performance of the optimization algorithms under different situations under the same conditions was demonstrated. The controllers used in the AVR system are proportional-integral-derivative controller (PID), fractional-order PID (FOPID) and FOPIDD, a version of the FOPID controller with an additional second derivative operator. The parameters of these controllers were optimized with the zebra optimization algorithm (ZOA), dandelion optimization algorithm (DO) and flower pollination optimization algorithm (FPA). In the optimization process, the time-domain integral of absolute error (ITAE) and Zve-Lee Gaing (ZLG) objective functions, which are constructed as opposed to the error-based approach are used. The obtained controller-purpose function-algorithm performances are compared with time domain analysis, convergence curve, box plots and other statistical methods.

Keywords: Automatic voltage regulator, Controller design, Meta-heuristic algorithm, ITAE, ZLG.

¹Kocaeli Üniversitesi, Enerji Sistemleri Mühendisliği Bölümü, Kocaeli, Türkiye, 246171001@kocaeli.edu.tr

²Karadeniz Teknik Üniversitesi, Enerji Sistemleri Mühendisliği Bölümü, Trabzon, Türkiye, boracavdar@ktu.edu.tr

*Sorumlu Yazar/Corresponding Author

1. Giriş

Sistem kararlılığı, elektrik şebekelerinin kesintisiz ve güvenli bir şekilde elektrik tedarik edebilmesi için oldukça önemlidir. Kararlılığın bozulması durumunda şebekelerde kesintiler oluşur ve verim düşer. Bu nedenle şebekeye kaliteli ve sürekli enerji sağlanmalıdır (Kundur ve ark., 2004). Senkron generatörler kesintisiz ve yüksek kaliteli elektrik enerjisi sağlamada şebekenin önemli bir parçasıdır (Cavdar ve ark., 2023). Senkron generatörlerde frekans ve terminal geriliminin belirli sınırlar içerisinde tutulması gerekir. Frekans ve gerilimde bozulmalar meydana gelirse, güç sistemleri elemanlarının işlevselliği etkilenir. Frekans ve gerilimi belirli sınırlar içerisinde tutmak için aktif ve reaktif güç kontrolü yapılması gerekir. Gerilim kararlılığını terminal gerilimden sorumlu olan otomatik gerilim regülatörü (AVR) sağlar. Frekans kararlılığı ise senkron generatörlerin primer, sekonder ve tersiyer frekans kontrolü ile sağlanır (Çavdar ve ark., 2023). Bu makalede, senkron generatörlerde terminal gerilimini istenilen değerde tutabilen AVR sistemi üzerinde çalışılmaktadır.

AVR sisteminin amacı, elektrik güç sistemindeki senkron generatörün terminal gerilimini en kısa sürede belirlenmiş değer aralığına çekmek ve o aralıkta tutmaktır (Micev ve ark., 2020; Sahib, 2015a). Terminal gerilimini düzenlemek için generatörün uyarımı kontrol altına alınır, bu sayede bozucu etki ortadan kaldırılır (Saadat, 1999). Güç sistemlerinde yük değişiklikleri ve generatör alan sargılarının yüksek endüktansı, AVR'nin etkin ve hızlı tepki vermesini zorlaştırır (Bingul & Karahan, 2018). Bu nedenle AVR sisteminin düzgün çalışmasını sağlamak ve dinamik performansını arttırmak için bir denetleyici kullanılması gerekir. Literatürde AVR sisteminin kontrolü için birçok denetleyici kullanılmaktadır. Bunlardan en popüler olanı oransal-integral-türevsel (PID) denetleyicisidir. PID denetleyicisi, kolay uygulanabilirliği ve yüksek performansı sebebiyle birçok kontrol uygulamaları alanında sıklıkla tercih edilmektedir (El-Deen ve ark., 2015). Ek olarak, ikinci dereceden türevli operatörlü PID (PIDD²) (Sahib, 2015b), türev filtreli PID (PIDF) (Bingul & Karahan, 2018), kesir dereceli PID (FOPID) (Micev ve ark., 2020), bulanık mantık tabanlı PID (Fuzzy-PID) (Bingul & Karahan, 2018), 2 serbestlik dereceli PI (2DOF-PI) (Gozde, 2020) denetleyici türleri de literatürde AVR sistemlerinde tercih edilmiştir. PIDD² gibi denetleyicilerin kullanımı, AVR sisteminin performansını arttırmak için çeşitli denetleyicilerin geliştirilmesine yol açmıştır. Bu denetleyici çalışmalarına örnek olarak; $PI^{\lambda}D^{\mu}$ (Jumani ve ark., 2020), $PI^{\lambda}DND^2N^2$ (Moschos & Parisses, 2022), $PI^{\lambda}D^{\mu}D^2$ (Tabak, 2021b), $PI^{\lambda}DND^{\mu}$ (Paliwal ve ark., 2021), $PI^{\lambda}D^{\mu_1}D^{\mu_2}$ (Tabak, 2021a) verilebilir.

Denetleyicilerin seçimi her ne kadar kritik öneme sahipse, denetleyici parametrelerin optimum değerlerini belirlemek de o kadar önemlidir. Denetleyici parametreleri, manuel ayarlama yöntemlerle ve meta-sezgisel algoritmalarla belirlenebilir. Ziegler-Nichols ve Cohen-Coon, manuel ayarlama yöntemleri arasında en yaygın kullanılan tekniklerdir (Ekinci & Hekimoglu, 2019). Bu yöntemler, denetleyici parametrelerinin belirlenmesinin zorluğu, sistem yanıtında salınımlar, kalıcı hal hatası ve

aşım görülmesi nedeniyle günümüzde tercih edilmemektedir (Ghosh ve ark., 2021). Meta-sezgisel algoritmalar kolay uygulanabilirliği, optimum çözümü hızlı bir şekilde bulabilmesi gibi avantajlara sahiptir. Bu nedenle birçok mühendislik alanında sıklıkla kullanılmaktadır. AVR sistemi için denetleyicilerin parametrelerini en iyi şekilde optimize edebilmek amacıyla literatürde birçok meta-sezgisel arama algoritmaları kullanılmıştır. Bunlardan bazıları; Parçacık sürüsü optimizasyonu (PSO) (Sahib, 2015a), çoklu evren optimizasyonu (MVO) (13), denge optimize edici (EO) (Tabak, 2021a), sinüs kosinüs algoritması (SCA) (Ayas & Sahin, 2021), öğretme-öğrenme temelli optimizasyon (TLBO) (Eltag & Zhang, 2021), ağaç tohum algoritması (TSA) (Kose, 2020), jaya optimizasyon algoritması (JOA) (Bhookya & Jatoth, 2020), geliştirilmiş jaya optimizasyon algoritması (IJOA) (Bhookya & Jatoth, 2020), karga arama algoritması (CSA) (Bhullar ve ark., 2020), geliştirilmiş karga arama algoritması (ECSA) (Bhullar ve ark., 2020), çiçek tozlaşma algoritması (FPA) (Zhou ve ark., 2020), hibrit parçacık sürüsü optimizasyonu - yerçekimsel arama algoritması (PSO-GSA) (Zhou ve ark., 2020), gri kurt optimizasyonu (GWO) (Zhou ve ark., 2020), hibrit karınca kolonisi optimizasyonu - Nelder Mead metodu (ACO-NM) (Blondin ve ark., 2019), uyum arama algoritması (HSA) (Mosaad ve ark., 2018), yerel tek modlu örnekleme algoritması (LUS) (Mosaad ve ark., 2018), salp sürüsü optimizasyon algoritması (SSA) (Khan ve ark., 2019), çekirge optimizasyon algoritması (GOA) (Khan ve ark., 2019) şeklindedir.

Denetleyicinin ve algoritmanın performansını değerlendirmek için amaç fonksiyonlarına ihtiyaç duyulur. Amaç fonksiyonlarının seçimi oldukça önemlidir. Çünkü denetleyicinin performansını seçilen amaç fonksiyonu büyük ölçüde etkilemektedir (Ekinci & Hekimoğlu, 2019). Literatürde sıklıkla kullanılan amaç fonksiyonları; zaman ağırlıklı mutlak hatanın integrali (ITAE), zaman ağırlıklı hatanın karesinin integrali (ITSE), hata karesinin integrali (ISE), mutlak hatanın integrali (IAE) örnek olarak verilebilir. Bu amaç fonksiyonların yanı sıra Zve-Lee Gaing (ZLG) amaç fonksiyonu da literatürde sıklıkla kullanılmaktadır (Gaing, 2004). Tablo 1' de literatürdeki güncel AVR çalışmaları bu çalışmalarda kullanılan denetleyici, optimizasyon algoritması ve amaç fonksiyonu gösterilmektedir.

Tablo 1. 2018-2022 yılları arası AVR çalışmaları

Yıl	Ref.No.	Denetleyici	Optimizasyon	Amaç Fonksiyonu
2022	(Dogruer & Can, 2022)	Fuzzy – PID	GA	$a(ITAE) + \beta(\text{peak} - 1)$
2021	(Altbawi ve ark., 2024)	$PI^\lambda D^\mu$	GBO	ITAE
2021	(Tabak, 2021b)	PID, $PI^\lambda D^\mu$, $PIDD^2$, $PI^\lambda D^\mu D^2$	MVO	ZLG
2021	(Paliwal ve ark., 2021)	$PI^\lambda DND^\mu$	EO	ITAE, ITSE, IAE, ISE
2021	(Tabak, 2021a)	PID, $PI^\lambda D^\mu$, $PIDD^2$, $PI^\lambda D^{\mu_1} D^{\mu_2}$,	EO	ZLG
2021	(Eltag & Zhang, 2021)	PID, S. T. Fuzzy – PID, PIDN, S. T. Fuzzy – PIDN	TLBO	ZLG
2020	(Jumani ve ark., 2020)	$PI^\lambda D^\mu$	JOA	ITAE
2020	(Kose, 2020)	PID	TSA	$a_1(ITAE/IAE/ITSE/ISE) + a_2(t_s) + a_3(M_p)$
2020	(Bhookya & Jatoth, 2020)	PID, $PI^\lambda D^\mu$	JOA, IJOA	IAE, ITSE, ISE, ZLG
2020	(Bhullar ve ark., 2020)	PID	CSA, ECSA	$a \left(IAE + \int u(t) dt \right) + (1 - a)(t_s + t_r)$
2020	(Sikander & Thakur, 2020)	PID	CSA	ZLG, $(1 - a)ITAE + aM_p$
2020	(Zhou ve a., 2020)	PID	SCA, FPA, PSOGSA, GWO, SHO, ISHO	ITAE
2019	(Khan ve ark., 2019)	$PI^\lambda D^\mu$	SSA, PSO, ABC, GOA	ITAE
2019	(Blondin ve ark., 2019)	PID	ACO-NM	ZLG
2018	(Mosaad ve ark., 2018)	PID, $PIDD^2$	HSA, TLBO, LUS	ISE
	Önerilen Çalışma	PID, FOPID, FOPIDD	ZOA, DO, FPA	ZLG, ITAE

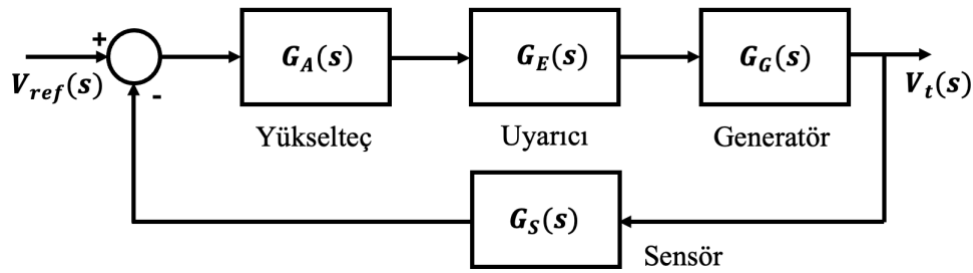
Bu çalışmada, AVR sisteminde PID, FOPID ve FOPIDD denetleyicileri kullanılarak bu denetleyicilerin parametreleri Zebra Optimizasyon Algoritması (ZOA), Karahindiba Optimizasyon Algoritması (DO) ve Çiçek Tozlaşma Algoritması (FPA) ile optimize edilmiştir. Optimizasyon sırasında tüm denetleyiciler için iki farklı amaç fonksiyonu ITAE ve ZLG kullanılmıştır. Her algoritma aynı uygunluk fonksiyonu değerlendirme sayısı kadar iterasyona tabi tutulmuştur. İterasyon sayıları yakınsama eğrilerinin davranışlarını sabitlediği noktaya göre belirlenmiştir. Burada kullanılan iterasyon sayıları PID için 5000, FOPID için 10000 ve FOPIDD için ise 15000'dir. Ayrıca her koşullar bağımsız şekilde 10 kez tekrarlanmış hem en iyi sonuçlar için geçici yanıt analizleri

paylaşmış hem de 10 koşu için elde edilen sonuçlar istatistiksel olarak analiz edilmiştir. Bu sayede literatürde ilk defa DO, ZOA ve FPA algoritmalarının performansları AVR sistemi için detaylı olarak aynı koşullar altında karşılaştırılmıştır.

2. Materyal ve Metot

2.1. AVR Sistem Modeli

AVR sistemi, generatörün terminal gerilimini belirli bir seviyede tutmak için kullanılır. Şekil 1’ de AVR sisteminin blok diyagramı verilmiştir.



Şekil 1. AVR sisteminin blok diyagramı

Şekil 1’ de görüldüğü üzere AVR sistemi dört ana bileşenden meydana gelmektedir. Bu bileşenler, yükselteç, uyarıcı, generatör ve sensördür. AVR sisteminin çalışma prensibi aşağıdaki gibi beş adımda özetlenebilir.

- 1. Adım: Terminal gerilimi sensör tarafından ölçülür ve düzenlenir.
- 2. Adım: Sensör çıkışındaki gerilim değeri ile referans değeri karşılaştırılarak hata sinyali üretilir.
- 3. Adım: Üretilen hata sinyali denetleyici tarafından kontrol edilir.
- 4. Adım: Kontrol sinyali yükselteç sayesinde yükseltilir.
- 5. Adım: Generatörün terminal gerilimi uyarıcı ile ayarlanır.

AVR sistemindeki bileşenlerin parametreleri farklı değer aralıklarına sahiptir. Bu değerlerin sınırları ve genellikle seçilen parametre değerleri, literatürdeki çalışmalar temel alınarak Tablo 2’ de verilmiştir (Çavdar ve ark., 2023; Güvenç ve ark., 2016; Zeng ve ark., 2015).

Tablo 1. AVR sisteminin deęer aralıkları ve parametre deęerleri

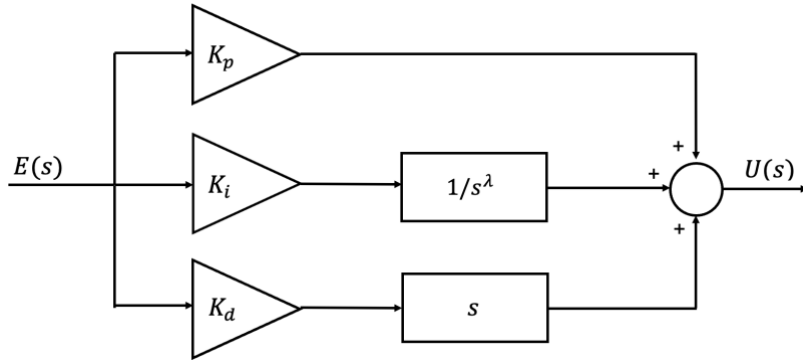
AVR Bileşenleri	Zaman Sabiti Deęer Aralığı	Kazanç Katsayısı Deęer Aralığı	Seçilen Deęerler
Yükselteç	$0.02 \leq \tau_A \leq 0.1$	$10 \leq K_A \leq 40$	$\tau_A = 0.1, K_A = 10$
Uyarıcı	$0.4 \leq \tau_E \leq 1$	$1 \leq K_E \leq 10$	$\tau_E = 0.4, K_E = 1$
Generatör	$1 \leq \tau_G \leq 2$	$0.7 \leq K_G \leq 1$	$\tau_G = 1, K_G = 1$
Sensör	$0.001 \leq \tau_S \leq 0.06$	$0.9 \leq K_S \leq 1.1$	$\tau_S = 0.01, K_S = 1$

Tabloda, K_A yükselteç kazancını, τ_A yükselteç zaman sabitini, K_E uyarıcı kazancını, τ_E uyarıcı zaman sabitini, K_G generatör kazancını, τ_G generatör zaman sabitini, K_S sensör kazancını, τ_S ise sensör zaman sabitini temsil etmektedir.

2.2. Denetleyiciler

2.2.1. PID Denetleyici

PID denetleyici basit ve kararlı yapıya sahip olması nedeniyle literatürde sıklıkla kullanılır (Dhanasekaran ve ark., 2020). Denetleyici üç farklı parametreye sahiptir. Şekil 3'te PID denetleyicisinin blok diyagramı verilmiştir.

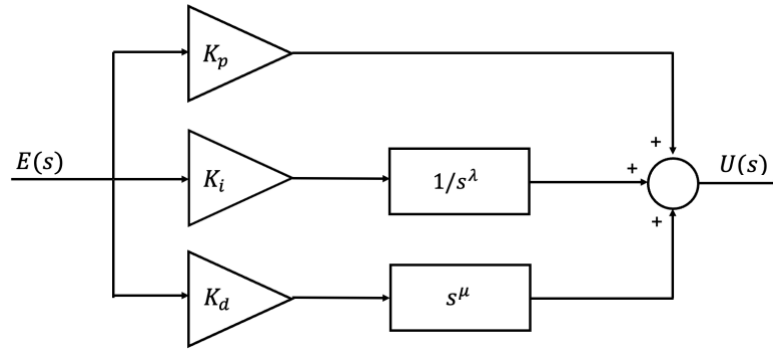
**Şekil 3.** PID denetleyicisinin blok diyagramı

Burada K_p oransal kazanç sabitini, K_i integral kazanç sabitini, K_d ise türevsel kazanç sabitini temsil etmektedir. PID denetleyicisinin transfer fonksiyonu Denklem 1' de verilmiştir.

$$G_{PID} = K_p + \frac{K_i}{s} + K_d s \quad (1)$$

2.2.2. Kesir Dereceli PID Denetleyici

Kesir dereceli PID denetleyicisi (FOPID), Podlubny tarafından önerilmiş olup, literatürde sıklıkla tercih edilen denetleyicilerden biridir (Moschos & Parisses, 2022). FOPID denetleyicilerin bu kadar popüler olmasının nedeni geleneksel denetleyicilerden daha iyi performans sunmasıdır. PID denetleyicisinden daha karmaşık bir yapıya sahiptir. Denetleyicide beş farklı parametre mevcuttur. Şekil 4' te FOPID denetleyicisinin blok diyagramı verilmiştir.



Şekil 4. FOPID denetleyicisinin blok diyagramı

Burada, μ ve λ gerçekte pozitif tamsayılarıdır. Kesir dereceli denetleyici tasarlayabilmek için kesirli kalkülüse ihtiyaç vardır. Literatürde en çok kullanılan kesirli kalkülüs yöntemi Oustaloup yakınsamasıdır (Oustaloup ve ark., 2000). Denklem 2' de bant geçiren filtreye dayalı Oustaloup yakınsama yöntemi verilmiştir (Çelik, 2021; Xue ve ark., 2006).

$$s^\alpha \approx \omega_h^\alpha \prod_{k=-N}^N \frac{s + \omega_z}{s + \omega_p} \quad \text{for } 0 < \alpha < 1 \quad (2)$$

Burada, N kutup çiftleri sayısıdır ve görevi analog filtre derecesini $(2N + 1)$ belirlemektir. ω_z sıfırları, ω_p kutupları, α integralin ya da türevin derecesini temsil etmektedir. Sıfırlar ve kutuplar aşağıdaki gibi tanımlanmaktadır;

$$\omega_p = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+N+\frac{1}{2}(1+\alpha)}{2N+1}}, \omega_z = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+N+\frac{1}{2}(1-\alpha)}{2N+1}} \quad (3)$$

Burada, ω_h ve ω_b sırasıyla frekans bandı üst ve alt sınırlarıdır. Çalışmada frekans bandı alt sınırı 10^{-4} rad/s , üst sınır ise 10^4 rad/s olarak belirlenmiştir. Oustaloup yaklaşımı kullanılarak oluşturulan FOPID denetleyicisinin transfer fonksiyonu Denklem 4' te verilmiştir.

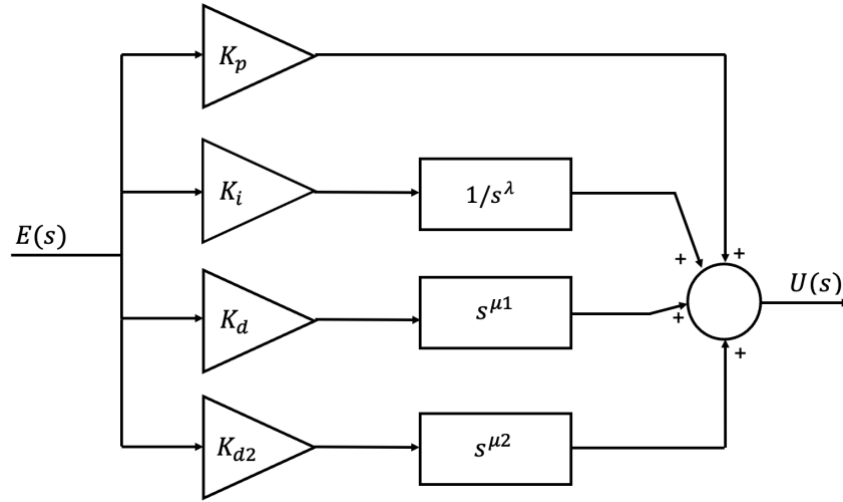
$$G_{FOPID} = K_p + K_i s^{-\lambda} + K_d s^{\mu} \quad (4)$$

2.2.3. FOPIDD Denetleyici

Literatürdeki AVR sistemleri çalışmaları incelendiğinde FOPID, PIDD² gibi çeşitli denetleyiciler tercih edilmiştir. FOPID denetleyicisi tanıtıldığı günden bu yana birçok mühendislik uygulamalarında sıkça kullanılmıştır. PIDD² denetleyicisi ise AVR sistemlerinde başarılı sonuçlar elde ederek literatüre katkılarda bulunmuştur. Ancak kesirli kalkülüs ile PIDD² denetleyicisinin birleştirilmesi ile elde edilen FOPIDD denetleyicisi AVR sistemlerinde çok fazla kullanılmamıştır (Tabak, 2021b). Denklem 5' te FOPIDD denetleyicisinin transfer fonksiyonu verilmiştir.

$$G_{FOPIDD} = K_p + K_i s^{-\lambda} + K_d s^{\mu_1} + K_{d2} s^{\mu_2} \quad (5)$$

Denklem 5' te görüldüğü üzere FOPIDD denetleyicisinde yedi farklı parametre bulunmaktadır. Şekil 5'te FOPIDD denetleyici için blok diyagramı verilmiştir.



Şekil 5. FOPIDD denetleyicisinin blok diyagramı

2.3. Algoritmalar

Meta-sezgisel algoritmalar, özellikle mühendislik başta olmak üzere birçok alanda problem çözmedeki başarıları sayesinde son yıllarda büyük bir popülerlik kazanmıştır. Bu algoritmalar, kullanım kolaylığı, hızlı uyarlanabilme ve optimal çözüme etkili bir şekilde yakınsama gibi birçok yeteneği ile öne çıkmaktadır. Bu doğrultuda, önerilen çalışmada literatürde daha önce kullanılmamış ZOA algoritmasının yanı sıra, literatürde sıklıkla kullanılan güçlü ve rekabetçi algoritmalar DO ve FPA algoritmaları da tercih edilmiştir. Bir sonraki alt başlıklarda, önerilen algoritmalar hakkında detaylı bilgilere yer verilmiştir.

2.3.1. Zebra Optimizasyonu Algoritması

ZOA zebraların doğada hayatta kalma stratejilerinden ilham alınarak tasarlanan sürü tabanlı algoritmadır (Trojovska ve ark., 2022). Zebraların doğada sergilediği iki temel davranışı vardır. Bunlar yiyecek arama ve avcılardan korunma stratejileridir (Trojovska ve ark., 2022). ZOA'nın ilk aşamasında, zebra popülasyonundaki en iyi zebra öncü zebra olarak seçilir. Kalan üyeler ise arama uzayında konumlandırılır (Danell ve ark., 2006). Oluşturulan konumu güncellemek için Denklem 6 ve 7'deki matematiksel ifadeler kullanılarak güncellenir.

$$x_{ij}^{new,P1} = x_{ij} + r. (PZ_j - I. x_{ij}) \quad (6)$$

$$X_i = \begin{cases} X_{ij}^{new,P1}, & F_i^{new,P1} < F_i; \\ X_i, & else, \end{cases} \quad (7)$$

Burada; $X_{ij}^{new,P1}$ birinci aşamaya bağlı j. zebranın yeni konumunu, $x_{ij}^{new,P1}$ j. boyuttaki değerini, $F_i^{new,P1}$ amaç fonksiyonun değerini, PZ seçilmiş öncü zebrayı, PZ_j j. boyutunu, r [0,1] arasında seçilmiş olan rastgele değerini, I ise $I = round(1 + rand)$ matematiksel ifadesiyle hesaplanan, $rand$ değeri [0,1] arası seçilen rastgele sayıyı temsil etmektedir. Böylece $I \in \{1,2\}$ arasında değer almış olur. Eğer $I=2$ olursa zebra popülasyonunda çok fazla değişikliğe sebep olur.

ZOA'nın ikinci aşamasında, avcılarının türüne göre oluşturulmuş savunma mekanizmalarından faydalanılarak popülasyondaki zebraların konumu güncellenir. Zebraların başlıca avcıları genellikle aslanlardır. Zebralar aslanlardan kaçarken, zikzak ve rastgele dönüş hareketleri yapar (Wilson ve ark., 2018). Çita, leopar ve yaban köpekleri gibi diğer yırtıcılara karşı ise bu stratejileri kullanmazlar. Bu yırtıcılara karşı, kafalarını karıştırarak ve korkutarak onları uzaklaştırmaya çalışırlar. Bu iki stratejinin matematiksel ifadeleri S_1 ve S_2 olmak üzere Denklem 7' de verilmiştir. Konum güncellemesi ise Denklem 8' de verilmiştir.

$$x_{ij}^{new,P2} = \begin{cases} S_1: x_{ij} + R \cdot (2r - 1) \cdot (1 - \frac{t}{T} \cdot x_{ij}), & P_s < 0; \\ S_2: x_{ij} + r \cdot (AZ_j - I \cdot x_{ij}), & else, \end{cases} \quad (8)$$

$$X_i = \begin{cases} X_i^{new,P2}, & F_i^{new,P2} < F_i; \\ X_i, & else, \end{cases} \quad (9)$$

Denklem 8 ve 9' da $X_i^{new,P2}$ ikinci aşamadaki i. zebanın yeni konumunu, $x_{ij}^{new,P2}$ j. boyuttaki değerini, $F_i^{new,P2}$ amaç fonksiyonun değerini, t iterasyon sayısını, T maksimum iterasyon sayısını, R 0,01 değerine eşit olan sabit bir katsayıyı, P_s [0,1] arasında rastgele seçilen iki senaryodan birini seçme olasılığını, AZ saldırıya uğramış zebraı, AZ_j ise j. boyuttaki değerini temsil etmektedir.

2.3.2. Karahindiba Optimizasyon Algoritması

DO sürü zekâsı ilhamlı bir optimizasyon algoritmasıdır (Zhao ve ark., 2022). DO algoritması, karahindiba tohumunun rüzgâr ile taşınarak farklı noktalara yayılma sürecini model alır. Bu süreç üç farklı aşamaya ayrılır. İlk aşama, tohumların hava akımının ve sürüklenme kuvvetinin etkisiyle spiral bir şekilde yukarı doğru hareket ettiği yükselme evresidir. Bu aşama genellikle güneşli havalarda gerçekleşir. Buna ek olarak, yağmurlu havalarda karahindiba tohumları, rüzgârın etkisinin az olması nedeniyle daha çok buldukları yerin çevresinde yayılır. Bu yükseliş farkı, tohumların rüzgâr ve hava akımlarının etkisiyle rastgele konumlara yerleşmesini sağlayarak iki farklı arama senaryosu sunar ve bu süreç sonunda yeni karahindiba bitkilerinin oluşumu gerçekleşir.

Havanın açık olduğu durumda rüzgârın log-normal dağılıma uygun olduğu kabul edilir. Böylece tohumların daha uzak mesafelere taşınma olasılığı artar. Bu durum DO algoritmasının keşif aşamasını oluşturmaktadır. Tohumların yükseklerle doğru çıkması rüzgâr hızıyla orantılıdır. Denklem 10' da matematiksel ifadesi verilmiştir.

$$X_{t+1} = X_t + \alpha \cdot v_x \cdot v_y \cdot \ln Y \cdot (X_s - X_t) \quad (10)$$

Burada, X_t tohumun t. İterasyondaki konumunu, X_s t. İterasyonda arama uzayındaki rastgele konumunu, $\ln Y$ ise log-nominal dağılımı ifade etmektedir. X_s değerini elde ederken Denklem 11' deki matematiksel ifade kullanılmaktadır (Zhao ve ark., 2022).

$$X_s = rand(1, Dim) \cdot (x_j^{max} - x_j^{min}) + x_j^{min} \quad (11)$$

Denklem 10' da belirtilen α değeri, arama adım uzunluğunu ayarlayan uyarlanabilir bir parametredir. Bu değerın matematiksel ifadesi Denklem 11' de verilmiştir (Zhao ve ark., 2022).

$$\alpha = rand. \left(\frac{1}{T^2} t^2 - \frac{2}{T} t + 1 \right) \quad (12)$$

v_x ve v_y değerleri karahindiba kaldırma katsayısı bileşenlerini ifade etmektedir. Bu değerleri hesaplamak için Denklem 13-15' teki matematiksel ifadeler kullanılmaktadır (Zhao ve ark., 2022).

$$r = \frac{1}{e^\theta} \quad (13)$$

$$v_x = r \cdot \cos \theta \quad (14)$$

$$v_y = r \cdot \sin \theta \quad (15)$$

Burada, θ $[-\pi, \pi]$ arasında rastgele bir sayıdır.

Havanın yağışlı olduğu durumda tohumlar, yeterince yükselemedikleri için uzak mesafelere doğru uçamazlar. Bu durumun matematiksel ifadesi Denklem 16' da verilmiştir.

$$X_{t+1} = X_t \cdot k \quad (16)$$

Burada k değeri karahindibanın yerel arama alanını düzenlemek için kullanılan sayıdır. Bu değerın matematiksel ifadesi Denklem 17' de verilmiştir.

$$k = 1 - rand. q \quad (17)$$

$$q = \frac{1}{T^2 - 2T + 1} t^2 - \frac{2}{T^2 - 2T + 1} t + \frac{1}{T^2 - 2T + 1} \quad (18)$$

Yükselme durumundaki karahindiba tohumlarının davranışının matematiksel ifadesi Denklem 19' da verilmiştir (Zhao ve ark., 2022).

$$X_{t+1} = \begin{cases} \{X_t + \alpha \cdot v_x \cdot v_y \cdot \ln Y \cdot (X_s - X_t), & randn < 1.5 \\ X_t \cdot k, & diğ er \end{cases} \quad (19)$$

Burada $randn$, rastgele üretilen bir sayıdır.

İkinci aşama olan alçalma aşamasında, yükselen tohumlar uzayda yönlerini sürekli olarak düzenleyecek şekilde alçalırlar. Hareket yörüngesi, Brown hareketi ile taklit edilir. Bu aşama keşif aşamasına karşılık gelir. Denklem 20' de matematiksel ifadesi verilmiştir (Zhao ve ark., 2022).

$$X_{t+1} = X_t - \alpha \cdot \beta_t \cdot (X_{mean_t} - \alpha \cdot \beta_t \cdot X_t) \quad (20)$$

Burada, β_t Brown hareketini belirten rastgele bir sayısını, X_{mean_t} ise i . iterasyondaki popülasyonun ortalama konumunu ifade etmektedir. Ortalama konumu Denklem 21' deki matematiksel ifade ile hesaplanmaktadır (Zhao ve ark., 2022).

$$X_{mean_t} = \frac{1}{pop} \sum_{i=1}^{pop} X_i \quad (21)$$

Algoritmanın son aşaması iniş aşamasıdır. Bu aşamada, tohumların büyüebilmesi için rastgele konumlara iniş yaparlar. Bu aşamanın matematiksel ifadesi Denklem 22' de verilmiştir.

$$X_{t+1} = X_{elite} - levy(\lambda) \cdot \alpha \cdot (X_{elite} - X_t \cdot \delta) \quad (22)$$

Burada, X_{elite} en iyi çözümü, $levy(\lambda)$ Levy uçuşunu, δ ise 0'dan 2'ye doğrusal artan bir fonksiyonu ifade etmektedir. Levy uçuşunun matematiksel ifadesi Denklem 23'te verilmiştir.

$$Levy(\lambda) = s \times \frac{\omega \times \sigma}{|t|^{\frac{1}{\beta}}} \quad (23)$$

Burada, β [0,2] arasında rastgele seçilen bir sayıya, s değeri 0,001 olan sabit bir sayıya, ω ve t [0,1] arasında rastgele seçilen bir sayıya karşılık gelir. σ değerinin matematiksel ifadesi Denklem 24'te verilmiştir.

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \quad (24)$$

Burada β değeri 1,5 olan sabit bir sayıyı ifade etmektedir.

2.3.3. Çiçek Tozlaşma Algoritması

FPA Xin-She Yang tarafından geliştirilen doğadan ilham alan bir optimizasyon algoritmasıdır (Yang, 2012). Bu algoritma, bitkilerin tozlaşma sürecinden ilham alır. Tozlaşma süreci dört ana kuraldan meydana gelir. Doğal tozlaşma, böcekler, rüzgâr ve su gibi doğal etkenler vasıtasıyla gerçekleşen uzun mesafeli polen yolculuğunu ifade eder. Yerel tozlaşma, tozlaşmanın aynı bitki veya komşu bitkiler arasında meydana geldiği süreçtir. Bu kısımda, belirli komşuluk sınırlarında arama yapılır. Çiçeklerin etkili tozlaşması, hayatta kalma ve yayılma yetenekleriyle doğrudan ilişkilidir. En iyi çözümler, diğerlerine göre daha cazip çiçek olarak kabul edilir ve arama süreci bu çözümlerin etrafında toplanma eğilimi gösterir. Tozlaşma olasılığı algoritmanın arama sürecinde hem geniş alanları keşfetmesini hem de mevcut çözümler üzerinde ince ayar yapmasına imkân tanır. Dört farklı kural doğrultusunda, FPA algoritması yerel ve küresel arama süreçleri arasında denge kurarak farklı optimizasyon problemlerinde etkili olmayı başarır.

Gerçek zamanda, her türün birden fazla çiçeğe ve her çiçeğin milyarlarca polen gametine sahip olmasına rağmen, algoritmanın basitliği açısından her türün yalnızca bir çiçeği ve bu çiçeğin de sadece bir polen gameti ürettiği varsayılmıştır. Böylece polenlerin tek tek tanımlama ve ayırt etme gereksinimi ortadan kaldırılmıştır.

Denklem 25'te çiçek sabitinin matematiksel ifadesi verilmiştir.

$$x^{t+1} = x_i^t + \gamma \cdot L(\lambda)(g_* - x_i^t) \quad (25)$$

Burada x^{t+1} çözüm vektörünü, g_* mevcut en iyisini, γ ise adım boyutunu ayarlama faktörüdür. Levy dağıtımı tozlaşma kuvvetini karşılamak için kullanılır. Böcekler daha uzak yerlere uçarken, böceklerin hareketi Levy dağılımına göre gösterilebilir. Levy'nin matematiksel ifadesi Denklem 26'da verilmiştir.

$$L \sim \frac{\lambda \Gamma(\lambda) \cdot \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0) \quad (26)$$

Burada $\Gamma(\lambda)$ standart gama fonksiyonudur ve s adım büyüklüğündedir. Bu dağılım $s > 0$ büyük adımlar için geçerlidir. Teoride $s_0 \gg 0$ olması gerekir. Ancak uygulamada s_0 değeri 0,1 kadar küçük bir değer de olabilir.

2.4. Amaç Fonksiyonu

Amaç fonksiyonları, optimizasyon sürecinde önemli bir rol oynar. Burada ele alınan problemde denetleyici parametrelerinin optimum değerlerinin belirlenmesi için kullanılır. Optimizasyon problemlerinde amaç fonksiyonu genellikle hata değerini ifade eder. Amaç fonksiyonunun değeri ne kadar az olursa, denetleyicinin parametreleri o kadar iyi ayarlanmış olur. Böylece sistem performansında iyileştirmeler meydana gelir (Gaing, 2004). AVR sisteminin performansı, zaman bölge analizi yöntemi ile belirlenir. Zaman bölge analizi ile tespit edilen yerleşme zamanı (t_s), yükselme zamanı (t_r) ve maksimum aşım (M_p) gibi geçici yanıt parametreleri minimum değere indirgenerek AVR sisteminin performansının iyileştirilmesi hedeflenir.

ITAE amaç fonksiyonu, t_s , t_r ve M_p parametrelerini iyileştirir. Optimizasyon sırasında, ITAE ne kadar küçük bir değerde olursa, denetleyici parametreleri o kadar iyi ayarlanmıştır. Denklem 27’de ITAE amaç fonksiyonunun matematiksel ifadesi verilmiştir.

$$ITAE = \int_0^t t|e(t)|dt \quad (27)$$

Burada, e hatayı t ise hata zamanını ifade etmektedir.

ZLG amaç fonksiyonu, hem geçici yanıtları (t_s , t_r , M_p) hem de kararlı durum yanıtı (E_{ss}) gibi çıkış parametrelerini içermesinden dolayı başarılı sonuçlar göstermektedir. Denklem 28’de ZLG amaç fonksiyonunun matematiksel ifadesi verilmiştir.

$$ZLG = (1 - e^{-\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r) \quad (28)$$

Burada, E_{ss} kalıcı durum hatasını, β ağırlık faktörünü ifade etmektedir.

AVR çalışmalarında, hata tabanlı birçok amaç fonksiyonları kullanılmaktadır. Önerilen çalışmada ITAE ve ZLG amaç fonksiyonlarının tercih edilme nedeni, daha başarılı sonuçlar elde edilmesidir. ITAE amaç fonksiyonu, oturma süresini kısaltma ve aşımı azaltma konusundaki üstün performansı sayesinde diğerlerine göre öne çıkmaktadır. ZLG amaç fonksiyonunda ise hem geçici yanıtları hem de kararlı durum yanıtını içeren parametreleri kapsadığı için tercih edilmiştir (Gaing, 2004).

3. Bulgular ve Tartışma

Bu bölümde, optimizasyon algoritmaları sırasıyla PID, FOPID ve FOPIDD denetleyicileri için karşılaştırılmaktadır. Üç denetleyicinin performans değerlendirme kriterleri benzer aşamalardan oluşmaktadır. Bu aşamalar aşağıda sırasıyla verilmektedir.

- ZLG ve ITAE için elde edilen zaman bölge analizi sonuçlarının değerlendirilmesi
- ITAE VE ZLG'deki algoritmaların yakınsama eğrilerinin değerlendirilmesi
- Algoritmaların 10 bağımsız çalışma sonucunda elde edilen kutu grafiklerinin değerlendirilmesi
- Algoritmaların 10 bağımsız çalışma sonucunda istatistiksel olarak değerlendirilmesi

Bu çalışmada, literatürde sıklıkla kullanılan denetleyici parametre aralığı dikkate alınmıştır. Bu sayede elde edilen sonuçlar literatürdeki çalışmalar ile karşılaştırılabilir. Tablo 4' te üç denetleyici için sınır değer aralığı verilmiştir.

Tablo 4. Denetleyici sınır değerleri.

	K_p	K_i	K_{d1}	K_{d2}	λ	μ_1	μ_2
PID	0,001-3	0,001-3	0,001-3	-	1	1	-
FOPID	0,001-3	0,001-3	0,001-3	-	0,1-1,5	0,1-1,5	-
FOPIDD	0,001-3	0,001-3	0,001-3	0,001-3	0,1-1,5	0,1-1,5	0,1-2,5

3.1. PID Denetleyicisi için Algoritmaların Performansı

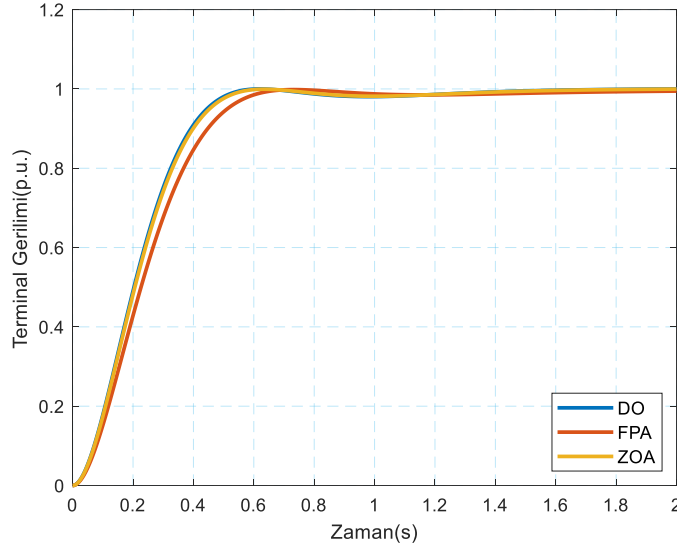
3.1.1. Algoritmaların PID-ZLG için Performansı

DO, FPA ve ZOA algoritmaları, PID-ZLG için çalıştırılarak zaman bölgesindeki değerler alınmıştır. Bu değerler Tablo 5'te verilmiştir.

Tablo 5. PID-ZLG için zaman bölge analizi ve denetleyici parametreleri.

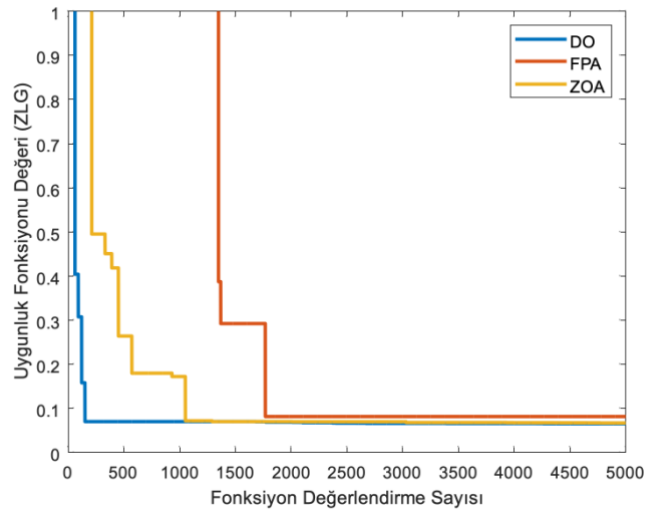
Algoritmalar	En İyi Değer	K_p	K_i	K_d	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,0651	0,6006	0,4202	0,2029	0,498	0,321	0
FPA	0,0815	0,5503	0,3636	0,1708	0,581	0,369	0
ZOA	0,0668	0,5933	0,4133	0,1985	0,509	0,328	0

Tablo 5 incelendiğinde, en iyi yerleşme ve yükselme zamanını DO algoritması elde etmiştir. Üç algortmada da aşım görülmemektedir. ZLG için en iyi değerler sırasıyla, DO, ZOA ve FPA algoritmasıdır. Şekil 6'da algoritmaların zaman bölge grafikleri verilmiştir.



Şekil 6. PID-ZLG için sistemin geçici yanıtı.

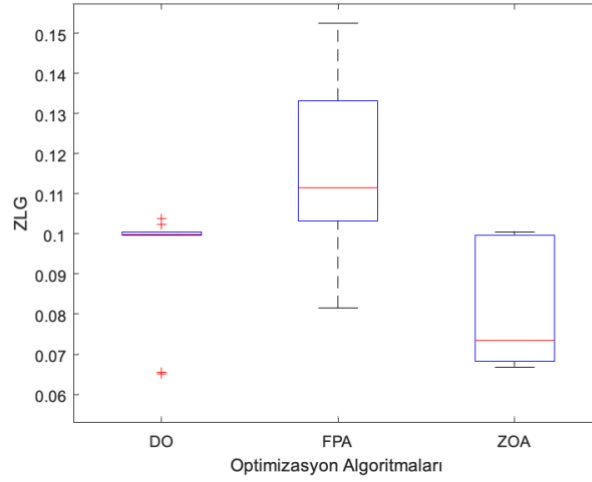
Şekil 6’da DO ve ZOA algoritması ile elde edilen yanıt benzer sürede referansa otururken FPA algoritması ile elde edilen sonuçların daha geç referansa ulaştığı görülmektedir. Şekil 7’de algoritmaların en iyi sonuçları elde ettiği yakınsama eğrileri verilmiştir.



Şekil 7. PID-ZLG için yakınsama eğrisi.

Şekil 7 incelendiğinde, DO algoritması 0-500 arasında fonksiyon değerlendirme sayısı arasında ZOA algoritması 1000 değerlendirme sayısında, FPA ise 2000 fonksiyon değerlendirme sayısında minimum değere ulaştığı görülmektedir.

Bir algoritmanın performansını test ederken, en iyi değer yanında diğer tüm sonuçları da değerlendirmek önemlidir. Şekil 8’de üç algoritmanın kutu grafiği verilmiştir.



Şekil 8. PID-ZLG için kutu grafiği.

DO algoritmasının aykırı değerler dışında lokal minimuma takıldığı Şekil 8'den açıkça görülmektedir. ZOA algoritması DO ve FPA algoritmalarına göre daha tutarlı ve başarılı sonuç ortaya koymaktadır.

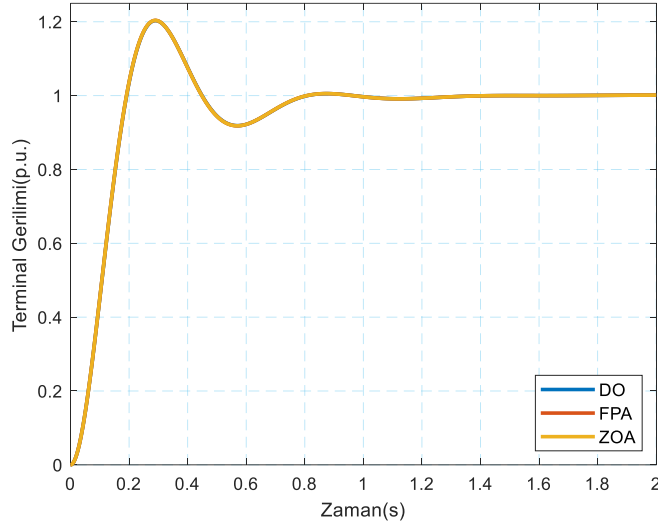
3.1.2. Algoritmaların PID-ITAE için Performansı

DO, FPA ve ZOA algoritmaları, PID-ITAE için çalıştırılarak zaman bölgesindeki değerler alınmıştır. Bu değerler Tablo 6'da verilmiştir.

Tablo 6. PID-ITAE için zaman bölge analizi ve denetleyici parametreleri.

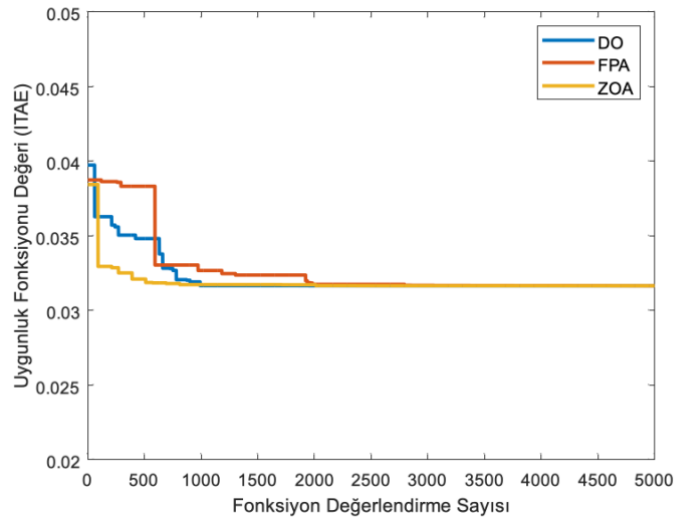
Algoritmalar	En İyi Değer	K_p	K_i	K_d	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,0316	1,4249	1,0065	0,5447	0,738	0,129	20,30
FPA	0,0316	1,4280	1,0091	0,5462	0,737	0,129	20,372
ZOA	0,0316	1,4277	1,0090	0,5464	0,737	0,129	20,368

Tablo 6'da en iyi ITAE değeri üç algortmada aynı olmasına rağmen en az aşım DO algoritması elde etmiştir. Sonuçlar oldukça birbirine yakındır Şekil 9'da da bu durum açıkça görülmektedir.



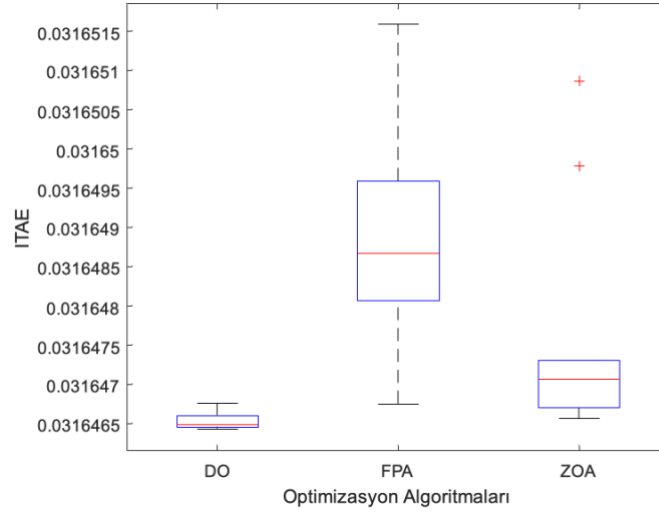
Şekil 9. PID-ITAE için sistemin geçici yanıtı.

Şekil 10’da algoritmaların en iyi sonuçların elde edildiği koşullar için yakınsama eğrileri verilmiştir.



Şekil 10. PID-ITAE için yakınsama eğrisi.

Şekil 10’ da ZOA ve DO algoritmaların 2000 fonksiyon değerlendirme sayısına ulaşmadan önce yakınsadığı görülmektedir. FPA algoritması ise 2000 fonksiyon değerlendirme sayısından sonra minimum değere yakınsadığı görülmektedir. Şekil 11’de üç algoritmanın kutu grafiği verilmiştir.



Şekil 11. PID-ITAE için kutu grafiği.

Kutu grafikleri incelendiğinde, en az hata değer aralığına sahip algoritmanın DO olduğu açıkça görülmektedir.

3.1.3. PID-ZLG ve PID-ITAE İstatistiksel Analiz

PID denetleyici kullanılarak algoritmaların 10 koşu sonucunda hem ZLG hem de ITAE için diğer bir istatistiksel analiz yöntemi olarak ortalama, standart sapma ve en iyi değer Tablo 7’de ele alınmaktadır.

Tablo 7. PID-ZLG ve PID-ITAE için istatistiksel analiz.

Algoritma	En iyi değer	ZLG		En iyi değer	ITAE	
		Ortalama	Standart Sapma		Ortalama	Standart Sapma
DO	0,0651	0,0936	0,0150	0,0316	0,0316	1,24E-07
FPA	0,0815	0,1166	0,0205	0,0316	0,0316	1,42E-06
ZOA	0,0668	0,0807	0,0147	0,0316	0,0316	1,49E-06

ZLG amaç fonksiyonunda en iyi ortalama değeri ZOA algoritması ile edilmiştir. ITAE amaç fonksiyonunda ise en iyi ortalama değeri üç algorithmada aynı çıkmıştır. Standart sapma değerleri incelendiğinde, ZLG amaç fonksiyonunda en iyi değeri DO algoritması, ITAE amaç fonksiyonunda ise en iyi değeri FPA algoritması sağlamıştır.

3.2. FOPID Denetleyicisi için Algoritmaların Performansı

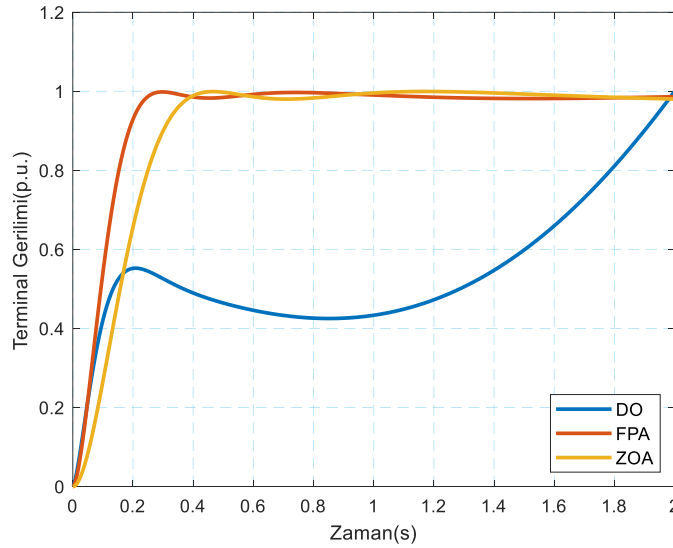
3.2.1. Algoritmaların FOPID-ZLG için Performansı

Tablo 8’ de FOPID-ZLG için en iyi değerleri ve denetleyici parametreleri verilmiştir.

Tablo 8. FOPID-ZLG için zaman bölge analizi ve denetleyici parametreleri.

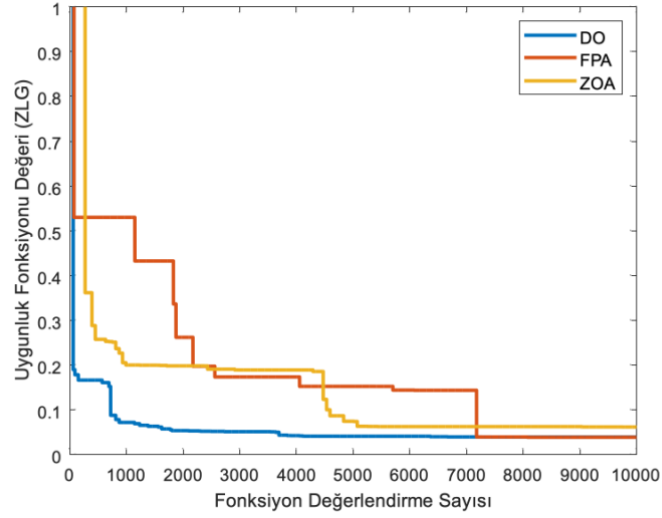
Algoritmalar	En İyi Değer	K_p	K_i	K_d	λ	μ	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,0394	0,0010	0,3614	0,1657	1,5	1,5	-	-	-
FPA	0,0388	1,7418	0,3848	0,3459	1,3948	1,2824	0,239	0,157	0
ZOA	0,0617	0,5719	0,6417	0,3177	0,6379	1,0376	0,379	0,245	0

Tablo 8’de en iyi ZLG değeri DO algoritması verdiği görülmektedir. FPA ve ZOA algoritmalarında aşım gözlemlenmemektedir. En kısa yerleşme ve yükselme zamanını FPA algoritması elde etmiştir. Şekil 12’de FOPID-ZLG için sistemin geçici yanıtları verilmiştir.



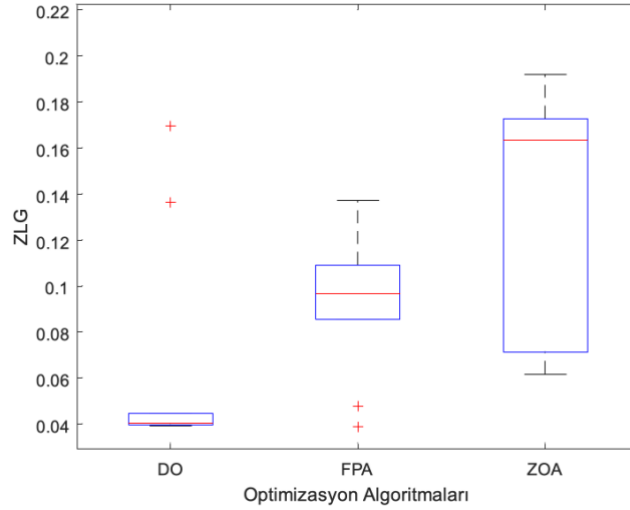
Şekil 12. FOPID-ZLG için sistemin geçici yanıtı.

Şekil 12’ de FPA ve ZOA algoritmasının çıkış sinyali referansa otururken DO algoritması referansa ulaşmayı başaramamış diğer bir deyişle kararlı bir sonuç ortaya koyamamıştır. Şekil 13’te algoritmaların en iyi sonuçları elde ettiği yakınsama eğrileri verilmiştir.



Şekil 13. FOPID-ZLG için yakınsama eğrisi.

Şekil 13'te, DO algoritması 4000 fonksiyon değerlendirme sayısından önce minimum değere ulaşmış fakat bu değerle kararlı bir sistem yanıtı elde edilememektedir. ZOA algoritması 5000 fonksiyon değerlendirme sayısında ulaşırken, FPA algoritması 8000 fonksiyon değerlendirme sayısından önce minimum değeri yakalamıştır. Şekil 14'te üç algoritmanın kutu grafiği verilmiştir.



Şekil 14. FOPID-ZLG için kutu grafiği.

Şekil 14 incelendiğinde, bu problem özelinde DO algoritmasının daha kararlı sonuçlar ortaya koyduğu ama bulduğu bu sonuçların kontrol sinyali için uygun yanıtlar sağlamadığı ortaya koyulmaktadır.

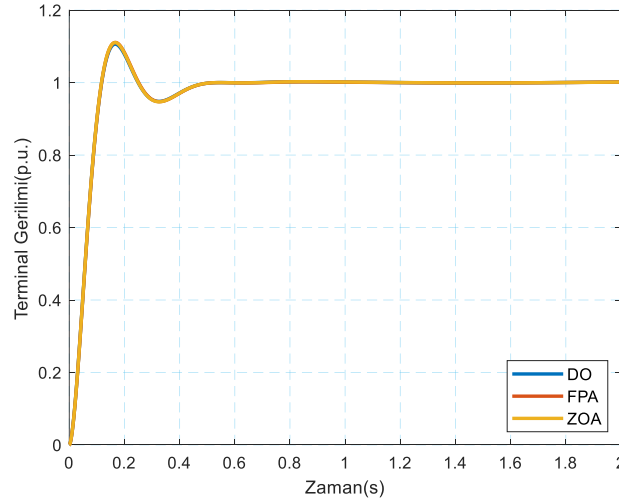
3.2.2. Algoritmaların FOPID-ITAE için Performansı

Tablo 9’ da FOPID-ITAE için algoritmalarla elde edilen en iyi değerler ve denetleyici parametreleri verilmiştir.

Tablo 9. FOPID-ITAE için zaman bölge analizi ve denetleyici parametreleri.

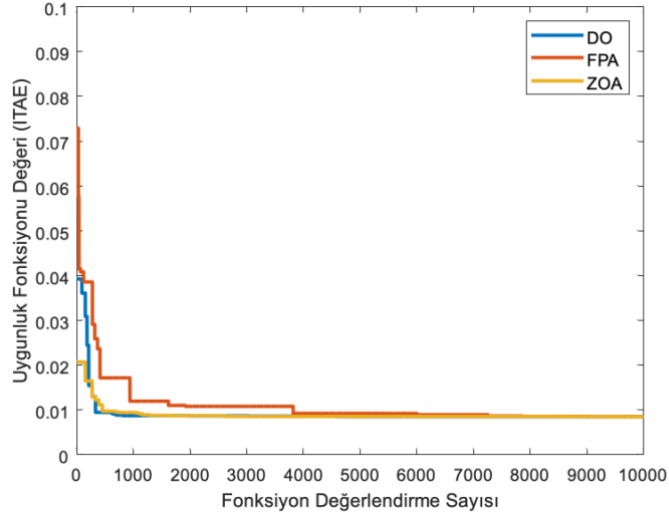
Algoritmalar	En İyi Değer	K_p	K_i	K_d	λ	μ	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,0085	3,0000	0,9537	0,6388	1,4155	1,3013	0,420	0,081	10,596
FPA	0,0085	3,0000	0,9573	0,6460	1,4072	1,2959	0,422	0,081	11,083
ZOA	0,0085	3,0000	0,9683	0,6475	1,3915	1,2970	0,421	0,081	11,040

Tablo 9’ da en iyi yerleşme zamanını ve aşımı DO algoritması elde etmiştir. Tablodaki sonuçlar doğrultusunda elde edilen değerlerin birbirine yakın olduğu görülmektedir. Şekil 15’te FOPID-ITAE için sistemin geçici yanıtı verilmiştir.



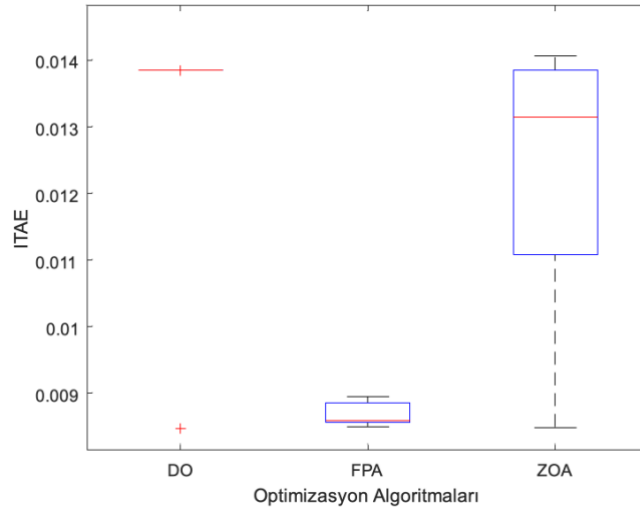
Şekil 15. FOPID-ITAE için sistemin geçici yanıtı.

Tablo 9’da da görüldüğü gibi Şekil 15’te üç algoritma ile benzer sistem yanıtları elde edilmiştir. Şekil 16’da algoritmaların en iyi sonuçları elde ettiği yakınsama eğrileri verilmiştir.



Şekil 16. FOPID-ITAE için yakınsama eğrisi.

Şekil 16' da DO ve ZOA algoritmasının 1000 fonksiyon değerlendirme sayısından önce minimum değere ulaştığı görülmektedir. FPA algoritması ise 4000 fonksiyon değerlendirme sayısında minimum değere yakınsamaktadır. Şekil 17'de üç algoritmanın kutu grafiği verilmiştir.



Şekil 17. FOPID-ITAE için kutu grafiği.

Kutu grafiği incelendiğinde, en iyi değerler sırasıyla DO, FPA ve ZOA olduğu görülmektedir. DO algoritması ile elde edilen aykırı değer olarak yorumlanan değer en iyi sonucu elde ettiği fakat algoritmanın diğer iki algoritmadan da bu problem bazında başarısız olduğu bu problem için en başarılı diğer bir deyişle en kararlı yanıtları FPA algoritmasının verdiği söylenebilir.

3.2.3. FOPID-ZLG ve FOPID-ITAE için İstatistiksel Analiz

FOPID denetleyicisinde 10 kez çalıştırılan algoritmaların hem ZLG hem de ITAE için istatistiksel analizi Tablo 10'da gösterilmektedir.

Tablo 10. FOPID-ZLG ve FOPID-ITAE için istatistiksel analiz.

Algoritma	ZLG			ITAE		
	En iyi değer	Ortalama	Standart Sapma	En iyi değer	Ortalama	Standart Sapma
DO	0,0394	0,0631	0,0480	0,0085	0,0133	0,0017
FPA	0,0388	0,0940	0,0317	0,0085	0,0087	0,0002
ZOA	0,0617	0,1369	0,0523	0,0085	0,0123	0,0020

Tablo 10'da görüldüğü üzere, ZLG ve ITAE amaç fonksiyonlarında en iyi ortalama değerini DO algoritması elde etmiştir. Standart sapma değerlerine göre hem ZLG hem de ITAE amaç fonksiyonunda en iyi değeri FPA algoritması sağlamıştır.

3.3. FOPIDD Denetleyicisi için Algoritmaların Performansı

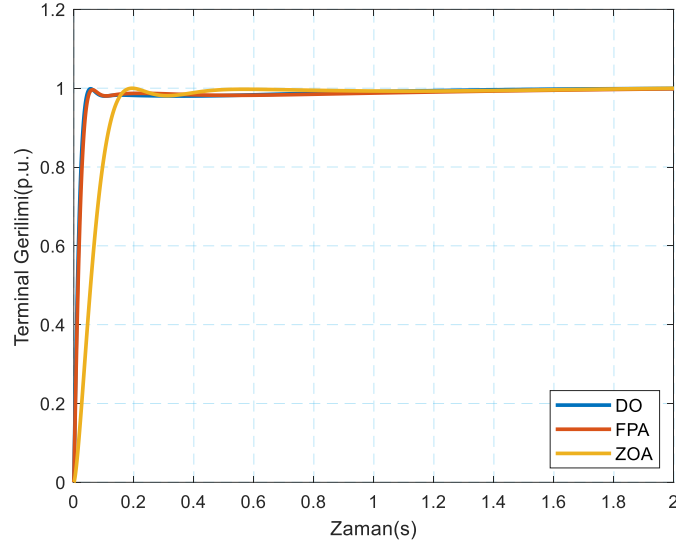
3.3.1. Algoritmaların FOPIDD-ZLG için Performansı

Tablo 11' de FOPIDD-ZLG için en iyi değerler ve denetleyici parametreleri verilmiştir.

Tablo 11. FOPIDD-ZLG için zaman bölge analizi ve denetleyici parametreleri.

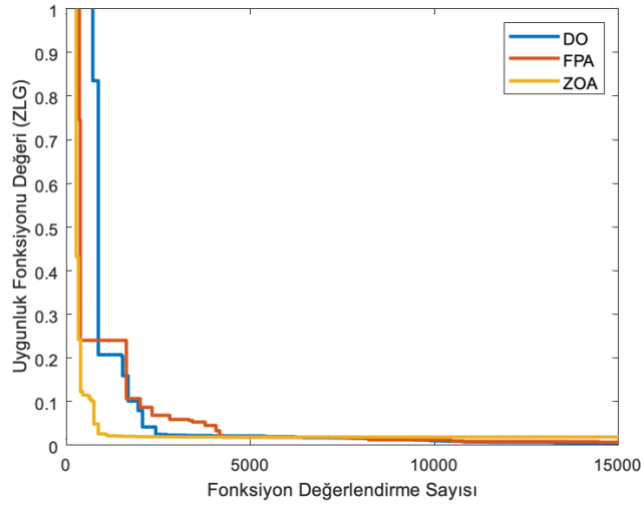
Algoritmalar	En İyi Değer	K_p	K_i	K_{d_1}	K_{d_2}	λ	μ_1	μ_2	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,0053	2,4560	3,0	2,9538	0,2134	0,7747	0,9204	2,0113	0,042	0,029	0
FPA	0,0066	2,9696	2,1162	2,8885	0,2360	1,0435	0,8567	1,9629	0,048	0,033	0
ZOA	0,0186	1,2224	1,1261	1,2481	0,3083	1,0886	0,4285	1,5197	0,153	0,103	0

Tablo 11' de en iyi yerleşme ve yükselme zamanını DO algoritması elde ettiği açıkça görülmektedir. DO, FPA ve ZOA algoritmalarında aşım görülmemektedir. En iyi ZLG değeri DO algoritması elde etmiştir. Şekil 18'de FOPIDD-ZLG için sistemin geçici yanıtı paylaşılmaktadır.



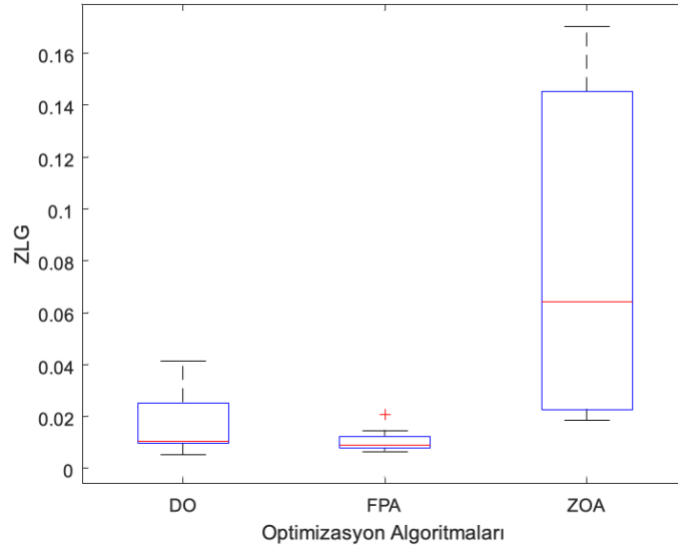
Şekil 18. FOPIDD-ZLG için sistemin geçici yanıtı.

Şekil 18 incelendiğinde DO ve FPA algoritmalarıyla elde edilen sistem çıkış sinyallerinin birbirine yakın yanıt verdiği görülmektedir. ZOA algoritmasıyla elde edilen çıkış sinyali ise referansa daha geç oturmuştur. Şekil 19'da algoritmaların en iyi sonuçlarının yakınsama eğrileri verilmiştir.



Şekil 19. FOPIDD-ZLG için yakınsama eğrisi.

Yakınsama eğrisine göre, üç algortmada 5000 fonksiyon değerlendirme sayısından önce minimum değere yakınsamayı başarmıştır. Şekil 20'de üç algoritmanın kutu grafiği verilmiştir.



Şekil 20. FOPIDD-ZLG için kutu grafiği.

Şekil 20' de en az hata değeri aralığına sahip algoritma FPA olurken en fazla hata değeri aralığını ZOA algoritması vermiştir.

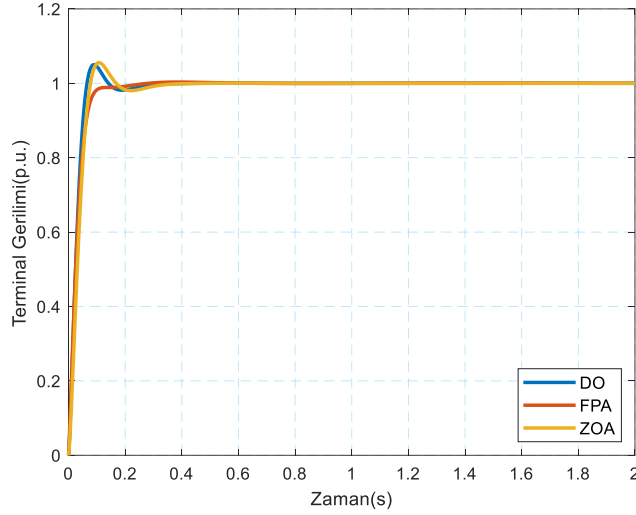
3.3.2. Algoritmaların FOPIDD-ITAE için Performansı

Tablo 12' de FOPIDD-ITAE için en iyi değerler ve denetleyici parametreleri verilmiştir.

Tablo 12. FOPIDD-ITAE için zaman bölge analizi ve denetleyici parametreleri.

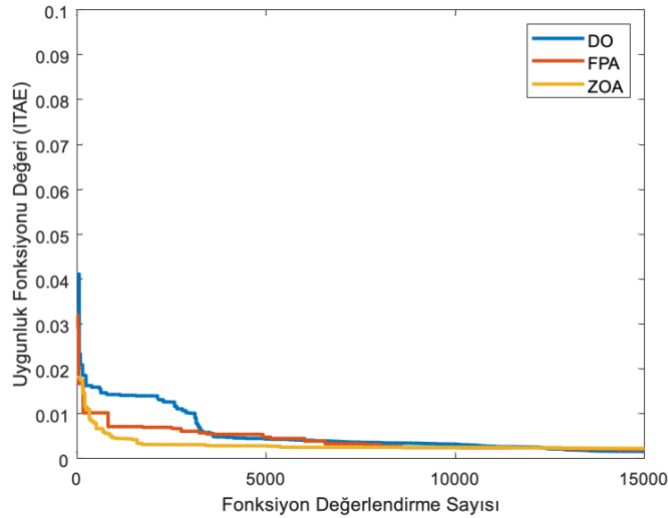
Algoritmalar	En İyi Değer	K_p	K_i	K_{d_1}	K_{d_2}	λ	μ_1	μ_2	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,0016	2,8724	3,0	2,1335	0,3409	0,8889	0,7337	1,7420	0,122	0,045	5,01
FPA	0,0020	2,5597	2,7428	1,8882	0,2123	0,8660	0,7976	1,8579	0,096	0,057	0,33
ZOA	0,0022	2,5035	2,5517	1,8350	0,3514	0,9130	0,6859	1,6730	0,148	0,053	5,55

Tablo 12 incelendiğinde, en iyi aşımı FPA algoritması vermiştir. ITAE için en iyi değeri DO algoritması ile elde edilmiştir. Şekil 21'de FOPIDD-ITAE için sistemin geçici yanıt grafiği verilmiştir.



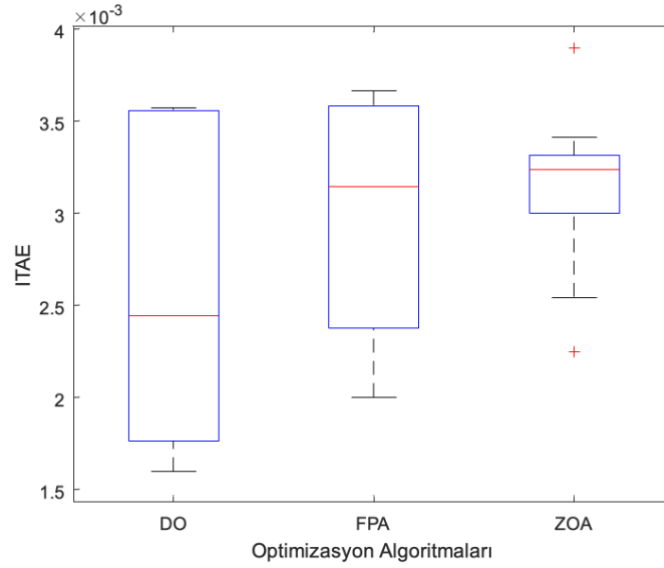
Şekil 21. FOPIDD-ITAE için sistemin geçici yanıtı.

Şekil 21’de verilen geçici yanıtlara göre DO ve ZOA algoritmasının çıkış sinyali benzer yanıt verirken FPA algoritmasının çıkış sinyali daha kısa sürede yerleşerek referansa ulaşmıştır. Şekil 22’de algoritmaların en iyi sonuçları elde ettiği yakınsama eğrileri verilmiştir.



Şekil 22. FOPIDD-ITAE için yakınsama eğrisi.

Yakınsama eğrisine göre, ZOA algoritması 5000 fonksiyon değerlendirme sayısından önce minimum değeri yakalamayı başarmıştır. DO ve FPA algoritması ise 10000 fonksiyon değerlendirme sayısından önce minimum değere yakınsamıştır. Şekil 23’te üç algoritmanın kutu grafiği verilmiştir.



Şekil 23. FOPIDD-ITAE için kutu grafiği.

Şekil 23 incelendiğinde sonuç aralığının sırasıyla DO, FPA ve ZOA algoritmalarına doğru azaldığı görülmektedir. ZOA algoritmasıyla elde edilen en iyi sonucunda bir aykırı değer olduğu ortaya koyulmaktadır.

3.3.3. FOPIDD-ZLG ve FOPID-ITAE için İstatistiksel Analiz

Tablo 13' te her iki amaç fonksiyonu için algoritmaların en iyi değer, ortalama ve standart sapmaları verilmiştir.

Tablo 13. FOPIDD-ZLG ve FOPIDD-ITAE için istatistiksel analiz

Algoritma	ZLG			ITAE		
	En iyi değer	Ortalama	Standart Sapma	En iyi değer	Ortalama	Standart Sapma
DO	0,0053	0,0166	0,0118	0,0016	0,0026	0,0009
FPA	0,0066	0,0108	0,0044	0,0020	0,0030	0,0007
ZOA	0,0186	0,0789	0,0624	0,0022	0,0031	0,0005

Tablo 13'e göre, ZLG amaç fonksiyonunda en iyi ortalama değer FPA algoritması elde ederken, ITAE amaç fonksiyonunda en iyi ortalama değer DO algoritması sağlamıştır. Standart sapmalar incelendiğinde ise, ZLG amaç fonksiyonunda en iyi standart sapma değerini FPA algoritması elde ederken, ITAE amaç fonksiyonunda en iyi standart sapma değerini ZOA algoritması sağlamıştır.

4. Sonuçlar ve Öneriler

Bu çalışmada DO, FPA ve ZOA algoritmaları ele alınarak bu algoritmaların aynı koşullar altında farklı durumlarda nasıl performans sergiledikleri değerlendirilmiştir. Algoritmaların performansı, PID, FOPID ve FOPIDD denetleyicileri ile karşılaştırılmıştır. Ayrıca bu algoritmaların performansları aynı denetleyiciler farklı amaç fonksiyonları kullanılarak da test edilmiştir. Çalışmada elde edilen tüm sonuçlar için zaman bölge analizi yapılmış ve yakınsama grafikleri, kutu grafikleri, farklı istatistiksel sonuçlar ortaya koyulmuştur. Bu çalışmadan elde edilen algoritma performansları ve çıkarılan sonuçlar aşağıda maddeler halinde özetlenmiştir.

- Algoritmaların denetleyici açısından performansının genel olarak değerlendirilmesi: Denetleyiciler kompleks hale geldikçe algoritmaların performanslarının daha belirgin hale geldiği sonucu varılmaktadır. PID denetleyicisi için üç algoritmanın da elde ettiği en iyi sonuçlar birbirine oldukça yakınken FOPID ve FOPIDD denetleyicilerinde bu farkın açıldığı söylenebilir.

- Algoritmaların istatistiksel açıdan performansının genel olarak değerlendirilmesi: ZLG ile ITAE amaç fonksiyonları karşılaştırıldığında algoritmaların ZLG amaç fonksiyonunda elde ettiği sonuçların daha çok saptığı ITAE amaç fonksiyonunda sapma aralığının küçük olduğu daha kararlı yanıtlar elde ettiği gözlemlenmiştir.

- Algoritmaların yakınsama grafikleri açısından performansının genel olarak değerlendirilmesi: PID denetleyicisinden FOPIDD denetleyicisine doğru gittikçe daha fazla iterasyon sayısına ihtiyaç duyulduğu görülmektedir. Bu durum ITAE ile ZLG arasında da açıkça görülmektedir ZLG amaç fonksiyonu daha fazla iterasyona ihtiyaç duymaktadır.

- Algoritmaların performanslarının karşılaştırılması: PID denetleyicisi için ITAE amaç fonksiyonunda en iyi sonucu tüm algoritmalar elde ederken ZLG amaç fonksiyonunda DO algoritması en iyi sonucu elde etmiştir. FOPID denetleyicisi için en iyi sonucu ITAE amaç fonksiyonunda üç algoritma da elde etmişken ZLG amaç fonksiyonunda en iyi sonucu FPA algoritması elde etmiştir. FOPIDD denetleyicisi için hem ITAE amaç fonksiyonunda hem de ZLG amaç fonksiyonunda en iyi sonucu DO algoritması elde etmiştir.

Elde edilen sonuçları daha sağlıklı karşılaştırabilmek için literatürdeki bazı çalışmalar ile karşılaştırmak gerekir. ZLG amaç fonksiyonu için Tablo 14'te literatürdeki PID, FOPID ve FOPIDD denetleyicileri için yerleşme, yükselme zamanı ve aşımalar karşılaştırılmıştır.

Tablo 14. PID, FOPID ve FOPIDD denetleyicileri için literatür karşılaştırması

Referans	PID			FOPID			FOPIDD		
	t_s (%2) (s)	t_r (s)	M_p (%)	t_s (%2) (s)	t_r (s)	M_p (%)	t_s (%2) (s)	t_r (s)	M_p (%)
DO	0,498	0,321	0	-	-	-	0,042	0,029	0
FPA	0,581	0,369	0	0,239	0,157	0	0,048	0,033	0
ZOA	0,509	0,328	0	0,379	0,245	0	0,153	0,103	0
(Can ve ark., 2023)	0,456	0,301	0,918	0,220	0,150	1,293	0,081	0,049	0,129
(Tabak, 2021b)	0,507	0,326	0,002	0,349	0,107	1,029	0,090	0,055	0,169
(Moschos ve ark., 2022)	0,448	0,295	1,000	0,460	0,144	0,185	0,067	0,045	0

Tablo 14 incelendiğinde literatürdeki bazı çalışmalar yerleşme ve yükselme zaman parametrelerinde ön planda olmasına rağmen referansa otururken aşım meydana gelmiştir. DO algoritması ise üç denetleyicide de aşım olmadan referansa oturmuştur. FOPIDD denetleyicisinde ise literatürdeki çalışmalardan çok daha başarılı sonuç vermiştir.

Sonuç olarak her problem için üç algoritmadan birinin doğrudan en iyi olarak değerlendirilemeyeceği fakat genel olarak DO algoritmasının elde ettiği en iyi sonuçlar ve kararlı yanıtlar bakımından diğer algoritmalarından bir adım önde olduğu söylenebilir.

Yazarların Katkısı

Tüm yazarlar çalışmaya eşit katkıda bulunmuştur.

Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

Kaynaklar

- Altbawi, S. M. A., Mokhtar, A. S. Bin, Jumani, T. A., Khan, I., Hamadneh, N. N., & Khan, A. (2024). Optimal design of Fractional order PID controller based Automatic voltage regulator system using gradient-based optimization algorithm. *Journal of King Saud University - Engineering Sciences*, 36(1). <https://doi.org/10.1016/j.jksues.2021.07.009>
- Ayas, M. S., & Sahin, E. (2021). FOPID controller with fractional filter for an automatic voltage regulator. *Computers and Electrical Engineering*, 90. <https://doi.org/10.1016/j.compeleceng.2020.106895>
- Bhookya, J., & Jatoh, R. K. (2020). Improved Jaya algorithm-based FOPID/PID for AVR system. *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 39(4). <https://doi.org/10.1108/COMPEL-08-2019-0319>

- Bhullar, A. K., Kaur, R., & Sondhi, S. (2020). Enhanced crow search algorithm for AVR optimization. *Soft Computing*, 24(16). <https://doi.org/10.1007/s00500-019-04640-w>
- 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). <https://doi.org/10.1016/j.jfranklin.2018.05.056>
- Blondin, M. J., Sicard, P., & Pardalos, P. M. (2019). Controller Tuning Approach with robustness, stability and dynamic criteria for the original AVR System. *Mathematics and Computers in Simulation*, 163. <https://doi.org/10.1016/j.matcom.2019.02.019>
- Can, Ö., Andiç, C., Ekinçi, S., & Izci, D. (2023). Enhancing transient response performance of automatic voltage regulator system by using a novel control design strategy. *Electrical Engineering*, 105(4). <https://doi.org/10.1007/s00202-023-01777-8>
- Cavdar, B., Dincer, K., Baslik, S., Sahin, E. , & Nuroglu, F. M. (2023). A novel objective function design and detailed analysis for the AVR-LFC system. *Springer India*, 4(48), 229. <https://doi.org/10.1007/s12046-023-02292-zS>
- Çavdar, B., Şahin, E., Akyazı, Ö., & Nuroğlu, F. M. (2023). A novel optimal $PI\lambda I\lambda 2D\mu 1D\mu 2$ controller using mayfly optimization algorithm for automatic voltage regulator system. *Neural Computing and Applications*, 35(27). <https://doi.org/10.1007/s00521-023-08834-0>
- Çelik, E. (2021). Design of new fractional order PI–fractional order PD cascade controller through dragonfly search algorithm for advanced load frequency control of power systems. *Soft Computing*, 25(2). <https://doi.org/10.1007/s00500-020-05215-w>
- Danell, K. , Bergström, R. , & Duncon, P. ,& P. J. (2006). Large herbivore ecology ecosystem dynamics and conservation. *Choice Reviews Online*, 44(04). <https://doi.org/10.5860/choice.44-2102>
- Dhanasekaran, B., Siddhan, S., & Kaliannan, J. (2020). Ant colony optimization technique tuned controller for frequency regulation of single area nuclear power generating system. *Microprocessors and Microsystems*, 73. <https://doi.org/10.1016/j.micpro.2019.102953>
- Dogruer, T., & Can, M. S. (2022). Design and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm. *Transactions of the Institute of Measurement and Control*, 44(9). <https://doi.org/10.1177/01423312211066758>
- Ekinçi, S., & Hekimoglu, B. (2019). Improved Kidney-Inspired Algorithm Approach for Tuning of PID Controller in AVR System. *IEEE Access*, 7. <https://doi.org/10.1109/ACCESS.2019.2906980>
- El-Deen, A. T., Hakim Mahmoud, A. A., & El-Sawi, A. R. (2015). Optimal PID tuning for DC motor speed controller based on genetic algorithm. *International Review of Automatic Control*, 8(1). <https://doi.org/10.15866/ireaco.v8i1.4839>
- Eltag, K., & Zhang, B. (2021). Design Robust Self-tuning FPIDF Controller for AVR System. *International Journal of Control, Automation and Systems*, 19(2). <https://doi.org/10.1007/s12555-019-1071-8>
- Gaig, Z. L. (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion*, 19(2). <https://doi.org/10.1109/TEC.2003.821821>
- Ghosh, A., Ray, A. K., Nurujjaman, M., & Jamshidi, M. (2021). Voltage and frequency control in conventional and PV integrated power systems by a particle swarm optimized Ziegler–Nichols based PID controller. *SN Applied Sciences*, 3(3). <https://doi.org/10.1007/s42452-021-04327-8>
- Gozde, H. (2020). Robust 2DOF state-feedback PI-controller based on meta-heuristic optimization for automatic voltage regulation system. *ISA Transactions*, 98. <https://doi.org/10.1016/j.isatra.2019.08.056>
- Güvenç, U., Yiğit, T., Işık, A. H., & Akkaya, I. (2016). Performance analysis of biogeography-based optimization for automatic voltage regulator system. *Turkish Journal of Electrical Engineering and Computer Sciences*, 24(3). <https://doi.org/10.3906/elk-1311-111>
- Jumani, T. A., Mustafa, M. W., Hussain, Z., Md. Rasid, M., Saeed, M. S., Memon, M. M., Khan, I., & Nisar, K. S. (2020). Jaya optimization algorithm for transient response and stability enhancement of a fractional-order PID based automatic voltage regulator system. *Alexandria Engineering Journal*, 59(4). <https://doi.org/10.1016/j.aej.2020.03.005>
- Khan, I. A., Alghamdi, A. S., Jumani, T. A., Alamgir, A., Awan, A. B., & Khidrani, A. (2019). Salp swarm optimization algorithm-based fractional order pid controller for dynamic response and stability enhancement of an automatic voltage regulator system. *Electronics (Switzerland)*, 8(12). <https://doi.org/10.3390/electronics8121472>
- Kose, E. (2020). Optimal Control of AVR System with Tree Seed Algorithm-Based PID Controller. *IEEE Access*, 8. <https://doi.org/10.1109/ACCESS.2020.2993628>
- Kundur, P., Paserba, J., Ajarapu, V., Andersson, G., Bose, A., Canizares, C., Hatziargyriou, N., Hill, D., Stankovic, A., Taylor, C., Van Cutsem, T., & Vittal, V. (2004). Definition and classification of power

- system stability. *IEEE Transactions on Power Systems*, 19(3). <https://doi.org/10.1109/TPWRS.2004.825981>
- Micev, M., Čalasan, M., & Oliva, D. (2020). Fractional order PID controller design for an AVR system using Chaotic Yellow Saddle Goatfish Algorithm. *Mathematics*, 8(7). <https://doi.org/10.3390/math8071182>
- Mosaad, A. M., Attia, M. A., & Abdelaziz, A. Y. (2018). Comparative Performance Analysis of AVR Controllers Using Modern Optimization Techniques. *Electric Power Components and Systems*, 46(19–20). <https://doi.org/10.1080/15325008.2018.1532471>
- Moschos, I., & Parissos, C. (2022). A novel optimal PI λ DND $2N_2$ controller using coyote optimization algorithm for an AVR system. *Engineering Science and Technology, an International Journal*, 26. <https://doi.org/10.1016/j.jestch.2021.04.010>
- Oustaloup, A., Levron, F., Mathieu, B., & Nanot, F. M. (2000). Frequency-band complex noninteger differentiator: Characterization and synthesis. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(1). <https://doi.org/10.1109/81.817385>
- Paliwal, N., Srivastava, L., & Pandit, M. (2021). Equilibrium optimizer tuned novel FOPID-DN controller for automatic voltage regulator system. *International Transactions on Electrical Energy Systems*, 31(8). <https://doi.org/10.1002/2050-7038.12930>
- Saadat, H. (1999). *Power system analysis* (Vol. 2).
- Sahib, M. A. (2015a). A novel optimal PID plus second order derivative controller for AVR system. *Engineering Science and Technology, an International Journal*, 18(2). <https://doi.org/10.1016/j.jestch.2014.11.006>
- Sahib, M. A. (2015b). Engineering Science and Technology , an International Journal A novel optimal PID plus second order derivative controller for AVR system. *Engineering Science and Technology, an International Journal*, 18(2).
- Sikander, A., & Thakur, P. (2020). A new control design strategy for automatic voltage regulator in power system. *ISA Transactions*, 100. <https://doi.org/10.1016/j.isatra.2019.11.031>
- Tabak, A. (2021a). A novel fractional order PID plus derivative (PI λ D μ D μ_2) controller for AVR system using equilibrium optimizer. *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*. <https://doi.org/10.1108/COMPEL-02-2021-0044>
- Tabak, A. (2021b). Maiden application of fractional order PID plus second order derivative controller in automatic voltage regulator. *International Transactions on Electrical Energy Systems*, 31(12). <https://doi.org/10.1002/2050-7038.13211>
- Trojovska, E., Dehghani, M., & Trojovsky, P. (2022). Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2022.3172789>
- Wilson, A. M., Hubel, T. Y., Wilshin, S. D., Lowe, J. C., Lorenc, M., Dewhirst, O. P., Bartlam-Brooks, H. L. A., Diack, R., Bennitt, E., Golabek, K. A., McNutt, J. W., Curtin, N. A., & West, T. G. (2018). Biomechanics of predator-prey arms race in lion, zebra, cheetah and impala. *Nature*, 554(7691). <https://doi.org/10.1038/nature25479>
- Xue, D., Zhao, C., & Chen, Y. Q. (2006). A modified approximation method of fractional order system. 2006 *IEEE International Conference on Mechatronics and Automation, ICMA 2006*, 2006. <https://doi.org/10.1109/ICMA.2006.257769>
- Yang, X. S. (2012). Flower pollination algorithm for global optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7445 LNCS. https://doi.org/10.1007/978-3-642-32894-7_27
- Zeng, G. Q., Chen, J., Dai, Y. X., Li, L. M., Zheng, C. W., & Chen, M. R. (2015). Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. *Neurocomputing*, 160. <https://doi.org/10.1016/j.neucom.2015.02.051>
- Zhao, S., Zhang, T., Ma, S., & Chen, M. (2022). Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence*, 114. <https://doi.org/10.1016/j.engappai.2022.105075>
- Zhou, G., Li, J., Tang, Z., Luo, Q., & Zhou, Y. (2020). An improved spotted hyena optimizer for PID parameters in an AVR system. *Mathematical Biosciences and Engineering*, 17(4). <https://doi.org/10.3934/MBE.2020211>