



# Uygunluk Mesafe Dengesi Tabanlı Yapay Ekosistem Optimizasyon Algoritması ile Otomatik Gerilim Regülatörü Sistemi için Denetleyici Tasarımı

## Controller Design for Automatic Voltage Regulator System with Fitness Distance Balance Based Artificial Ecosystem Optimization Algorithm

<sup>1</sup>Muhammet DEMİRBAŞ , <sup>2</sup>Serhat DUMAN , <sup>3</sup>M. Kenan DÖŞOĞLU 

<sup>1</sup>Kastamonu Üniversitesi, Tosya Meslek Yüksekokulu, Tosya/Kastamonu, Türkiye

<sup>2</sup>Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bandırma/Balıkesir, Türkiye

<sup>3</sup>Düzce Üniversitesi, Mühendislik Fakültesi, Merkez/Düzce, Türkiye

<sup>1</sup>muhammetdemirbas@kastamonu.edu.tr, <sup>2</sup>sduman@bandirma.edu.tr

<sup>3</sup>kenandosoglu@duzce.edu.tr

Araştırma Makalesi/Research Article

### ARTICLE INFO

#### Article history

Received : 30 March 2023

Accepted : 2 May 2023

#### Keywords:

Automatic Voltage Regulator (AVR), Fitness Distance Balance (FDB), Artificial Ecosystem based Optimization (AEO), FOPID

### ABSTRACT

Devices used in power systems are designed to operate at their rated voltages. Variations in the terminal voltages of synchronous generators can shorten the life of these devices. To eliminate these variations, synchronous generators are equipped with Automatic Voltage Regulator (AVR) systems that regulate the excitation currents. The design of controllers used in AVR systems is an important issue for system performance. In this study, PID-AVR and FOPID-AVR systems were designed using the Artificial Ecosystem based Optimization (AEO) algorithm developed with the Fitness Distance Balance (FDB) method. The designed systems were compared with similar studies in the literature in terms of step response and frequency response. The analysis results showed the superiority of the performances of the PID and FOPID controller AVR systems designed with the FDB-AEO algorithm.

© 2023 Bandırma Onyedi Eylül University, Faculty of Engineering and Natural Science. Published by Dergi Park. All rights reserved.

### MAKALE BİLGİSİ

#### Makale Tarihleri

Gönderim : 30 Mart 2023

Kabul : 2 Mayıs 2023

#### Anahtar Kelimeler:

Otomatik Gerilim Regülatörü (OGR), Uygunluk Mesafe Dengesi (FDB), Yapay Ekosistem Tabanlı Optimizasyon (AEO), FOPID

### ÖZET

Güç sistemlerinde kullanılan cihazlar nominal gerilimlerinde çalışmak üzere tasarlanmıştır. Senkron generatörlerin terminal gerilimlerinde yaşanan değişimler bu cihazların ömürlerinin kılmasına neden olabilir. Bu değişimleri ortadan kaldırmak için senkron generatörler uyarım akımlarını düzenleyen Otomatik Gerilim Regülatörü (OGR) sistemleri ile donatılmaktadır. OGR sistemlerinde kullanılan denetleyicilerin tasarımı sistem performansı için önemli bir konudur. Bu çalışmada Uygunluk Mesafe Dengesi (FDB) yöntemi ile geliştirilmiş Yapay Ekosistem Tabanlı Optimizasyon (AEO) algoritması kullanılarak PID-OGR ve FOPID-OGR sistemlerinin tasarımı yapılmıştır. Tasarımı yapılan sistemler adım yanıtı ve frekans yanıtı açısından literatürde daha önce yapılmış olan benzer çalışmalarla karşılaştırılmıştır. Analiz sonuçları FDB-AEO algoritması ile tasarlanan PID ve FOPID denetleyicili OGR sistemlerinin performanslarındaki üstünlüğü göstermiştir.

© 2023 Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi. Dergi Park tarafından yayınlanmaktadır. Tüm Hakları Saklıdır.

ORCID ID: <sup>1</sup>0000-0002-5223-1279

<sup>2</sup>0000-0002-1091-125X

<sup>3</sup>0000-0001-8804-7070

## 1. GİRİŞ

Güç sistemlerinde kullanılan cihazlar nominal gerilimlerinde çalışmak için tasarlanmaktadır. Sistemde meydana gelen gerilim dalgalanmaları bu cihazların ömürlerinin azalmasına neden olabilir. Gerilimde meydana gelen değişiklikler reaktif güç talebiyle doğrudan ilişkilidir. Reaktif güç talebinin karşılanamadığı durumlarda gerilimde çökmeler meydana gelebilir. İletim ve Dağıtım sistemlerinde reaktif güç talebi reaktif güç kompanzasyon elemanları ile karşılanırken, üretim bölgelerinde, enerji üretimini sağlayan senkron generatörlerin Otomatik Gerilim Regülatörleri (OGR'ler) ile kullanılması ile karşılanmaktadır. OGR'ler beraber kullanıldıkları senkron generatörlerin terminal gerilimlerini ölçer. Ölçtükları bu gerilim değerini referans gerilim ile karşılaştırarak hata sinyali üretir. Hata sinyali bir denetleyici vasıtasıyla işlenir ve elde edilen sonuca göre senkron generatörün uyarım akımı değiştirilerek gerilim kontrol edilmiş olur [1-2]. OGR sistemlerinde en çok kullanılan denetleyici modeli basit kontrol yapısı nedeniyle Oransal-İntegral-Türevsel (PID)'dir [3-9]. Ancak sistemin dinamik yanıtını iyileştirmek amacıyla araştırmacılar kesir dereceli PID (FOPID) denetleyicisini önermişlerdir [10-15]. FOPID denetleyicisi klasik PID denetleyicisine ek olarak iki adet ( $\lambda$  ve  $\mu$ ) kontrol parametresi içerir. Bu durum FOPID'nin PID'ye göre daha esnek ayarlanabilmesine olanak sağlar. Ancak sahip olduğu ek kontrol parametreleri nedeniyle PID'ye göre daha karmaşık bir yapıya sahiptir. Bu nedenle kontrol parametrelerinin ayarlanma süreci de daha karmaşıktır.

OGR sistemlerinde kullanılan denetleyicilerin kontrol parametrelerinin ayarlanması araştırmacıların karşılaştıkları önemli bir konudur. Kontrol parametrelerinin belirlenebilmesi için Ziegler-Nichols [16-17], Root-Locus [18] gibi klasik yöntemler kullanılmış olsa da, son yıllarda metasezgisel optimizasyon algoritmalarının da gelişmesiyle kontrol parametresi belirleme işlemi bu algoritmalar ile yapılmaya başlanmıştır. Parçacık Sürüsü Optimizasyonu (PSO) [19], Yerçekimi Arama Algoritması (GWO) [20], Stokastik Fraktal Arama (SFS) [21], Simbiyotik Organizma Arama (SOS) [22], Yapay Ekosistem Tabanlı Optimizasyon (AEO) [23], Sinüs Cosinüs Algoritması [24] bu amaçla kullanılan metasezgisel algoritmalarından bazılarıdır. Bu optimizasyon algoritmalarının hepsi senkron generatör terminal geriliminin yükselme zamanı, yerleşme zamanı, en büyük yüzde aşım gibi adım yanıtı özelliklerini iyileştirmeyi amaçlamaktadır. Ancak algoritmaların hiçbiri en iyi adım yanıtı sonucunu vermeyi garanti etmez. Bu nedenle farklı algoritmalarla OGR sisteminin tasarımı araştırmacılar için önemli bir çalışma konusunu oluşturmaktadır.

Metasezgisel optimizasyon algoritmaları en iyi adım yanıtını veren kontrol parametrelerini çeşitli amaç fonksiyonları kullanarak aramaktadır. Çalışmalarda kullanılan bazı amaç fonksiyonları, Hatanın Mutlak Değerinin İntegrali (IAE) [5,20,25], Hatanın Karesinin İntegrali (ISE) [20,25,26], Zamanın İntegrali ile Çarpılan Mutlak Hata (ITAE) [27-29], Zaman Ağırlıklı Karesel Hatanın İntegrali (ITSE)'dir [21,30,31]. Kullanılan amaç fonksiyonu algoritmanın performansını büyük ölçüde etkilemektedir. Bu nedenle bazı araştırmacılar amaç fonksiyonu geliştirme çalışmalarına da odaklanmıştır [32-34].

Araştırmacılar tarafından geliştirilen metasezgisel optimizasyon algoritmalarının birçoğunun ortak problemi çözüm adayına erken yakınsayarak en uygun çözüme ulaşamamalarıdır. Bu sorunu ortadan kaldırmak amacıyla 2020 yılında mevcut optimizasyon algoritmaların seçim yöntemi üzerinde geliştirme yapmak için Uygunluk Mesafe Dengesi (FDB) yöntemini önerilmiştir. Önerilen yöntem ilk olarak SOS algoritması üzerinde uygulanmıştır [35]. FDB yönteminin etkisinin gözlemlenmesi yöntemin farklı algoritmalara uygulanmasına neden olmuştur. FDB-SFS [36] ve FDB-AEO [37] bu yöntemle geliştirilen algoritmaların bazılarıdır.

Açıklanan bilgiler değerlendirildiğinde OGR sistemlerinin en uygun şekilde tasarımının güç sistemlerinin kararlı bir şekilde çalışması açısından önemi açıktır. Bu bağlamda yapılan çalışmanın literatüre olan katkısı aşağıdaki gibi özetlenebilir.

- PID ve FOPID denetleyicilerinin tasarımı OGR sistemi için yapılmıştır.
- 2022 yılında geliştirilen FDB-AEO algoritması [37] ilk kez OGR sistemi tasarımı için kullanılmıştır.
- FDB-AEO algoritması ile OGR sistemi için tasarımı yapılan PID denetleyicisinin performansı Kaotik tabanlı PSO (CPSO) [38], SCA [39] ve SFS [21] algoritmalarının performansları ile karşılaştırılmıştır.
- FDB-AEO algoritması ile OGR sistemi için tasarımı yapılan FOPID denetleyicisinin performansı ise Çoklu Evren Optimizasyon Algoritması (MVO) [40], Salp Sürüsü Optimizasyon Algoritması (SSA) [12] ve Jaya Optimizasyon Algoritması (JOA) [13] algoritmalarının performansları ile karşılaştırılmıştır.
- Tasarımı yapılan sistemler adım yanıtı ve bode analizi ile incelenerek üstün performansları gösterilmiştir.

Bu çalışmanın bölümleri aşağıdaki gibidir.

- Bölüm 2'de FDB-AEO algoritması, OGR sistemi ve amaç fonksiyonu sunulurken çalışmada kullanılan metodlar detaylı bir şekilde açıklanmıştır.
- Bölüm 3'de çalışmada yapılan araştırmaya ve bu araştırmanın sonucunda elde edilen bulgulara yer verilmiştir.
- Son olarak Bölüm 4'de, çalışmada elde edilen sonuçlara yer verilerek değerlendirme ve öneriler sunulmuştur.

## 2. MATERYEL VE METOD

### 2.1. FDB-AEO Algoritması

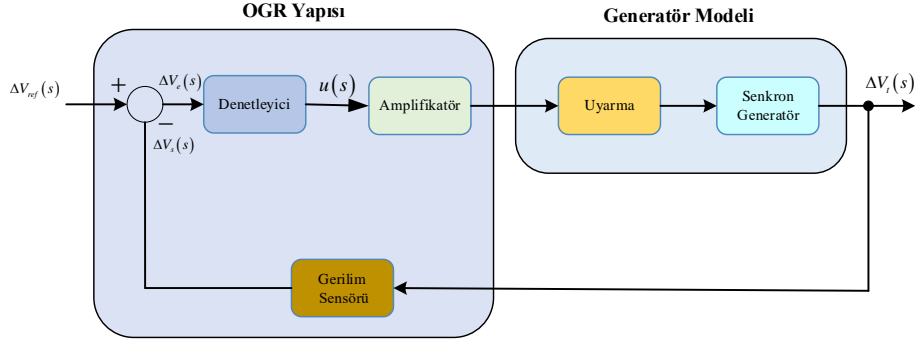
AEO algoritması canlı organizmaların üretim, tüketim ve ayrıştırma özelliklerini taklit edecek şekilde Zhao ve arkadaşları tarafından 2020 yılında geliştirilmiştir [23]. Algoritma oluşturduğu ekosistemde organizmaları üreticiler, tüketiciler ve ayrıştırıcılar olarak gruplandırmaktadır. Üreticiler, ihtiyaç duyduğu besini fotosentez yoluyla elde eden bitkileri içermektedir. Tüketiciler ise ihtiyaç duyduğu enerjiyi diğer tüketici ve üretici gruplarından elde etmek durumunda olan hayvanları içermektedir. Hayvanlar; otobur, etçil ve hem etçil hem otçul olan hepciller olarak gruplara ayrılmaktadır. Ekosistemde bakteri ve mantarlardan oluşan gruba ayrıştırıcılar denir. Bu grup ölen organizmaların kalıntılarını kullanarak su ve mineral gibi besinlere dönüştürür. Üreticiler bu besinleri tekrar kullanarak besin zinciri döngüsünü yeniden başlatır. AEO algoritması, üreticilerden ayrıştırıcılara kadar olan bu besin akışını taklit etmektedir. Ekosistemdeki üretici, tüketici ve ayrıştırıcılar algoritmanın operatörlerini oluşturur. Ekosistemi oluşturan canlı organizmalar çözüm adaylarını oluştururken, bu canlıların enerji düzeyi ise uygunluk değerine karşılık gelmektedir. Çözüm adaylarından bir tanesi üretici, bir tanesi ayrıştırıcı ve kalanları tüketici olarak gruplandırılır. Tüketiciler ise kendi arasında etçil, otçul ve hepcil olarak eşit biçimde ayrılır [23]. AEO algoritmasında en düşük enerji seviyesine veya uygunluk değerine sahip birey ayrıştırıcı olarak kullanılmaktadır. Ancak tek bir ayrıştırıcının kullanılması besin zincirindeki enerji akışını tam olarak yansıtmamaktadır. En düşük enerji düzeyine sahip bireyle birlikte daha yüksek enerji seviyesine sahip bir bireyin kullanılması ayrıştırıcıları dengeleyerek ekolojik dengenin daha iyi temsil edilmesini sağlayabilmektedir [37]. 2020 yılında Kahraman ve arkadaşları tarafından önerilen FDB yöntemi [35], AEO algoritmasının ayrıştırma sürecini iyileştirmek amacıyla kullanılmıştır [37]. Yapılan geliştirmede AEO algoritmasının ayrıştırma sürecinde kullanılan bireye ek olarak, arama sürecine katkıda bulunabilecek birey FDB yöntemi ile seçilerek sürece dahil edilmektedir [37]. Geliştirilen FDB-AEO algoritmasının sözde kodu Tablo 1’de verilmiştir.

**Tablo 1.** FDB-AEO algoritmasının sözde kodu [37].

|     |  |
|-----|--|
| 1.  | <b>Başla</b>   |
| 2.  | Ekosistemin oluşturulması: $X_i$ çözümleri rastgele oluşturulur.                           |
| 3.  | <b>for</b> $i=1:n$ <b>do</b>   |
| 4.  | Uygunluk değeri $fit_i$ ve $X_{best}$ değerleri hesaplanır.                                |
| 5.  | <b>end for</b>   |
| 6.  | <b>while</b> (arama süreci yaşam döngüsü <b>do</b> : amaç fonksiyonu değerlendirme sayısı) |
| 7.  | $X_i$ konumu güncellenir. //Üretim aşaması   |
| 8.  | <b>for</b> $i=2:n$ <b>do</b> //Tüketim aşaması   |
| 9.  | <b>If</b> ( $rand < 1/3$ ) $X_i$ konumu otçul olarak güncellenir.                          |
| 10. | <b>Else If</b> ( $1/3 < rand < 2/3$ ) $X_i$ konumu etçil olarak güncellenir.               |
| 11. | <b>Else</b> $X_i$ konumu hepcil olarak güncellenir.  |
| 12. | <b>End If</b>  |
| 13. | <b>End for</b>   |
| 14. | <b>for</b> $i=1:n$ <b>do</b>   |
| 15. | Uygunluk değeri $fit_i$ ve $X_{best}$ değeri güncellenir.                                  |
| 16. | <b>End for</b>   |
| 17. | <b>for</b> $i=1:n$ <b>do</b> //Ayrıştırma aşaması  |
| 18. | Her bireyin mesafesi hesaplanır.   |
| 19. | Her bireyin FDB puanı hesaplanır.  |
| 20. | <b>End for</b>   |
| 21. | $D_p$ ve $S_p$ vektörleri oluşturulur.   |
| 22. | <b>for</b> $i=1:n$ <b>do</b>   |
| 23. | $X_i$ konumu güncellenir.  |
| 24. | Uygunluk değeri $fit_i$ hesaplanır.  |
| 25. | <b>end for</b>   |
| 26. | Bulunan en iyi $X_{best}$ değeri güncellenir.  |
| 27. | <b>End while</b>   |
| 28. | En iyi çözüm getirilir.  |

### 2.2. Otomatik Gerilim Regülatörü (OGR)

Güç sistemlerinde enerji üretimini sağlayan senkron generatörlerin uyartım akımını kontrol ederek terminal gerilimlerinde meydana gelen değişimleri düzenleyen cihazlara Otomatik Gerilim Regülatörleri (OGR) denir. OGR sistemleri Amplifikatör, Uyarma, Senkron Generatör ve Sensör yapıları ile birlikte modellenir. OGR sisteminin blok diyagramı Şekil 1’de verilmiştir.



Şekil 1. OGR sisteminin blok diyagramı.

OGR sisteminin blok diyagramında; referans gerilimi  $\Delta V_{ref}(s)$ , hata gerilimi  $\Delta V_e(s)$ , kontrol sinyali  $u(s)$  ve sensör gerilimi  $\Delta V_s(s)$  ve senkron generatörün terminal gerilimi  $\Delta V_t(s)$  ile gösterilmiştir. OGR sistemini oluşturan Amplifikatör, Uyarma, Senkron Generatör ve Gerilim Sensörü bileşenlerinin transfer fonksiyonları sırasıyla Denklem 1, 2, 3 ve 4'te verilmiştir.

$$G_{Amplifikatör}(s) = \frac{K_a}{1 + sT_a} \quad (1)$$

$$G_{Uyarma}(s) = \frac{K_e}{1 + sT_e} \quad (2)$$

$$G_{Generatör}(s) = \frac{K_g}{1 + sT_g} \quad (3)$$

$$G_{Sensör}(s) = \frac{K_s}{1 + sT_s} \quad (4)$$

Burada,  $K_a$ ,  $K_e$ ,  $K_g$  ve  $K_s$  kazanç sabitleri,  $T_a$ ,  $T_e$ ,  $T_g$  ve  $T_s$  zaman sabitleridir. Bu değerler literatürden yola çıkılarak  $K_a=10$ ,  $K_e=1$ ,  $K_g=1$ ,  $K_s=1$ ,  $T_a=0.1s$ ,  $T_e=0.4s$ ,  $T_g=1s$  ve  $T_s=0.01s$  olarak belirlenmiştir [5-7,23]. Denetleyici modeli içermeyen OGR sisteminin transfer fonksiyonu Denklem 5'de verilmiştir [41].

$$G_{OGR}(s) = \frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1s + 10}{0.0004s^4 + 0.0454s^3 + 0.555s^2 + 1.51s + 11} \quad (5)$$

OGR sistemlerinde en sık kullanılan PID denetleyicisinin  $K_P$ ,  $K_I$  ve  $K_D$  olmak üzere 3 adet kontrol parametresi bulunmaktadır. Bu kontrol parametrelerinin optimizasyon metodları ile belirlenmesi ile tasarımı gerçekleştirilir. PID denetleyicisinin transfer fonksiyonu Denklem 6'da verilmiştir.

$$G_{PID}(s) = K_P + \frac{K_I}{s} + K_D s \quad (6)$$

Burada;  $K_P$  orantısal kazanç,  $K_I$  integral kazancı,  $K_D$  türev kazancıdır. PID denetleyicili OGR sisteminin transfer fonksiyonu Denklem 7'de verilmiştir [41].

$$G_{OGR-PID}(s) = \frac{0.1K_D s^2 + (0.1K_P + 10K_D)s^2 + (0.1K_I + 10K_P)s + 10K_I}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + (1.51 + 10K_D)s^2 + (1 + 10K_P)s + 10K_I} \quad (7)$$

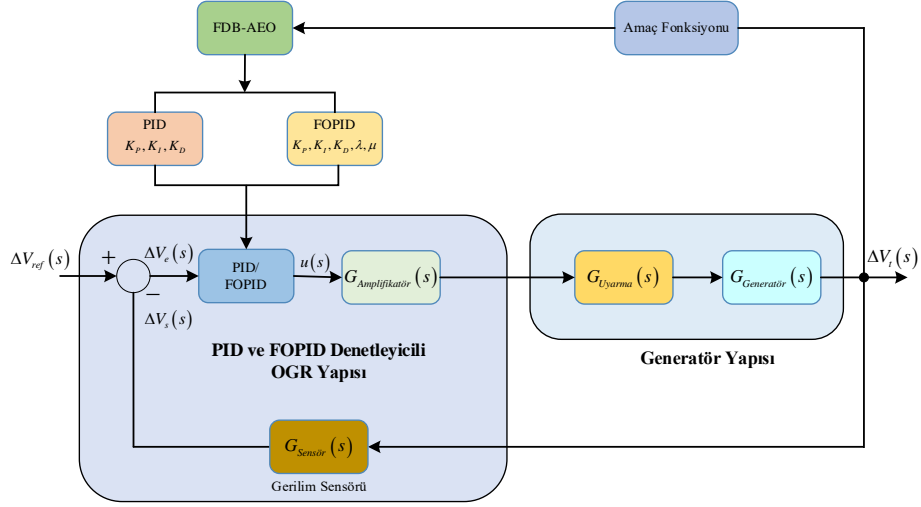
FOPID denetleyicisinin  $K_P$ ,  $K_I$ ,  $K_D$ ,  $\lambda$  ve  $\mu$  olmak üzere 5 adet kontrol parametresi bulunmaktadır. PID denetleyicisinden farklı olarak sahip olduğu iki kesir derecesi parametresi nedeniyle tasarımı daha zordur. Bu 5 parametre optimizasyon metodları ile belirlenerek tasarımı gerçekleştirilir. FOPID denetleyicisinin transfer fonksiyonu Denklem 8'de verilmiştir.

$$G_{FOPID}(s) = K_P + \frac{K_I}{s^\lambda} + K_D s^\mu \quad (8)$$

Burada; PID denetleyicisinden farklı olarak;  $\lambda$  ve  $\mu$  kesir derecelerdir. FOPID denetleyicili OGR sisteminin transfer fonksiyonu Denklem 9'da verilmiştir [42].

$$G_{OGR-FOPID}(s) = \frac{0.1K_D s^{\lambda+\mu+1} + 10K_D s^{\lambda+\mu} + 0.1K_P s^{\lambda+1} + 10K_P s^{\lambda} + 0.1K_I s + 10K_I}{0.0004s^{\lambda+4} + 0.0454s^{\lambda+3} + 0.555s^{\lambda+2} + 1.51s^{\lambda+1} + 10K_D s^{\lambda+\mu} + (10K_P + 1)s^{\lambda} + 10K_I} \quad (9)$$

Kontrol parametreleri FDB-AEO algoritması ile belirlenen PID ve FOPID denetleyicili OGR sisteminin kapalı çevrim transfer fonksiyonu blok diyagramı Şekil 2’de verilmiştir.



Şekil 2. Kontrol parametreleri FDB-AEO algoritması ile ayarlanan PID ve FOPID denetleyicili OGR sisteminin blok diyagramı.

Şekil 2’de gösterilen PID ve FOPID denetleyicili OGR sisteminin kontrol parametreleri FDB-AEO algoritması ile belirlenerek sistemin çıkış yanıtı  $\Delta V_t(s)$  elde edilir. Elde edilen çıkış yanıtının ideal çıkış yanıtı olan  $\Delta V_{ref}(s)$  ile karşılaştırılması ile hata fonksiyonu elde edilir. Hata fonksiyonunun küçüklüğü kontrol edilen sistemin referans modele yakınlığı ile ilişkilidir. Ancak algoritmanın adım yanıtı açısından da sistemin performansını değerlendirmesi gerekmektedir. Bu nedenle Şekil 2’de gösterildiği gibi her bir döngüde algoritma sistemin çıkış yanıtından yola çıkarak amaç fonksiyonunu hesaplar. Sonlandırma kriteri sağlanana kadar bu amaç fonksiyonunun en düşük değerini elde etmeye çalışır. En küçük amaç fonksiyonunun elde edilmesini sağlayan kontrol parametreleri denetleyicilerin kontrol parametresi olarak belirlenmiş olur.

### 2.3. Amaç Fonksiyonu

OGR sisteminde kullanılan denetleyici parametrelerini belirleyen FDB-AEO algoritması da tüm metasezgisel optimizasyon algoritmalarında olduğu gibi bu işlemi senkron generatör terminal geriliminin yükselme zamanı, yerleşme zamanı ve en büyük yüzde aşım değerlerini değerlendirerek yapmaktadır. Bu nedenle minimizasyonu yapılacak olan amaç fonksiyonunun seçimi en uygun çözüm adayına ulaşmak için önemli bir konudur. Literatürde en çok kullanılan amaç fonksiyonlarından olan IAE, ISE, ITAE ve ITSE Denklem 10-13 arasında verilmiştir.

$$IAE = \int_0^t |e(t)| dt \quad (10)$$

$$ISE = \int_0^t |e^2(t)| dt \quad (11)$$

$$ITAE = \int_0^t t |e(t)| dt \quad (12)$$

$$ITSE = \int_0^t t |e^2(t)| dt \quad (13)$$

Burada; t, simülasyon zamanını ve e(t), t zamanında kalıcı durum hatasını ifade etmektedir. Amaç fonksiyonu seçiminin algoritma performansına etkisi araştırmacıları yeni amaç fonksiyonları belirleme konusundaki çalışmaları yapmaya itmiştir. Bu bağlamda Denklem 14’ de verilen amaç fonksiyonunun [34], OGR sistemi tasarımı için oldukça iyi performans sergilediği görülmüştür.

$$AF = W_1.OS + W_2.t_s + W_3.ITAE + W_4.ITSE \quad (14)$$

Burada; OS, en büyük yüzde aşım ve  $t_s$ , yerleşme zamanınıyken,  $W_1$ ,  $W_2$ ,  $W_3$  ve  $W_4$  amaç fonksiyonunda kullanılan bileşenlerin ağırlıklarını ifade eden katsayılarıdır. Bu ağırlık katsayıları sırasıyla 2, 2, 5 ve 5 olarak seçilmiştir.

### 3. ARAŞTIRMA VE BULGULAR

Bu çalışmada PID ve FOPID denetleyicili OGR sistemi tasarımı; Intel Core i7-11370H işlemcili, 3.30GHz CPU ve 64 GB RAM'e sahip bilgisayar ile Matlab 2022a ortamında gerçekleştirilmiştir. FDB algoritmasının ayar parametreleri, ekosistem boyutu: 50 ve amaç fonksiyonu değerlendirme sayısı 200 olarak belirlenmiştir. FDB-AEO algoritması, Denklem 14'de verilen amaç fonksiyonu ile PID ve FOPID denetleyici parametrelerini Tablo 2'de belirtilen sınırlarda aramıştır.

**Tablo 2.** Belirlenen kontrol parametresi sınırları.

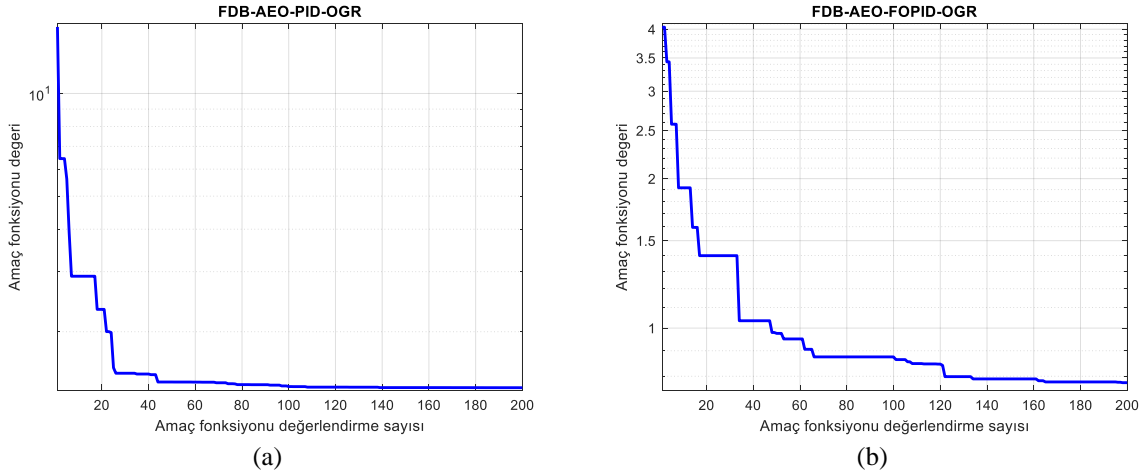
| Denetleyici | Kontrol Parametreleri |          |          |           |        |
|-------------|-----------------------|----------|----------|-----------|--------|
|             | $K_P$                 | $K_I$    | $K_D$    | $\lambda$ | $\mu$  |
| PID         | [0, 2]                | [0, 2]   | [0, 2]   | -         | -      |
| FOPID       | [0.2, 2]              | [0.2, 2] | [0.2, 2] | [1, 2]    | [1, 2] |

Her bir denetleyici için arama 20 döngü boyunca yapılmıştır. Bu 20 döngüde AF'nin en düşük değerini veren kontrol parametreleri denetleyicilerin kontrol parametresi olarak belirlenmiştir. Denetleyiciler için belirlenen kontrol parametreleri Tablo 3'de verilmiştir.

**Tablo 3.** FDB-AEO algoritmasının belirlediği kontrol parametreleri.

| Denetleyici | Kontrol Parametreleri |        |        |           |        |
|-------------|-----------------------|--------|--------|-----------|--------|
|             | $K_P$                 | $K_I$  | $K_D$  | $\lambda$ | $\mu$  |
| PID         | 0.5959                | 0.4024 | 0.1967 | -         | -      |
| FOPID       | 1.4385                | 0.6061 | 0.3023 | 1.1077    | 1.2271 |

En iyi döngülerde AF'nin bulunduğu değer, PID denetleyicisi için 1.3708 iken FOPID denetleyicisi için 0.76106'dır. En iyi döngüde elde edilen yakınsama eğrileri Şekil 3'de verilmiştir.



**Şekil 3.** Amaç Fonksiyonu (AF)'nin yakınsama eğrileri; (a) PID, (b) FOPID.

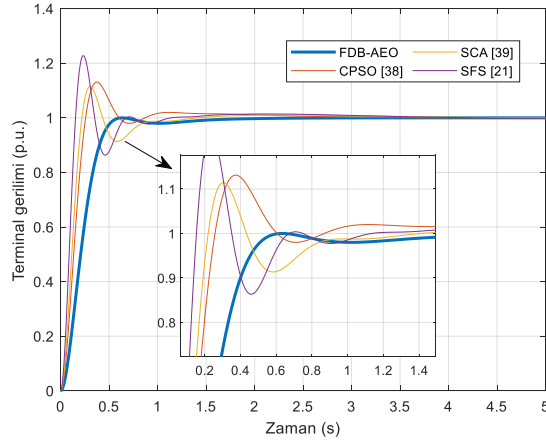
AF'nin 20 döngüde bulunan en düşük değeri, en büyük değeri, ortalama değeri ve standart sapması Tablo 4'de verilmiştir.

**Tablo 4.** AF'nin 20 döngü boyunca elde edilen değerleri.

| Önerilen Yaklaşım | En düşük | En büyük | Ortalama | Standart sapma |
|-------------------|----------|----------|----------|----------------|
| FDB-AEO-PID       | 1.3708   | 0.3711   | 1.3709   | 0.00011234     |
| FDB-AEO-FOPID     | 0.76106  | 0.78241  | 0.77386  | 0.0059817      |

#### 3.1. Adım Yanıtı

FDB-AEO algoritması ile tasarımı yapılan PID-OGR sisteminin adım yanıtı Şekil 4'de ve bu adım yanıtı ile elde edilen veriler Tablo 5'de verilmiştir.

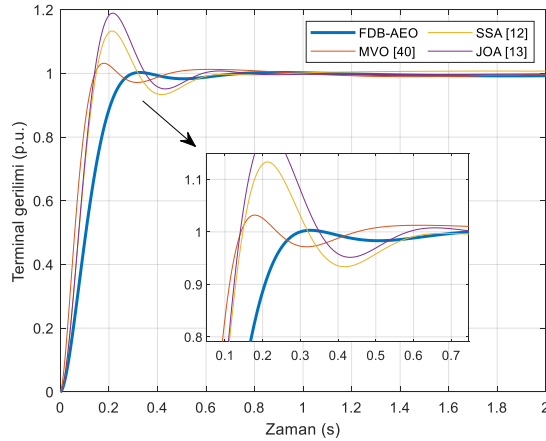


Şekil 4. FDB-AEO algoritması ile tasarımı yapılan PID-OGR sisteminin adım yanıtı.

Tablo 5. FDB-AEO algoritması ile tasarımı yapılan PID-OGR sisteminin adım yanıtı sonuçları.

| Algoritma                          | Tr (0.1 → 0.9) | Ts ( $\pm 2\%$ ) | OS%     |
|------------------------------------|----------------|------------------|---------|
| <b>FDB-AEO (Önerilen yaklaşım)</b> | 0.3279         | 0.5098           | 0.0039  |
| CPSO [38]                          | 0.1743         | 0.5640           | 13.1100 |
| SCA [39]                           | 0.1481         | 0.8415           | 11.4244 |
| SFS [21]                           | 0.1030         | 0.5840           | 22.8000 |

Adım yanıtı sonuçlarına göre FDB-AEO algoritması ile tasarımı yapılan PID-OGR sisteminin 0.5098 saniye yerleşme zamanı ve %0.0039 maksimum yüzde aşım değerlerinin literatürde daha önce önerilen çalışmalara göre daha iyi olduğu görülmektedir. 0.3279 saniye yükselme zamanı ile bu çalışmada önerilen yaklaşımın diğer çalışmaların gerisinde kaldığı görülse de bu durum maksimum yüzde aşım değerinde gösterdiği üstün performanstan dolayı kabul edilebilir. Çünkü güç sistemlerinde kullanılan cihazlar nominal gerilimlerinde çalışmak için tasarlanmışlardır. Literatürde daha önce yapılan çalışmalarda yükselme zamanının düşük olmasının etkisi maksimum yüzde aşım değerinin yüksek olmasına neden olmuştur. Bu durum güç sistemlerinde kullanılan cihazlar için olumsuz etki oluşturabilir. FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sisteminin adım yanıtı Şekil 5'de ve bu adım yanıtı ile elde edilen veriler Tablo 6'da verilmiştir.



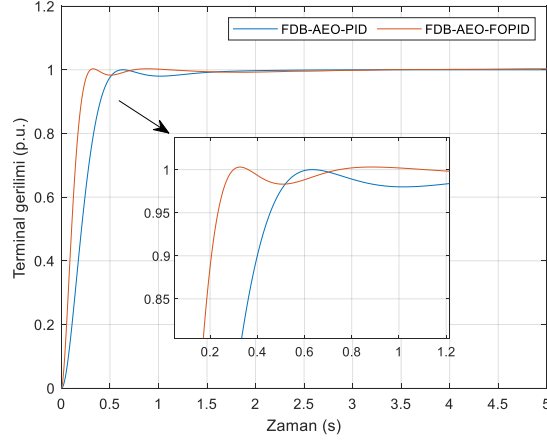
Şekil 5. FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sisteminin adım yanıtı.

Tablo 6. FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sisteminin adım yanıtı sonuçları.

| Algoritma                          | Tr (0.1 → 0.9) | Ts ( $\pm 2\%$ ) | OS%    |
|------------------------------------|----------------|------------------|--------|
| <b>FDB-AEO (Önerilen yaklaşım)</b> | 0.1726         | 0.2627           | 0.1072 |
| MVO [40]                           | 0.1075         | 0.3493           | 1.0295 |
| SSA [12]                           | 0.0981         | 0.551            | 15.5   |
| JOA [13]                           | 0.0827         | 0.453            | 13.2   |

Adım yanıtı sonuçlarından FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sisteminin 0.2627 ile yerleşme zamanının ve %0.1072 ile en büyük yüzde aşım değerinin literatürde önerilen yaklaşımlara göre daha iyi olduğu görülmektedir. Yükselme zamanı açısından MVO [40], SSA [12] ve JOA [13] algoritmalarının sırasıyla 0.1075, 0.0981 ve 0.0827 ile daha iyi sonuç vermesine rağmen en büyük yüzde aşım değerlerinin yüksek olması ve

yerleşme zamanlarının da sırasıyla 0.3493, 0.551 ve 0.453 ile FDB-AEO algoritması ile tasarlanan sistemin gerisinde kalması FDB-AEO algoritmasının üstün performansını göstermektedir. Denetleyici modelleri arasında karşılaştırma yapmak amacıyla tasarımı yapılan ve diğer çalışmalara göre üstün performansları gösterilen PID ve FOPID denetleyicili OGR sistemlerinin adım yanıtları tek bir görsel halinde Şekil 6'da gösterilmiştir. Adım yanıtı sonuçları ise Tablo 7'de verilmiştir.



Şekil 6. Tasarımı yapılan PID ve FOPID denetleyicili OGR sistemlerinin adım yanıtları.

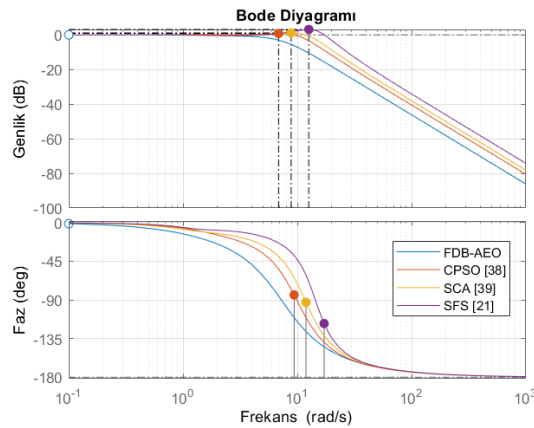
Tablo 7. Tasarımı yapılan PID ve FOPID denetleyicili OGR sistemlerinin adım yanıtı sonuçları

| Önerilen yöntem | Tr (0.1 → 0.9) | Ts ( $\pm 2\%$ ) | OS%    |
|-----------------|----------------|------------------|--------|
| FDB-AEO-PID     | 0.3279         | 0.5098           | 0.0039 |
| FDP-AEO-FOPID   | 0.1726         | 0.2627           | 0.1072 |

FDB-AEO algoritması ile tasarımı yapılan PID ve FOPID denetleyicili OGR sistemlerinin adım yanıtı sonuçlarına göre; FOPID denetleyicili OGR sisteminin, 0.1726 saniye yükselme zamanı ve 0.2627 saniye yerleşme zamanı ile PID denetleyicili sisteme göre üstün olduğu görülmektedir. En büyük yüzde aşım konusunda %0.1072 ile kötü bir performans sergilemeyen FOPID denetleyicili sistemin senkron generatörler ile birlikte kullanılmasının uygun olacağı görülmektedir.

### 3.2. Bode Analizi

FDB-AEO algoritması ile tasarımı yapılan PID-OGR sisteminin bode diyagramı Şekil 7'de ve frekans yanıtı Tablo 8'de verilmiştir.



Şekil 7. FDB-AEO algoritması ile tasarımı yapılan PID-OGR sisteminin bode diyagramı.

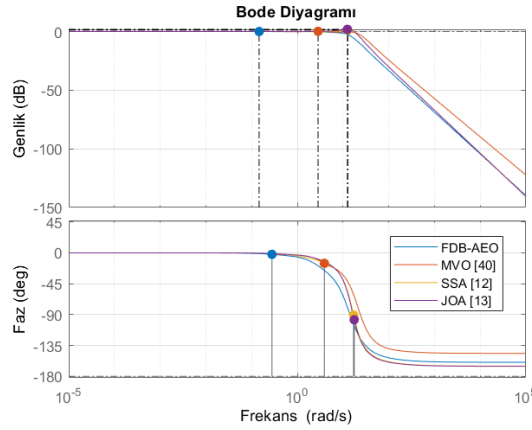
Bode analizi ile tasarlanan sistemin frekans yanıtı elde edilerek kararlılık düzeyi yorumlanmış olur. Bode diyagramlarından sistemin tepe kazancı (dB), faz marjı (derece), gecikme marjı (s) ve bant genişliği elde edilir. Tepe kazancı genlik grafiğinin en büyük değerini, faz marjı, kararsızlık oluşmadan yaşanabilecek en büyük faz kaymasını ve gecikme marjı, kararsızlık durumunun atlatılabilmesi için sistemin kaldırabileceği en büyük gecikme süresini ifade eder. Bant genişliği ise sistemin yükselme zamanı ile ilişkilidir. Bant genişliğinin yüksek olması daha düşük sürelerdeki yükselme zamanını temsil eder.



**Tablo 8.** FDB-AEO algoritması ile tasarımı yapılan PID-OGR sistemlerinin frekans yanıtı sonuçları.

| Algoritma                          | Tepe kazancı (dB)   | Faz marjı (derece)   | Gecikme marjı (s) | Bant genişliği (rad/s) |
|------------------------------------|---|----------------------|-------------------|------------------------|
| <b>FDB-AEO (Önerilen yaklaşım)</b> | $-1.93 \times 10^{-14}$ (at $2.03 \times 10^{-17}$ rad/s) | -180 (at 0 rad/s)    | sonsuz            | 6.7800                 |
| CPSO [38]                          | 0.789 (at 6.85 rad/s)                                     | 95.8 (at 9.38 rad/s) | 0.178             | 12.2672                |
| SCA [39]                           | 1.09 (at 9.17 rad/s)                                      | 87.3 (at 11.9 rad/s) | 0.128             | 14.8214                |
| SFS [21]                           | 3.11 (at 12.6 rad/s)                                      | 62.4 (at 17.2 rad/s) | 0.0634            | 19.8210                |

FDB-AEO algoritması ile tasarlanan PID denetleyicisi,  $-1.93 \times 10^{-14}$  dB ile en düşük tepe kazancına, -180 derece ile en iyi faz marjına ve sonsuz gecikme marjı ile en iyi gecikme marjına sahiptir. 0.3279 yükselme zamanı ile diğer algoritmalarla tasarlanan sistemlerin gerisinde kalması bant genişliğine de yansımıştır. Ancak adım yanıtı sonuçlarında da belirtildiği gibi bu durum en büyük yüzde aşım değerindeki üstün performans dikkate alındığında kabul edilebilir. FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sisteminin bode diyagramı Şekil 8’de ve frekans yanıtı Tablo 9’da verilmiştir.

**Şekil 8.** FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sisteminin bode diyagramı.**Tablo 9.** FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sistemlerinin frekans yanıtı sonuçları.

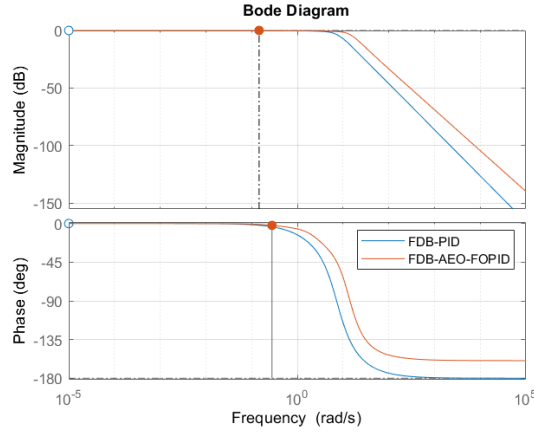
| Algoritma                          | Tepe kazancı (dB)       | Faz marjı (derece)   | Gecikme marjı (s) | Bant genişliği (rad/s) |
|------------------------------------|-------------------------|----------------------|-------------------|------------------------|
| <b>FDB-AEO (Önerilen yaklaşım)</b> | 0.0168 (at 0.146 rad/s) | 178 (at 0.279 rad/s) | 11.1              | 13.3                   |
| MVO-FOPID [40]                     | 0.161 (at 2.85 rad/s)   | 165 (at 3.91 rad/s)  | 0.735             | 24.1                   |
| SSA-FOPID [12]                     | 1.31 (at 12.8 rad/s)    | 89.3 (at 17 rad/s)   | 0.0914            | 21.2                   |
| JOA-FOPID [13]                     | 1.88 (at 12.5 rad/s)    | 82.5 (at 17.5 rad/s) | 0.082             | 21.3                   |

FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sistemi 0.0168 dB ile en düşük tepe kazancına, 178 derece ile en büyük faz marjına, 11.1 saniye ile en büyük gecikme marjına ve 13.3 rad/s ile en düşük bant genişliğine sahiptir. Bu durum tasarımı yapılan FOPID-OGR sisteminin kıyaslama yapılan sistemler arasında kararsızlık durumunda yaşanabilecek faz kayması açısından ve kararsızlığı kaldırılması için sistemin kaldırabileceği en büyük gecikme süresi açısından en iyi durumda olduğunu göstermektedir. Bant genişliğinin düşük olması, yükselme zamanının diğer sistemlere göre daha büyük olduğunu göstermektedir. Bu durum en büyük yüzde aşım ve yerleşme zamanı açısından tasarlanan sistemin üstünlüğü düşünüldüğünde kabul edilebilir. Tasarımı yapılan ve diğer çalışmalara göre üstün performansları gösterilen PID ve FOPID denetleyicili OGR sistemlerinin bode diyagramları Şekil 9’da gösterilmiştir. Frekans yanıtı sonuçları ise Tablo 10’da verilmiştir.

Tasarlanan sistemlerin kararlılık düzeylerini gösteren frekans yanıtları incelendiğinde PID-OGR sisteminin  $-1.93 \times 10^{-14}$  dB ile tepe kazancı, -180 derece ile faz marjı ve sahip olduğu sonsuz gecikme marjı ile FOPID-OGR sistemine göre daha üstün olduğu görülmektedir. Ancak FOPID-OGR sisteminin kararlılık düzeyi, literatürde önerilen diğer çalışmaların durumları da göz önüne alındığında oldukça iyi durumdadır. FOPID-OGR sisteminin adım yanıtı sonuçları bakımından PID-OGR sistemine göre üstün olduğu da düşünüldüğünde senkron generatörlerde kullanılmasının uygun olacağı sonucuna varılmıştır.

**Tablo 10.** Tasarımı yapılan PID ve FOPID denetleyicili OGR sistemlerinin frekans yanıtları.

| Önerilen yöntem | Tepe kazancı (dB)   | Faz marjı (derece)   | Gecikme marjı (s) | Bant genişliği (rad/s) |
|-----------------|---|----------------------|-------------------|------------------------|
| FDB-AEO-PID     | $-1.93 \times 10^{-14}$ (at $2.03 \times 10^{-17}$ rad/s) | -180 (at 0 rad/s)    | sonsuz            | 6.7800                 |
| FDB-AEO-FOPID   | 0.0168 (at 0.146 rad/s)                                   | 178 (at 0.279 rad/s) | 11.1              | 13.3                   |



Şekil 9. Tasarımı yapılan PID ve FOPID denetleyicili OGR sistemlerinin bode diyagramları.

#### 4. SONUÇ

Senkron generatörlerin terminal gerilimlerinde meydana gelen gerilim dalgalanmalarının önüne geçmek için senkron generatörler OGR sistemleri ile birlikte kullanılmaktadır. Bu çalışmada OGR sistemlerinde kullanılan denetleyici modellerinden olan PID ve FOPID denetleyicilerinin tasarımı FDB-AEO algoritması kullanılarak yapılmıştır. Tasarlanan denetleyicilerin performansı literatürde önerilen diğer çalışmalarla karşılaştırılmıştır. Çalışmadan elde edilen sonuçlar şu şekilde özetlenebilir;

- FDB-AEO algoritması ilk kez OGR sistemi tasarımı için kullanılmıştır. Algoritmanın gösterdiği üstün performans OGR sistemi tasarımı için kullanıma uygun olduğunu gösterdiği gibi diğer mühendislik problemlerinin çözümü için de kullanıma değer olduğunu göstermiştir.
- FDB-AEO algoritması ile tasarımı yapılan PID-OGR sistemi; CPSO [38], SCA [39] ve SFS [21] algoritmaları ile tasarımı yapılan PID-OGR sistemleri ile adım yanıtı ve frekans yanıtı açısından karşılaştırılmıştır. Karşılaştırma sonucunda FDB-AEO algoritması ile tasarımı yapılan sistemin diğer sistemlere göre üstün olduğu görülmüştür.
- FDB-AEO algoritması ile tasarımı yapılan FOPID-OGR sistemi; MVO [40], SSA [12] ve JOA [13] algoritmaları ile tasarımı yapılan FOPID-OGR sistemleri ile adım yanıtı ve frekans yanıtı açısından karşılaştırılmıştır. FDB-AEO algoritması ile tasarımı yapılan sistemin karşılaştırılan sistemlerden üstün olduğu görülmüştür.
- Tasarımı yapılan PID ve FOPID denetleyicilerinin performansı birbirleri ile de karşılaştırılmıştır. Karşılaştırma sonucunda adım yanıtı açısından FOPID, frekans yanıtı açısından PID denetleyicisinin performansı daha iyi bulunmuştur. FDB-AEO algoritması ile tasarımı yapılan FOPID denetleyicisinin frekans yanıtı benzer çalışmalardan daha iyi olduğu için adım yanıtında gösterdiği üstün performans da göz önüne alındığında OGR sistemlerinde kullanılmasının uygun olacağı sonucuna ulaşılmıştır.

#### Yazar Katkıları

Yazarlar eşit oranda katkı sağlamıştır.

#### Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

#### KAYNAKÇA

- [1] S. Habib, G. Abbas, T.A. Jumani, A.A. Bhutto, S. Mirsaeidi, and E.M. Ahmed "Improved whale optimization algorithm for transient response, robustness, and stability enhancement of an automatic voltage regulator system," *Energies*, vol. 15, no. 14, 5037, 2022.
- [2] B. Ozgenc, M.S. Ayas, and I.H. Altas "Performance improvement of an AVR system by symbiotic organism search algorithm-based PID-F controller," *Neural Computing and Applications*, vol. 34, pp. 7899-7908, 2022.
- [3] A. Sikander, and P. Thakur "A new control design strategy for automatic voltage regulator in power system," *ISA transactions*, vol. 100, pp. 235-243, 2020.
- [4] M. Çalasan, M. Micev, Ž. Djurovic, and H.M.A. Mageed "Artificial ecosystem-based optimization for optimal tuning of robust PID controllers in AVR systems with limited value of excitation voltage," *The International Journal of Electrical Engineering and Education*, 2020.
- [5] A.K. Bhullar, R. Kaur, and S. Sondhi "Enhanced crow search algorithm for AVR optimization. *Soft Computing*," *Soft Computing*, vol. 24, pp. 11957-11987, 2020.

- [6] P. Govindan "Evolutionary algorithms-based tuning of PID controller for an AVR system," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 3, pp. 3047-3056, 2020.
- [7] E. Köse "Optimal control of AVR system with tree seed algorithm-based PID controller," *IEEE Access*, vol. 8, pp. 89457-89467, 2020.
- [8] N. Pachauri "Water cycle algorithm-based PID controller for AVR," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 39, no. 3, pp. 551-567, 2020.
- [9] M. Micev, M. Čalasan, and D. Oliva "Design and robustness analysis of an Automatic Voltage Regulator system controller by using Equilibrium Optimizer algorithm," *Computers and Electrical Engineering*, vol. 89, p. 106930, 2021.
- [10] A. Sikander, P. Thakur, R.C. Bansal, and S. Rajasekar "A novel technique to design cuckoo search based FOPID controller for AVR in power systems," *Computers and Electrical Engineering*, vol. 70, pp. 261-274, 2018.
- [11] B. Bourouba, S. Ladaci, and H. Schulte "Optimal design of fractional order  $PI^{\lambda}D^{\mu}$  controller for an AVR system using Ant Lion Optimizer," *IFAC-PapersOnLine*, vol. 52, no. 13, 200205, 2019.
- [12] I.A. Khan, A.S. Alghamdi, T.A. Jumani, A. Alamgir, A.B. Awan, and A. Khidrani "Salp swarm optimization algorithm-based fractional order PID controller for dynamic response and stability enhancement of an automatic voltage regulator system, vol. 8, no. 12, pp. 1472, 2019.
- [13] T.A. Jumani, M.W. Mustafa, Z. Hussain, M.M. Rasid, M.S. Saeed, M.M. Memon, I. Khan, and K.S. Nisar "Jaya optimization algorithm for transient response and stability enhancement of a fractional-order PID based automatic voltage regulator system," *Alexandria Engineering Journal*, vol. 59, no. 4, pp. 2429-2440, 2020.
- [14] S.M.A. Altbawi, A.S.B. Mokhtar, T.A. Jumani, I. Khan, N.N. Hamadneh, and A. Khan "Optimal design of Fractional order PID controller based Automatic voltage regulator system using gradient-based optimization algorithm", *Journal of King Saud University-Engineering Sciences*, 2021.
- [15] V.K. Munagala, and R.K. Jatoth "Improved fractional  $PI^{\lambda}D^{\mu}$  controller for AVR system using Chaotic Black Widow algorithm," *Computers and Electrical Engineering*, vol. 97, p. 107600, 2022.
- [16] J.G. Ziegler, and N.B. Nichols "Optimum settings for automatic controllers," *Transactions of the American society of mechanical engineers*, vol. 64, no. 8, pp. 759-765, 1942.
- [17] C.C. Hang, B. K.J. Åström, and W.K. Ho "Refinements of the Ziegler–Nichols tuning formula," In *IEE Proceedings D (Control Theory and Applications)*, vol. 138, no. 2, pp. 111-118, 1991.
- [18] Q.G. Wang, T.H. Lee, H.W. Fung, Q. Bi, and Y. Zhang "PID tuning for improved performance," *IEEE Transactions on control systems technology*, vol. 7, no. 4, pp. 457-465, 1999.
- [19] M.A. Sabib, and B.S. Ahmed "A new multiobjective performance criterion used in PID tuning optimization algorithms," *Journal of advanced research*, vol. 7, no. 1, pp. 125-134, 2016.
- [20] H. Rezaei, O. Bozorg-Haddad, and X. Chu "Grey wolf optimization (GWO) algorithm," *Advanced optimization by nature-inspired algorithms*, pp. 81-91, 2018.
- [21] E. Çelik "Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system," *Neural Computing and Applications*, vol. 30, pp. 1991-2002, 2018.
- [22] B. Ozgenc, M.S. Ayas, and I.H. Altas "Performance improvement of an AVR system by symbiotic organism search algorithm-based PID-F controller," *Neural Computing and Applications*, vol. 34, pp. 7899-7908, 2022.
- [23] W. Zhao, L. Wang, and Z. Zhang "Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm," *Neural Computing and Applications*, vol. 32, pp. 9383-9425, 2020.
- [24] M.S. Ayas, and E. Sahin "FOPID controller with fractional filter for an automatic voltage regulator," *Computers and Electrical Engineering*, vol. 90, p. 106895, 2021.
- [25] R. Pradhan, S.K. Majhi, and B.B. Pati "Design of PID controller for automatic voltage regulator system using Ant Lion Optimizer," *World Journal of Engineering*, vol.15, no.3, pp. 373-387, 2018.
- [26] A.M. Mosaad, M.A. Attia, and A.Y. Abdelaziz "Comparative performance analysis of AVR controllers using modern optimization techniques," *Electric Power Components and Systems*, vol. 46, no. 19-20, pp. 2117-2130, 2018.
- [27] A.G. Suri babu, B.T. Chiranjeevi "Implementation of fractional order PID controller for an AVR system using GA and ACO optimization techniques," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 456-461, 2016.
- [28] Y. Zhou, J. Zhang, X. Yang, and Y. Ling "Optimization of PID controller based on water wave optimization for an automatic voltage regulator system," *Information Technology and Control*, vol. 48, no. 1, pp. 160-171, 2019.
- [29] I.A. Khan, A.S. Alghamdi, T.A. Jumani, A. Alamgir, A.B. Awan, and A. Khidrani "Salp swarm optimization algorithm-based fractional order PID controller for dynamic response and stability enhancement of an automatic voltage regulator system," *Electronics*, vol. 8, no. 12, pp. 1472, 2019.
- [30] I. Eke, M. Saka, H. Gozde, Y. Arya, and M.C. Toplamacioglu "Heuristic optimization based dynamic weighted state feedback approach for 2DOF PI-controller in automatic voltage regulator," *Engineering Science and Technology, an International Journal*, vol. 24, no. 4, pp. 899-910, 2021.
- [31] S. Ekinci, A. Demirören, H.L. Zeynelgil, and, S. Kaya "Design of PID controller for automatic voltage regulator system through kidney-inspired algorithm," *GU J Sci, Part C*, vol. 7, no. 2, pp. 383-398, 2019.

- [32] E. Çelik, and R. Durgut “Performance enhancement of automatic voltage regulator by modified cost function and symbiotic organisms search algorithm,” *Engineering science and technology, an international journal*, vol. 21, no. 5, pp. 1104-1111, 2018.
- [33] M.S. Ayas “Design of an optimized fractional high-order differential feedback controller for an AVR system,” *Electrical Engineering*, vol. 101, pp. 1221-1233, 2019.
- [34] V. Rajinikanth, and S.C. Satapathy “Design of controller for automatic voltage regulator using teaching learning based optimization,” *Procedia Technology*, vol. 21, pp. 295-302, 2015.
- [35] H.T. Kahraman, S. Aras, and E. Gedikli “Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms,” *Knowledge-Based Systems*, vol. 190, 105169, 2020.
- [36] S. Aras, E. Gedikli, and H.T. Kahraman “A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization,” *Swarm and Evolutionary Computation*, vol. 61, 100821, 2021.
- [37] Y. Sonmez, S. Duman, H.T. Kahraman, M. Kati, S. Aras, and U. Guvenc “Fitness-distance balance based artificial ecosystem optimisation to solve transient stability constrained optimal power flow problem,” *Journal of Experimental and Theoretical Artificial Intelligence*, 2022.
- [38] N. Anwar, A. Hanif, M.U. Ali, and A. Zafar “Chaotic-based particle swarm optimization algorithm for optimal PID tuning in automatic voltage regulator systems,” *Electrical Engineering & Electromechanics*, vol. 1, pp. 50-59, 2021.
- [39] B. Hekimoğlu “Sine-cosine algorithm-based optimization for automatic voltage regulator system,” *Transactions of the Institute of Measurement and Control*, vol. 41, no. 6, pp. 1761-1771, 2019.
- [40] A. Tabak “Maiden application of fractional order PID plus second order derivative controller in automatic voltage regulator,” *International Transactions on Electrical Energy Systems*, vol. 31, no. 12, e13211, 2021.
- [41] S. Panda, B.K. Sahu, and P.K. Mohanty “Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization,” *Journal of the Franklin Institute*, vol. 349, no. 8, pp. 2609-2625, 2012.
- [42] A. Tabak “A novel fractional order PID plus derivative ( $PI^{\lambda}D^{\mu}D^{\mu_2}$ ) controller for AVR system using equilibrium optimize,” *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 40, no. 3, pp. 722-743, 2021.