



## Kızıl Tilki Optimizasyon Algoritması ile DA-DA Alçaltıcı Tip Dönüştürücü Kontrolör Katsayılarının Optimizasyonu

### Optimization of Buck Converter Controller Coefficients with Red Fox Optimization Algorithm

<sup>1</sup>Ersagun Kürşat YAYLACI , <sup>2</sup>Ahmet Erdem YILMAZ , <sup>3</sup>Hatice Nur ÖZDEŞ 

<sup>1,2,3</sup>Karabük Üniversitesi, Mühendislik Fakültesi, Karabük, Türkiye

<sup>1</sup>ekyaylaci@karabuk.edu.tr, <sup>2</sup>aeilmz07@gmail.com,

<sup>3</sup>haticenur9878@gmail.com

Araştırma Makalesi/Research Article

#### ARTICLE INFO

##### Article history

Received : 30 April 2022

Accepted : 12 July 2022

##### Keywords:

Red Fox Optimization, PI Controller, Buck Converter, Particle Swarm Optimization

#### ABSTRACT

In this study, closed-loop output voltage control of Buck converter, which is widely used in devices that need dc voltage, has been made. It is seen that meta-heuristic methods have been used quite a lot recently for the determination of controller coefficients. By using red fox optimization algorithm, steady-state error, which is one of the control performance criteria of the system, is tried to be improved with PI controller. The objective function used for the desired improvement process; is calculated by taking the average of the sum of the absolute values of the error, the sum of the squares of the error, and the sum of the time-weighted squares of the error. This study is carried out in MATLAB/Simulink environment and the obtained results are analyzed comparatively by the particle swarm optimization method. According to the analysis results obtained, the objective function fitness values for the red fox and particle swarm optimization methods were obtained as 1.1181 and 1.1045, respectively, and they showed close performances. When the working times are compared, the particle swarm optimization algorithm can find the optimum two times faster than the red fox optimization algorithm.

© 2022 Bandırma Onyedi Eylül University, Faculty of Engineering and Natural Science. Published by Dergi Park. All rights reserved.

#### MAKALE BİLGİSİ

##### Makale Tarihleri

Gönderim : 30 Nisan 2022

Kabul : 12 Temmuz 2022

##### Anahtar Kelimeler:

Kızıl Tilki Optimizasyonu, PI Kontrolör, Alçaltıcı Tip DA-DA Dönüştürücü, Parçacık Sürü Optimizasyonu

#### ÖZET

Bu çalışmada, da gerilime ihtiyaç duyulan cihazlarda yaygın olarak kullanılan alçaltıcı tip DA-DA dönüştürücünün kapalı çevrim çıkış gerilim kontrolü yapılmıştır. Kontrolör katsayılarının belirlenmesi için meta-sezgisel yöntemlerin son zamanlarda oldukça fazla kullanıldığı görülmektedir. PI kontrolörü ile kızıl tilki optimizasyon yöntemi kullanılarak sistemin kontrol performans kriterlerinden sürekli hal hatası iyileştirilmeye çalışılmıştır. Yapılmak istenen iyileştirme işlemi için kullanılan amaç fonksiyonu; zamanla ağırlıklandırılmış hata karelerinin toplamı, hata mutlak değerlerinin toplamı ve hata karelerinin toplamı değerlerinin ortalaması alınarak hesaplanmıştır. Bu çalışma MATLAB/Simulink ortamında gerçekleştirilmiş olup elde edilen sonuçlar parçacık sürü optimizasyonu yöntemi ile karşılaştırmalı olarak analiz edilmiştir. Elde edilen analiz sonuçlarına göre kızıl tilki ve parçacık sürüsü optimizasyon yöntemleri için amaç fonksiyonu uygunluk değerleri sırasıyla 1,1181 ve 1,1045 olarak elde edilmiş olup birbirine yakın performans sergilemişlerdir. Çalışma süreleri kıyaslandığında ise parçacık sürü optimizasyon algoritması, kızıl tilki optimizasyon algoritmasından yaklaşık iki kat daha hızlı optimum bulabilmektedir.

© 2022 Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi. Dergi Park tarafından yayımlanmaktadır. Tüm Hakları Saklıdır.

ORCID ID: <sup>1</sup>0000-0003-0358-5617

<sup>2</sup>0000-0002-8408-5536

<sup>3</sup>0000-0002-3198-3412

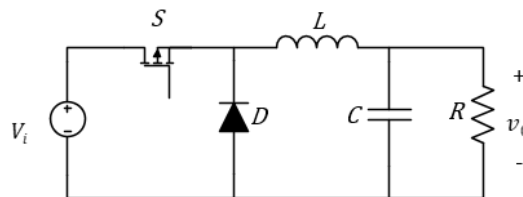
## 1. GİRİŞ

Doğadaki canlıların yaşayış biçimleri meta-sezgisel optimizasyon yöntemlerine ilham kaynağı olmaktadır. Doğada sürü halinde yaşayan canlıların sergiledikleri sezgisel hareketler, avlanma şekilleri ve yaşayış tarzları matematiksel olarak modellenerek çeşitli problemlerin çözümü için farklı optimizasyon algoritmaları geliştirilmiştir [1, 2]. Parçacık Sürü Optimizasyonu (PSO), Gri Kurt Optimizasyonu (GKO), Kızıl Tilki Optimizasyonu (KTO) algoritmaları bu tür algoritmalara örnek olarak verilebilir [3–5]. PSO algoritması 1995 yılında Kennedy ve Eberhart tarafından geliştirilmiş olup doğrusal olmayan problemlerin çözümünde de sıkça tercih edilmektedir [6]. Kızıl tilki optimizasyonu algoritması 2021 yılında Dawid Połap ve Marcin Woźniak tarafından kızıl tilkilerin doğadaki yaşayış ve avlanma biçimlerinden esinlenilerek geliştirilmiş bir optimizasyon algoritmasıdır. KTO algoritması PSO'ya benzer bir şekilde sürü zekasına dayanmakta ve amaç fonksiyonuna en uygun olan tilkinin konumu kullanılarak sürü içerisindeki tilkilerin yeniden konumlandırılması yapılmaktadır. Buna ek olarak avını çevreleyen bir avlanma modeli ile avlanma tamamlanmaktadır [5]. KTO yöntemi şimdiye dek çeşitli problemlerin çözümünde kullanılmıştır [7–13]. Örneğin 2022 yılında yapılan bir çalışmada, deri kanserinin teşhisinde KTO tabanlı geliştirilmiş algoritma kullanılmıştır [7]. Bir başka çalışmada uçak yakıt hücresi, süper kapasitör ve pil enerji depolama biriminden oluşan hibrit sistem için sistemin verimliliğini arttırmak amacıyla Kaos Kızıl Tilki Optimizasyonu algoritması adı verilen KTO tabanlı geliştirilmiş bir meta-sezgisel algoritma kullanılmıştır [8]. Bir diğer çalışmada ise KTO algoritmasında geliştirmeler yapılarak elde edilen algoritma ile Olağanüstü Öğrenme Makineleri (Extreme Learning Machines, ELM) yöntemi hibrit kullanılmıştır [9]. Neng Zhu ve arkadaşları yaptıkları çalışmada sıfır enerjili binalarda yer alan yenilenebilir enerji yapıları için maliyet ve model tasarımında, geliştirilmiş KTO yöntemi ile çok kriterli optimizasyon ve optimum model için Gri Çok Düzeyli Tam Değerlendirme yaklaşımının entegrasyonunu sağlamışlardır [10]. 2022 yılında yapılan bir çalışmada PV hücre/modellerinde kullanılacak olan tek ve çift diyot model parametrelerinin güvenilir seçilmesi amacıyla Runge Kutta yöntemi ile KTO ve yeni çıkan diğer dört optimizasyon yöntemi karşılaştırılmıştır [11]. Katı oksit yakıt hücresindeki bilinmeyen değişkenlerin optimum şekilde tanımlanması amaçlanan bir çalışmada ise KTO yöntemi tercih edilerek optimizasyon sağlanmıştır [12]. Bir başka çalışmada ise yapay zeka uygulamalarında kullanılan birleşik öğrenme yönteminin doğruluğunu arttırmak için KTO yöntemi kullanılmıştır [13].

Endüstride sıklıkla kullanılan doğru akım dönüştürücülerinden biri olan alçaltıcı tip DA-DA dönüştürücü (Buck Converter, BC), girişine uygulanan gerilimi çıkışına düşürerek aktaran anahtarlamalı bir dönüştürücü devresidir. BC devresinin çıkış gerilim kontrolü, çıkışına bağlanacak cihazların problemsiz çalışması için önem arz etmektedir. Bu çalışma kapsamında BC çıkış gerilim kontrolü için PI (Oransal-İntegral) kontrolör kullanılmıştır. PI kontrolöre ilişkin katsayıların belirlenmesi bir optimizasyon problemi olarak kabul edilebilir [6]. Günümüze dek en uygun katsayılar için optimizasyon yöntemlerinin kullanıldığı bir çok farklı çalışma yapılmıştır [14–16]. Bu çalışmalardan birinde DA-DA yükseltici dönüştürücü için PI kontrolörü kullanılmış ve Guguk kuşu optimizasyonu yöntemi ile PI kontrolörünün katsayıları belirlenmiştir [14]. Bir diğer çalışmada ise alçaltıcı-yükseltici DA-DA dönüştürücünün çıkış gerilim kontrolü için PI kontrolörü uygulanmış ve PI katsayıları PSO yöntemi ile belirlenmiştir [15]. BC devresi için PI kontrolörü kullanılan başka bir çalışmada ise optimum katsayıları Karışık Kurbağa Sıçrayan Algoritması (Shuffled Frog Leaping Algorithm, SFLA) ile bulunmuştur [16]. Literatürde henüz KTO yöntemi ile kontrolör katsayısı optimizasyonuna ilişkin bir çalışmaya rastlanmamıştır. Bu çalışmada, KTO ile BC devresinin çıkış gerilim kontrolü için kullanılan PI kontrolörün katsayıları optimize edilmiştir. Yöntem için sürekli hal hatası referans alınarak amaç fonksiyonu uygunluk değerinin minimizasyonu amaçlanmıştır. Önerilen yöntem ile daha önce literatürde oldukça fazla kullanılan PSO algoritması; uygunluk değeri, iterasyon sayısı ve simülasyon süreleri bakımından karşılaştırılmıştır. Çalışmanın geri kalanında 2. bölüm alçaltıcı tip DA-DA dönüştürücüye ayrılmıştır. 3. ve 4. bölümde sırasıyla PSO ve KTO yöntemleri anlatılmış, 5. ve 6. bölümde ise sırasıyla bulgular ve sonuç kısımlarına yer verilmiştir.

## 2. ALÇALTICI TİP DA-DA DÖNÜŞTÜRÜCÜ

BC devresi, elektronik cihazların birçoğunda, güç kaynakları ve motor hız denetimi uygulamalarında sıklıkla kullanılmaktadır [17]. Girişine uygulanan gerilimi anahtarlama elemanı ile çıkışına düşürerek ileten BC devresi, anahtarlama elemanının iletim ve kesim durumuna göre iki farklı çalışma konfigürasyonuna sahiptir. Ayrıca BC sürekli ve süreksiz akım modu olarak iki farklı dinamik yapıya sahip olabilir. Bu çalışma kapsamında sürekli akım modunda çalışması sağlanmaktadır [6]. BC elektriksel eşdeğer devresi Şekil 1'de gösterilmiştir.



Şekil 1. Alçaltıcı tip DA-DA dönüştürücü.

Anahtar açık durumda iken BC eşdeğer devresi Şekil 2.a'da gösterildiği gibi olup çıkışında elde edilen akım ve gerilim Denklem 1 ve 2 ile hesaplanabilmektedir:

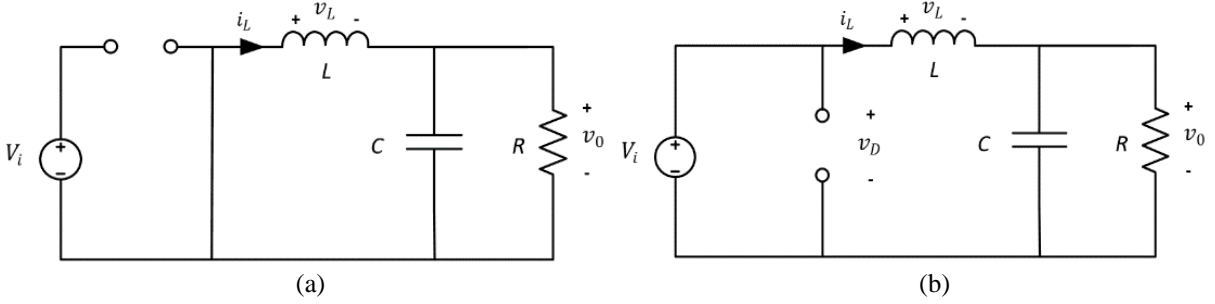
$$\frac{di_L}{dt} = -\frac{v_o}{L} \quad (1)$$

$$\frac{dv_o}{dt} = \frac{1}{C} \left( i_L - \frac{v_o}{R} \right) \quad (2)$$

Anahtar kapalı durumda iken ise BC eşdeğer devresi Şekil 2.b'de gösterildiği gibi olup akım ve çıkış gerilimi Denklem 3 ve 4'te ifade edilen denklemler ile hesaplanabilmektedir:

$$\frac{di_L}{dt} = \frac{1}{L} (V_i - v_o) \quad (3)$$

$$\frac{dv_o}{dt} = \frac{1}{C} \left( i_L - \frac{v_o}{R} \right) \quad (4)$$



Şekil 2. BC elektriksel eşdeğer devresi a) Anahtar açık iken, b) Anahtar kapalı iken.

Çalışmada kullanılan BC devresi parametreleri Tablo 1'de gösterilmiştir.

Tablo 1. Parametre değerleri.

Tanım	Sembol	Parametre
Giriş Gerilimi	$V_i$	9 V
Çıkış Gerilimi	$V_o$	5 V
İndüktör	L	555 $\mu$ H
Kapasitör	C	25 $\mu$ F
Yük Direnci	R	10 $\Omega$
Anahtarlama Frekansı	$f_s$	20 kHz

### 3. PARÇACIK SÜRÜ OPTİMİZASYONU

PSO algoritması, doğadaki yaşamlarını sürüler halinde sürdüren canlıların yaşayış biçimlerinden esinlenerek geliştirilmiştir. Sürü içinde bilgi paylaşımını ve sürü dayanışmasını esas almaktadır. Sürülerin beslenmek ve güvenli bir barınak bulmak için sergiledikleri rastgele veya sezgisel davranışların, sürüyü amacına kolay ve hızlı bir şekilde ulaştırdığı görülmüştür. Parçacıklar, daha önceki tecrübelerinden ve sürünün en iyi bireyinden faydalanarak kendilerini sürü içerisindeki en iyi konuma ayarlamaya çalışırlar [6].

Algoritma esas olarak aşağıdaki adımlardan meydana gelir [3];

1. Başlangıçta rastgele atanan başlangıç konumları ve hızları ile sürü oluşturulur.
2. Sürüde bulunan tüm parçacıklar için amaç fonksiyonundan uygunluk değerleri hesaplanır.
3. Her bir parçacığın, bulunduğu iterasyon için lokal en iyi değeri belirlenir.
4. Bulunulan iterasyon için parçacıkların lokal en iyi değerlerinin arasından global en iyi değeri ifade eden "gbest" değeri seçilir.
5. Konum ve hız değerleri gbest değerine göre her iterasyonda güncellenir.

Hız ve konum değerleri sırasıyla Denklem 5 ve 6'da ifade edildiği gibi hesaplanmaktadır [18].

$$v_{iD}^{(t+1)} = w \cdot v_{iD}^t + c_1 r_1 (p_{iD}^t - x_{iD}^t) + c_2 r_2 (g_i^t - x_{iD}^t) \quad i = 1, 2, 3, \dots, m \quad (5)$$

$$x_{iD}^{t+1} = x_{iD}^t + v_{iD}^{t+1} \quad (6)$$

Burada  $c_1$  ve  $c_2$  ölçeklendirme katsayıları,  $p_{iD}^t$  lokal en iyi konum,  $g_i^t$  ise global en iyi konumdur. Parçacık sayısı  $i$  ile ifade edilirken,  $t$  iterasyon sayısını belirtmektedir.  $r_1$  ve  $r_2$  çarpanları her iterasyonda güncellenen, [0,1] aralığında rastgele atanan çarpanlardır [18]. Atalet ağırlığı  $w$  ile ifade edilip yapılmak istenen yavaşlatma işleminin oranını belirlemektedir. Bu değer için algoritmanın başında bir maksimum ve minimum noktası belirlenir ve iterasyon boyunca  $w$  değeri  $w_{max}$ 'dan  $w_{min}$ 'e doğru doğrusal bir azalma gösterir. Bu azalma işlemi ile sistemin

başlangıcında  $w$  global arama performansında artış sağlarken, sistemin sonlanmasına doğru lokal arama performansında artış sağlanmaktadır. Atalet ağırlığı Denklem 7’de gösterildiği gibi hesaplanmaktadır [6].

$$w = w_{\max} - i \left( \frac{w_{\max} - w_{\min}}{i_{\max}} \right) \quad (7)$$

$w_{\max}$  ve  $w_{\min}$  değerleri çalışmaların genelinde sırasıyla 0,9 ve 0,4 kullanılmaktadır [19].

Algoritmanın arama işlemi sırasında, parçacıkların arama uzayının dışına çıkmasını önlemek için sistemin başında arama uzayı için bir alt ve üst sınır belirlenir. Sınır değerleri, sistemin optimuma daha hızlı ulaşmasını sağlamak için önemli bir etmendir. Parçacıkların  $v_{\max}$  ve  $v_{\min}$  sınırları Denklem 8 ve 9’da belirtildiği gibi belirlenmektedir [20].

$$v^{\max} = k \cdot x^{\max} \quad 0,1 \leq k \leq 0,5 \quad (8)$$

$$v^{\min} = -v^{\max} \quad (9)$$

Konum ve hız değerleri, belirlenen sınırlar dışında bir değer gelmesi halinde Denklem 10’a göre güncellenir [21].

$$\text{Eğer } v^{t+1} > v^{\max} \text{ ise } v^{t+1} = v^{\max}$$

$$\text{Eğer } v^{t+1} < v^{\min} \text{ ise } v^{t+1} = v^{\min}$$

$$\text{Eğer } x^{t+1} > x^{\max} \text{ ise } x^{t+1} = x^{\max} \quad (10)$$

$$\text{Eğer } x^{t+1} < x^{\min} \text{ ise } x^{t+1} = x^{\min}$$

#### 4. KIZIL TILKI OPTİMİZASYONU

Dawid Połap ve Marcin Woźniak tarafından 2021 yılında geliştirilen KTO, kızıl tilkilerin doğadaki yaşam tarzlarından ve avlanmalarından esinlenilerek geliştirilmiştir. Her bir tilki sürüsü kendi alfa çiftinin egemenliği altında tek bir bölgeyi paylaşmaktadır. Sürüdeki genç üyeler büyüdüklerinde eğer başka bir bölgeyi kontrol etme şansı yakalarlarsa sürüden ayrılarak kendi sürülerini kurabilmektedirler. Bölgede yemek için arama yapan bir tilki, avlayabileceği bir av fark ettiğinde ona en etkili saldırıyı yapabilecek yakınlığa gelene kadar ava fark ettirmeden yaklaşır. Kızıl tilkinin avlanması iki safhada gerçekleşmektedir; birinci safhada bölge üzerinde av araması yapılır ve bu safha global arama olarak isimlendirilir. İkinci safhada bölge üzerinde fark edilen ava; saldırmadan önce olabildiğince yaklaşılır ve bu safha ise lokal arama olarak isimlendirilmektedir [5].

Tüm iterasyonlar boyunca tilki sayısı sabit kalmaktadır. Tilkilerin her birinin  $n$  koordinattaki bir noktası  $\bar{x} = (x_0, x_1, \dots, x_{n-1})$  şeklinde gösterilmektedir. Her bir tilki  $(\bar{x}_j^t)^t$  gösterimi ile ifade edilmektedir. Burada  $j$  boyut indisine karşılık gelmektedir. Amaç fonksiyonundan optimum değerleri bulmak için tilkilerin çözüm uzayı sınırları içerisinde dolaştıkları kabul edilmektedir [5].

##### 4.1. Yemek Arama (Global Arama)

Sürü içerisindeki tüm tilkiler, diğer üyelerin hayatta kalması için önem arz etmektedir. Farklı bölgelerin keşfi ve lokal habitatta yemek olmaması gibi durumlar sonucunda tilkiler uzak noktalara giderek av arayışlarını sürdürürler. Bu keşif eylemi sırasında edindikleri bilgi, tecrübe gibi kazanımları sürünün hayatta kalması ve gelişmesi için diğer tilkilerle paylaşırlar. Bölgenin keşfi, bütün bireylerin uygunluk değerine göre modellenir. Sürüdeki en iyi tilkinin, amaç fonksiyonuna göre arazi üzerindeki en uygun noktayı keşfettiği varsayılır ve en iyi tilki bu noktayı ailenin diğer üyeleri ile paylaşır. Bu sebepten dolayı, ilk olarak bireyler amaç fonksiyonuna göre sıralanır ve sürü içerisindeki en iyi tilki  $(\bar{x}^{en\ iyi})^t$  olarak atanır. Daha sonra her bir tilkinin  $(\bar{x}^{en\ iyi})^t$  ile olan öklid uzaklığı hesaplanır. Bu hesaplama işlemi Denklem 11 ile gösterildiği gibi yapılır ve Denklem 12’ye bağlı olarak tüm tilkiler yeniden konumlandırılırlar [5].

$$d((\bar{x}^i)^t, (\bar{x}^{en\ iyi})^t) = \sqrt{\|(\bar{x}^i)^t - (\bar{x}^{en\ iyi})^t\|} \quad (11)$$

$$(\bar{x}^i)^t = (\bar{x}^i)^t + \alpha \cdot \text{sign}((\bar{x}^{en\ iyi})^t - (\bar{x}^i)^t) \quad (12)$$

Burada  $\alpha \in (0, d((\bar{x}^i)^t, (\bar{x}^{en\ iyi})^t))$  aralığında sürüdeki tüm tilkiler için rastgele atanan ölçeklendirme parametresidir. Denklem 12 ile hesaplanan yeni konum değerleri, bir önceki konum değerinin uygunluğu ile kıyaslanır. Elde edilen yeni konum değerinin uygunluğu, önceki konumuna göre daha iyi ise yeni konum korunur. Aksi halde tilki eski konumuna geri döner [5].

##### 4.2. Lokal Habitattan Geçiş (Lokal Arama)

Kızıl tilki arama yaptığı bölgede bir av bulduğunda, kendisini ava fark ettirmeden yaklaşabildiği kadar yaklaşır ve avını dairesel olarak çevreler. Ava yeterince yaklaştığında ise ava saldırır. Bu yaklaşım sırasında tilkinin hareketini modellemek için  $\mu$  parametresi, her iterasyonda tüm bireyler için 0 ile 1 arasında rastgele atanır [5].

$$\begin{cases} \text{Ava yaklaş} & \mu > 0,75 \\ \text{Konumunu koru ve saklan} & \mu \leq 0,75 \end{cases} \quad (13)$$

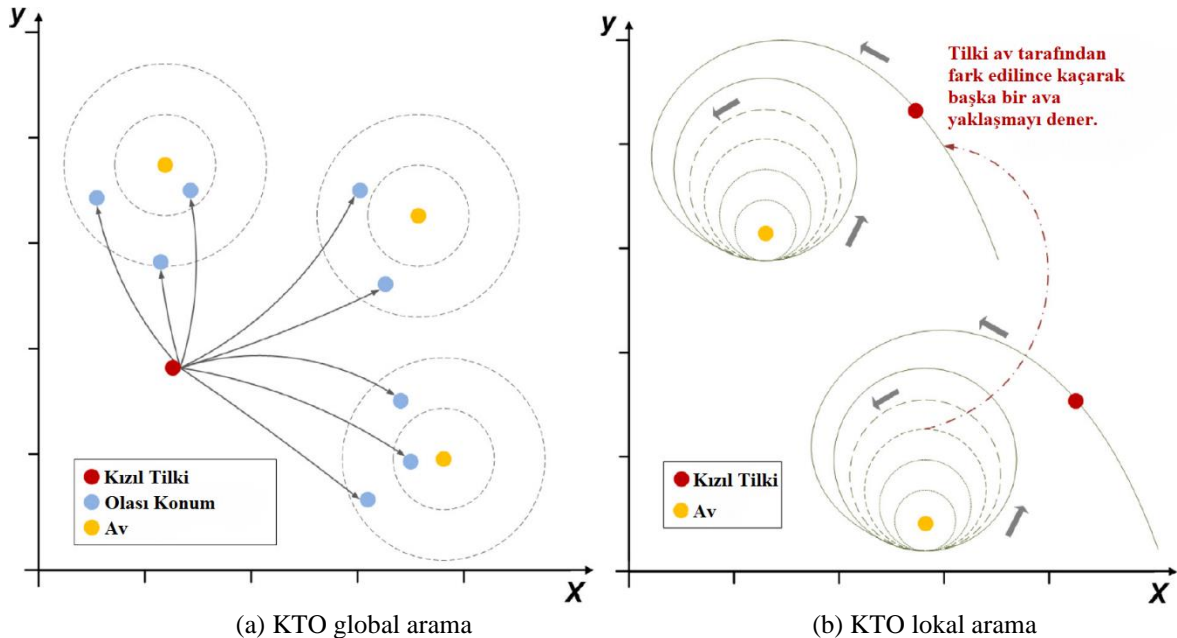
Denklem 13 ile belirtilen ilk koşula göre,  $\mu$  katsayısının 0,75'ten büyük olması halinde tilkiler harekete geçmektedirler. Hareket sırasında tilkilerin gözlem yarıçapı iki parametre ile temsil edilmektedir;  $a \in (0, 0,2)$  aralığında ölçeklendirme parametresidir ve tüm tilkiler için her iterasyonda rastgele atanarak güncellenmektedir.  $\phi_0$  ise  $(0, 2\pi)$  aralığında, tilkilerin gözlem açısını belirlemek amacıyla algoritmanın başlangıcında bütün tilkiler için rastgele atanmaktadır. Burada tilkilerin hareketinin modellenmesinde değiştirilmiş Cochleoid denklemi kullanılmaktadır. Bu sayede dairesel bir harekete olanak sağlanmaktadır. Tilkinin görüş yarıçapı Denklem 14 ile belirtildiği gibi tanımlanır [5].

$$r = \begin{cases} a \frac{\sin(\phi_0)}{\phi_0} & \text{eğer } \phi_0 \neq 0 \\ \theta & \text{eğer } \phi_0 = 0 \end{cases} \quad (14)$$

Burada belirtilen  $\theta$  değeri, sis, yağmur, kar gibi olumsuz hava koşullarını temsil etmektedir. Bu değer iterasyonun başında bir kere 0 ile 1 aralığında rastgele atanmaktadır. Tilkilerin arazi üzerindeki hareketleri Denklem 15 ile ifade edildiği gibi hesaplanmaktadır [5].

$$\begin{cases} x_0^{yeni} = ar \cdot \cos(\phi_1) + x_0^{gerçek} \\ x_1^{yeni} = ar \cdot \sin(\phi_1) + ar \cdot \cos(\phi_2) + x_1^{gerçek} \\ x_2^{yeni} = ar \cdot \sin(\phi_1) + ar \cdot \sin(\phi_2) + ar \cdot \cos(\phi_3) + x_2^{gerçek} \\ \dots \\ x_{n-2}^{yeni} = ar \cdot \sum_{k=1}^{n-2} \sin(\phi_k) + ar \cdot \cos(\phi_{n-1}) + x_{n-2}^{gerçek} \\ x_{n-1}^{yeni} = ar \cdot \sin(\phi_1) + ar \cdot \sin(\phi_2) + \dots + ar \cdot \sin(\phi_{n-1}) + x_{n-1}^{gerçek} \end{cases} \quad (15)$$

Burada  $\phi_1, \phi_2, \dots, \phi_{n-1}$  değerleri  $(0, 2\pi)$  aralığında her tilki için rastgele seçilir. Belirtilen bu hareket ile tilki, avını saptadıktan sonra yaklaşılabildiği kadar yakınına yaklaşır. Tilki, avını kendini fark etmesi durumunda başka bir av seçerek aynı şekilde yaklaşmaya çalışır. Bu hareket tilkinin lokal arama safhası için modellenmiştir. Belirtilen yaklaşma hareketi Şekil 3 ile gösterilmiştir [5].



**Şekil 3.** (a) KTO global arama: Kızıl tilkinin avın etrafını çevrelemek ve avlanmaya başlamak için olası konumları. Habitat içerisinde geçiş sırasında, her tilki olası bir av arar ve ona doğru hareket eder. (b) KTO lokal arama: Kızıl tilkinin avlanma sırasında avını aldatmak için gerçekleştirdiği dönme hareketinin görseli. Tilki avına olabildiğince yaklaşmaya çalışır, ancak fark edilirse başka bir ava benzeri şekilde saldırı gerçekleştirir [5].

### 4.3. Sürüden Ayrılma ve Sürüye Katılma

Kızıl tilkiler doğada birçok tehlike ile karşı karşıya kalmaktadırlar. Lokal yaşama alanlarında yiyecek sıkıntısı çekmeleri sonucunda, yaşam alanlarını terk ederek farklı yeni alanlarda arama yapmak zorunda kalabilirler. Bu

durum tilkiler açısından tehlike arz etmektedir. Bir diğer tehlike ise insanların tilkileri avlamasıdır. Fakat doğada bu olaylar neticesinde zeki olan tilkiler bu olumsuz koşullardan etkilenmez ve arazi üzerinde yiyecek bulabilir, avcılardan kaçabilir veya kendilerine yeni bir sürü oluşturmak için zemin hazırlayabilirler [5].

Her iterasyon sonunda amaç fonksiyonundan aldığımız değerlere göre popülasyondaki en kötü bireylerin %5'i belirlenir. Belirlenen bu tilkilerin avcılar tarafından avlandığı veya sürüyü terk ettiği varsayılır. Popülasyondaki tilki sayısını sabit tutmak için alfa çifti tarafından oluşturulmuş yaşam alanı modeli kullanılarak sürüye yeni tilkiler eklenir. Bu işlem için her iterasyonda  $(\bar{x}^{(1)})^t$  ve  $(\bar{x}^{(2)})^t$  ile ifade edilen sürünün en iyi iki tilkisi olan alfa çifti seçilir. Seçilen alfa çifti kullanılarak Denklem 16 ile habitat merkezi hesaplanır [5].

$$(\text{habitat}^{\text{(merkez)}})^t = \frac{(\bar{x}^{(1)})^t + (\bar{x}^{(2)})^t}{2} \quad (16)$$

Daha sonra alfa çifti arasındaki öklid uzaklığının kare kökü hesaplanarak habitatın çapı Denklem 17 ile gösterildiği gibi hesaplanır [5].

$$(\text{habitat}^{\text{(cap)}})^t = \sqrt{\|(\bar{x}^{(1)})^t - (\bar{x}^{(2)})^t\|} \quad (17)$$

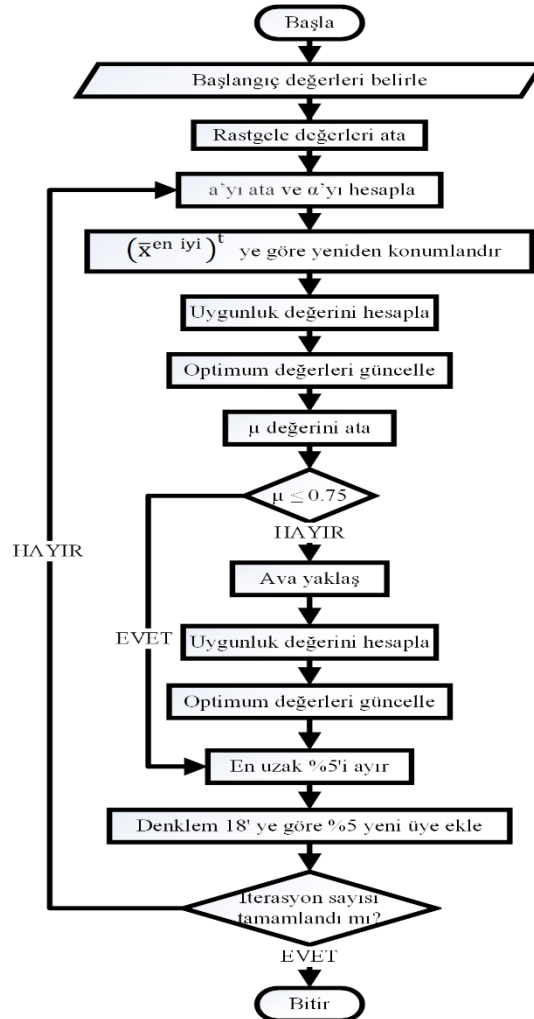
İterasyonların her birinde  $(0,1)$  arasında rastgele atanan  $\kappa$  parametresine göre sürüye yeni katılacak olan tilkilerin katılım şekli belirlenir. Bu seçim Denklem 18 ile belirtildiği gibi yapılmaktadır [5].

$$\begin{cases} \text{Yeni göçebe tilkiler} & \text{eğer } \kappa \geq 0,45 \\ \text{Alfa çiftinin üremesi} & \text{eğer } \kappa < 0,45 \end{cases} \quad (18)$$

İlk durumda habitata seçilecek yeni tilkiler habitatın dışı ile arama uzayının sınırları arasında rastgele seçilir. İkinci durumda ise sürüye yeni katılacak olan tilkilerin alfa çiftinden meydana geldiği varsayılmaktadır. Bu koşul ile sürüye yeni eklenen tilkilerin konumu Denklem 19 ile ifade edildiği gibi hesaplanmaktadır [5].

$$(\bar{x}^{\text{(alfa çiftinden üreyen)}})^t = \kappa \frac{(\bar{x}^{(1)})^t + (\bar{x}^{(2)})^t}{2} \quad (19)$$

Kızıl tilki algoritmasına ait akış diyagramı Şekil 4 ile ifade edildiği gibidir.



Şekil 4. KTO algoritma akış şeması.



## 5. BULGULAR

Belirtilen yöntemler farklı iterasyon ve parçacık sayıları ile eşit koşullarda çalıştırılarak karşılaştırılmıştır. Yapılan karşılaştırma işleminde amaç fonksiyonu uygunluk değeri, sistemin çalışma süresi, yükselme zamanı ve yerleşme zamanı göz önünde bulundurulmuştur. Amaç fonksiyonu sırasıyla Denklem 20, 21 ve 22'de belirtilen hataların mutlak değerlerinin toplamı (HMT), hataların kareleri toplamı (HKT) ve zamanla ağırlıklandırılmış hataların kareleri toplamı (ZHKT) ifadelerinin ortalaması alınarak Denklem 23 ile ifade edildiği gibi kullanılmıştır.

$$HMT = \sum_{k=1}^q |e(k)| \quad (20)$$

$$HKT = \sum_{k=1}^q e^2(k) \quad (21)$$

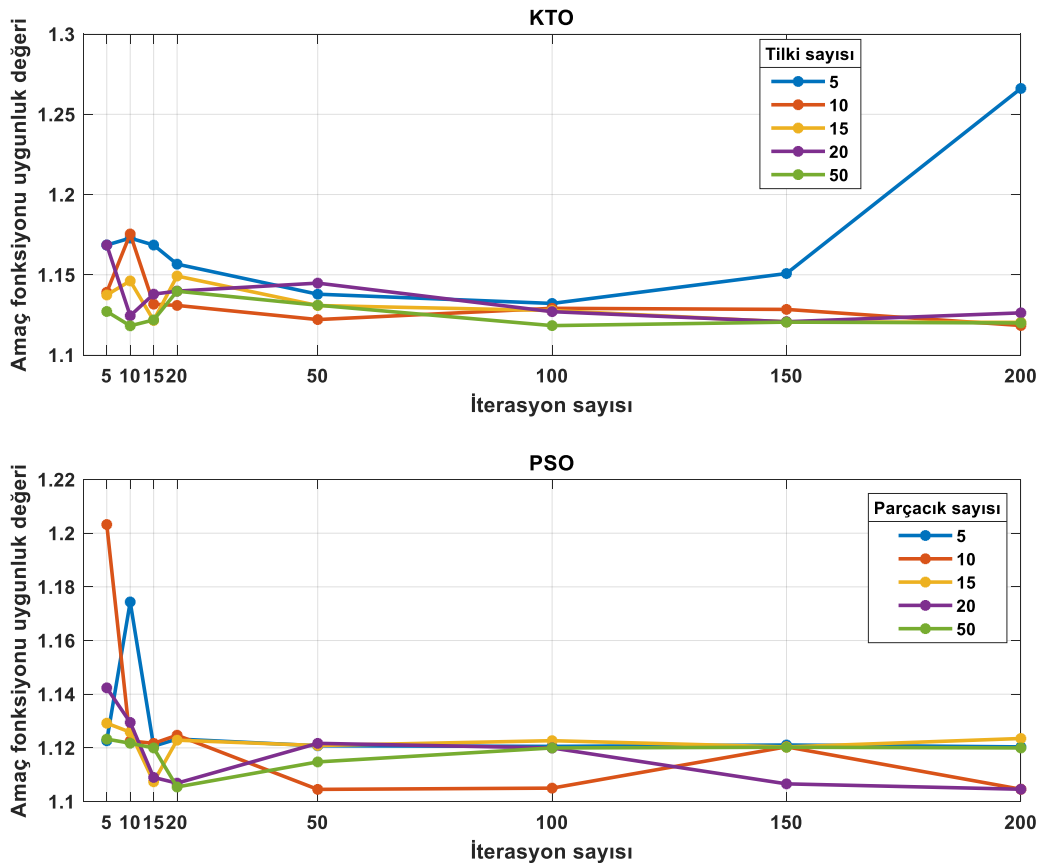
$$ZHKT = \sum_{k=1}^q t. e^2(k) \quad (22)$$

$$\text{Amaç fonksiyonu} = \frac{HMT+HKT+ZHKT}{3} \quad (23)$$

Bu karşılaştırma kıstaslarından amaç fonksiyonu uygunluk değeri, yerleşme zamanı ve yükselme zamanı değerleri her iki algortmada da yakın değerler gelmekteyken, PSO algoritmasının çalışma süresinin KTO algoritmasına kıyasla daha kısa sürdüğü gözlemlenmiştir. Amaç fonksiyonu uygunluk değeri; tilki/parçacık sayısı ve iterasyon sayısına göre sırasıyla KTO ve PSO için Şekil 5 ile verilmiştir. Şekil 5'te görüldüğü gibi KTO algoritması için optimum değerler 50 tilki 10 iterasyon ile elde edilirken, PSO algoritmasında 10 parçacık 50 iterasyon ile elde edilmiştir. Sistemde yapılacak olan testler bu değerler ile gerçekleştirilmiştir. Optimum değerlere ilişkin  $K_p$  ve  $K_i$  değerleri Tablo 2 ile gösterilmiştir.

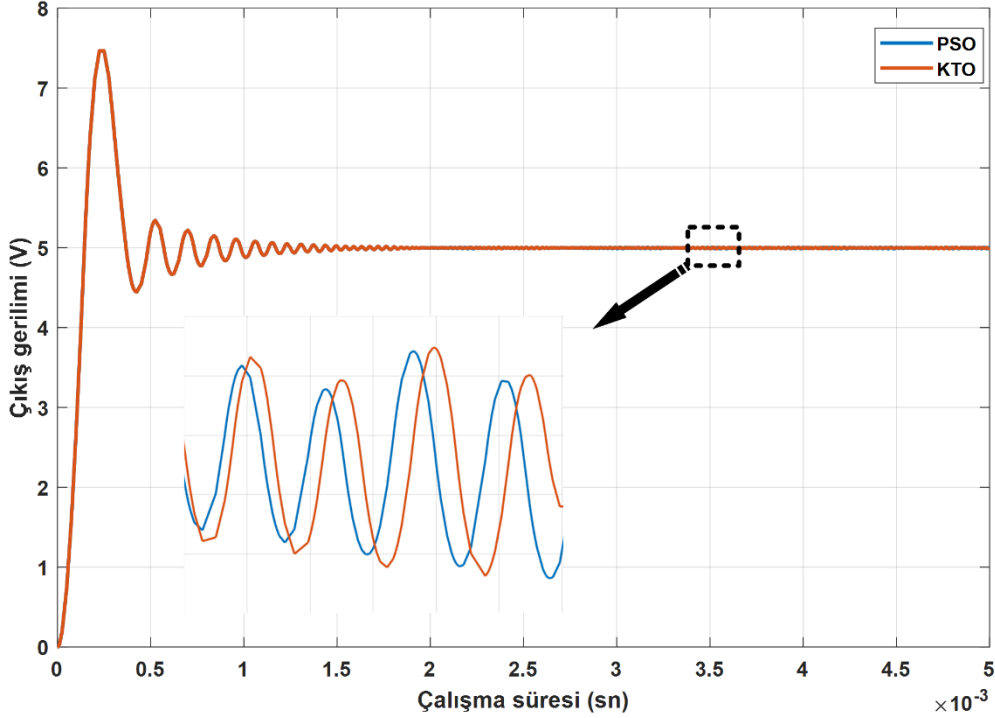
**Tablo 2.** Optimum  $K_p$  ve  $K_i$  değerleri.

	KTO	PSO
$K_p$	367,206	336,237
$K_i$	52,290	0,658

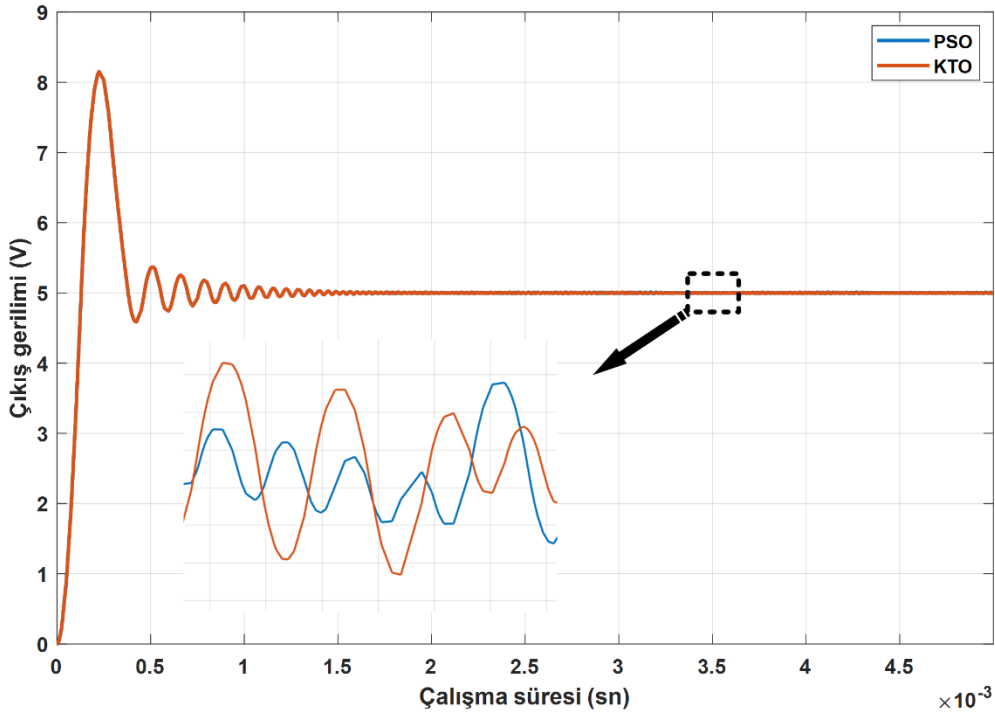


**Şekil 5.** Amaç fonksiyonu uygunluk değerlerinin karşılaştırılması.

Tüm koşullar her iki algoritmada da test edilerek optimum katsayılar bulunmuştur. Seçilen optimum değerlerin sistemdeki toleransını gözlemleyebilmek için bu değerler ile sistem farklı koşullarda test edilmiştir. Yapılan ilk çalışmada giriş gerilim değerine  $\pm\%20$  değişim uygulanarak sonuçlar gözlemlenmiştir. Her üç karşılaştırmada da kayda değer bir fark gözlemlenmemiştir. Burada kullanılan yerleşme zamanı  $\%2$  kriterine göre seçilmiştir. Giriş gerilimi 9, 10,8 ve 7,2 V değerlerinde iken elde edilen grafiksel sonuçlar sırasıyla Şekil 6, 7 ve 8 ile sunulmuştur. Ayrıca sonuçları daha detaylı sunabilmek için performans kriterleri tablolaştırılarak Tablo 3 ile sunulmuştur.



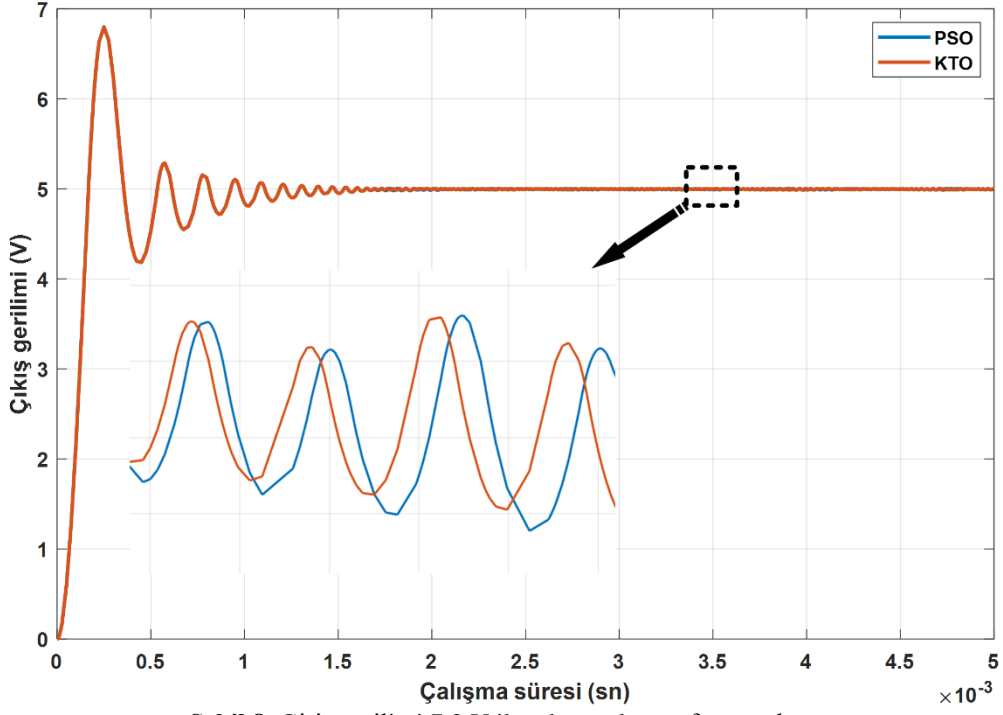
Şekil 6. Normal koşullar altında kontrolör performansları.



Şekil 7. Giriş gerilimi 10,8 V iken kontrolör performansları.

Bir diğer çalışmada çıkış gerilim referansı için  $\pm\%20$  değişim uygulanarak kontrolör performansları test edilmiştir. Çıkış gerilim referansı 4 ve 6 V iken sonuçlar Şekil 9 ve 10 ile verilmiştir. Yine daha detaylı sonuçlar Tablo 4 ile sunulmuştur. Yapılan test için de elde edilen sonuçlar oldukça yakındır.

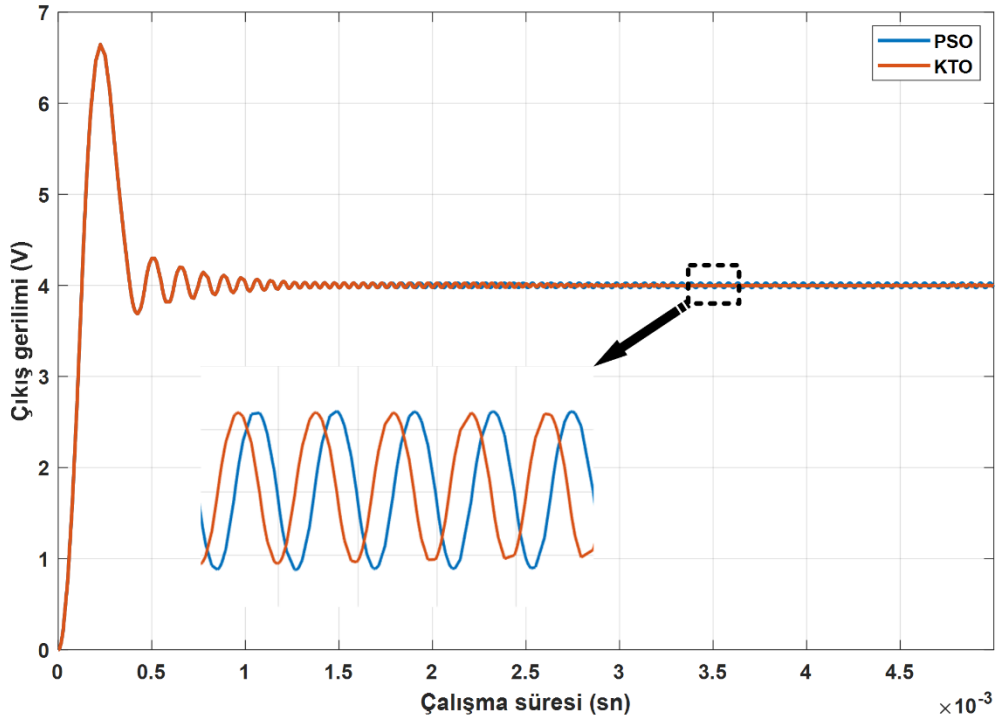




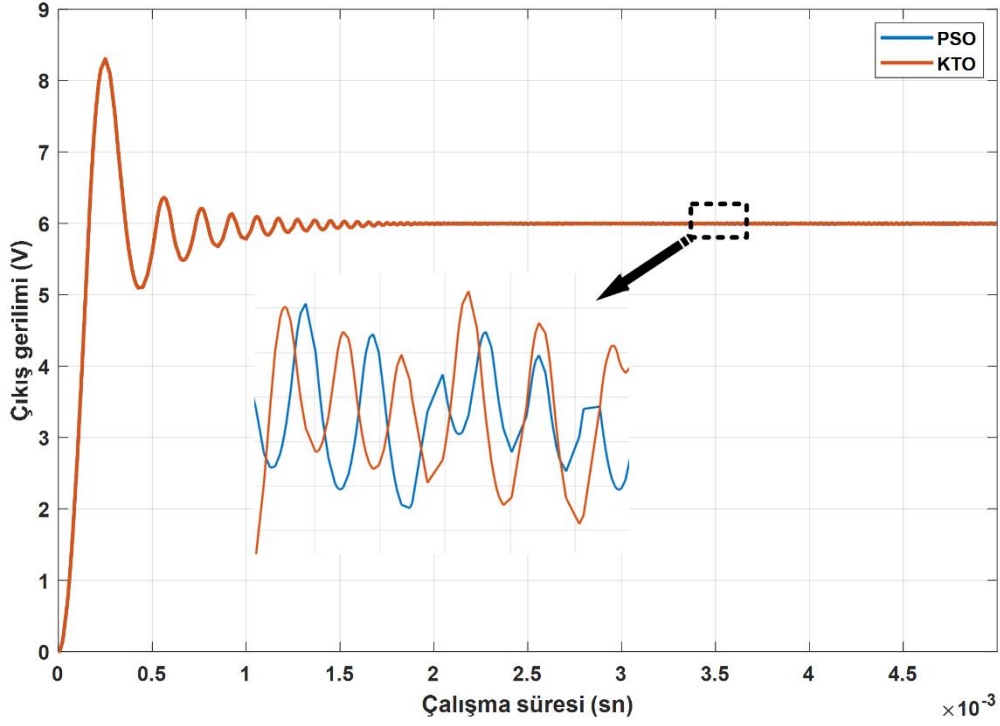
Şekil 8. Giriş gerilimi 7,2 V iken kontrolör performansları.

Tablo 3. Giriş gerilim değerinin  $\pm\%20$  değişimi.

Giriş gerilimi	KTO			PSO		
	Uygunluk değeri	$t_s$ (ms)	$t_r$ ( $\mu$ s)	Uygunluk değeri	$t_s$ (ms)	$t_r$ ( $\mu$ s)
9 V	1,1181	1,054	94,729	1,1045	1,054	94,727
10,8 V	1,1924	0,859	84,990	1,6789	0,868	84,989
7,2 V	1,9416	1,059	109,907	1,9280	1,059	109,905



Şekil 9. Çıkış gerilimi referansı 4 V iken kontrolör performansları.

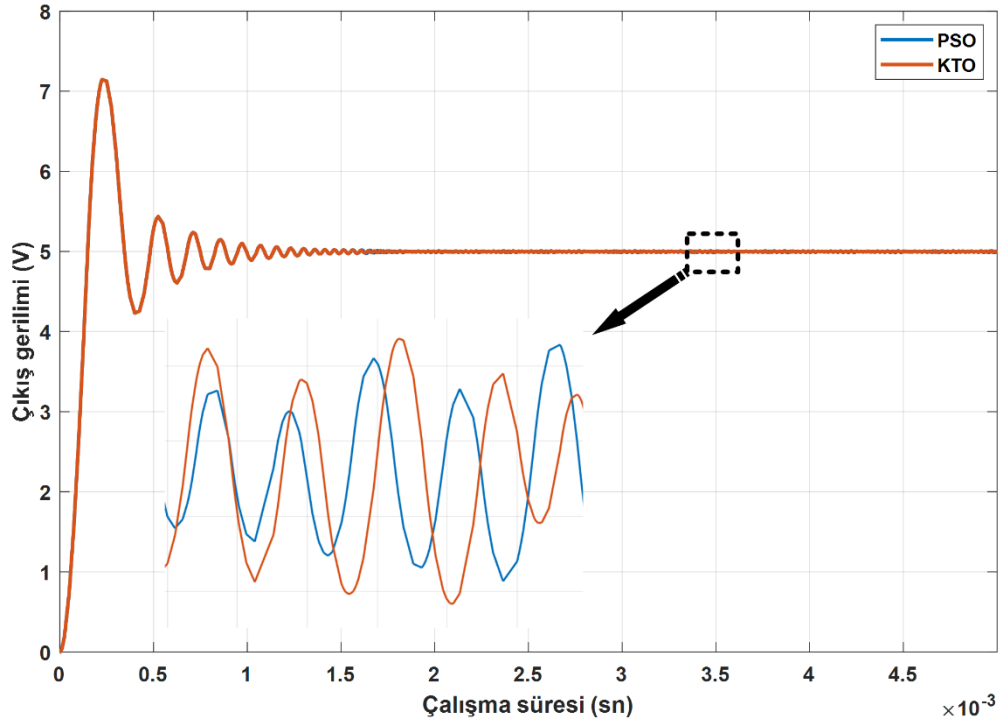


Şekil 10. Çıkış gerilimi referansı 6 V iken kontrolör performansları.

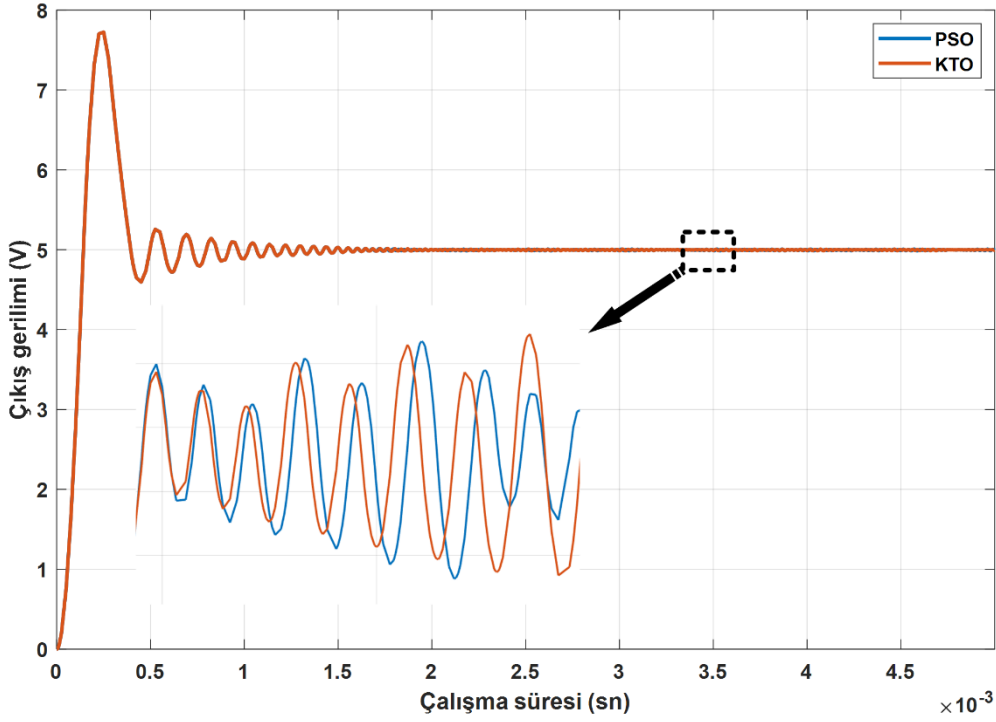
Tablo 4. Çıkış gerilim değerinin  $\pm\%20$  değişimi.

Çıkış gerilimi	KTO			PSO		
	Uygunluk değeri	$t_s$ (ms)	$t_r$ ( $\mu$ s)	Uygunluk değeri	$t_s$ (ms)	$t_r$ ( $\mu$ s)
5 V	1,1181	1,054	94,729	1,1045	1,054	94,727
6 V	2,9770	1,219	106,699	2,5080	1,219	107,296
4 V	0,7181	0,941	82,915	0,7949	0,940	82,913

Yapılan son çalışmada ise giriş gerilimi ve çıkış gerilimi referansı sabit iken yük direncinin  $\pm\%20$  değişimi için kontrolör performansı incelenmiştir. Sonuçlar Şekil 11, 12 ve Tablo 5 ile sunulmuştur.



Şekil 11. Yük direnci 8  $\Omega$  iken kontrolör performansları.

Şekil 12. Yük direnci 12  $\Omega$  iken kontrolör performansları.Tablo 5. Yük direnci değerinin  $\pm\%20$  değişimi.

Yük direnci	KTO			PSO		
	Uygunluk değeri	$t_s$ (ms)	$t_r$ ( $\mu$ s)	Uygunluk değeri	$t_s$ (ms)	$t_r$ ( $\mu$ s)
10 $\Omega$	1,1181	1,054	94,729	1,1045	1,054	94,727
12 $\Omega$	0,8411	0,857	92,945	0,8297	0,945	92,944
8 $\Omega$	1,9633	0,979	97,858	1,7231	0,979	97,856

## 6. SONUÇ

DA-DA alçaltıcı tip dönüştürücü birçok elektronik cihazda bulunmakta olup kararlı bir çıkış gerilimi, bulunduğu sistemin düzgün ve kararlı çalışması için oldukça önemlidir. Bu amaçla kontrolörün önemi artmaktadır. Bu çalışma kapsamında BC için optimum kontrolör katsayılarının elde edilmesi amacıyla literatüre yeni kazandırılan KTO yöntemi kullanılmıştır. Bu amaçla daha önce hiç kullanılmamış olan KTO yönteminin literatüre kazandırılabilmesi ve yeni çalışmalara ışık tutacağı düşünülmektedir. Önerilen yöntem ile kontrolör performansı; giriş geriliminde, yük direncinde ve çıkış gerilim referansında meydana gelebilecek değişikliklere karşı test edilmiştir. Ayrıca elde edilen sonuçlar literatürde en yaygın kullanılan meta-sezgisel yöntemlerden birisi olan PSO yöntemi ile karşılaştırılmıştır. Yapılmak istenen minimizasyon işleminin sonucunda PSO ve KTO algoritmalarının birbirine yakın performans sergilediği gözlemlenmiştir. MATLAB/Simulink ortamında gerçekleştirilen simülasyon işlemlerinin sonucunda PSO algoritması çalışma süresi bakımından KTO algoritmasına göre yaklaşık iki kat daha iyi bir performans sergilemiş ve uygunluk değerini 1,1045 bularak sistemin daha kararlı bir çıkış geriliminin olmasını sağlamıştır.

### Yazar Katkıları

Bütün yazarlar eşit oranda katkı sağlamıştır.

### Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması yoktur.

### KAYNAKÇA

- [1] Y. Atalan Ayaz, "Ambulance Service for Hospital Selection: Optimization with Discrete-Event Simulation Application for Yozgat Province Of Turkey", Mühendislik Bilim. ve Araştırmaları Derg., vol. 3, no. 1, pp. 112–122, 2021.
- [2] G. Mühürücü, A. Mühürücü, E. Köse, and A. Kuyumcu, "Parameter Optimization of PI Controller by PSO for Optimal Controlling of A Buck Converter's Output", in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), pp. 1–6, 2017.

- [3] M. Y. Özsağlam ve M. Çunkaş, "Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması", *J. Polytech.*, vol. 11, no. 4, pp. 299–305, 2008.
- [4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.
- [5] D. Połap and M. Woźniak, "Red fox optimization algorithm", *Expert Syst. Appl.*, vol. 166, no. 114107, 2021.
- [6] E. K. Yaylacı, A. E. Yılmaz, ve H. N. Özdeş, "Geliştirilmiş Parçacık Sürü Optimizasyon Algoritması ile Da-Da Alçaltıcı Tip Dönüştürücünün PI Katsayılarının Optimizasyonu," in 2nd International Conference on Applied Engineering and Natural Sciences, pp. 257–261, 2022.
- [7] Z. Fu, J. An, Q. Yang, H. Yuan, Y. Sun, and H. Ebrahimian, "Skin cancer detection using Kernel Fuzzy C-means and Developed Red Fox Optimization algorithm," *Biomed. Signal Process. Control*, vol. 71, no. 103160, 2022.
- [8] Z. Huo, S. J. Liu, and H. Ebrahimian, "Aircraft Energy Management System Using Chaos Red Fox Optimization Algorithm," *J. Electr. Eng. Technol.*, vol. 17, no. 1, pp. 179–195, 2022.
- [9] M. Zhang, Z. Xu, X. Lu, Y. Liu, Q. Xiao, and B. Taheri, "An Optimal Model Identification for Solid Oxide Fuel Cell Based on Extreme Learning Machines Optimized by Improved Red Fox Optimization Algorithm," *Int. J. Hydrogen Energy*, vol. 46, no. 55, pp. 28270–28281, 2021.
- [10] N. Zhu, X. Liu, Q. Dong, and D. Rodriguez, "Optimization of zero-energy building by multi-criteria optimization method: A case study," *J. Build. Eng.*, vol. 44, no. 102969, pp. 1–13, 2021.
- [11] D. Yousri et al., "Modified Interactive Algorithm Based on Runge Kutta Optimizer for Photovoltaic Modeling: Justification Under Partial Shading and Varied Temperature Conditions," *IEEE Access*, vol. 10, pp. 20793–20815, 2022.
- [12] R. Luo and M. Shafiee, "The Application of Metaheuristics in Optimal Parameter Identification of Solid Oxide Fuel Cell," *Energy Reports*, vol. 7, pp. 2563–2573, 2021.
- [13] D. Połap and M. Woźniak, "Meta-Heuristic as Manager in Federated Learning Approaches for Image Processing Purposes," *Appl. Soft Comput.*, vol. 113, no. 107872, 2021.
- [14] A. Mamizadeh, N. Genc, and R. Rajabioun, "Optimal Tuning of PI Controller for Boost DC-DC Converters Based on Cuckoo Optimization Algorithm," in 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA), pp. 677–680, 2018.
- [15] S. Vadi, F. Batincan Gurbuz, R. Bayindir, and S. Sagioglu, "Optimization of PI Based Buck-Boost Converter by Particle Swarm Optimization Algorithm," in 2021 9th International Conference on Smart Grid (icSmartGrid), pp. 295–301, 2021.
- [16] E. Köse, G. Mühürçü, A. Mühürçü, and B. Sevim, "SFLA Based PI Parameter Optimization for Optimal Controlling of A Buck Converter's Voltage," in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), pp. 1–5, 2017.
- [17] O. Kapukaya, "Adaptif Parçacık Sürü Optimizasyonu ve Karınca Kolonisi Algoritması Kullanarak Öz Ayarlamalı PID Denetleyici Tasarımı," *Fen Bilimleri Enstitüsü, Kahramanmaraş*, 2019.
- [18] H. Gözde, M. C. Taplamacıoğlu, İlhan Kocaarslan, ve M. A. Şenol, "İki Bölge Ara-Isıtmalı Termal Güç Sisteminin Yük-Frekans Kontrolü için Parçacık Sürüsü Optimizasyonu Tabanlı PI-Kontrolör," *Isı Bilim. ve Tek. Derg.*, vol. 30, no. 1, pp. 13–21, 2010.
- [19] O. T. Altınöz and H. Erdem, "Evaluation Function Comparison of Particle Swarm Optimization for Buck Converter," in International Symposium on Power Electronics, Electrical Drives, Automation and Motion, pp. 798–802, 2010.
- [20] H. Aygun and H. Demirel, "Comparison of PSO-PID, FLC and PID in A Circulating Fluidized Bed Boiler," in 7th International Conference on Electrical and Electronics Engineering, pp. 355–359, 2011.
- [21] H. Aygün, "Akışkan Yataklı Buhar Kazanının Yatak Sıcaklığının Parçacık Sürüsü Optimizasyonu Tabanlı PID Kontrolör (PSO-PID) le Kontrolü," *Fen Bilimleri Enstitüsü, Karabük*, 2011.