TOK2026 -ANKARA

IFAC Türkiye

TOK 2026

Otomatik Kontrol Ulusal Toplantısı

Turkish National Meeting on Automatic Control

BİLDİRİLER KİTABI



8–10 Ekim 2026

Ankara Üniversitesi · Ankara

tok2026.ankara.edu.tr

Editörler: Abdurrahim Toktaş · Tufan Kumbasar · Fethi Candan

Kilitlenmeyi Önleyici Fren Sistemleri için Metasezgisel Algoritmalar Kullanılarak Optimize Edilmiş PID Kontrolcü Tasarımı
Design of an Optimized PID Controller for Anti-lock Brake Systems Using Metaheuristic Algorithms

Yunus Tezel, Mumin Tolga Emirler

Kontrol ve Otomasyon Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, İstanbul
{yunus.tezel, emirler}@yildiz.edu.tr

Özetçe

Kilitlenmeyi Önleyici Fren Sistemi (ABS), acil frenleme sırasında tekerlek kilitlenmesini önleyerek araç güvenliğini artırır. Bu çalışmada, MATLAB/Simulink'in yerleşik "Modeling an Anti-Lock Braking System" örneği temel alınarak oluşturulan bir çeyrek araç ABS modelinde tekerlek kayma oranını kontrol etmek amacıyla bir PID kontrolcü tasarlanmıştır. PID kontrolcü kazançları, üç farklı metasezgisel algoritma kullanılarak optimize edilmiştir: Genetik Algoritma (GA), Parçacık Sürüsü Optimizasyonu (PSO) ve Gri Kurt Optimizasyonu (GWO). Her bir algoritmanın performansı, durma mesafesi ve durma süresini içeren özel tasarlanmış bileşik bir maliyet fonksiyonu temel alınarak değerlendirilmiştir. Simülasyon sonuçları, her üç optimizasyon tekniğinin de kontrol performansını önemli ölçüde iyileştirdiğini; GWO'nun ise değişen yol tutunma koşulları altında maliyet fonksiyonunu minimize etmede avantaj sunduğunu göstermektedir. Bu karşılaştırmalı analiz, ABS kontrolünde metasezgisel parametre ayarlama yöntemlerinin etkinliğini vurgulamaktadır.

Abstract

The Anti-lock Braking System (ABS) enhances vehicle safety by preventing wheel lockup during emergency braking. In this study, a PID controller is designed to regulate wheel slip ratio in a quarter-vehicle ABS model implemented using MATLAB/Simulink's built-in "Modeling an Anti-Lock Braking System" example. The PID controller gains are optimized using three different metaheuristic algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Gray Wolf Optimizer (GWO). The performance of each algorithm is evaluated based on a custom composite cost function that includes stopping distance and stopping time. The simulation results show that all three optimization techniques significantly improve control performance, with GWO offering an advantage in minimizing the cost function under varying road adhesion conditions. This comparative analysis highlights the effectiveness of metaheuristic tuning methods for ABS control.

1. Giriş

Kilitlenmeyi Önleyici Fren Sistemleri (ABS), ani veya aşırı frenleme durumlarında tekerlek kilitlenmesini önlemek için tasarlanmış, modern otomotiv güvenliğinin kritik bileşenleridir. ABS, fren basıncını ayarlayarak, özellikle ıslak, buzlu veya engebeli yol yüzeyleri gibi düşük sürtünmeli koşullarda sürücünün direksiyon kontrolünü ve aracın kararlılığını korumasına olanak tanır [1]. ABS'deki temel kontrol stratejisi, tekerlek kayma oranını istenen referans aralığında tutarak optimal boyuna lastik kuvvetinin oluşmasını sağlamak, durma mesafesini en aza indirmek ve aracın manevra kabiliyetini korumaktır.

Parametre ayarları deneme-yanılma yoluyla yapılan Bang-Bang ve PID gibi geleneksel ABS kontrolcüleri, tasarımsal basitlikleri ve pratik uygulanabilirlikleri nedeniyle yaygın olarak kullanılmaktadır [2]. Ancak bu tür geleneksel yöntemler, değişken yol koşulları altında ABS'nin doğrusal olmayan ve zamanla değişen dinamiklerine uyum sağlamakta yetersiz kalabilmektedir. Geleneksel yöntemlerin bu kısıtlarını gidermek amacıyla, son yıllarda literatürde akıllı optimizasyon algoritmalarına yönelik ilgi giderek artmaktadır. Örneğin, Domingues vd. PID kontrollü bir ABS'yi optimize etmek için NSGA-II, MODA ve MSSA gibi çok amaçlı metasezgisel algoritmalar kullanmış ve özellikle yeni önerdikleri OBLI-MSSA yönteminin durma mesafesi ile sürüş konforunu dengelemede üstün bir performans sergilediğini göstermişlerdir [3]. Benzer şekilde, Sejal vd. bir çeyrek araç modelinde PID katsayılarını ayarlamak amacıyla Gri Kurt Optimizasyonu (GWO) algoritmasını kullanmış, GWO ile ayarlanmış PID kontrolcünün geleneksel yöntemlere kıyasla durma mesafesini ve durma süresini önemli ölçüde azalttığını kanıtlamışlardır [4].

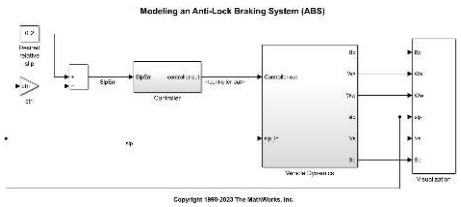
Geliştirilen çeşitli yaklaşımlara rağmen, PID kontrolcüler, gerçek zamanlı gömülü sistemlerde güvenilir ve kolay uygulamabilir bir kontrol stratejisi olarak geçerliliğini korumaktadır. Ancak, elde edilecek kontrol performansı temel olarak PID katsayılarının hassas bir şekilde ayarlanmasına dayanmaktadır. Genetik Algoritma (GA), Parçacık Sürüsü Optimizasyonu (PSO) ve Gri Kurt Optimizasyonu gibi metasezgisel yöntemler; optimizasyon problemlerini gradyan bilgisine ihtiyaç duymadan çözebilme yetenekleri sayesinde, PID katsayılarının ayarlanmasında son derece etkili araçlar sunmaktadır [5].

Bu çalışmada, bir ABS sistemine ait PID kontrol katsayı-

larının ayarlanması amacıyla, GA, PSO ve GWO metasezgisel optimizasyon algoritmalarının karşılaştırmalı bir analizi sunulmaktadır. Gerçekleştirilen simülasyon çalışmalarında, MATLAB/Simulink kütüphanesinde yer alan "Modeling an Anti-Lock Braking System" örneği temel alınmıştır. Literatürdeki mevcut çalışmalarda kontrolcü parametrelerinin optimizasyonu için genellikle kayma oranı referans takip hatasını minimize etmeye odaklanan amaç fonksiyonları tercih edilmektedir [2, 4]. Geleneksel hata takibi yaklaşımından farklı olarak bu çalışmada, sistemin fiziksel performans kriterlerini doğrudan yansıtan durma mesafesi ve durma süresini içeren bileşik bir maliyet fonksiyonu tasarlanmış ve birden fazla algoritmanın etkinliği bu metrik üzerinden karşılaştırmalı olarak analiz edilmiştir. Simülasyon sonuçları tüm algoritmaların kontrol performansını iyileştirdiğini gösterirken, GWO'nun tasarlanan maliyet fonksiyonunu minimize etmede diğer yöntemlere kıyasla daha başarılı bir performans sergilediğini ortaya koymaktadır.

2. ABS Modeli

Değişen yol koşullarının frenleme performansı üzerindeki etkilerini incelemek ve önerilen PID kontrolcü tasarımlarını değerlendirmek amacıyla, literatürde yaygın olarak kullanılan standart bir çeyrek araç ABS modeli tercih edilmiştir. Sistem dinamiklerinin modellenmesinde MATLAB/Simulink standart kütüphanesinde yer alan ABS örneği [6] temel alınmış olup, kurulan simülasyon altyapısına ait blok diyagramı Şekil 1'de sunulmuştur.



Şekil 1: Kilitlenmeyi önleyici fren sistemi MATLAB/Simulink modeli.

Frenleme sürecindeki temel araç dinamiklerini içeren bu çeyrek araç modeli; tekerlek kayması, lastik-yol etkileşimi ve fren eyleyici dinamiği gibi alt sistemleri dikkate almaktadır. Model, fren torkunun tekerlek kayma oranına bağlı olarak sürekli ayarlandığı kapalı çevrim bir kontrol yapısına sahiptir. Bu yapıdaki temel amaç, yol tutunmasını maksimize etmek ve durma mesafesini en aza indirmek için kayma oranını optimal değer (kuru yol koşullarında genellikle 0.2 civarı) etrafında tutmaktır.

Model; tekerlek açısal hızı, araç hızı, kayma oranı ve frenleme kuvveti gibi temel araç değişkenlerini simüle eder. Frenleme kuvveti, μ-kayma eğrisi ve tekerlek üzerindeki dikey yük aracılığı ile hesaplanabilir. Simülasyon çalışmalarında kullanılan çeyrek araç ABS modeline ait temel fiziksel parametreler Tablo 1'de sunulmuştur.

Sonuç olarak bu Simulink altyapısı; farklı PID kontrolcü konfigürasyonlarının test edilmesi ve önerilen metasezgisel optimizasyon algoritmalarının kontrol senaryolarındaki performanslarının karşılaştırmalı olarak analiz edilebilmesi için uygun bir test ortamı sunmaktadır.

Tablo 1: Çeyrek Araç ABS Modeli Fiziksel Parametreleri

Sembol	Parametre	Değer (Birim)
m	Çeyrek Araç Kütlesi	730 kg
I	Tekerlek Atalet Momenti	6.78 kg·m^2
R_r	Tekerlek Yarıçapı	0.38 m
v_0	Başlangıç Hızı	77 km/h
P_{Bmax}	Maksimum Fren Torku	2034 Nm
T_B	Fren Sistemi Zaman Sabiti	0.01 s
g	Yerçekimi İvmesi	9.81 m/s^2
K_f	Sürtünme/Kazanç Katsayısı	1

3. ABS Kontrol Sistemi

Kilitlenmeyi Önleyici Fren Sistemleri'nde hedeflenen optimal yol tutuşunu sağlamak, doğrusal olmayan tekerlek dinamiklerini yönetebilecek kararlı bir kontrol altyapısı gerektirir. Bu doğrultuda, PID kontrolcüler; yapısal sadelikleri, uygulama kolaylıkları ve endüstriyel sistemlerdeki kanıtlanmış güvenilirlikleri nedeniyle halen en yaygın tercih edilen yaklaşımların başında gelmektedir.

PID kontrolcü, hedeflenen kayma oranı ile tekerlekten ölçülen anlık kayma oranı arasındaki hatayı minimize etme prensibiyle çalışır. Kontrolcü çıkış sinyali; oransal, integral ve türev terimlerinin bir kombinasyonu olarak aşağıdaki gibi elde edilmektedir:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \qquad (1)$$

burada $e(t)$, referans kayma oranı ile ölçülen kayma oranı arasındaki fark olarak tanımlanan kontrol hatasını ifade ederken; K_p, K_i ve K_d sırasıyla oransal, integral ve türev kazanç katsayılarını göstermektedir.

PID kazançlarının deneme-yanılma yöntemiyle ayarlanması yaygın bir yaklaşım olsa da; ABS dinamiklerinin doğrusal olmayan ve zamanla değişen yapısı nedeniyle bu durum genellikle optimal olmayan bir kontrol performansıyla sonuçlanmaktadır. Bu nedenle, bu çalışmada PID kontrolcü katsayıları; GA, PSO ve GWO olmak üzere üç farklı metasezgisel optimizasyon yöntemi kullanılarak otomatik olarak ayarlanmıştır. İlgili algoritmalar, kontrolcü kazanç uzayını etkin bir şekilde taramak ve durma mesafesi ile durma süresinin ağırlıklı birleşiminden oluşan maliyet fonksiyonunu minimize etmek amacıyla kullanılmıştır.

Optimize edilmiş PID kontrolcü, tekerlek kayma oranını düzenlemek ve değişen yol koşulları altında tekerlek kilitlenmesini önlemek için fren torkunu dinamik olarak ayarlar. Literatürdeki çalışmalar, metasezgisel yöntemlerle uygun şekilde ayarlanan PID kontrolcülerin, deneme-yanılma tabanlı stratejilere kıyasla ABS performansını önemli ölçüde iyileştirdiğini göstermektedir [7]. Önerilen kontrol yapısına ait maliyet fonksiyonu, durma mesafesi ve durma süresinin toplamı olarak şu şekilde formüle edilmiştir:

$$J = \begin{cases} k_1 \cdot d + k_2 \cdot t + P, & t \geq 15s \\ k_1 \cdot d + k_2 \cdot t, & \text{aksi halde} \end{cases} \quad (2)$$

burada J maliyet fonksiyonunu, d durma mesafesini (m), t durma süresini (s), k_1 ve k_2 ağırlık katsayılarını ve P ise belirlenen ceza değerini temsil etmektedir. Bu çalışmada, her iki metriğin birimsel farklılıklarını dengelemek amacıyla $k_1 = 1$, $k_2 = 10$ ve ceza değeri 500 olarak belirlenmiştir.

4. Metasezgisel Algoritmalar

Metasezgisel algoritmalar, temelini doğadaki süreçlerden veya zeki davranışlardan alan optimizasyon yöntemleridir [9]. Geleneksel analitik yöntemlerin yetersiz kaldığı veya hesaplama maliyetinin çok yüksek olduğu karmaşık problemlerde, makul bir süre içerisinde en iyiye yakın ve yüksek kaliteli çözümleri başarıyla sunarlar. Bu özellikleri sayesinde; karmaşık matematiksel fonksiyonların optimizasyonu, mobil robotlar için yörünge planlaması, mühendislik tasarım problemleri, endüstriyel süreçlerin optimizasyonu, gezgin satıcı problemi ve çizge (graf) tabanlı problemler başta olmak üzere oldukça geniş bir uygulama yelpazesinde yüksek etkinlik göstermişlerdir [8].

Metasezgisel algoritmalar, önceden tanımlanmış bir amaç fonksiyonunu optimize etmek amacıyla çok sayıda iterasyon boyunca çalışır. Bu algoritmalar genellikle rastgele seçilen bir veya birden fazla başlangıç noktasından hareket ederek, daha iyi çözümler keşfetmek üzere arama uzayında bir arama süreci simüle ederler. Bu süreç boyunca potansiyel çözümler üretilir, değerlendirilir ve mevcut en iyi çözümle karşılaştırılır. Belirlenen optimizasyon kriteri sağlanana kadar bu iteratif süreç devam eder.

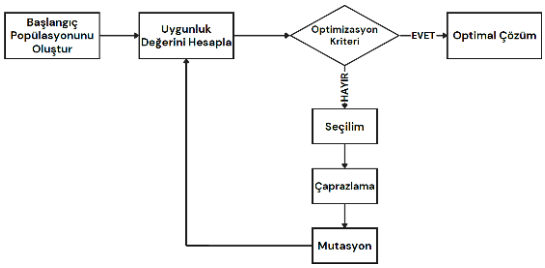
4.1. Genetik Algoritma

Genetik algoritma 1974 yılında J. Holland ve arkadaşları tarafından geliştirilmiştir. Darwin'in evrim teorisini temel alır ve genetikte olduğu gibi en iyi olanın hayatta kalması veya güçlünün zayıfı yenmesi felsefesine dayanır [10]. Doğal evrim sürecinden esinlenen bu algoritma; seçim, çaprazlama ve mutasyon gibi temel genetik operatörleri kullanmaktadır.

Algoritma temelde en uygun bireylerin hayatta kalmasını hedeflese de; rastgele seçilen çözümleri seçim ve çaprazlama operatörleri vasıtasıyla birleştirerek stokastik bir arama süreci yürütür. Doğası gereği paralel bir arama tekniği olan GA, çözüm uzayının tamamını eşzamanlı olarak tarama yeteneğine sahiptir. Bu karakteristik özelliği, algoritmanın yerel minimum veya maksimum noktalara takılmadan küresel en iyi çözümü bulma ihtimalini artırmaktadır.

Algoritmanın adım adım işleyişini özetleyen Şekil 2'deki akış diyagramında görüldüğü üzere süreç, bir popülasyonun başlatılmasıyla başlar. Bu başlangıç popülasyonundaki her birey, önceden tanımlanmış bir uygunluk fonksiyonu üzerinden değerlendirildikten sonra optimizasyon kriteri kontrol edilir. Belirlenen kritere henüz ulaşılamamışsa, popülasyondan bireyleri seçmek için bir seçilim süreci gerçekleştirilir. Bu seçilim aşamasında, daha yüksek uygunluk değerlerine sahip bireylerin seçilme olasılığı daha yüksektir ve aynı birey birden fazla kez seçilebilir.

Seçilim aşaması tamamlandıktan sonra, bireyler rastgele



Şekil 2: Genetik algoritma akış şeması.

çiftler halinde eşleştirilir ve yeni nesiller üretmek üzere çaprazlama işlemi uygulanır. Ardından, yeni oluşturulan bu bireylerdeki her bir gen düşük bir olasılıkla mutasyona uğratılır. Mutasyon operatörünün temel amacı, popülasyondaki genetik çeşitliliği korumak ve algoritmanın arama uzayındaki tek bir nokta etrafında erken yakınsamasını önlemektir. Mutasyon aşamasından sonra, yeni bireylerin uygunluk değerleri hesaplanır ve optimizasyon kriteri yeniden değerlendirilir. Bu iteratif süreç, belirlenen optimizasyon kriteri sağlanana dek devam eder.

4.2. Parçacık Sürüsü Optimizasyonu

1995 yılında Kennedy ve Eberhart tarafından literatüre kazandırılan PSO algoritmasında, her bir parçacığın istenen çözüme yakınlığı bir maliyet fonksiyonu ile belirlenir [11]. Optimal değere ulaşmayı amaçlayan bu algoritmada her parçacık; hem kendi edindiği deneyimleri hem de sürü içerisinde en iyi performansı gösteren parçacığın deneyimini temel alarak durumunu günceller. Bu sürecin bir parçası olarak algoritma, her iterasyonda parçacıkların bireysel en iyi değerini ve sürünün en iyi değerini günceller. Konum güncelleme işlemini gerçekleştirebilmek için her bir parçacığın hızı aşağıdaki gibi hesaplanmaktadır:

$$v_i(t+1) = w \cdot v_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)). \quad (3)$$

Denklem (3)'te $v_i(t)$, i. parçacığın mevcut hızını ifade ederken; $v_i(t+1)$ güncellenmiş hızını göstermektedir. $x_i(t)$, i. parçacığın konumunu belirtmektedir. $pbest_i$, i. parçacığın bireysel en iyi konumunu; $gbest$ ise ilgili iterasyona kadar tüm sürü tarafından bulunmuş olan küresel en iyi konumu ifade eder. Parametrelerden w atalet ağırlığını, c_1 bilişsel katsayıyı ve c_2 sosyal katsayıyı temsil etmektedir. r_1 ve r_2 ise $[0, 1]$ aralığında düzgün dağılımlı rastgele sayılardır.

Atalet ağırlığı w, parçacığın mevcut hareket yönünü korumasını sağlarken; bilişsel bileşen c_1, parçacığın kendi edindiği deneyimlere dayanarak konumunu güncellemesine olanak tanır. Sosyal bileşen c_2 ise parçacığın, sürü içerisinde en iyi performansı gösteren bireyin tecrübesinden faydalanmasına yardımcı olur. r_1 ve r_2 rastgele katsayıları, sürünün yerel bir optimuma erken yakınsamasını engellemede kritik bir rol oynar. Ardından, Denklem (4)'te verilen formül kullanılarak, hesaplanan bu hız değeri parçacığın konumuna eklenir ve böylece yeni çözüm elde edilmiş olur.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

burada $x_i(t+1)$, i. parçacığın güncellenmiş konumunu ifade etmektedir. Bir parçacığın hız vektörü; atalet, bilişsel ve sosyal olmak üzere üç temel bileşenden meydana gelir. Her bir iterasyonda PSO, o ana kadar elde edilen bireysel en iyi değer ile küresel en iyi değeri kullanarak parçacığın hızını günceller. Ardından, elde edilen bu yeni hız değeri parçacığın mevcut konumuna eklenerek güncel konumu belirlenir.

Bir parçacığın hızının aşırı derecede artması durumunda çözüm, arama uzayının dışına çıkabilir. Bu nedenle, parçacık hızları için alt ve üst sınırlar belirlenmesi zorunludur. Her bir parçacığın hızının $[-V_{max}, V_{max}]$ aralığında sınırlandırılması önerilmekte ve V_{max} değerinin, parçacığın alabileceği maksimum konum değerinin 0.1 ile 1.0 katı arasında seçilmesi tavsiye edilmektedir [12]. Eğer bu sınırlandırma uygulanmazsa, Denklem (3)'te yer alan r_1 ve r_2 rastgele ağırlık faktörleri hız değerlerinin kontrolsüzce artmasına yol açabilir ve bu durum, parçacık konumlarının sonsuza doğru ıraksamasına yol açabilir.

PSO algoritmasının işleyişi, önceden tanımlanmış bir optimizasyon kriteri karşılanana kadar devam eder. Sürüye ait başlangıç hızları sıfır olarak ayarlanabileceği gibi rastgele değerler de atanabilir. Denklem (3)'te yer alan bilişsel katsayı c_1 ve sosyal katsayı c_2 değerleri genellikle 2 olarak kabul edilmektedir.

4.3. Gri Kurt Optimizasyonu

GWO, doğadaki gri kurtların (Canis lupus) liderlik hiyerarşisinden ve avlanma mekanizmasından esinlenerek geliştirilmiş popülasyon tabanlı metasezgisel bir algoritmadır. Bu yöntem, 2014 yılında Mirjalili vd. [13] tarafından literatüre kazandırılmıştır. GWO'da kurtların sosyal hiyerarşisi dört temel kategori kullanılarak modellenmektedir: alfa (α), beta (β), delta (δ) ve omega (ω). Optimizasyon sürecinde elde edilen en iyi üç çözüm sırasıyla α, β ve δ olarak kabul edilir ve bu çözümler, arama süreci boyunca popülasyonun geri kalanına (ω) rehberlik eder.

Gri kurtların avı kuşatma, avlanma ve ava saldırma eylemlerini içeren karakteristik avlanma davranışı, arama ajanlarını optimal çözümlere yönlendirmek amacıyla matematiksel olarak modellenmiştir. Avı kuşatma davranışı aşağıdaki gibi formüle edilmektedir:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|. \quad (5)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}. \quad (6)$$

burada $\vec{X}_p$, hedef avın konum vektörünü; $\vec{X}$ ise arama yapan bir gri kurdun anlık konum vektörünü ifade etmektedir. $\vec{A}$ ve $\vec{C}$ ise aşağıdaki gibi hesaplanan katsayı vektörleridir:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}. \quad (7)$$

$$\vec{C} = 2\vec{r}_2. \quad (8)$$

Denklem (7)'de yer alan $\vec{a}$ vektörünün değeri, iterasyonlar boyunca 2'den 0'a lineer olarak azalmaktadır. $\vec{r}_1$ ve $\vec{r}_2$ ise $[0, 1]$ aralığında düzgün dağılımlı rastgele vektörlerdir. Kurulan bu mekanizma, algoritmanın erken iterasyonlarda arama uzayını geniş çaplı taramasına yani keşfetmesine, ilerleyen iterasyonlarda ise elde edilen umut verici çözümleri detaylıca işlemesine yani sömürmesine olanak tanımaktadır.

Avın muhtemel konumunu tahmin edebilmek amacıyla kurtlar; hiyerarşideki en iyi üç kurdun (α, β ve δ) konumlarını referans alarak kendi konumlarını aşağıdaki gibi güncellerler:

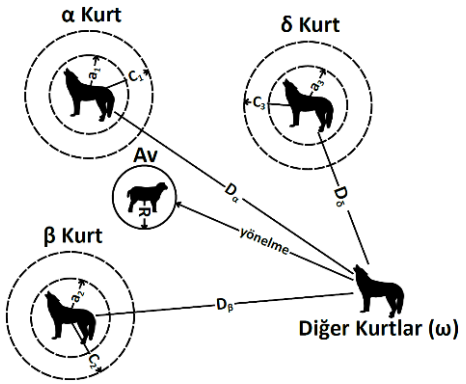
$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1|\vec{C}_1\vec{X}_\alpha - \vec{X}|. \quad (9)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2|\vec{C}_2\vec{X}_\beta - \vec{X}|. \quad (10)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3|\vec{C}_3\vec{X}_\delta - \vec{X}|. \quad (11)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \quad (12)$$

En iyi üç çözümün bu şekilde ortalamasının alınması, tüm arama ajanlarının arama uzayındaki umut verici bölgelere yönlendirilmesini sağlar. Algoritmanın keşif ve sömürü dengesi, adaptif $\vec{a}$ parametresi aracılığıyla dinamik olarak ayarlanmaktadır. GWO'nun genel yapısı Şekil 3'te gösterilmektedir.



Şekil 3: Gri Kurt Optimizasyonu algoritmasında konum güncelleme mekanizması.

Optimizasyon süreci; maksimum iterasyon sayısına ulaşılması veya maliyet fonksiyonunun yakınsaması gibi bir durdurma koşulu sağlanana kadar iteratif olarak devam eder. Sonlandırma kriteri başarıyla sağlandığında, alfa (α) kurdu tarafından elde edilen çözüm, optimal sonuç olarak kabul edilir.

5. Benzetim Çalışmaları

Bu çalışmada, Kilitlenmeyi Önleyici Fren Sisteminin kontrolü için Genetik Algoritma, Parçacık Sürüsü Optimizasyonu ve Gri Kurt Optimizasyonu yöntemleri kullanılarak optimize edilmiş PID kontrolör tasarımları sunulmuş ve sistem performansları karşılaştırmalı olarak analiz edilmiştir. Kontrolcü performansını artırmak amacıyla, yalnızca durma mesafesine odaklanmak yerine durma mesafesi ve durma süresini eşzamanlı olarak minimize eden bileşik bir maliyet fonksiyonu tanımlanmıştır. Dikkat edilmesi gereken en önemli nokta, algoritmaların optimizasyon sürecinin ve PID katsayılarının bulunmasının yalnızca kuru asfalt (yüksek tutunma) koşulları altında gerçekleştirilmiş olmasıdır. Adil bir karşılaştırma yapabilmek adına her üç algoritmada da popülasyon büyüklüğü 10 olarak belirlenmiş ve optimizasyon süreci 20 iterasyon boyunca çalıştırılmıştır. Ayrıca, arama uzayını sınırlandırmak ve fiziksel olarak uygulanabilir sonuçlar elde etmek amacıyla PID katsayıları $[0.01, 200]$ aralığında

kısıtlanmıştır. Bu süreç sonucunda her bir algoritma için elde edilen en iyi PID katsayıları Tablo 2'de sunulmaktadır. Tabloda PSO'nun K_i katsayısında alt sınıra yakınsayarak sistemi pratikte bir PD kontrolcüye dönüştürdüğü görülmektedir. Ancak bu durum algoritmanın karakteristik bir davranışı değil, yalnızca en yüksek performansı sergileyen çalıştırmanın sonucudur.

Tablo 2: PID Katsayıları

	GA	PSO	GWO
$\mathbf{K_p}$	155.2399	200	168.6651
$\mathbf{K_i}$	156.5278	0.01	118.8754
$\mathbf{K_d}$	16.1016	26.5293	25.1308

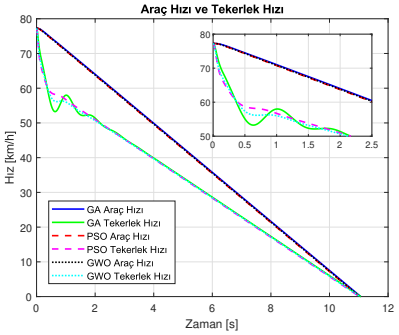
Elde edilen sonuçlar Tablo 3 üzerinden analiz edildiğinde; 5 bağımsız çalışmanın ortalamasına göre PSO ve GWO algoritmalarının maliyet fonksiyonunu minimize etmede birbirine oldukça yakın ve GA'ya kıyasla çok daha iyi bir performans sergilediği açıkça görülmektedir. Özellikle GA'nın ilk iki denemesinde maliyet değerinin 1000 sınırını aşmasının temel nedeni, aracın hedeflenen 15 saniye içerisinde durmayı başaramayarak maliyet fonksiyonundan 500 ceza puanı almış olmasıdır. Çalışma süreleri bakımından incelendiğinde ise GA ve GWO algoritmalarının işlem sürelerinin birbirine yakın seviyelerde olduğu görülmektedir. Bununla birlikte, hem elde edilen maliyet değeri hem de hesaplama süresi eşzamanlı olarak değerlendirildiğinde GWO algoritması genel performans açısından öne geçmektedir.

Tablo 3: Optimizasyon Algoritmaları için Maliyet Değeri ve Çalışma Süresi

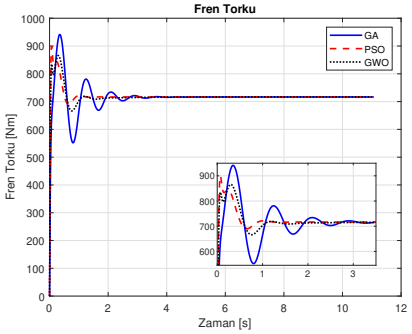
GA		PSO		GWO		
Süre	**Maliyet**	**Süre**	**Maliyet**	**Süre**	**Maliyet**	
93.47	1146.96	97.69	599.94	90.39	599.54	**1**
90.84	1149.90	95.51	599.15	90.66	601.38	**2**
90.52	602.12	95.71	598.92	90.76	599.26	**3**
90.43	602.41	95.65	599.76	90.93	599.58	**4**
90.56	603.45	95.66	599.12	90.38	599.57	**5**
91.16	820.97	96.05	599.38	90.62	599.86	**Ort.**

Optimize edilen PID katsayılarının simülasyon sonuçları ilk olarak tasarlandığı kuru asfalt zemini için değerlendirilmiştir. Kuru zeminde zamana karşı araç ve tekerlek hızını gösteren Şekil 4, fren torku değişimlerini ifade eden Şekil 5 ve toplam duruş mesafesini sunan Şekil 6, tüm algoritmaların kararlı bir frenleme sağladığını doğrulamaktadır. PSO ve GWO, referans hız takibinde pürüzsüz bir dinamik sistem tepkisi oluştururken, fren torkundaki istenmeyen osilasyonları GA'ya kıyasla ciddi oranda sönümleyerek frenleme performansını iyileştirmiştir.

Çalışmanın kritik aşamalarından biri olan ve kontrolörün değişen zemin şartlarına karşı dayanıklılığının test edildiği aşamada, sadece kuru asfalt senaryosuna göre elde edilen bu PID katsayıları doğrudan ıslak asfalt (düşük tutunma) modeline uygulanmıştır. Bu yaklaşımın temel sebebi, yüksek tutunmalı kuru zeminde ayarlanan PID katsayılarının, zorlu ve kaygan yol koşullarında da tekerlek kilitlenmesini önleyerek başarılı bir sonuç verdiğini kanıtlamaktır. Islak zemin senaryosuna ait araç ve tekerlek hızını gösteren Şekil 7, fren torku değişimlerini sunan Şekil 8 ve durma mesafesini ifade eden Şekil 9 incelendiğinde,



Şekil 4: Kuru asfaltta araç ve tekerlek hızının zamana göre değişimi.



Şekil 5: Kuru asfaltta fren torkunun zamana göre değişimi.

kontrolcülerin sistem kararlılığını başarıyla sürdürdüğü görülmektedir. Özellikle GWO ve PSO, belirgin şekilde düşen yüzey tutunmasına rağmen araç dengesini kaybetmeden etkin ve güvenilir bir frenleme sağlamıştır.

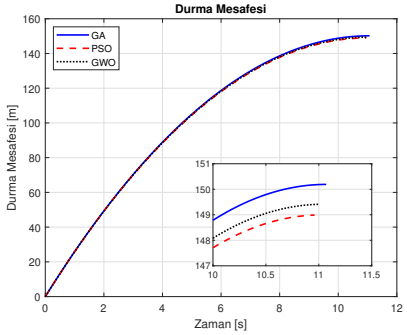
6. Sonuçlar

Gerçekleştirilen karşılaştırmalı analizler, ABS kontrolünde GA, PSO ve GWO algoritmalarının PID katsayılarını başarılı bir şekilde optimize ettiğini kanıtlamaktadır. Çözüm kalitesi açısından GWO'nun PSO ile eşdeğer ve GA'ya kıyasla üstün bir performans sergilediği; çalışma hızı bakımından ise PSO'ya karşı belirgin bir avantaj sağladığı görülmektedir. Bunun yanı sıra, yalnızca kuru zemin koşullarına göre optimize edilen kontrolcünün ıslak zeminde de değişen şartlara uyum sağlayarak sistem kararlılığını koruması, bu yöntemin araç güvenliği uygulamaları için elverişli bir strateji olduğunu ortaya koymaktadır.

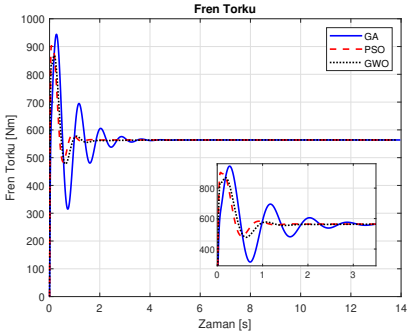
7. Kaynakça

[1] R. Bosch GmbH, *Bosch Automotive Handbook*, 8th ed. Cambridge, MA, USA: Bentley Publishers, 2011.

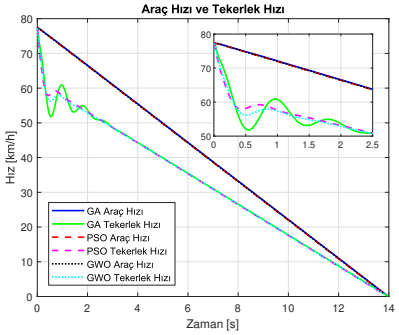
[2] O. Mohammed, H. Metered, and T. Abdelaziz, "Optimi-



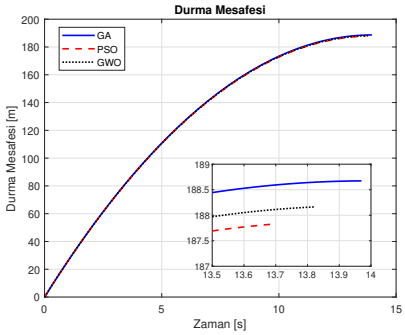
Şekil 6: Kuru asfaltta durma mesafesi karşılaştırması.



Şekil 8: Islak asfaltta fren torkunun zamana göre değişimi.



Şekil 7: Islak asfaltta araç ve tekerlek hızının zamana göre değişimi.



Şekil 9: Islak asfaltta durma mesafesi karşılaştırması.

zed PID Controller Using Genetic Algorithm for Anti-lock Brake System," *SAE Technical Paper 2023-01-0696*, 2023, doi: 10.4271/2023-01-0696.

[3] P. H. L. S. P. Domingues, R. Z. Freire, L. S. Coelho, and H. V. H. Ayala, "Bio-Inspired Multiojective Tuning of PID-Controlled Antilock Braking Systems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Bari, Italy, 2019, pp. 888–895.

[4] S. Sejal, S. Kumar, and R. M. U. Bosch, "Study of Anti-Lock Braking System Using GWO Tuned PID Controller," in *Proc. IEEE Renew. Energy Sustain. E-Mobility Conf. (RESEM)*, 2023.

[5] S. B. Joseph *et al.*, "Metaheuristic Algorithms for PID Controller Parameters Tuning: Review, Approaches and Open Problems," *Heliyon*, vol. 8, no. 10, e09399, 2022.

[6] MathWorks, "Modeling an Anti-Lock Braking System," MathWorks Simulink Documentation. [Online]. Available: https://www.mathworks.com/help/simulink/slref/modeling-an-anti-lock-braking-system.html [Accessed: August 12, 2026].

[7] V. Chopra, S. K. Singla, and L. Dewan, "Comparative Analysis of Tuning a PID Controller using Intelligent Methods," *Acta Polytechnica Hungarica*, vol. 11, no. 8, pp. 235–249, 2014.

[8] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. Frome, UK: Luniver Press, 2010.

[9] D. Karaboğa, *Yapay Zeka Optimizasyon Algoritmaları*. Ankara, Türkiye: Nobel Akademik Yayıncılık, 2020.

[10] G. Bekdaş, S. K. Kara, and A. T. Atay, *Meta-Sezgisel Optimizasyon Yöntemleri ve Mühendislik Uygulamaları*. Ankara, Türkiye: Seçkin Yayıncılık, 2021.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[12] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. Part I: background and development," *Natural Computing*, vol. 6, no. 4, pp. 467–484, 2007.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014.