

Çift Rotorlu MIMO Sistemi için Sürü Zekâsı Tabanlı LQGI Kontrolcü Tasarımı

Mehmet BELİ¹ , Handan GÜZ² / 9¹DEMİR² 

*^{1,2}İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği,
İskenderun, Hatay

(Alınış / Received: 05.08.2025, Kabul / Accepted: 25.08.2025, Online Yayınlanma / Published Online: 30.08.2025)

Anahtar Kelimeler

Çift rotorlu MIMO sistem,
TRMS,
LQG,
Meta-sezgisel algoritma,
Sürü zekâsı tabanlı

Öz: Bu çalışmada, çift rotorlu çok girişli çok çıkışlı (MIMO) sistem (TRMS) üzerinde, integral etkili Doğrusal-Karesel-Gauss (LQGI) optimal kontrol yaklaşımı uygulanmış; kontrolcü parametre optimizasyonu için popülasyon tabanlı meta-sezgisel algoritmaların sürü zekâsı tabanlı sınıfı değerlendirilmiştir. Esinlenen doğal hayvan topluluklarının hareketlerine göre dört ana gruptan seçilen Gri Kurt Algoritması (GWO), Balina Optimizasyon Algoritması (WOA), Yapay Arı Kolonisi (ABC) ve Parçacık Sürü Optimizasyonu (PSO), LQGI kontrolcünün Q ve R parametrelerinin eniyilemesinde kullanılmıştır. Ayrıca, sistemin geçici rejim davranışı, izleme başarımı ve kontrol sinyali etkinliğini birlikte değerlendiren çok amaçlı bir maliyet fonksiyonu tanımlanarak optimizasyon süreci yönlendirilmiştir. Geliştirilen kontrolcülerin performansı, optimize edilmemiş LQGI kontrolcüyle karşılaştırılarak değerlendirilmiştir. Sonuçlar, önerilen sürü zekâsı tabanlı algoritmaların hem yükseliş (pitch) hem de sapma (yaw) hareketlerinde yörünge takibinde maksimum aşımı, yükselme ve yerleşme sürelerini azaltarak performansı geliştirdiğini ve ani darbeye karşı sistemin kararlılığını artırdığını göstermektedir. Çalışmadan elde edilen sonuçlara göre sürü zekâsı tabanlı optimizasyon algoritmalarının, gelişmiş kontrolcü tasarımı için etkili ve esnek bir araç olduğu ortaya konmuştur.

Design of a Swarm Intelligence-Based LQGI Controller for a Twin Rotor MIMO System

Keywords

Twin rotor MIMO System,
TRMS,
LQG,
Metaheuristic algorithm,
Swarm intelligence-based

Abstract: In this study, the swarm intelligence-based class of population-based metaheuristic algorithms was evaluated for parameter optimization of the Linear Quadratic Gaussian with Integral action (LQGI) optimal control approach applied to the twin rotor multi-input multi-output (MIMO) system (TRMS). Based on the behavior of naturally inspired animal groups, four main algorithms were selected: Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Artificial Bee Colony (ABC), and Particle Swarm Optimization (PSO), which were employed to optimize the Q and R parameters of the LQGI controller. In addition, a multi-objective cost function that takes into account the system's transient response behavior, tracking performance, and control signal efficiency was defined to guide the optimization process. The performance of the developed controllers was evaluated by comparing them with a non-optimized LQGI controller. The results indicate that the proposed swarm intelligence-based algorithms improve performance in both pitch and yaw motions by reducing maximum overshoot, rise time, and settling time in trajectory tracking, while also enhancing the system's stability against sudden disturbances. According to the findings of the study, it was demonstrated that swarm intelligence-based optimization algorithms are effective and flexible tools for advanced controller design.

*İlgili Yazar, email: handan.gursoy@iste.edu.tr

1. Giriş

Son yıllarda, insansız hava araçları (İHA) ve helikopter gibi hava araçlarının savunma sanayinde giderek yaygınlaşması, bu karmaşık sistemlerin güvenli ve etkili şekilde kontrol edilmesi ihtiyacını ortaya çıkarmıştır. Bu nedenle, bu sistemlerin dinamik davranışlarının modellenmesi ve gelişmiş kontrol yöntemlerinin geliştirilmesi önem kazanmıştır. Bu modellerden biri olan Çift Rotorlu Çok Girişli-Çok Çıkışlı Sistem (ing. Twin Rotor MIMO System - TRMS), Feedback Instruments Ltd. tarafından geliştirilmiş olup, bir helikopterin çok girişli, çok çıkışlı (MIMO) yapısını ve karmaşık dinamik etkileşimlerini basitleştirilmiş biçimde temsil eden fiziksel bir sistemdir.

TRMS modelinin kontrolünü karmaşıklaştıran en önemli etkenlerden biri, sistemin sahip olduğu çok girişli-çok çıkışlı yapısıdır. Yükseliş (pitch ve sapma (yaw) birbirine doğrudan bağlıdır; bu yüzden bir eksenindeki değişiklik, diğer eksen etkilemektedir. Bu çapraz etkiler, kontrol stratejilerinin tasarımını zorlaştırmakta ve klasik tek girişli-tek çıkışlı (SISO) kontrol yöntemlerinin yetersiz kalmasına neden olmaktadır. Ayrıca, TRMS'in rotor dinamikleri, sistemdeki gürültü ve dış etkilere karşı duyarlılığı gibi faktörler de kontrol problemini daha karmaşık hale getirmektedir. Bu nedenle, TRMS gibi MIMO sistemlerin kontrolünde değişkenler arası etkileşimlerin doğru modellenmesi ve bu karmaşıklığa uygun, çok değişkenli kontrol yöntemlerinin uygulanması gerekmektedir. Bu karmaşık yapıyı kontrol altına alabilmek için literatürde klasik yöntemler yanında gelişmiş çeşitli kontrolcü tasarımları da önerilmiştir [1-6].

Klasik kontrol yöntemlerinden en bilinen PID methodu, TRMS kontrolünde de sıklıkla kullanılmıştır. Sivadasan ve Iruthayarajan, TRMS sisteminde nonlinear PID kontrolcü parametrelerini evrimsel algoritmalarla optimize etmiş ve PSO algoritmasının diğer yöntemlere göre daha iyi performans sağladığını göstermiştir [1]. Benzer şekilde, Takeş ve ark. tarafından yapılan çalışmada PI-PD denetleyici, klasik PID'ye göre daha iyi geçici yanıt verdiği vurgulanmıştır [2]. Bunun yanında, Abukan ve Almalı, FOPID ve FOSTSMC gibi gelişmiş kontrol yöntemleriyle TRMS'in doğrusal olmayan yapısına çözüm sunmuştur [3].

Klasik ve gelişmiş kontrol yöntemlerinin yanı sıra, optimal kontrol yaklaşımları da TRMS'in karmaşık yapısına çözüm üretmek amacıyla yaygın olarak uygulanmaktadır. Bu kapsamda, LQR ve LQI tabanlı tasarımların, hem geçici hem kararlı durumda üstün performans sağladığı çeşitli çalışmalarla ortaya konmuştur [4-6]. Daha ileri düzey bir yaklaşım olan LQG kontrol, hem durum geri beslemesi hem de rastgele gürültü altındaki tahmin problemi için optimal bir çözüm sunması nedeniyle literatürde öne çıkan yöntemlerden biri olmuştur. Geetha Madhuri ve Rao ise, LQG ve integral etkili LQGI yapılarını karşılaştırmış ve LQGI'nin daha iyi performans sağladığını raporlamıştır [7]. Ayrıca, Guardeno ve ark. tarafından quadrotor sistemleri için LQR/LQG temelli MIMO PID kontrolcü tasarımı yapılmış, bu yaklaşım sensör hataları, model belirsizlikleri ve dış bozuculara karşı dayanıklılık sağlamıştır [8].

Son yıllarda, literatürdeki çalışmalar dikkate alındığında TRMS kontrolü için durum geri beslemeli bir optimal kontrol yaklaşımı olan LQG kontrolcü yöntemi tercih edilmiştir. LQR/LQG gibi kontrol yöntemlerinde ağırlık matrisleri Q ve R matrislerinin seçimi, kontrol performansını doğrudan etkiler. Bu nedenle, son yıllardaki çalışmalarda çok sayıda meta-sezgisel optimizasyon algoritması, bu problemi çözmek için önerilmiştir. Örneğin Katebi ve ark., dalgacık tabanlı LQR kontrolünü farklı algoritmalarla optimize etmiş ve en iyi sonucu İmparatorluk Rekabeti Algoritması ile elde etmiştir [9]. Benkhoud ve Bouallague insansız hava aracı sistemine uyguladıkları LQG kontrolöründe en iyi sonucu Su Döngüsü Optimizasyon Algoritması ile almışlardır [10]. Batista ve ark. manipülatör için LQG+PSO kombinasyonunun performans açısından diğerlerine üstün olduğunu göstermişlerdir [11]. Lin ve ark., MR sönümleyicilerle deprem titreşimlerini azaltmak için LQG kontrolörünü Balina Optimizasyonu Algoritması (ing. Whale Optimization Algorithm - WOA) ile optimize edip ANFIS ile destekleyerek en iyi sonuca ulaşmıştır [12]. Kumar ve ark. tarafından yapılan çalışmada, TRMS için LQG kontrolörün Q ve R parametreleri B Dovžan, D., Logar, V., & Škrjanc, I. (2016). Evolving fuzzy model (eFuMo) method for on-line fuzzy model learning with application to monitoring system. Simulation Notes Europe, 26(4), 205–220. Bakteri Beslenme Optimizasyonu algoritmasıyla optimize edilmiştir [13]. Souza ve ark. relüktans hız kontrolünde LQG ve PID kontrolörlerini meta-sezgisel yöntemlerle optimize etmiş ve en iyi sonucu LQG – Genetik Algoritma (GA) almışlardır [14]. Önen ve ark. çalışmada, çift sarkaçlı ters sarkaç sistemi için LQR kontrolcüsünün parametreleri GA, PSO ve ABC algoritmalarıyla optimize edilip karşılaştırılmış; GA en hızlı, ABC ise en düzgün kontrol tepkisini sağlamıştır [15]. Rivera-Ruiz ve arkadaşları, iki tekerlekli denge robotu için LQR kontrolcüsünün optimizasyonunda diferansiyel evrim algoritmasının diğer meta-sezgisel yöntemlere göre en iyi performansı gösterdiğini ortaya koymuştur [16].

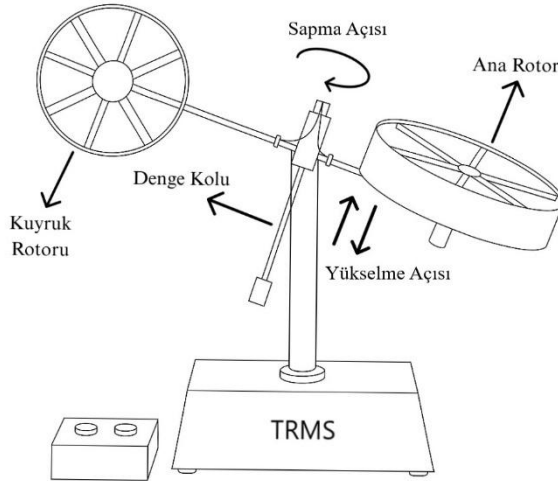
Bu çalışmada, optimizasyon aşamasında popülasyon tabanlı meta-sezgisel algoritmaların bir alt sınıfı olan sürü zekâsı tabanlı optimizasyon yöntemleri ele alınmıştır. Büyük bir kısmı doğadan ilham alan meta-sezgisel

algoritmalar, problemi çözmek veya keşif süreci yürütmek için kullanılan, optimum çözümü garanti etmeyen ancak pratikte yeterli doğrulukta sonuçlar üretebilen yöntemlerdir [17, 18]. Literatürde metasezgisel algoritmalar; Evrimsel Tabanlı (ör. GA), Sürü Zekâsı Tabanlı (ör. PSO), İnsan Davranışı Tabanlı (ör. Sosyal Davranış Algoritmaları) ve Fiziksel Süreç Tabanlı (ör. Simüle Edilmiş Tavlama Algoritması) olmak üzere dört ana gruba ayrılmaktadır [19]. Son yıllarda, özellikle hayvan davranışlarından ilham alan sürü zekâsı tabanlı algoritmalar, literatürde etkili sonuçlar vermiş ve bu başarı, çok sayıda yenilikçi algoritmanın geliştirilmesine de zemin hazırlamıştır. Bu bağlamda, çalışmamızda sürü zekâsı tabanlı algoritmalar tercih edilmiştir; çünkü bu yöntemler, bireylerin etkileşiminden doğan kolektif zekâyı kullanarak, TRMS gibi MIMO ve dinamik sistemlerde güçlü bir optimizasyon kabiliyeti sunmaktadır. Söz konusu algoritmalar, esinlendikleri hayvan topluluklarının doğal yaşam ortamları ve hareket dinamiklerine göre genel olarak dört alt grup olarak ele alınmıştır. TRMS sistemi üzerinde uygulanacak LQGI kontrolcü parametrelerini optimize etmek için her gruptan bir adet temsilci optimizasyon algoritması seçilmiştir. Bu kapsamda; kullanılan algoritmalar Gri Kurt Algoritması (ing. Grey Wolf Optimizer - GWO), Balina Optimizasyon Algoritması (ing. Whale Optimization Algorithm - WOA), Yapay Arı Kolonisi (ing. Artificial Bee Colony - ABC) ve Parçacık Sürü Optimizasyonu (ing. Particle Swarm Optimization - PSO) olarak belirlenmiştir. Sonuçlarda, herhangi bir optimizasyon algoritması kullanılmadan elde edilen nominal LQGI yaklaşımının sonuçlarına göre yükseliş ve sapma hareketlerinde yörünge takibinin çok daha kısa sürede gerçekleştirilebildiğini gözlemlenmiştir.

Bu çalışmada, 2. bölümde TRMS sisteminin genel matematiksel modeli verilmiştir. Bölüm 3'te, LQG ve LQGI kontrolcü temellerini açıklanırken, Bölüm 4'te sürü zekâsı tabanlı optimizasyon algoritmalarından seçilen dört yöntemin ayrıntılarına yer verilmiştir. Bölüm 5'te ise simülasyon sonuçları sunulmuş ve elde edilen bulgular tartışılmıştır. Sonuç bölümünde ise çalışma kapsamında elde edilen genel değerlendirmelere yer verilmiştir.

2. TRMS Matematiksel Modeli

TRMS, yatay düzlemde serbestçe hareket edebilen bir çubuk üzerine yerleştirilmiş ana rotor ve kuyruk rotorundan oluşur ve genel yapısı Şekil 1'de verilmiştir. Her iki rotor da birer doğru akım motoruyla çalıştırılmaktadır. Sistemin merkezine yerleştirilmiş bir karşı ağırlık (denge kolu) sayesinde sistem dengede tutulmakta ve sönümlleme etkisi sağlanmaktadır. Modelin dikey eksenindeki açısal hareketi yükseliş açısı, yatay eksenindeki açısal hareketi ise sapma açısı olarak tanımlanmaktadır.



Şekil 1. TRMS sistemi genel yapısı

TRMS sisteminde ana ve kuyruk rotorlarının hareketi, karşılıklı etkileşimler ile modellenmiştir. TRMS modelinin rotorları, eş kabul edilen birer doğru akım (DA) motorla çalıştırılmaktadır ve motorların eşit oldukları varsayılmıştır. DA motorlar, kontrol sinyalinden sistemin fiziksel tepkisine giden ilk bağlantıdır. Elektriksel zaman sabiti ihmal edilerek sadeleştirilmiş modeli şu şekildedir:

$$v - k_v \omega - Ri = 0 \quad (1)$$

Sistem dört adet rijit ile modellenir: ana rotor, kuyruk rotoru, gövde ve pivot. Kinetik ve potansiyel enerji ifadeleri kullanılarak, Euler-Lagrange yöntemiyle hareket denklemleri türetilir.

Yükseliş Açısı Hareket Denklemi :

$$J_p \ddot{\theta} + (M_y L_m g + M_t L_t g + M_b L_b g + M_c L_c g) \cos \theta + (K_{F_m} r_m \omega_m + K_{F_t} r_t \omega_t) \cos \theta + (b_m + b_t) \dot{\theta} = 0 \quad (2)$$

Bu denklemde; J_p yükseliş hareketi için toplam atalet momentini, θ yükseliş açısı, M_y ana rotorun kütlesi, M_t kuyruk rotorunun kütlesi, M_b gövdenin kütlesi ve M_c karşı ağırlığın kütlesi olarak kullanılmıştır. L_m, L_t, L_b ve L_c ise sırasıyla ana rotor, kuyruk rotoru, gövde ve karşı ağırlığın ağırlık merkezine olan uzaklıklarını belirtir. g yerçekimi ivmesi, K_{F_m} ana rotorun kaldırma katsayısı, K_{F_t} ise kuyruk rotorunun kaldırma katsayısıdır. r_m ve r_t , ana ve kuyruk rotoru için moment kolunu (veya yarıçapı) ifade eder. ω_m ana rotorun, ω_t ise kuyruk rotorunun açısal hızıdır. b_m ve b_t , sırasıyla ana ve kuyruk rotorlarının sönümleme katsayılarıdır. Son olarak, $\dot{\theta}$ yükseliş açısının açısal hızı ve $\ddot{\theta}$ yükseliş açısının açısal ivmesi de modelde yer almaktadır.

Sapma Açısı Hareket Denklemi :

$$J_y * \ddot{\psi} + (K_{F_m} * r_m * \omega_m - K_{F_t} * r_t * \omega_t) * L_m * \cos(\theta_{pitch}) + b_y * \dot{\psi} = 0 \quad (3)$$

Bu denklemde, J_y sapma hareketi için toplam atalet momenti, ψ sapma açısı, $\dot{\psi}$ sapma açısının açısal hızı ve $\ddot{\psi}$ sapma açısının açısal ivmesi olarak kullanılmaktadır [20].

Bu çalışmada kullanılan TRMS sistemine ait bazı model parametreleri Tablo 1’de sunulmuştur.

Tablo 1. TRMS model parametreleri

Parametre	Değer	Birim	Kısa Açıklama
L_m	0.254	m	Ana rotor kol uzunluğu
L_t	0.275	m	Kuyruk rotor kol uzunluğu
K_{F_m}	0.0202	$N \cdot m \cdot A^{-1}$	Ana motor moment sabiti
K_{F_t}	0.0202	$N \cdot m \cdot A^{-1}$	Kuyruk motor moment sabiti
K_{um}	8.5	—	Ana motor kazancı
K_{ut}	6.5	—	Kuyruk motor kazancı
J_p	6.1644e-2	$kg \cdot m^2$	Yükseliş eksenli atalet momenti
J_y	2.1117e-2	$kg \cdot m^2$	Sapma eksenli atalet momenti
b_m	4.2150e-4	$N \cdot m \cdot rad^{-1} \cdot s$	Yükseliş viskoz sürtünme katsayısı
b_t	1.1243e-4	$N \cdot m \cdot rad^{-1} \cdot s$	Sapma viskoz sürtünme katsayısı
G_z	-0.25254	$N \cdot m$	Yükseliş eksenli yerçekimi momenti
G_y	0.15032	$N \cdot m$	Sapma eksenli yerçekimi momenti

3. LQG Matematiksel Modeli

LQG kontrolcü, doğrusal sistemler için temel olarak LQR kontrol yapısı ile Kalman filtresini birleştiren bir optimal kontrol yöntemidir. LQR kontrolcünün amacı sistem çıkışını istenen referans değerinde tutarken kontrol sinyalini de minimize etmektir. Uygun seçilen Q ve R matrisleriyle beraber sistem üzerinden istenen denge yakalanır. Kalman filtresi ise gürültülü ve model belirsizliği olan durumlarda sistemin gerçek durumlarını tahmin eder. Bu

iki yapının birleşmesiyle LQG kontrolcü hem optimal kontrol performansını hem de gürültü ve belirsizliklere karşı dayanıklılık sağlar.

Sistem modelinin durum-uzayı gösterimi şu şekildedir:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= Cx(t) + v(t)\end{aligned}\tag{4}$$

Burada; $x(t)$ sistemin gerçek durumunu, $u(t)$ kontrol girdisini, $y(t)$ ölçülen gürültülü çıkışı, $w(t)$, $v(t)$ sırasıyla sistem ve ölçüm gürültüsünü ve A, B, C ise sistem matrislerini göstermektedir.

LQG kontrolü temel iki adımda tasarlanır :

- **LQR : Durum Geri Beslemesi**

Sistemin tüm durum bilgisi varsa, klasik LQR ile kontrol yasası şu şekilde tanımlanır :

$$u(t) = -K\hat{x}(t)\tag{5}$$

Bu kontrolcünün optimal geri besleme kazancı K , aşağıdaki sürekli zamanlı Riccati denklemi kullanılarak elde edilir:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0\tag{6}$$

Ardından :

$$K = R^{-1}B^T S\tag{7}$$

Burada; durum değişkenlerinin ağırlık matrisi Q ve kontrol girişlerinin ağırlık matrisi R seçimi, kontrol kazancı K 'yi belirlemek için doğrudan kullanılır.

- **Kalman Filtresi : Durum Tahmini**

Durum doğrudan gözlemlenemediğinde, Kalman filtresi ile sistemin durumu tahmin edilir.

$$\dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Ly(t)\tag{8}$$

Burada $\hat{x}(t)$, tahmin edilen durum vektörüdür. L kazancı ise aşağıdaki Riccati denklemi ile hesaplanır:

$$AP + PA^T - PC^T V^{-1} CP + W = 0\tag{9}$$

Ardından :

$$L = PC^TV^{-1} \quad (10)$$

Yukarıdaki denklemlerde; P , Kalman filtresi Riccati denkleminin çözümünü, W , proses gürültüsü kovaryans matrisini ve V , ölçüm gürültüsü kovaryans matrisini ifade etmektedir. Ayrıca, L , tahmin doğruluğunu optimize eden kazançtır ve W, V, A, C ile ilgilidir.

Sonuç olarak, LQG Kontrol Yasası elde edilirken durum tahmini ve durum geri beslemesi birleştirilir ve aşağıda verilen şekilde elde edilir.

$$\dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Ly(t) \quad (11)$$

$$u(t) = -K\hat{x}(t)$$

Bu yapı sayesinde; L , ölçüm gürültüsüne rağmen doğru tahmini sağlar ve K , tahmin edilen durumu kullanarak sistemi optimal şekilde kontrol eder [21].

3.1. İntegral Etkili LQG Kontrol Yaklaşımı (LQGI)

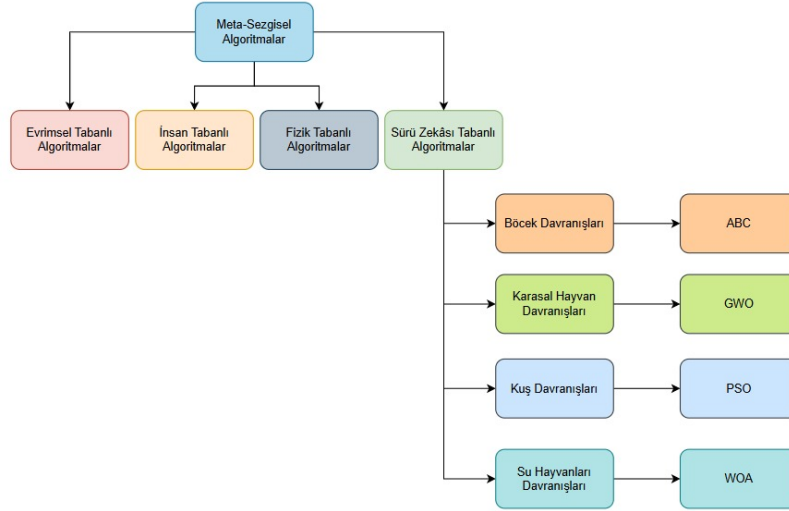
Bu çalışmada LQG kontrol yapısına ek olarak integral etkili bir yaklaşım uygulanmıştır. Böylece sistemin referans takibi daha hassas bir hale getirilmiştir. LQGI kontrol yasaşı şu şekildedir :

$$u(t) = -K \cdot [\hat{x}(t); x_i(t)]^T \quad (12)$$

Burada; $\hat{x}(t)$, Kalman filtresi ile tahmin edilen sistem durumunu, $x_i(t)$, hata integralini ifade eder. Ayrıca, K , LQGI kazanç matrisi olup hem durum hem integral bileşenini kapsayacak şekilde genişletilmiştir. Bu sayede, LQGI kontrolcü ile sistemin referans takibi performansı artırılmış olur.

4. Sürü Zekâsı Tabanlı Optimizasyon Algoritmaları

Popülasyon tabanlı optimizasyon algoritmalarının bir alt başlığı olan sürü zekâsı tabanlı optimizasyon algoritmaları, günümüzde literatürde en yaygın kullanılan optimizasyon yöntemlerinden biridir. Bu algoritmaların temelinde, bireysel olarak sınırlı yeteneklere sahip olan çok sayıda ajanın, birlikte hareket ettiklerinde doğadaki sürü davranışlarına benzer şekilde bir kolektif zekâ sergilemeleri yatmaktadır [22]. Karınca, arı, kuş ve balık gibi çeşitli hayvanların doğadaki koordineli hareketlerinden ilham alınarak geliştirilen bu algoritmalar, ajanların iş birliği ve basit kurallar çerçevesindeki etkileşimleri sayesinde, karmaşık problemlerde dahi etkili ve başarılı çözümler üretebilmektedir. Sürü zekâsına dayalı bu algoritmalar, esinlendikleri hayvan topluluklarının doğal yaşam ortamlarına ve hareket biçimlerine göre bazı kaynaklarda genel olarak dört ana sınıf şeklinde sunulmaktadır ve Şekil 2’de bu sınıflar özetlenmiştir [19; 23].



Şekil 2. Popülasyon tabanlı optimizasyon algoritmalarının genel sınıflandırması ve bazı sürü zekâsı tabanlı algoritmalar

4.1. Gri Kurt Optimizasyon Algoritması (GWO)

Gri Kurt Optimizasyon Algoritması (GWO), Mirjalili ve çalışma arkadaşları tarafından 2014 yılında, gri kurtların sürü hâlindeki sosyal davranışları ve avlanma tekniklerinden esinlenilerek geliştirilen bir meta-sezgisel optimizasyon yöntemidir [24, 25]. Gri kurtlar; lider olan alfa, lidere destek olan beta, karar verme süreçlerine katkı sunan delta ve en alt düzeydeki omega bireylerden oluşan bir hiyerarşi içinde yaşarlar. Avlanma süreçleri takip, kuşatma ve saldırı olmak üzere üç temel aşamadan meydana gelir.

Algoritma 1 Gri Kurt Optimizasyon Algoritması (GWO)

```

1: Başlat:
2:   Popülasyonu (kurt pozisyonları) rastgele oluştur.
3:   Alpha, Beta ve Delta kurtların pozisyonlarını ve skorlarını (başlangıçta çok büyük değer) belirle.
4:   for iterasyon = 1 to MaxIter do
5:     for her kurt  $i$  do
6:       Pozisyonu sınırlar içinde tut.
7:       Amaç fonksiyonunu hesapla (fitness).
8:       if fitness, Alpha'dan iyiye then
9:         Alpha'yı güncelle.
10:      else if fitness, Beta'dan iyiye then
11:        Beta'yı güncelle.
12:      else if fitness, Delta'dan iyiye then
13:        Delta'yı güncelle.
14:      end if
15:    end for
16:     $a$  parametresini lineer olarak 2'den 0'a indir.
17:    for her kurt  $i$  do
18:      Alpha, Beta ve Delta'nın etkisiyle yeni pozisyon hesapla.
19:      Bu üç yeni pozisyonun ortalamasını al ve yeni pozisyonu belirle.
20:    end for
21:    En iyi Alpha skorunu kaydet.
22:  end for
23:  Çıktı: En iyi Alpha pozisyonu ve skoru.

```

Algoritma, bu doğal süreci model alır. En iyi üç çözüm sırasıyla alfa, beta ve delta olarak belirlenir ve diğer çözümler bu liderlerin etrafında yönlendirilerek güncellenir. Her bir çözüm avın konumunu temsil eder; sürü ise iterasyonlar boyunca avın (optimum çözümün) yerini tahmin etmeye çalışır. Zamanla çözümler, en başarılı

bireylerin etrafında yoğunlaşarak daha kaliteli sonuçlar elde eder. Bu sayede GWO, arama alanının farklı bölgelerini keşfederken aynı zamanda mevcut en iyi çözümler üzerinde yoğunlaşarak etkin bir yakınsama sağlar. GWO algoritmasının temel adımları Algoritma 1’de özetlenmiştir.

4.2. Balina Optimizasyon Algoritması (WOA)

Balina Optimizasyon Algoritması (WOA), Seyyedi Mirjalili ve Andrew Lewis tarafından 2016 yılında, kambur balinaların “baloncuk ağıyla avlanma” yöntemi esas alınarak geliştirilmiş bir meta-sezgisel algoritmadır [26]. Kambur balinalar, avlarını çevreleyip spiral hareketlerle kuşatarak avlanırlar. Algoritma da bu davranışı taklit ederek en iyi bulunan çözümü av olarak belirler ve diğer çözümler bu hedefin etrafında konumlarını günceller. Ayrıca balinalar, rastgele hareketlerle yeni av alanları da arar; böylece sadece mevcut en iyi çözüme yakın bölgeleri değil, arama alanının daha geniş bölgelerini de keşfeder. Pozisyon güncellemeleri ve hareketler, avın konumunu tahmin eden parametrelerle kontrol edilir. Bu sayede algoritma, hem keşif hem de sömürü yeteneklerini dengede tutarak optimum çözüme ulaşana kadar iteratif şekilde çalışır. WOA’nın işleyişi, aşağıda Algoritma 2’de adım adım verilmiştir.

Algoritma 2 Balina Optimizasyon Algoritması (WOA)

```

1: Başlat:
2:   Popülasyonu (balina pozisyonları) rastgele oluştur.
3:   Lider balinanın pozisyonunu ve skorunu çok büyük (sonsuz) olarak başlat.
4:   for iterasyon = 1 to MaxIter do
5:     for her balina i do
6:       Pozisyonu sınırlar içinde tut.
7:       Amaç fonksiyonunu hesapla (fitness).
8:       if fitness, liderin skorundan iyiye then
9:         Lider balınayı güncelle.
10:      end if
11:    end for
12:    a parametresini 2’den 0’a doğru lineer olarak azalt.
13:    a2 parametresini -1’den -2’ye doğru lineer olarak azalt.
14:    for her balina i do
15:      Rastgele sayı p üret.
16:      if p < 0.5 then
17:        if  $|A| \geq 1$  then
18:          Rastgele seçilen bir balınaya doğru hareket et.
19:        else
20:          Lider balınaya doğru hareket et.
21:        end if
22:      else
23:        Spiral hareketle liderin etrafında dön.
24:      end if
25:    end for
26:    En iyi lider skorunu kaydet.
27:  end for
28: Çıktı: En iyi lider balina pozisyonu ve skoru.

```

4.3. Yapay Arı Kolonisi (ABC) Algoritması

Yapay Arı Kolonisi (ABC) Algoritması, 2005 yılında Derviş Karaboğa tarafından, bal arılarının doğadaki besin arama davranışlarından esinlenerek geliştirilmiş bir meta-sezgisel optimizasyon algoritmasıdır [27]. Gerçek arılar, yüksek kaliteli bir besin kaynağı bulduklarında bu bilgiyi kolonideki diğer arılara iletir ve diğer arılar da bu kaynağa yönelir. Ayrıca uzun süre verim sağlamayan kaynaklar terk edilir ve arılar yeni kaynaklar aramak için rastgele alanlara yönelir. Bu sosyal iş birliği ve arama stratejisi, çözüm uzayında hem keşif hem de verimli alanlarda yoğunlaşmayı sağlar.

Algoritmada bu davranış şu şekilde modellenir: Her çözüm bir besin kaynağıdır ve çalışan arılar mevcut çözümler üzerinde küçük değişiklikler yaparak yeni çözümler üretir. Bu bilgiler izleyici arılara aktarılır ve onlar da en verimli çözümleri seçerek arama yapar. Eğer bir çözüm uzun süre geliştirilmezse, keşifçi arılar devreye girerek yeni ve rastgele çözümler oluşturur. Böylece algoritma, hem iyi çözümleri geliştirir hem de yeni alanları keşfederek global optimuma ulaşmayı hedefler. ABC algoritmasının akışına ilişkin detaylı adımlar Algoritma 3’de sunulmaktadır.

Algoritma 3 Yapay Arı Kolonisi (ABC) Algoritması

```

1: Başlat:
2:   Popülasyonu (arı pozisyonlarını) rastgele oluştur.
3:   Her arının maliyetini hesapla.
4:   En iyi çözümü belirle.
5:   Trial sayaçlarını (deneme sayaçları) sıfırla.
6: for iterasyon = 1 to MaxIter do
7:   İşçi Arılar (Employed Bees) Aşaması:
8:   for her arı i do
9:     Başka bir arı seç.
10:    Rastgele hızlanma ile yeni pozisyon oluştur.
11:    if yeni pozisyon daha iyiysen then
12:      Pozisyonu güncelle.
13:      Trial sayacını sıfırla.
14:    else
15:      Trial sayacını artır.
16:    end if
17:  end for
18:  Onlooker Arılar Aşaması:
19:  Arıların fitness değerlerine göre seçim olasılıklarını hesapla.
20:  for seçilen arılar do
21:    İşçi arılar gibi pozisyon güncelle.
22:  end for
23:  Scout Arılar Aşaması:
24:  for her arı i do
25:    if trial sayacı belirli limiti aşarsa then
26:      Rastgele yeni pozisyon ata.
27:      Trial sayacını sıfırla.
28:    end if
29:  end for
30:  En iyi çözümü güncelle.
31: end for
32: Çıktı: En iyi pozisyon ve maliyet.

```

4.4. Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), 1995 yılında Kennedy ve arkadaşları tarafından, kuş sürülerinin yiyecek arama ve hareket etme davranışlarından esinlenerek geliştirilmiş bir meta-sezgisel optimizasyon yöntemidir [28]. Gerçek kuş sürülerinde bireyler kendi en iyi deneyimlerini ve sürünün bulduğu en iyi konumu hatırlar, hareketlerini buna göre ayarlarlar. Böylece sürü, farklı bölgeleri keşfederken aynı zamanda en iyi çözümün etrafında yoğunlaşır.

Modelde her parçacık (ajan) bir çözümü temsil eder ve hızları, kendi geçmiş en iyi pozisyonları ile sürünün en iyi pozisyonu arasındaki farklara göre güncellenir. Parçacıklar böylece sürekli hareket ederek arama alanında ilerler. Algoritmanın temel mantığı, bireysel deneyim ile grup bilgisini dengeli kullanarak hem keşif hem de yoğunlaşmayı sağlayıp optimum çözüme ulaşmaktır ve uygulama adımları Algoritma 4’de ayrıntılı olarak gösterilmiştir.

Algoritma 4 Parçacık Sürü Optimizasyonu (PSO) Algoritması

```

1: Başlat:
2:   Popülasyonu (parçacık pozisyonlarını) rastgele oluştur.
3:   Tüm parçacıkların hızlarını sıfırla.
4:   Her parçacığın maliyetini hesapla.
5:   Her parçacığın kendi en iyi pozisyonunu ve maliyetini kaydet.
6:   Global en iyi pozisyonu ve maliyeti belirle.
7:   for iterasyon = 1 to MaxIter do
8:     for her parçacık  $i$  do
9:       Hızını, kişisel ve global en iyiye göre güncelle.
10:      Hızını sınırlar içinde tut.
11:      Pozisyonunu güncelle.
12:      Pozisyonunu sınırlar içinde tut.
13:      Maliyetini hesapla.
14:      if yeni maliyet, kişisel en iyiden iyiyse then
15:        Kişisel en iyiyi güncelle.
16:      end if
17:      if kişisel en iyi, global en iyiden iyiyse then
18:        Global en iyiyi güncelle.
19:      end if
20:    end for
21:    Atalet ağırlığını ( $w$ ) belirli bir oranda azalt.
22:  end for
23: Çıktı: En iyi global pozisyon ve maliyet.

```

5. Simülasyon Sonuçları ve Bulgular

Bu bölümde, TRMS sistemi üzerinde tasarlanan LQGI kontrolcünün ayarları ve elde edilen simülasyon sonuçları verilecek ve bulgular tartışılacaktır.

5.1. LQGI Kontrolcü ve Sistem Parametre Ayarlaması

TRMS sistemi ve tasarlanan LQGI kontrolcü parametrelerinin ayarlanmasında kullanılan değerler aşağıda verilen Tablo 2'deki gibidir.

Tablo 2. Sistem ve kontrolcü parametreleri

Parametre	Değer
Boyut	10 (8Q + 2R)
Alt Sınır	1e-3, 1e-3, 1e-3, 1e-3, 1e-3, 1e-3, 1e-3, 1e-3, 0.001, 0.001
Üst Sınır	1e5, 1e5, 1e5, 1e5, 1e5, 1e5, 1e5, 1e5, 5, 5
Maliyet Fonksiyonu	J
Simülasyon Süresi	30 saniye
Popülasyon Boyutu	50
İterasyon Sayısı	100
Tekrar Sayısı	20

Performans ölçütü olarak yeni bir maliyet fonksiyonu önerilmiştir. Önerilen maliyet fonksiyonu, LQGI kontrolcüsünün optimizasyonunda sistemin dinamik ve kontrol performansını çok kriterli olarak değerlendirmek amacıyla oluşturulmuştur. Fonksiyon; yükseliş ve sapma eksenlerindeki yerleşme zamanlarını, zamana bağlı mutlak hata integralini (ITAE) ve kontrol sinyalinin ortalama enerji değerini dikkate alır. Her bir performans ölçütü, sistem tasarımında istenen önceliklere göre belirlenen ağırlık katsayıları ile çarpılmıştır. Böylece, elde edilen toplam maliyet, hem hızlı cevap hem de düşük hata ve minimum kontrol enerjisi sağlayan bir LQGI parametre optimizasyonu hedefler.

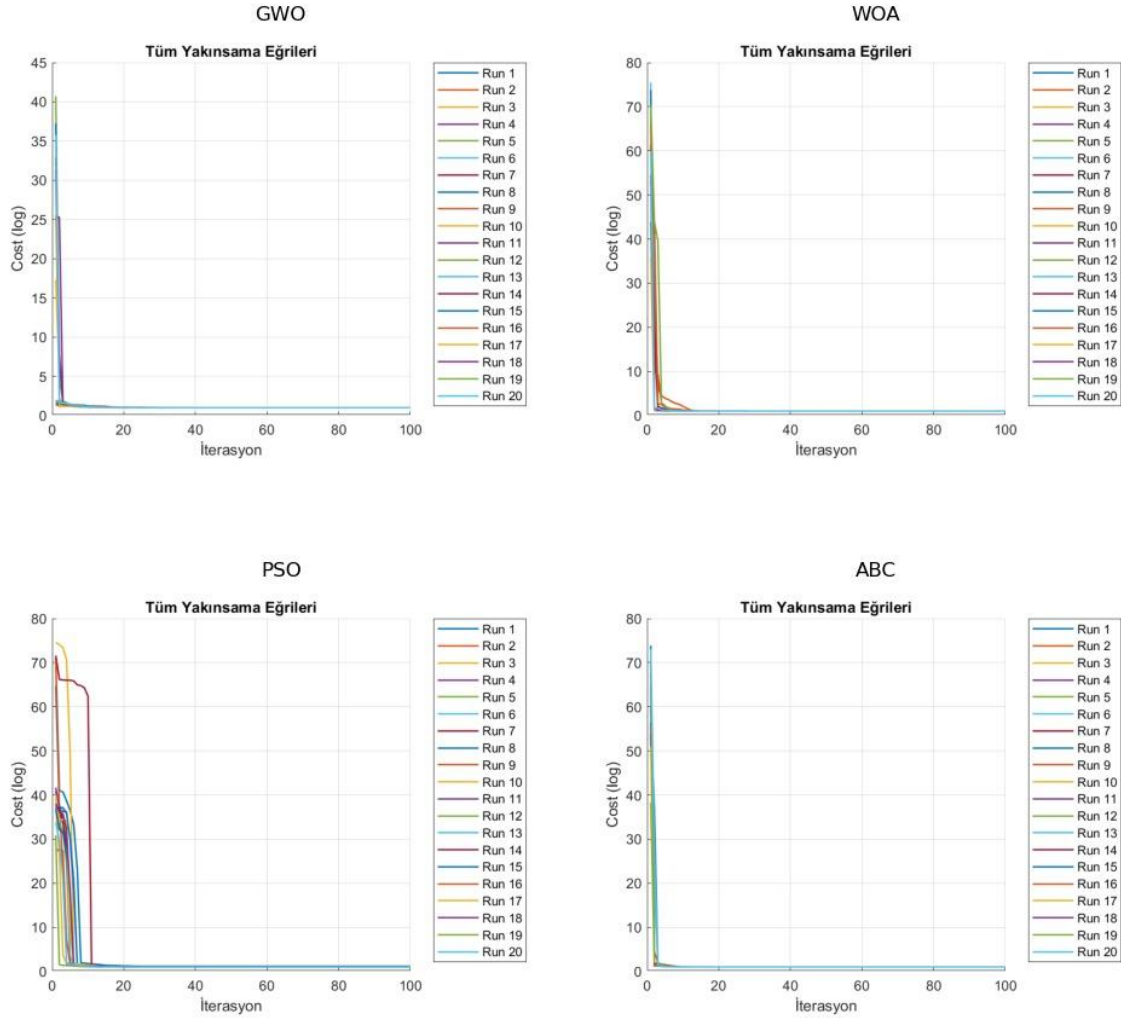
$$J = w_{ts\psi} t_{s\psi} + w_{ts\phi} t_{s\phi} + w_{ITAE} ITAE + w_u U_{effort} \quad (13)$$

Bu denklemde, $t_{s\psi}$, $t_{s\phi}$ yükseliş ve sapma için yerleşme zamanı, U_{effort} kontrol çabasının ortalama enerji (RMS) değeri, $w_{ts\psi}$, $w_{ts\phi}$, w_{ITAE} , w_u ağırlık katsayılarını ifade etmektedir. Ayrıca, ITAE'nin değerinin denklemi aşağıdaki denklemde verilmiştir.

$$ITAE = \int_0^{\infty} t|e(t)|dt \quad (14)$$

5.2. Optimizasyon Algoritmalarının İstatistiksel Sonuçları

Şekil 3'de algoritmaların yakınsama grafikleri verilmiştir. Grafiklerden görüldüğü üzere, tüm algoritmaların maliyet fonksiyonunda hızlı bir düşüş yaşandığı ve genellikle ilk 10 iterasyon içerisinde çok düşük değerlere ulaşıldığı gözlemlenmektedir. Bu durum, ilgili optimizasyon algoritmalarının problem için oldukça etkili şekilde parametre optimizasyonu gerçekleştirebildiğini göstermektedir. Özellikle GWO ve ABC algoritmalarının daha kararlı ve hızlı bir şekilde yakınsadığı, bazı runlarda PSO ve WOA'nın ise daha geç yakınsadığı veya lokal minimumlara takılabildiği görülmektedir.



Şekil 3. Algoritmaların yakınsama performansları

Tablo 3'de ise optimizasyon algoritmalarının istatistiksel sonuçları verilmiştir. Tabloda verilen sonuçlar incelendiğinde, ABC algoritması en düşük ortalama maliyet değerine (0.9945) ulaşarak optimizasyon başarımı açısından en iyi performansı göstermiştir. GWO algoritması ise en düşük standart sapmaya (0.0003) sahip olup, farklı çalıştırmalarda en kararlı sonuçları üretmiştir. WOA algoritması, düşük maliyet ve tutarlılık açısından

rekabetçi bir performans sunarken, PSO algoritması ise yüksek standart sapma ve ortalama maliyet değerleri nedeniyle diğer algoritmalarla kıyasla daha az tutarlı ve etkin bir sonuç vermiştir. Bu bulgular, LQGI kontrolcüsü parametre optimizasyonu için doğadan esinlenen algoritmaların yüksek başarı potansiyeline sahip olduğunu göstermektedir.

Tablo 3. Optimizasyon algoritmalarının istatistiksel sonuçları

Algoritma	Maksimum Maliyet	Minimum Maliyet	Ortalama Maliyet	Maliyet Std. Sapması
GWO	1.0091	1.0077	1.0080	0.0003
WOA	0.9994	0.9937	0.9948	0.0013
PSO	1.1836	0.9935	1.0355	0.0488
ABC	0.9960	0.9940	0.9945	0.0004

5.3. Simülasyon Sonuçları

Tablo 4 ve Tablo 5’de, GWO, WOA, PSO ve ABC algoritmalarının LQGI kontrolcüsü için optimize ettiği Q ve R matris parametreleri sunulmuştur. Q matrisi, sistem durum değişkenlerine verilen ağırlıkları, R matrisi ise kontrol girişlerine verilen cezaı temsil etmektedir. Elde edilen bu değerler, her algoritmanın kontrolcü tasarımına yaklaşımındaki farklılıkları açıkça göstermektedir.

Bu sonuçlar, her bir optimizasyon algoritmasının sistem dinamiklerini ve kontrol girdilerini farklı şekilde ağırlıklandırarak ortaya koymaktadır. GWO, bazı durum değişkenlerine ve kontrol girdilerine dengeli bir ceza uygularken; WOA ve PSO, belirli kontrol girdilerini daha fazla sınırlamaktadır. ABC algoritması ise özellikle durum değişkenleri için bazı parametrelere yüksek ağırlık verirken, kontrol çabasına ise minimumda tutmaktadır. Bu parametre farklılıkları, sistemin dinamik yanıtında ve kontrolcü performansında belirleyici olmaktadır.

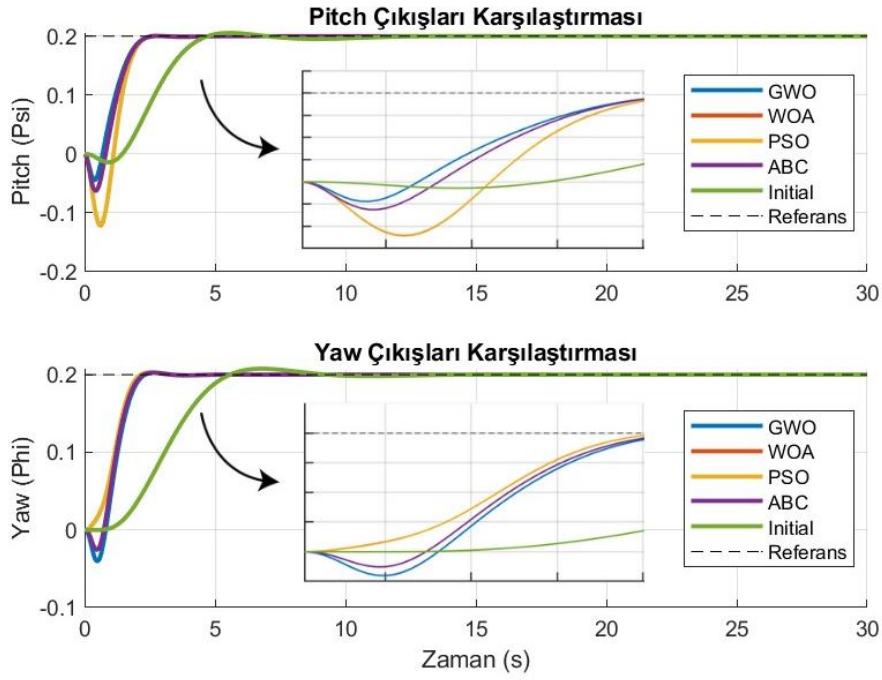
Tablo 4. LQGI kontrolcüsü için algoritmalar tarafından optimize edilen Q değerler tablosu

Algoritma	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
GWO	2.56e-03	1.70e-04	3.40e+00	3.00e-04	1.01e-04	5.00e-05	3.85e+04	6.50e+04
WOA	1.50e-04	1.50e-04	1.50e-04	1.53e-04	5.00e-05	5.00e-05	1.79e+04	6.50e+04
PSO	1.50e-04	1.50e-04	1.50e-04	1.50e-04	5.00e-05	5.00e-05	1.80e+04	6.50e+04
ABC	4.92e+00	1.50e-04	1.45e+00	2.80e-01	8.10e-05	5.00e-05	3.10e+04	6.50e+04

Tablo 5. LQGI kontrolcüsü için algoritmalar tarafından optimize edilen R değerler tablosu

Algoritma	R_1	R_2
GWO	0.01	0.01
WOA	0.11	0.00
PSO	0.13	0.00
ABC	0.00	0.00

Şekil 4’te yükseliş ve sapma açıları için simülasyon sonuçları verilmiştir. Sonuçlarda, tüm algoritmaların referans işareti olan 0.2 genlikli adım girişini kısa sürede ve minimum hatayla takip edebildiğini göstermektedir. GWO, WOA, PSO ve ABC algoritmaları ile optimize edilen kontrolcülerin tamamı, başlangıçta geçici rejimde bir miktar salınım göstermektedir; ancak bu salınımlar hızla sönümlenmekte ve sistemler kısa sürede kararlı hale gelmektedir. Özellikle GWO, WOA ve ABC algoritmalarının, aşım ve yerleşme zamanı açısından daha başarılı olduğu, PSO’nun ise biraz daha fazla aşım yaptığı gözlemlenmiştir. Referans ve başlangıç kontrolcüye kıyasla, optimize edilmiş kontrolcülerde izleme performansında anlamlı bir iyileşme sağlanmıştır.



Şekil 4. Yükseliş ve sapma açıları için simülasyon sonuçları

Tablo 6’da, algoritmaların yanı sıra nominal LQGI yaklaşımına ait geçici rejim performans değerleri de verilmiştir. Tabloya bakıldığında; tüm algoritmaların yükseliş ve sapma eksenlerinde maksimum aşım (M_p), yerleşme (t_s) ve yükselme (t_r) zamanı gibi geçici rejim performansları açısından birbirine çok yakın ve yüksek düzeyde başarılı sonuçlar verdiği görülmektedir. Özellikle GWO ve ABC algoritmaları, neredeyse sıfır aşım ve hızlı kararlı hale gelme özellikleriyle öne çıkmaktadır. WOA ve PSO ise özellikle yükselme zamanı açısından avantaj sağlarken, genel olarak tüm algoritmalar TRMS sisteminde LQGI parametre optimizasyonu için oldukça etkili çözümler sunmuştur. Buna karşın, nominal LQGI kontrol ile tasarlanan sistemde teorik olarak sıfır aşım sağlasa da, oldukça uzun yükselme ve yerleşme süreleriyle pratikte yetersiz bir performans sergilemiştir. Bu durum, optimizasyon yapılmadığında sistemin kararlı olmasına rağmen yavaş tepki verdiğini ve gerçek zamanlı uygulamalar açısından tercih edilebilir olmadığını göstermektedir. Ayrıca, J değerleri incelendiğinde, optimizasyon algoritmalarının yaklaşık 1 civarında elde ettiği sonuçlara karşın nominal LQGI kontrolcünde oldukça yüksek kalması, önerilen yaklaşımların etkinliğini açıkça ortaya koymaktadır.

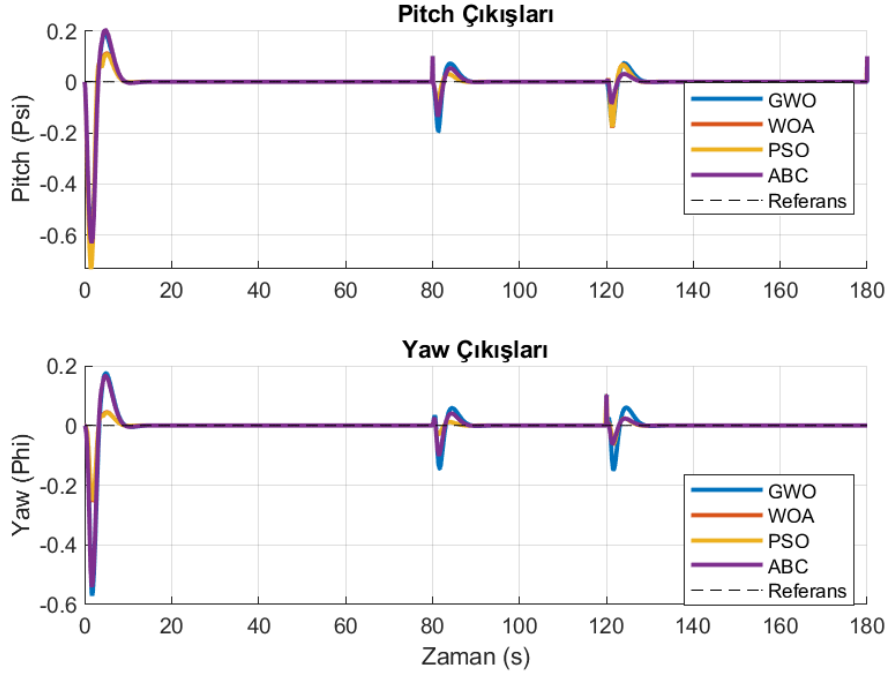
Tablo 6. Algoritmalarla göre yükseliş ve sapma eksenleri için geçici rejim performans kriterleri

Algoritma	J	Yükseliş			Sapma		
		M_p	t_s	t_r	M_p	t_s	t_r
GWO	1.01	2.62e-08	2.05	1.05	4.28e-03	1.95	0.90
WOA	0.99	4.28e-05	2.10	0.80	5.92e-03	1.80	1.05
PSO	0.99	4.21e-05	2.10	0.80	5.80e-03	1.80	1.05
ABC	0.99	1.49e-07	2.05	0.95	4.33e-03	1.90	0.90
Nominal LQGI	7.43	0.001	14.70	9.45	0.001	7.05	4.50

5.4 Sistemin Gürbüzülük Değerlendirmesi

Optimizasyon tabanlı geliştirilen LQGI kontrolcü yaklaşımının gürültü ve bozucu etkiler karşısında gürbüzülüğünü değerlendirebilmek amacıyla yükseliş ve sapma kanalları için iki adet darbe sinyali uygulanmıştır. Yükseliş kanalına 80. saniyede, sapma kanalına ise 120. saniyede başlayan ve 0.1 saniye süren, 0.1 genlikli darbeler uygulanarak sistemin bozulmuş durumlara karşı tepkisi gözlemlenmiştir.

Şekil 5’de, sistemin GWO, WOA, PSO ve ABC optimizasyon algoritmaları ile elde edilen LQGI kontrolcülerıyla ani bozucu girişlere karşı verdiği yanıtlar gösterilmektedir. Elde edilen sonuçlardan, optimize edilmiş tüm kontrolcülerin ani bozucu girişler karşısında sistem çıkışını hızlı bir şekilde referansa geri getirebildiği ve bozucunun etkisini kısa sürede bastırabildiği anlaşılmaktadır. Yükseliş ve sapma çıkışlarında, kontrolcülerin bozucu girişlerden sonra hızlı toparlandığı ve minimum kalıcı hata ile referansa ulaştığı görülmektedir. Genel olarak tüm bu sonuçların, kullanılan optimizasyon tabanlı LQGI kontrolcülerinin gürültü ve bozucu etkilere karşı oldukça gürbüz parametreler sağladığını göstermektedir.



Şekil 5. Farklı optimizasyon algoritmalarıyla elde edilen LQGI kontrolcülerinin ani bozucu girişler altındaki yükseliş ve sapma eksenli yanıtları

6. Tartışma ve Sonuç

Bu çalışmada, çift rotorlu MIMO sistem (TRMS) üzerinde LQGI optimal kontrolcü yaklaşımı uygulanmış ve bu kontrolcüye ait Q ve R matris parametrelerinin optimizasyonu için sürü zekâsı tabanlı meta-sezgisel algoritmaların etkinliği araştırılmıştır. Algoritma olarak literatürde başarımlı sıklıkla bahsedilen temel dört algoritma seçilmiştir. Bu algoritmalar; Gri Kurt Algoritması (GWO), Balina Optimizasyon Algoritması (WOA), Yapay Arı Kolonisi (ABC) ve Parçacık Sürü Optimizasyonu (PSO)’dur. Gerçekleştirilen simülasyonlar, optimize edilen LQGI kontrolcülerin, referans izleme ve ani bozucu girişini reddetme performansında klasik LQGI’ye göre belirgin üstünlük sağladığını ortaya koymuştur.

Simülasyonlarda elde edilen istatistiksel sonuçlar dikkate alındığında, özellikle ABC algoritmasının en düşük ortalama maliyet değeriyle (0.9945) optimizasyon başarımında öne çıktığı, GWO algoritmasının ise en düşük standart sapma (0.0003) ile en kararlı sonuçları ürettiği görülmüştür. WOA algoritması düşük maliyet ve tutarlılık açısından rekabetçi bir performans sergilerken, PSO algoritması yüksek standart sapma ve ortalama maliyet değerleri nedeniyle diğer algoritmalarla kıyasla daha değişken ve tutarsız sonuçlar vermiştir. Tüm bunlara ek olarak; yükseliş ve sapma eksenlerindeki maksimum aşım, yerleşme ve yükselme zamanı kriterleri dikkate alındığında, özellikle GWO ve ABC algoritmalarıyla elde edilen kontrolcülerin neredeyse sıfır aşım ve hızlı kararlılık avantajına sahip olduğu gözlemlenmiştir. WOA ve PSO ise yükselme zamanı bakımından rekabetçi performans sergilemiştir. Ek olarak, kontrolcülerin bozucu sinyallere karşı gürbüzlüğü de değerlendirilmiş ve tüm algoritmalarla tasarlanan kontrolcülerin ani bozucular sonrası hızlı şekilde referansa geri dönebildiği gösterilmiştir. Tüm bu bulgular, meta-sezgisel sürü zekâsı tabanlı algoritmaların hem istatistiksel olarak tutarlı hem de dinamik performans açısından yüksek başarı potansiyeline sahip olduğunu; LQG/LQR benzeri çok değişkenli optimal kontrolcülerin parametre optimizasyonunda etkin bir araç olarak kullanılabileceğini göstermektedir.

Tüm bu bulgular ışığında, gelecek çalışmalarda bu yaklaşımlar farklı MIMO sistemlerde, fiziksel deneysel

doğrulama veya farklı kontrol mimarileri ile genişletilebilir. Sonuç olarak, sürü tabanlı optimizasyon algoritmalarının, gelişmiş kontrolcü tasarımı için etkili ve esnek bir araç olduğu ortaya konmuştur.

Kaynakça

- [1] Sivadasan, J., & Willjuice Iruthayarajan, M. (2018). Tuning of nonlinear PID controller for TRMS using evolutionary computation methods. *Tehnički vjesnik*, 25(Supplement 1), 105-111.
- [2] Kaya, İ., Takeş, C. A., & Alyoussef, F. (2019). Parçacık Sürü Optimizasyonu Tabanlı PI-PD ile Twin Rotor Denetimi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 523-530.
- [3] Abukan, Y., & Almalı, M. N. (2023). Control of 2-Dof TRMS MIMO system using FOPID & FOSTSMC method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38(1), 605-615.
- [4] Phillips, A., Sahin, F. (2014). Optimal control of a twin rotor MIMO system using LQR with integral action. *World Automation Congress (WAC)*, 3-7 Ağustos, Hawaii, ABD, 114-119.
- [5] Pandey, S. K., Laxmi, V. (2014). Optimal control of twin rotor MIMO system using LQR technique. ss. 11-21. In: Babu, B. V., Das, K. N., ed. 2014. *Computational Intelligence in Data Mining, Volume 1: Proceedings of the International Conference on CIDM, 20-21 Aralık, Springer India, New Delhi*, 412s.
- [6] Chaudhary, S., & Kumar, A. (2019). Control of twin rotor MIMO system using PID and LQR controller. *Journal of Aircraft and Spacecraft Technology*, 3(1), 211-220.
- [7] Madhuri, C. S. G., Rao, G. R. (2016). Comparative Analysis of LQG and LQGI Controllers for Twin Rotor MIMO System. *International Journal of Engineering Research and Applications*, 6(5), 54-59.
- [8] Guardado, R., López, M. J., & Sánchez, V. M. (2019). MIMO PID controller tuning method for quadrotor based on LQR/LQG theory. *Robotics*, 8(2), 36.
- [9] Katebi, J., Shoaie-parchin, M., Shariati, M., Trung, N. T., & Khorami, M. (2020). Developed comparative analysis of metaheuristic optimization algorithms for optimal active control of structures. *Engineering with Computers*, 36(4), 1539-1558.
- [10] Benkhoud, K., & Bouallègue, S. (2018). Dynamics modeling and advanced metaheuristics based LQG controller design for a Quad Tilt Wing UAV. *International Journal of Dynamics and Control*, 6(2), 630-651.
- [11] Batista, J., Souza, D., dos Reis, L., Barbosa, A., & Rios, C. (2024). Dynamic model of a joint of the manipulator and LQG control optimized with GA and PSO metaheuristics. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 46(5), 309.
- [12] Lin, X., & Lin, W. (2022). Whale Optimization Algorithm-Based LQG-Adaptive Neuro-Fuzzy Control for Seismic Vibration Mitigation with MR Dampers. *Shock and Vibration*, 2022(1), 4060660.
- [13] Ranjan, K. V., Laxmi, V. (2022). Design of an optimal controller for a twin rotor MIMO system (TRMS). *International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 7-9 Nisan, Erode, Hindistan, 950-959.
- [14] Souza, D. A., de Mesquita, V. A., Reis, L. L., Silva, W. A., & Batista, J. G. (2021). Optimal LQI and PID synthesis for speed control of switched reluctance motor using metaheuristic techniques. *International Journal of Control, Automation and Systems*, 19(1), 221-229.
- [15] Önen, Ü., Cakan, A., & İlhan, I. (2019). Performance comparison of optimization algorithms in LQR controller design for a nonlinear system. *Turkish Journal of Electrical Engineering and Computer Sciences*, 27(3), 1938-1953.
- [16] Rivera-Ruiz, J. R., Rojas-Galvan, R., García-Martínez, J. R., Cruz-Miguel, E. E., Barra-Vázquez, O. A., Cruz-Orduña, M. I., ... & Rodríguez-Reséndiz, J. (2025). Performance Evaluation of Metaheuristics for LQR Controller Optimization: A Two-Wheel Balancing Robot Case Study. *IEEE Access*.
- [17] Özcan, N., Utku, S., & Berber, T. (2025). Artificial Circulation System Algorithm: A Novel Bio-Inspired Algorithm. *CMES-Computer Modeling in Engineering & Sciences*, 142(1).

- [18] Wong, W. K., Ming, C. I. (2019). A review on metaheuristic algorithms: recent trends, benchmarking and applications. 7th International Conference on Smart Computing & Communications (ICSCC), 28-30 Haziran, Sarawak, Malezya, 1-5.
- [19] Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, 243, 108457.
- [20] Tastemirov, A., Lecchini-Visintini, A., & Morales-Viviescas, R. M. (2017). Complete dynamic model of the Twin Rotor MIMO System (TRMS) with experimental validation. *Control Engineering Practice*, 66, 89-98.
- [21] Tang, Y., Zheng, Y., Li, N. (2021). Analysis of the optimization landscape of linear quadratic gaussian (LQG) control. *Learning for Dynamics and Control (L4DC)*, 7-8 Mayıs, PMLR, 599-610.
- [22] Fister Jr, I., Yang, X. S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.
- [23] Saravanan, G., Pazhanimuthu, C., & Naveen, P. (2025). Performance improvement of DC motor control system using PID controller with Kookaburra and Red Panda optimization algorithm. *Scientific Reports*, 15(1), 20021.
- [24] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- [25] Ozcan, N. (2025). Comprehensive Comparative Analysis of GWO and Its Variants for Solving Mechanical Optimization Problems. *Journal of Materials and Mechatronics: A*, 6(1), 150-169.
- [26] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- [27] Karaboga, D., & Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In *International fuzzy systems association world congress* (pp. 789-798). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [28] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.