

Döner Kanatlı İnsansız Hava Aracı için Tasarlanan Lineer Model Öngörülü Kontrolcünün Kazanç ve Ceza Katsayılarının Optimizasyonu

Optimization of Gain and Penalty Coefficients of a Rotary-Wing Unmanned Aerial Vehicle using Linear Model Predictive Controller

Ahmet Sabah¹, Ismail Bayezit¹

¹Uçak ve Uzay Bilimleri Fakültesi
İstanbul Teknik Üniversitesi, İstanbul
sabahah16@itu.edu.tr, bayezit@itu.edu.tr

Özetçe

Bu çalışma, dört-pervaneli konfigürasyonunda döner kanatlı bir İnsansız Hava Aracınının (İHA) stabilitesi ve kontrolü için tasarlanan lineer Model Öngörülü Kontrolcü'nün (MPC) kazanç ve ceza katsayılarının belirlenmesi sırasında en iyi performansı sağlayan uygun bir optimizasyon tekniğini bulmayı amaçlamaktadır. Çalışmanın içeriğinde; hava aracının modellenmesi, MPC'nin tasarım prensipleri, optimizasyon algoritmalarını karşılaştırmak için İHA'ya uygun bir yöntemin oluşturulması, optimizasyon yöntemleri ve bu yöntemlerin karşılaştırılmasının sonuçları yer almaktadır. Karşılaştırma için üç ana yöntem kullanılmıştır. Bu yöntem ve algoritmalar; küresel arama yöntemlerini temsilen Parçacık Sürü Optimizasyonu (Particle Swarm Optimization (PSO)) algoritması, konveks optimizasyon yöntemi olarak Simplex Optimizasyon algoritması ve bu ikisinin kombinasyonu olan Hibrit Optimizasyon. Sonuç olarak, doğruluk ve hesaplama maliyeti açısından Hibrit Optimizasyonun diğer yöntemlere göre daha uygun olduğu öngörülmüştür.

Abstract

This study is aiming to find a proper optimization technique that gives the best performance while tuning gain and penalty coefficients of the linear Model Predictive Controller (MPC) of a quadrotor model UAV. Modelling of the quadrotor, describing Model Predictive Controller, trajectory generation for comparing the optimization algorithms, defining the optimization methods, and results of the comparison of these methods are included in this study. Three main methods are used for the comparison: Particle Swarm Optimization (PSO) as Global Searching Method, the Simplex Search Method as Convex Optimization approach, and a combination of both used as Hybrid Method. Hybrid approach seem to be more appropriate and feasible than the other two methods in terms of certainty and computational cost.

1. Giriş

Son zamanlarda, Model Öngörülü Kontrol (MPC) teknikleri, araştırma çalışmalarında ve döner-kanatlı İHA ürünlerinde yaygın olarak kullanılmaktadır. MPC, 1970'ler ile 1980'ler

arasında geliştirilmiş olmasına rağmen, hesaplama maliyeti nedeniyle kontrol alanında tercih edilmemiştir. Ancak, bilgisayar bilimlerinin ve işlemci teknolojilerinin gelişmeleri sayesinde, günümüzde yaygın kullanılan modern kontrol yöntemlerinden biri haline gelmiştir [1]. MPC'nin önemli dezavantajlarından biri hesaplama karmaşıklığıdır. Bu nedenle, düşük kapasiteli donanımlar veya enerji kısıtlamaları olan sistemler için uygun değildir ve uygulamasında bazı zorlu çözümler gerektirir [2]. Öte yandan MPC, genellikle sistemdeki durumların ve girdilerin sınırlarını göz önüne alarak kontrol süreçlerini bu sınırlamalara uygun olarak yürütebilir [3]. Bu yüzden kısıtlamalar aşıldığı takdirde MPC; engellerden kaçınma, uygun yörünge oluşturma ve hızlı manevralar icra etme gibi birçok karmaşık sistem için uygun bir kontrol yöntemi haline gelebilir [4].

MPC yapısı tasarlanırken kazanç ve ceza (penalty) matrisinin belirlenmesi, genellikle sistem tepkisini gözlemleyerek deneme-yanılma gibi geleneksel yöntemlerle gerçekleştirilir. MPC'nin parametrelerini bu manuel yöntemlerle belirlemek, vakit kaybına ve yanlış değerler kullanılmasına yol açabilir [5]. Bu parametrelerin uygun bir şekilde belirlenmesi; tahmin ve kontrol zaman dilimi performansını, kontrol edilen durumların veya maliyet fonksiyonundaki parametrelerin ağırlıklarını ve girdi kısıtlamalarını etkiler [6]. Aynı zamanda, literatürde kazanç matrisinin değerlerini belirlerken farklı teknikler kullanılmaktadır; Pekiştirmeli Öğrenme (RL) uygulamaları, küresel arama algoritmaları, konveks optimizasyon yöntemleri, türev-tabanlı optimizasyon teknikleri vb. Bununla birlikte, hangi optimizasyon tekniklerinin hangi MPC yapılandırmaları için uygun olduğu net değildir; bu nedenle, farklı yöntemler ve algoritmalar çalışmanın sonunda karşılaştırılacaktır. Bu çalışmanın altı ana bölümü şunlardan oluşmaktadır: Model, Model Öngörülü Kontrolcü, Yörünge (Trajectory) Oluşturma, Optimizasyon Teknikleri, Bulgular ve Sonuçlar.

2. Model

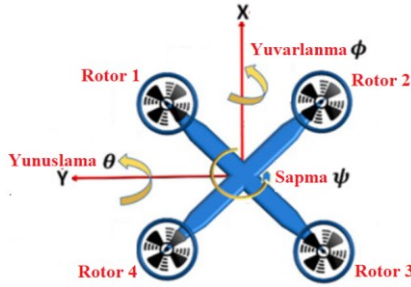
Kontrolcünün kazanç (Q) ve ceza (R) matrislerini optimize eden optimizasyon algoritmalarını daha verimli karşılaştırmak için analizlerde lineer döner kanatlı İHA modeli kullanılmıştır. Bu nedenle optimizasyon algoritmalarının oluşturdukları

haricinde, diğer etmenlerden oluşabilecek hesaplama yükü en aza indirilmiştir.

2.1. Manevralar

6 serbestlik dereceli (6DoF) döner kanatlı bir İHA; yukarı, aşağı, sol, sağ, ön ve arka yönlerde istenilen görevleri icra edebilir. Bu hava aracının boşluktaki hareketlerini tanımlarken; yuvarlanma (roll), yunuslama (pitch) ve sapma (yaw) manevraları kullanılır [7].

Dört rotorlu bir döner kanatlı İHA için motorların yerleşimi açısından birçok konfigürasyon vardır. X konfigürasyon şeklindeki motor yerleşimi, analizlerde kullanılacak olan İHA için seçilmiştir. Şekil 1’de X konfigürasyon, dört rotorlu döner kanatlı bir İHA’nın gövde eksenleri üzerindeki manevraları ve rotorlarının yerleşimi gösterilmiştir.



Şekil 1: X konfigürasyon döner kanatlı bir İHA'nın manevraları ve rotorlarının yerleşimi [18].

2.1. Doğrusal Hareket Denklemleri

Hava aracının modelini doğrusallaştırmak için; hareket denklemlerindeki lineer olmayan terimler (sinüs, kosinüs vb.), atalet momenti matrisindeki simetrik olmayan terimler (I_{xy} , I_{xz} , vb.) gibi parametreler ihmal edilir. Bu doğrusallaştırma sonucundaki hava aracının 12 durumunun türevlerinin denklemleri Eş. 1’de verilmiştir.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} -g\theta \\ g\phi \\ g - \frac{U_1}{m} \end{bmatrix}, \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{U_2}{I_{xx}} \\ \frac{U_3}{I_{yy}} \\ \frac{U_4}{I_{zz}} \end{bmatrix} \quad (1)$$

Bu denklemlerde x , y ve z hava aracının eksenlerdeki pozisyonlarını; v_x , v_y ve v_z eksenlerdeki lineer hızları; g yer çekimi ivmesini; ϕ yuvarlanma açısını; θ yunuslama açısını; ψ sapma açısını; U_1 dikey hareket girdisini; U_2 yuvarlanma hareketi girdisini; U_3 yunuslama hareketi girdisini; U_4 sapma hareketi girdisini; p yuvarlanma açısal hızını; q yunuslama açısal hızını; r sapma açısal hızını; I_{xx} x eksenine göre atalet momentini; I_{yy} y eksenine göre atalet momentini ve I_{zz} z eksenine göre atalet momentini göstermektedir [8].

Sonuç olarak bu döner kanatlı hava aracının sırasıyla üç eksenlerdeki pozisyonları, lineer hızları, Euler açıları ve açısal hızları olmak üzere toplam 12 durumunu gösteren vektör Eş. 2’de gösterilmiştir.

$$x = [x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r]^T \quad (2)$$

Lineer modeli tamamlamak için durum-uzay (state-space) gösterimi Eş. 3’te verilmiştir.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (3)$$

Burada x hava aracının durumlarını veren vektör, u girdilerin vektörü, y çıktılarının vektörü, A durum katsayılarının matrisi, B girdi katsayılarının matrisi ve C çıktı katsayılarının matrisidir [9]. Bu doğrusallaştırılmış denklemleri uygulamak için hava aracının gerekli parametreleri Tablo 1’de verilmiştir.

Tablo 1: Model Parametreleri

Parametre	Değer	Birim
m	16e-1	kg
I_{xx}	231e-4	kg*m ²
I_{yy}	231e-4	kg*m ²
I_{zz}	449e-4	kg*m ²

Bu parametreler ve denklemler ışığında durum-uzay gösterimini oluşturmak için kullanılan A ve B matrisleri Eş. 4 ve Eş. 5’te gösterilmiştir.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \quad (5)$$

3. Model Öngörülü Kontrolcü

Model Öngörülü Kontrolcü (Model Predictive Controller), temel olarak modele kontrol girdilerini vermeden kendi içerisinde bir simülasyon koşturur ve en iyi girdiyi bir minimizasyon problemi çözerek bulur. Bu simülasyon sırasında MPC, N adımlık bir tahmin ufku oluşturur ve N örnek zaman aralığı için model simülasyonunun kontrol girdilerinin dizisini hesaplar [10]. Bundan sonra, ilk giriş bir kontrolcü çıktısı olarak kullanılır. Eğer kontrolcüdeki model doğrusal ise, kontrolcüye doğrusal MPC denir; aksi takdirde, kontrolcü doğrusal olmayan bir MPC’dir. Ayrıca, minimizasyon problemini çözmek için çeşitli optimizasyon yöntemleri kullanılır.

3.1. Ayrık Zamanlı Dönüşüm

Eş. 3’te kullanılan durum-uzay gösterimi, türev tanımı içerdiği için sürekli zamanlı modellemeler için geçerlidir. MPC içerisindeki minimizasyon problemini çözmek için ayrık

zamanlı bir durum-uzay tanımı gereklidir ve sistemi ayrıştırmak için adım süresi (d_t) 0.01 saniye olarak belirlenmiştir. Sürekli zamandan ayrık zamana geçiş için gerekli olan çıkarım Forward Euler metodu kullanılarak Eş. 6'da gösterilmiştir.

$$\begin{aligned} x_{k+1} &= x_k + \dot{x}d_t \\ x_{k+1} &= x_k + (Ax_k + Bu_k)d_t \\ x_{k+1} &= (Ad_t + I)x_k + Bd_t u_k \\ x_{k+1} &= A_d x_k + B_d u_k \end{aligned} \quad (6)$$

Burada k parametrelerin adım numarası, A_d ayrık zamanlı A matrisi ve B_d ayrık zamanlı B matrisini göstermektedir [11].

3.2. MPC İçin Optimizasyon Problemi

MPC içerisindeki minimizasyon probleminin iki önemli parametresi; kazanç (Q) ve ceza (R) matrisleridir. Sistemin stabil olması için bu matrislerin pozitif yarı tanımlı (S_+) olması gerekmektedir. Nihai olarak kontrolcünün içerisindeki optimizasyon problemi Eş. 7'de gösterilmiştir.

$$\begin{aligned} \min_{X,U} \sum_{k=0}^{N-1} (x_{k+1}^{ref} - x_{k+1})^T Q (x_{k+1}^{ref} - x_{k+1}) + \Delta u_k^T R \Delta u_k \\ \text{subject to } x_{min} \leq x_k \leq x_{max} \quad x_k \in R^{12} \\ u_{min} \leq u_k \leq u_{max} \quad u_k \in R^4 \\ x_{k+1} = A_d x_k + B_d u_k \\ x_0 = x_{init} \end{aligned} \quad (7)$$

Burada N tahmin ufkunu, x_k hava aracının durumunu, x_k^{ref} referans yörüngesi, u_k kontrol girdisini ve Δu_k kontrol girdisinin k adımındaki değişimini göstermektedir. $R \in S_+^{4 \times 4}$ katsayı matrisi, ceza faktörü ekleyerek girdilerde ani ve önemli değişimler olmasını engeller. Aynı zamanda $Q \in S_+^{12 \times 12}$ katsayı matrisi de kazanç faktörü ekleyerek aracin durumlarında ani ve önemli değişimler olmasını engeller [11].

MPC içerisindeki minimizasyon probleminin çözümü için hava aracının durum ve girdi kısıtlarının tanımlanması gerekmektedir. Sırasıyla Eş. 8 ve Eş. 9'da bu tanımlar verilmiştir.

$$\begin{aligned} x_{max} &= [100, 100, 100, 10, 10, 10, \pi/5, \pi/5, \pi/5, \pi, \pi, \pi]^T \\ x_{min} &= -x_{max} \end{aligned} \quad (8)$$

Burada durumlardan pozisyonların birimi m, hızların birimi m/s, açılarının birimi rad ve açılal hızların birimi rad/s'dir.

$$\begin{aligned} u_{max} &= [mg, 1, 1, 1]^T \\ u_{min} &= -u_{max} \end{aligned} \quad (9)$$

Eş. 7'deki problemin çözümünün sonucunda N adımlık durum ve girdi değerleri gelecektir. Bu çözüm değerleri Eş. 10'da gösterilmiştir.

$$\begin{aligned} X^* &= [x_0^*, \dots, x_N^*] \\ U^* &= [u_0^*, \dots, u_{N-1}^*] \end{aligned} \quad (10)$$

Bu çözüm sonucunda bulunan girdi dizisinin ilk elemanı (u_0^*) kontrol edilmek istenen modele kontrolcü girdisi olarak verilir ve kontrol işlemi bu şekilde devam eder. Aynı zamanda içerideki optimizasyon problemini çözmek için IPOPT [12] algoritması kullanılmıştır.

3.3. MPC Parametreleri

Q ve R matrislerinin katsayı elemanları kontrolcünün performansını direk olarak etkiler. Bu elemanların değerleri değiştirilerek daha gürbüz veya daha hızlı kontrolcü tepkileri elde edilebilir. Aynı zamanda daha önce belirtildiği gibi sistemin stabil olması için bu matrislerin pozitif yarı tanımlı olmaları gerekmektedir. Bu durumun nedeni MPC içerisindeki minimizasyon probleminin çözümü için maliyet fonksiyonunun konveks olması gerekliliğidir. ($x^T A x$) şeklindeki ikinci

dereceden bir fonksiyonun konveks olması için de ara elemanın (A) pozitif yarı tanımlı bir matris olması gerekmektedir. Problemin sadeliği ve daha az hesaplama yükü için genellikle bu kazanç ve penaltı matrisleri diyagonal matrisler olarak seçilir. Matrislerin diyagonal formları Eş. 11'de gösterilmiştir.

$$\begin{aligned} Q &= \text{diag}([q_x, q_y, q_z, q_{v_x}, q_{v_y}, q_{v_z}, q_\phi, q_\theta, q_\psi, q_p, q_q, q_r]) \\ R &= \text{diag}([r_{U_1}, r_{U_2}, r_{U_3}, r_{U_4}]) \end{aligned} \quad (11)$$

Burada q_i , i durumunun katsayısını; r_i ise i girdisinin katsayısını temsil etmektedir. Bu katsayılar optimize edilerek kontrolcünün performansı belirlenebilir. Örnek olarak, eğer q_x 'in değerini q_y 'den daha büyük seçersek, kontrolcü x pozisyonuna ulaşırken y pozisyonuna göre daha az hata değeri verir. 12 tane durumların katsayıları ve 4 tane girdilerin katsayıları olmak üzere toplamda 16 tane katsayı optimum bir şekilde belirlenerek kontrolcünün performansı artırılabilir.

4. Yörünge Oluşturma

MPC içerisindeki x_{k+1}^{ref} terimi ve katsayı matrislerini optimize ederken kullanılacak optimizasyon yöntemlerini karşılaştırmak için hava aracına uygun bir yörünge oluşturmak gerekmektedir. Bu yörünge için de kontrol eforunu minimize edecek şekilde bir optimizasyon problemi kullanılmıştır ve bu problem Eş. 12'de tanımlanmıştır.

$$\min_{X,U} \sum_{k=0}^{N-1} u_k^T I u_k$$

$$\begin{aligned} \text{subject to } x_{min} \leq x_k \leq x_{max} \quad x_k \in R^{12} \\ u_{min} \leq u_k \leq u_{max} \quad u_k \in R^4 \\ x_{k+1} = A_d x_k + B_d u_k \\ x_0 = x_{init} \\ x_N = x_{final} \end{aligned} \quad (12)$$

Burada x_{init} döner kanatlı İHA'nın ilk durumları, x_{final} son andaki durumlarını, $I \in R^{4 \times 4}$ ve N toplam adım sayısını vermektedir. Bu adım sayısının denklemi (N) Eş. 13'te verilmiştir.

$$N = T/d_t \quad (13)$$

Buradaki T toplam simülasyon süresini vermektedir ki bu analizde 2 saniye olarak belirlenmiştir. Aynı zamanda yörüngenin ilk anındaki bütün durumlar 0 olarak belirlenmiştir ve son anında aracın ulaşması gereken durumlar Eş. 14'te gösterilmiştir.

$$x_{final} = [5, 6, 8, 0, 0, 0, 0, \pi/18, 0, 0]^T \quad (14)$$

Problemin çözümü için IPOPT [12] optimizasyon algoritması kullanılmıştır ve çıktı olarak 12 durumun; 0.01 saniye zaman aralıklarla, toplamda 2 saniye boyunca olması gereken değerler bulunmuştur. Bu değerlerin sembolik gösterim Eş. 15'te tanımlanmıştır.

$$X^* = [x_0, x_1, \dots, x_N] \quad (15)$$

5. Optimizasyon Teknikleri

Bu kısımda döner kanatlı hava aracının lineer model öngörülü kontrolcüsünde kullanılan kazanç ve ceza katsayılarının optimizasyonu için 3 farklı yaklaşım ele alınacaktır.

5.1. Maliyet Fonksiyonu

Optimizasyon tekniklerini karşılaştırmak için öncelikle probleme uygun bir maliyet fonksiyonu oluşturmak

gerekmektedir. Bu maliyet fonksiyonu, referans yörüngeyle uçağın anlık yörüngesi arasındaki fark baz alınarak oluşturulmuştur. 2 saniye boyunca her adımdaki fark ve son adımdaki farkın 5 ile çarpılması maliyet fonksiyonunu oluşturmuştur. Burada son adımın katsayı ile çarpılmasının nedeni; birinci önceliğin son anda istenilen durumlara ulaşmak, ikinci önceliğin ise bütün adımlar boyunca referans yörüngeye uygun hareket etmek olmasıdır. Bu optimizasyon problemi Eş. 16'da gösterilmiştir.

$$\min_{\lambda} \sum_{i=0}^N \epsilon_i + 5\epsilon_{final}$$

$$\text{subject to } 0 \leq \lambda_i \leq 10 \quad \lambda_i \in R^{16} \quad (16)$$

Burada ϵ_i i adımıdaki referans yörüngeyle aracın yörüngesi arasındaki farkı, ϵ_{final} son andaki farkı, N iterasyon sayısını ve λ kazanç ile penaltı matrislerinden oluşan vektörü temsil eder. Bu vektör Eş. 17'de gösterilmiştir.

$$\lambda = [q_x, q_y, \dots, r_{u_1}, r_{u_2}, \dots]^T \quad (17)$$

Bütün optimizasyon algoritmaları, karşılaştırmaları daha iyi yapabilmek için toplamda 8 iterasyon koşturulmuştur.

5.2. Global Arama Algoritmaları

Tasarımındaki kolaylıklar, istenmeyen türev operatörlerinin olmayışı ve standart türev azalımı (gradyan descent) yöntemlerine göre karşılaştırılabilir bir yakınsama performansından dolayı; küresel arama yöntemleri birçok araştırmacı tarafından tercih edilmektedir [13]. Küresel optimizasyon yöntemleri gerçek hayattaki karmaşık problemlerin çözümünde kullanılabilir. Bu yüzden robotik, kablosuz ağlar, iş takvimlerinin oluşturulması, nöral ağların sınıflandırılması ve eğitilmesi gibi birçok alanda bu optimizasyon yöntemleri boy göstermektedir [14]. Bu algoritmalar, optimum çözüme belirlenen limitler dahilinde ve ceza katsayılarıyla her adımda rastgele değerler üretirek ulaşırlar. Öte yandan bu belirli düzenler içerisindeki denemeyanılma işlemleri önemli derecede hesaplama maliyeti getirebilir.

Bu çalışmada küresel arama yöntemlerine örnek olarak Parçacık Sürü Optimizasyonu (PSO) algoritması tercih edilmiştir. Bu algoritma, doğadaki kuşların ve balıkların kolektif hareketlerinden esinlenerek rastgele değerler üretir. İki etmen bu algoritmanın geliştirilmesinde önemli rol oynar: bireysel ve küresel katsayılar. Başlangıçta belirlenen sayıda popülasyonlar oluşturulur ve bu popülasyonlar her iterasyonda değerler üretir ve her iterasyonda maliyet fonksiyonunu en aza indiren değer küresel en iyi değer olarak belirlenir. Sonraki adımlarda da bu en iyiye ulaşılan popülasyonlara doğru bir hız vektörü oluşturulur ve arama bu yönde devam eder. Bu noktada bireysel katsayılar arama bölgesinin genişliğini, küresel katsayılar ise en iyi noktaya odaklanmayı sağlar. Böylelikle optimum noktaya daha hızlı bir şekilde ulaşım sağlanır. Bu hız vektörü PSO algoritmasını diğer küresel algoritmalarından ayıran en önemli özelliktir [14].

5.3. Konveks Optimizasyon Algoritmaları

Konveks optimizasyon algoritmaları temelde iki prosten oluşur: yön belirleme ve aralık arama. Bu algoritmalar optimum noktaya giderken yönü belirlemek için her iterasyonda birinci türev (Jacobian), ikinci türev (Hessian) gibi terimlerden faydalanabilirler (türev-tabanlı). Aynı zamanda alanları veya aralıkları problemin kısıtlamalarını kullanarak belirleyip bu kısımların uç noktalarından daraltarak da optimum çözümlere giden konveks optimizasyon yöntemleri bulunmaktadır. Bu yöntemler küresel yöntemlere kıyasla daha hızlı çalışırlar. Öte

yandan gerçek hayattaki problemler genelde karmaşık ve konveks olmayan problemlerdir. Bu durum türev-tabanlı algoritmaların kolaylıkla lokal optimumda takılı kalmasına neden olur. Buradan kurtulmaları için optimum bir ilk değer verilmesi bu algoritmalar için mecburidir. Aynı zamanda çoğu problemin kolayca türevi tanımlanamaz [15]. Bu problemlerin konveks olmaması türev kullanmayan konveks optimizasyon algoritmaları için de bir sorun teşkil eder ve bu algoritmaların optimuma ulaşması için de problemin konveks bir alana çekilmesi gerekir.

Bu çalışmada konveks optimizasyon yöntemi olarak Simplex Metot seçilmiştir. Simplex Metot bir lineer programlama yöntemidir. Bu yöntemde eşitlik ve eşitsizlik kısıtlamaları belirlenir. Optimum nokta için eşitsizlik kısıtlamalarının kesişimi ile eşitlik kısıtının üzerindeki noktalar üzerinde maliyet fonksiyonu hesaplanır ve minimum değer olduğu nokta optimum nokta olarak bulunur [16]. Eğer problemler konveks ve bu tasarıma uygun ise algoritma hızlı bir şekilde optimum noktayı bulabilir. Gerçek hayattaki problemlerde ise bu yapıya uygun hale getirmek için problemlerin genellikle bazı işlemlerden geçmesi gerekir.

5.4. Hibrit Optimizasyon Algoritmaları

Hibrit algoritmalar genellikle küresel arama algoritmaları ile konveks optimizasyon algoritmalarının birleşimi şeklinde oluşturulur. Öncelikle karmaşık ve konveks olmayan problemler küresel algoritmalar sayesinde belirli sayıda iterasyonlardan geçerek konveks alana kadar daraltılır. Sonrasında küresel algoritmalarından gelen son çıktılar, konveks optimizasyon algoritmalarına ilk değer olarak verilir. Nihayetinde konveks alanda başlayan konveks optimizasyon algoritmaları hızlı bir şekilde optimum noktaya ulaşır. Küresel algoritmalar karmaşık alanlarda optimuma yakınsayabilir ama yavaş çalışırlar bu yüzden zaten konveks olan bir problemin çözümünü için kullanılmaları çok verimsizdir. Konveks optimizasyon algoritmaları ise hızlı bir şekilde optimum noktaya ulaşabilir ama karmaşık problemleri çözemez. Hibrit algoritmalar, konveks bölgeye kadar küresel algoritmaların kesinliğinden ve bu bölgeden sonra da konveks optimizasyon algoritmalarının çabukluğundan yararlanır [17]. Bu çalışmada Hibrit algoritma oluşturulurken ilk 3 iterasyon boyunca PSO ve son 5 iterasyon boyunca da Simplex Metot kullanılmıştır. Dördüncü iterasyonda PSO'dan çıkan sonuçlar Simplex Metot için ilk değerler olarak verilmiş ve sonrasında optimizasyon tamamlanmıştır.

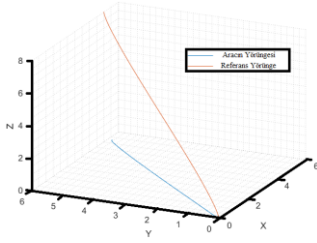
6. Simülasyon Sonuçları

Bu çalışmada analizler MATLAB kullanılarak yapılmıştır. Aynı zamanda MPC içerisindeki ve yörünge oluşumu için belirlenen minimizasyon problemlerinin çözümü, MATLAB için tasarlanan CasADi [19] kütüphanesi kullanılarak yapılmıştır.

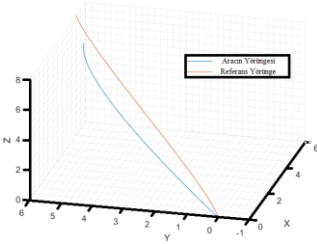
Q ve R matrisleri optimize edilmeden referans olması amacıyla diyagonal değerleri 1 olarak atanmıştır. Şekil 2'deki grafikte görüldüğü gibi girdi ve durumları ağırlıklandırmamak, özellikle dikey eksenle herhangi bir referans takibinin yapılamamasına neden olmuştur.

Q ve R matrislerinin Simplex Metot ile optimize edildikten sonra; hiç optimize edilmemelerine kıyasla hava aracının daha iyi bir performans sergilediği görülmüştür. Yine de dikey

eksende ve son durumda yeterince iyi bir referans takibinin yapılamadığı, Şekil 3'e bakılarak söylenebilir.

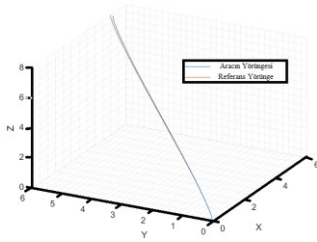


Şekil 2: Optimizasyon yapılmadan referans yörünge takibi.



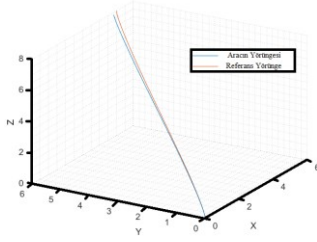
Şekil 3: Simplex Metot ile optimize edilen aracın referans yörünge takibi.

Öte yandan PSO ile bu proses yapıldığında İHA'nın neredeyse bire bir olarak verilen referans yörüngeyi takip ettiği Şekil 4'te görülebilmektedir.



Şekil 4: PSO ile optimize edilen aracın referans yörünge takibi.

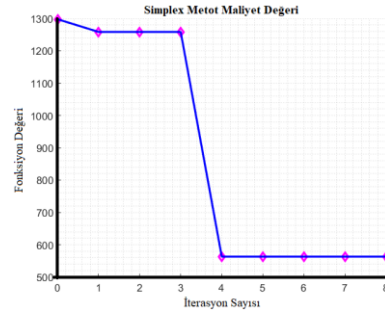
Son olarak optimizasyonu Hybrid Metot ile yaptıktan sonra aracın performansını incelediğimizde, Şekil 5'te de görülebileceği gibi PSO Metot kadar olmasa da kabul edilebilir bir referans takibi görülmektedir.



Şekil 5: Hibrit Metot ile optimize edilen aracın referans yörünge takibi.

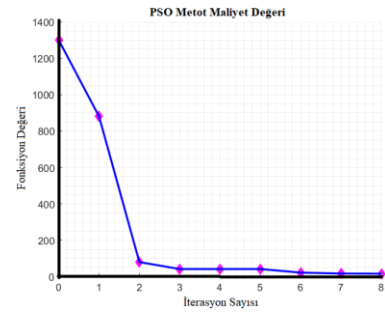
Öte yandan optimizasyon algoritmalarını karşılaştırabilmek için maliyet fonksiyonunu kaç adımda ve ne kadar minimize ettiklerini görmek önemli bir etmendir. Aynı zamanda maliyet performansındaki bu değişim referans takibi grafikleriyle de ilişkilendirilebilir. Şekil 6'da da görüldüğü gibi Simplex

algoritması, 4. iterasyonda önemli bir minimizasyon gerçekleşirse de maliyet fonksiyonunu 0 değerine yeterince yakınsatamamış ve sonraki iterasyonlarda da herhangi bir gelişim göstermeden lokal bir minimum noktasında takılı kalmıştır.



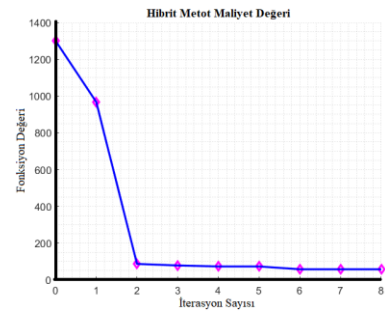
Şekil 6: Simplex Metot'un yakınsama grafiği.

Şekil 7'de de görüldüğü gibi aynı minimizasyon işleminde PSO algoritması, özellikle 2. iterasyonda maliyet fonksiyonunu önemli ölçüde azaltabilmiş ve diğer adımlarda da gelişim göstererek 0 değerine olabildiğince yaklaşmıştır.



Şekil 7: PSO'nun yakınsama grafiği.

PSO algoritmasındaki bu durumun bir benzeri Şekil 8'deki Hibrit Metot'un maliyet fonksiyonu grafiğinde de görülebilir. Aralarındaki tek fark 2. adımdan sonraki 0'a yakınsama miktarının Hibrit Metot için biraz daha düşük olmasıdır.



Şekil 8: Hibrit Metot'un yakınsama grafiği.

Bu 3 yöntemi karşılaştırmak için optimizasyon süreçlerinin toplam süresi ve 8 iterasyon sonundaki maliyet fonksiyonunun değerleri Tablo 2'de verilmiştir.

Tablo 2: Karşılaştırılma Parametreleri

Parametre	Simplex	PSO	Hybrid
İşlem Süresi	1012 s	7258 s	4239 s
Minimum Maliyet Değeri	562.9175	16.2302	57.1041

Analizlerin yapıldığı sistemin performans özellikleri şu şekildedir:

- Intel®Core™ i7-7700HQ CPU @ 2.80 GHz- 2.81 GHz
- 16 GB RAM x64 Windows 10 Pro

7. Sonuçlar

Bu çalışmanın amacı, 3 farklı optimizasyon yöntemi ile döner kanatlı bir İHA'ya uygulanan MPC'nin Q ve R matrislerini optimize etmek ve bu algoritmaların performansını karşılaştırmaktır. Her optimizasyon problemindeki gibi bu problemdeki en önemli kısımlardan birisi uygun bir maliyet fonksiyonu oluşturmaktır. Bu çalışmadaki maliyet fonksiyonu da hava aracına uygun şekilde tasarlanan bir referans yörüngeyle aracın yörüngesi arasındaki fark baz alınarak oluşturulmuştur. Optimizasyon algoritmaları uygulandıktan sonra karşılaştırma; referans yörünge takibi, maliyet fonksiyonunun minimize edilmesi ve işlem süresi açısından 3 farklı etmenle yapılmıştır.

Sonuç olarak en iyi yakınsama ve yörünge takibini PSO algoritması ve en kısa optimizasyon süresini Simplex Metot elde etmiştir. Bunun yanında PSO algoritması 7258 saniye gibi çok uzun bir işlem süresine, Simplex Metot ise 562.9175 gibi çok yüksek bir maliyet değerine sahiptir. Ayrıca Simplex Metot istenilen yörüngeyi uygun bir şekilde takip edememiştir. Hibrit Metot ise 3 değerlendirme unsurunda da görece kabul edilebilir değerlere sahiptir ve bu çalışma için optimizasyon algoritmaları arasındaki en uygun metot olarak görülmüştür.

İlerleyen çalışmalar için lineer model ne kadar işlem yükünü azaltsa da daha gerçekçi analizler elde edebilmek için lineer olmayan bir model kullanılabilir ve daha iyi bir karşılaştırma için farklı optimizasyon algoritmaları kullanılabilir.

Kaynakça

- [1] R. Susuki, F. Kawai, C. Nakazawa, T. Matsui, and E. Aiyoshi, "Parameter optimization of model predictive control using pso," in *2008 SICE Annual Conference*, 2008, pp. 1981–1988.
- [2] E. Böhn, S. Gros, S. Moe, T. A. Johansen, "Optimization of the model predictive control meta-parameters through reinforcement learning," *Engineering Applications of Artificial Intelligence*, Volume 123, Part A, 2023, 106211, ISSN 0952-1976.
- [3] S. Gros and M. Zanon, "Reinforcement Learning based on MPC and the Stochastic Policy Gradient Method," *2021 American Control Conference (ACC)*, New Orleans, LA, USA, 2021, pp. 1947-1952.
- [4] M. Kaymaz and N. Ure, "Obstacle Identification and Ellipsoidal Decomposition for Fast Motion Planning in Unknown Dynamic Environments." *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2021, pp. 1694-1700.
- [5] W. Edwards, G. Tang, G. Mamakoukas, T. Murphey and K. Hauser, "Automatic Tuning for Data-driven Model Predictive Control," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 7379-7385.
- [6] Q. Lu, L. D. Gonzalez, R. Kumar, and V. M. Zavala, "Bayesian optimization with reference models: A case study in mpc for hvac central plants," *Computers Chemical Engineering*, vol. 154, p. 107491, 2021.
- [7] M. Usman, "Quadcopter modelling and control with matlab/simulink implementation," Ph.D. dissertation, 2020.
- [8] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *2016 International Conference on Advanced Mechatronic Systems (ICAMEchS)*, 2016, pp. 498–503.
- [9] Raynaud, H-F., and A. Zergamoh. "State-space representation for fractional order controllers." *Automatica* 36.7 (2000): 1017-1021.
- [10] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved mpc," *IEEE Access*, vol. 7, pp. 161 064–161 073, 2019.
- [11] J. Lofberg, "Linear model predictive control," pp. 10–11, 2001.
- [12] A. Wachter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for largescale nonlinear programming," *Mathematical Programming*, vol. 01, no. 106, pp. 25–57, 2006.
- [13] A. Zbiciak and T. Markiewicz, "A new extraordinary means of appeal in the polish criminal procedure: the basic principles of a fair trial and a complaint against a cassatory judgment," *Access to Justice in Eastern Europe*, vol. 6, no. 2, pp. 1–18, mar 2023.
- [14] Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch Computat Methods Eng* 29, 2531–2561 (2022).
- [15] X. Wang, L. Xi, Y. Chen, S. Lai, F. Lin, and B. M. Chen, "Decentralized mpc-based trajectory generation for multiple quadrotors in cluttered environments," *Guidance, Navigation and Control*, vol. 01, no. 02, p. 2150007, 2021
- [16] Shamir, R. (1987). The Efficiency of the Simplex Method: A Survey. *Management Science*, 33(3), 301–334.
- [17] V. Kelner, F. Capitanescu, O. Léonard, L. Wehenkel, "A hybrid optimization technique coupling an evolutionary and a local search algorithm," *Journal of Computational and Applied Mathematics*, vol. 215, no. 2, p. 448-456, 2008.
- [18] S. Zahran, A. Moussa, and N. El-Sheimy, "Enhanced Drone Navigation in GNSS Denied Environment Using VDM and Hall Effect Sensor," *ISPRS Int. J. Geo-Inf.* vol. 8, no. 9, p. 169, 2019.
- [19] Andersson, J.A.E., Gillis, J., Horn, G. et al. CasADi: a software framework for nonlinear optimization and optimal control. *Math. Prog. Comp.* 11, 1–36 (2019).