

İNSANSIZ HAVA ARACININ GÖRSEL SERVO KONTROL İLE OTONOM HİZALANMASI VE OTONOM HEDEFE YÖNELİMİ

Mustafa Cihan SALMANER

Bilecik Şeyh Edebali Üniversitesi, Fen Bilimleri Enstitüsü

ORCID ID: 0009-0001-7035-4109

Tolga YÜKSEL

Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi, EE Mühendisliği, 11230, Bilecik

ORCID ID: 0000-0003-4425-7513

ÖZET

Bu çalışmada bir döner kanat insansız hava aracı (İHA) ile, özellikleri belirtilmiş bir hedefe otonom olarak yönelme, hedef karşısında hizalanma ve belirli süre boyunca hedef karşısında hizalanma sonunda hedefin içinde bulunan boşluktan geçmek için gerekli kontrolün yapılması amaçlanmıştır. Önerilen sistemde hedef, yer düzlemine dik içi boş bir dikdörtgendir. İHA yer düzlemine paralel hareket etmekte ve ön tarafında yer düzlemine dik eksenin görüntüsünü alabilecek kamera bulunmaktadır. Alınan kamera görüntüsünde soldan sağa doğru olan eksen x , yukarıdan aşağı doğru olan eksen y olarak tanımlanmıştır. İHA'nın ileri yönlü hızı V_{x+} , geri yönlü hızı V_{x-} , sağa yönlü hızı V_{y+} , sola yönlü hızı V_{y-} olarak tanımlanmıştır. Kontrol katmanında, görüntü merkez-hata vektöründen üretilen PID tabanlı komutlar ile İHA'nın hizalanması sağlanmıştır. Ayrıca kontrol denkleminde gövde hızları (V_x , V_y) geri beslemesi eklenen ve eklenmeyen iki yaklaşım karşılaştırma çerçevesinde tartışılmıştır.

Anahtar kelimeler: Döner kanat İHA, otonom uçuş, görüntü işleme, gerçek zamanlı hedef tespiti, PID kontrol

AUTONOMOUS ALIGNMENT AND TARGET ORIENTATION OF AN UNMANNED AERIAL VEHICLE USING VISUAL SERVO CONTROL

ABSTRACT

This study aims to control a rotary-wing unmanned aerial vehicle (UAV) to autonomously orient itself towards a specified target, align with the target, and pass through a gap within the target after a specified alignment time. In the proposed system, the target is a hollow rectangle perpendicular to the ground plane. The UAV moves parallel to the ground plane and has a camera on its front capable of capturing the image of the axis perpendicular to the ground plane. In the captured camera image, the axis from left to right is defined as x , and the axis from top to bottom is defined as y . The UAV's forward velocity is defined as V_{x+} , its backward velocity as V_{x-} , its rightward velocity as V_{y+} , and its leftward velocity as V_{y-} . In the control layer, the

alignment of the UAV is achieved using PID-based commands generated from the image center-error vector. Furthermore, two approaches, one with and one without adding body velocity (V_x , V_y) feedback to the control equation, are discussed within a comparative framework.

Keywords: Rotary-wing UAV, autonomous flight, image processing, real-time target detection, PID control.

1. GİRİŞ

İnsansız hava araçlarının (İHA) iç mekân görevlerinde güvenilir şekilde konumlanabilmesi ve belirli bir geometrik açıklıktan (kapı, pencere, hedef çerçeve vb.) geçebilmesi, arama-kurtarma, depo otomasyonu ve insan-robot etkileşimi gibi uygulamalar için önemlidir. İHA'ların otonom kontrolü üzerine yapılan çalışmalar son yıllarda önemli ölçüde artmıştır. Özellikle kamera sensörlerinin yaygınlaşması ile görüntü tabanlı kontrol yöntemleri İHA navigasyonunda yaygın olarak kullanılmaktadır. Görüntü tabanlı kontrol sistemlerinde kameradan elde edilen görsel bilgiler doğrudan kontrol algoritmasına geri besleme olarak verilmekte ve araç hareketi bu bilgiye göre düzenlenmektedir. Bu yöntem özellikle GPS sinyalinin bulunmadığı kapalı ortamlarda önemli avantajlar sağlamaktadır. Chen ve arkadaşları tarafından gerçekleştirilen çalışmada, görsel geri besleme kullanılarak hareketli bir hedefi takip edebilen bir İHA kontrol sistemi geliştirilmiştir ve sistemin hedefi başarılı şekilde takip edebildiği deneysel olarak gösterilmiştir [1].

Görüntü tabanlı kontrol sistemleri genel olarak Görüntü Tabanlı Görsel Servo Kontrol ve Pozisyon Tabanlı Görsel Servo Kontrol olmak üzere iki ana yaklaşım altında incelenmektedir. IBVS yönteminde kontrol sinyalleri doğrudan görüntü düzlemindeki hata değerleri kullanılarak hesaplanırken, PBVS yönteminde hedef nesnenin üç boyutlu konumu tahmin edilerek kontrol gerçekleştirilmektedir. Bu iki yaklaşımın performansı döner kanat platformlar üzerinde birçok çalışmada incelenmiştir. Ceren ve Altuğ tarafından yapılan çalışmada İHA için hibrit bir görsel servo kontrol sistemi geliştirilmiş ve yöntemin hedef takibi açısından etkili sonuçlar verdiği gösterilmiştir [2].

Görüntü tabanlı kontrol sistemlerinde algılama performansını artırmak amacıyla son yıllarda derin öğrenme yöntemleri de kullanılmaya başlanmıştır. Durdevic ve Arroyo tarafından gerçekleştirilen çalışmada derin sinir ağları kullanılarak görsel verilerden kontrol sinyali üretebilen bir sistem geliştirilmiştir. Çalışmada stereo kamera verileri kullanılarak hedef nesnenin konumu tahmin edilmiş ve elde edilen sonuçların görsel servo kontrol sistemleri için yeterli doğrulukta olduğu gösterilmiştir [3].

Görsel servo kontrol performansını artırmak amacıyla kontrol parametrelerinin adaptif şekilde ayarlanması da literatürde araştırılan konular arasında yer almaktadır. Fu ve arkadaşları tarafından yapılan çalışmada, görsel servo kontrol sisteminin kazanç parametreleri derin pekiştirmeli öğrenme algoritmaları kullanılarak optimize edilmiştir. Çalışma sonuçları, öğrenme tabanlı yaklaşımın klasik kontrol yöntemlerine göre daha stabil bir takip performansı sağlayabildiğini göstermektedir [4].

Ayrıca kamera tabanlı konum tahmini yöntemleri yalnızca tek bir İHA için değil, çoklu İHA sistemlerinde de kullanılabilir. Li ve arkadaşları tarafından gerçekleştirilen çalışmada, görüntü tabanlı konum tahmini kullanılarak otonom İHA navigasyonu ve kontrolü gerçekleştirilmiş ve görsel verinin navigasyon doğruluğunu önemli ölçüde artırdığı gösterilmiştir [5].

Literatürdeki çalışmalar incelendiğinde, görüntü tabanlı kontrol yöntemlerinin İHA'ların otonom navigasyonu, hedef takibi ve hassas konumlandırma görevlerinde etkin şekilde kullanılabildiği görülmektedir. Bununla birlikte görüntü gürültüsü, ölçüm gecikmesi ve hedef kaybı gibi problemler kontrol performansını etkileyen önemli faktörler arasında yer almaktadır. Bu nedenle birçok çalışmada görsel geri besleme ile farklı kontrol stratejileri veya ek geri besleme yöntemleri kullanılması önerilmektedir.

Bu çalışmada ise Parrot AR.Drone 2.0'ın ön kamerasından elde edilen görüntülerle, dışı yeşil içi kırmızı renk kodlu bir hedef çerçevenin tespit edilmesi ve İHA'nın hedefin merkezine hizalanarak çerçeve boşluğundan otonom geçişinin gerçekleştirilmesine odaklanılır. Hedef tasarımında iki renkli çerçeve kullanılması hem yanlış pozitifleri azaltmak hem de hedef doğrulamasını geometrik olarak güçlendirmek için tercih edilmiştir. İzleyen bölümde önerilen sistem ve deney düzeneği tanıtılmıştır.

2. ÖNERİLEN SİSTEM TANIMI VE DENEY DÜZENEGİ

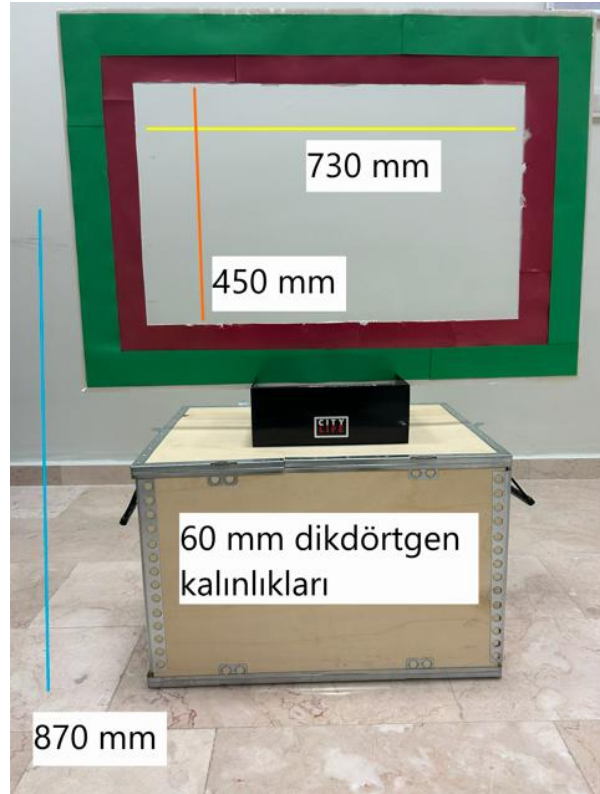
Önerilen sistem döner kanat İHA için tasarlanmıştır. Döner kanat İHA olarak Parrot AR.Drone 2.0 kullanılmıştır. Parrot AR.Drone maksimum yer düzlemine göre $\pm 15^{\circ}$ açı yaparak hareket etmektedir. 500 mAh 3S Lipo Batarya ile havada 7 dk kalabilmektedir. Bilgisayarda C# tabanlı, .Net Framework 4.8 ile yazılmış program ile İHA kontrolü sağlanmaktadır. İHA kablosuz ağ oluşturmaktadır. Bilgisayar ile bu ağa bağlanılıp yazılım ile komutlar gönderilmektedir. İHA'dan alınan gerçek zamanlı görüntü, aynı yazılımda görüntü işleme ile işlenir ve hedefin merkezi belirlenir. Görüntü işleme için EmguCV ve OpenCV fonksiyonları kullanılır.

Şekil 1'de AR.Drone 2.0 platformu ve fiziksel boyutları, Şekil 2'de hedef çerçeve ve deney düzeneği uzunluklarıyla beraber gösterilmiştir. Tablo 1'de AR.Drone ayrıntıları gösterilmiştir. Hedef; dış çerçevesi yeşil, iç çerçevesi kırmızı olan ve ortası boş bir dikdörtgenden

oluřmaktadır. İHA hedefin karřısına gelerek önce merkez hizalamasını saęlar, ardından ileri yönde komut ile açıklıktan geçmesi hedeflenmiřtir.



řekil 1. Parrot AR.Drone 2.0 platformu.



řekil 2. Deney hedefi

Tablo 1. AR.Drone Özellikleri

AR.Drone Özellikleri	Ayrıntılar
Ön Kamera	1280 x 720 piksel
Bağlantı	Wi-Fi bağlantı
Uçuş Süresi	7 dk
Açısal Eğim Kontrolü	-1.0 - 1.0

Önerilen sistem iki ana bileşenden oluşur: (i) algılama katmanı (görüntü işleme) ve (ii) kontrol katmanı (PID tabanlı komut üretimi). İHA'nın 1280×720 çözünürlüklü ön kamera görüntüsü işlenerek hedefin merkez konumu piksel koordinatlarında bulunur. Bu hata, İHA görüntü merkezine göre farkı piksel olarak bulunarak kontrol döngüsüne beslenir.

Önerilen sisteme dair yazılım C#/NET [11] ortamında gerçekleştirilmiştir. Görüntü akışı, AR.Drone 2.0 SDK [8] üzerinden alınmış ve Emgu CV [7] ile Mat yapıları üzerinde işlenmiştir. Renk ayrıştırma ve kontur analizi için OpenCV [6] fonksiyonları kullanılmıştır.

Görüntü işleme adımları olan balık gözü düzeltme, renk tabanlı ayrıştırma, kontur/köşe çıkarımı ve hedef doğrulama adımları alt bölümlerde özetlenmiştir. İşlem gerçek zamanlı çalışacak şekilde tasarlanmış ve Emgu CV (C#) ile OpenCV fonksiyonları kullanılmıştır.

2.1. Balık Gözü (FISHEYE) Düzeltme

AR.Drone 2.0'in ön kamerasında geniş açılı (balık gözü) lens nedeniyle özellikle görüntü kenarlarında belirgin radyal bozulma oluşur. Bu bozulma, dikdörtgen hedefin geometrik doğrulamasını ve köşe tespitini olumsuz etkiler. Bu nedenle görüntü, önceden elde edilmiş kamera iç parametreleri (K) ve bozulma katsayıları (D) kullanılarak düzeltilir. [12]

Kameranın içsel parametreleri K matrisi ile temsil edilmiştir. Bu matris odak uzaklıklarını (f_x, f_y) ve görüntü merkezini (c_x, c_y) içermektedir. Lens kaynaklı geometrik bozulmalar ise $D = [k_1, k_2, p_1, p_2, k_3]$ distorsiyon katsayıları ile modellenmiştir. Uygulamada Emgu CV ile "undistort/ remap" yaklaşımı tercih edilmiştir:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, D = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]$$
$$K = \begin{bmatrix} 620 & 0 & 320 \\ 0 & 620 & 180 \\ 0 & 0 & 1 \end{bmatrix}, D = [-0,24 \ 0,03 \ 0 \ 0 \ 0]$$

Bu bildiride, kalibrasyon parametrelerinin elde ediliŖi kapsam dıŖı bırakılmıŖ; hazır/ön tanımlı bozulma parametreleriyle geici düzeltme yapılmıŖtır. Bununla birlikte hazır parametre kullanımını hassas düzeltmeye engel olabileceđi belirtilmelidir. Hassas düzeltme için kameraya özel parametre, kameraya özel kalibrasyon yapılması gerekmektedir. Bu alıŖmada hazır parametrelerle yapılan düzeltme yeterli kabul edilmiŖtir. Ŗekil 3a’da düzeltme öncesi görüntü, Ŗekil 3b’de düzeltme sonrası görüntü paylaŖılmıŖtır. Özellikle hedef erevesindeki düzelme aıka görülmektedir.



Şekil 3.a) Düzeltme öncesi görüntü

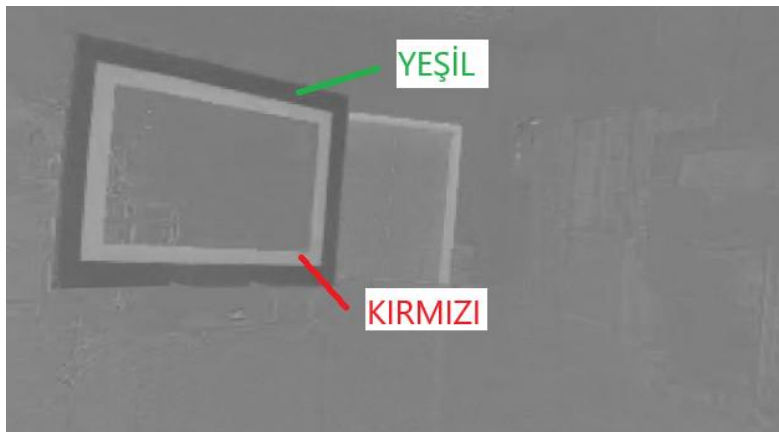


Şekil 3.b) Düzeltme sonrası görüntü

Şekil 3. Kalibrasyon ile görüntü düzeltme

2.2. Lab Renk Uzayında YeŖil ve Kırmızı ereve Tespiti

Renk tabanlı hedef tespitinde aydınlatma deđiŖimlerine karŖı daha kararlı olması nedeniyle RGB yerine CIELAB (Lab) renk uzayı kullanılmıŖtır. DüzleŖtirilmiŖ görüntü BGR’den Lab’ye dönüŖtürölür ve yeŖil ile kırmızı renkler için ayrı eŖikleme uygulanmıŖtır. Kırmızı ve yeŖil dikdörtgen için eŖik belirleme görüntüsü Ŗekil 4’te gösterilmiŖtir. Burada kırmızı ve yeŖil dikdörtgen için ayrı eŖikleme deđerleri ile dikdörtgenler arka plandan ayrılır. Gürültüyü azaltmak için Gauss bulanıklaŖtırma kullanılmıŖtır.[10] EŖik deđerleri ortam ışığına göre deneysel olarak belirlenmiŖ ve Tablo 2’de parametreleri sunulmuŖtur.



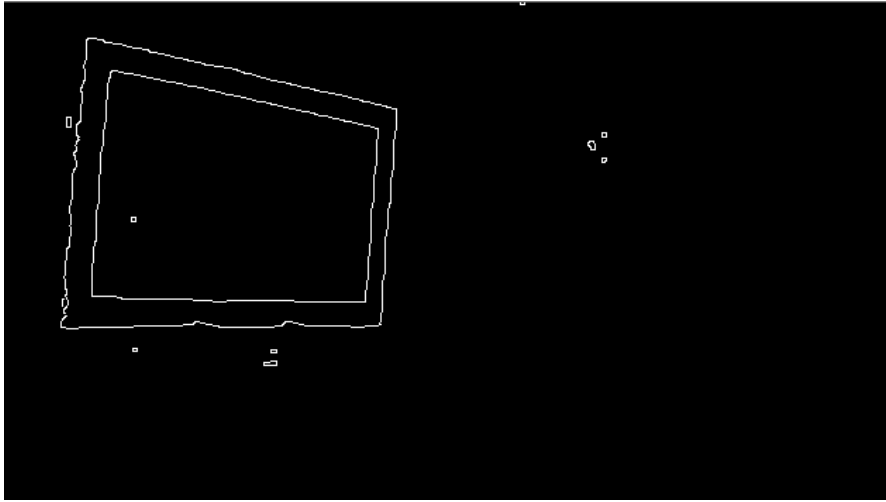
Şekil 4. Ayrı eŖikleme için kullanılan görüntü

Tablo 2. Renk ayrıştırma, düzeltme ve tespit işlemleri parametreleri

Renk	Lab Alt Eşik	Lab Üst Eşik	Son İşlem
Yeşil	118	255	Gauss,Canny,Find Contours
Kırmızı	142	255	Gauss,Canny,Find Contours

2.3. Kontur ve Köşe Noktalarının Çıkarılması

Eşiklenmiş ikili maskeler üzerinde kontur çıkarımı yapılmıştır. Kontur çıkarımı yapılmış görüntü Şekil 5'te gösterilmiştir. Her kontur için alan, dikdörtgensellik ve çokgen yaklaşımı kriterleriyle aday dikdörtgenler seçilmiştir. İdeal durumda hedef çerçeve için 4 köşeli bir çokgen elde edilmiştir. Köşe noktaları saat yönünde sıralanır ve geometrik testlerde kullanılmak üzere saklanmıştır.



Şekil 5. Yeşil dikdörtgen için uygulanmış kontur çıkarma işlemi görüntüsü

Yeşil çerçeve konturu dış sınırı, kırmızı çerçeve konturu ise iç sınırı temsil etmektedir. Her iki çerçeve için bulunan köşeler ile hedefin iç içe olma koşulu doğrulanmıştır.

2.4. Hedef Doğrulama ve Merkez Hesabı

Hedefin yanlış pozitiflere karşı sağlamlaştırılması için kırmızı dikdörtgenin yeşil dikdörtgen içinde olup olmadığı kontrol edilmiştir. Bu kontrol; kırmızı köşelerin yeşil köşelerin içinde kalma testi ile yapılmıştır.

Doğrulama sağlandığında hedef merkezinin piksel koordinatı, seçilen dikdörtgenin ağırlık merkezinden hesaplanır.

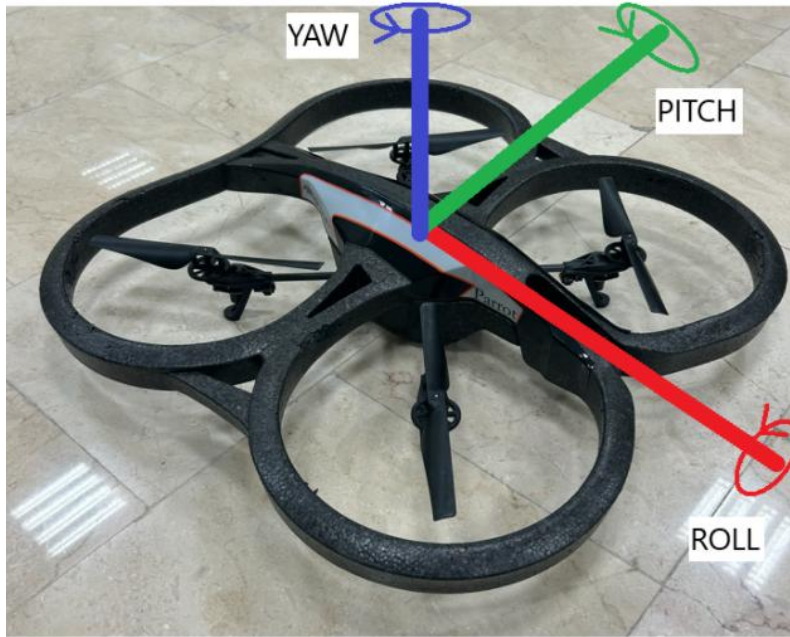
$$centerX = rect.X + rect.Width/2 \quad (1)$$

$$centerY = rect.Y + rect.Height/2 \quad (2)$$

Burada (centerX, centerY) hedef merkezidir. İHA görüntü merkezi ise (W/2, H/2) olarak alınır. İzleyen bölümde İHA'nın kontrol yöntemi gösterilmiştir.

3. KONTROL YÖNTEMİ

Bu bölümde hedef merkez-hatasından kontrol sinyallerinin üretimi açıklanmıştır. AR.Drone 2.0, gövde koordinat sisteminde roll (yanal), pitch (ileri-geri), yaw (dönüş) ve gaz (düşey) komutlarını kabul etmektedir. AR.Drone için bu komutlar Şekil 6'da görselleştirilmiştir. Çalışmada temel amaç, hedef merkezini görüntü merkezine hizalamaktır.



Şekil 6. AR.Drone Roll, Pitch, Yaw açıları

Bunun için görüntüdeki yatay hata ey roll/yaw ile, düşey hata ez gaz ile ve ileri yönde hareket (pitch) ex ileri-geri hata ile ilişkilendirilmiştir. Piksel hatası şu şekilde tanımlanır:

$$ex = 320 - centerX \quad (3)$$

$$ey = 180 - centerY \quad (4)$$

$$kenarHata = solkenar - sagkenar \quad (5)$$

$$ileriHata = 260 - ustKenar \quad (6)$$

ex bilgisayar ekran taramasında soldan sağa X eksen olduğu için yatay hatayı, ey bilgisayar ekran taramasında yukarıdan aşağı Y eksen olduğu için düşey hatayı temsil etmektedir. kenarHata, Roll eksen hareketini verebilmek için İHA'nın dikdörtgene göre açısını belli etmekte kullanılan hatayı temsil etmektedir.

İleriHata, Pitch eksen hareketi için gerekli hatayı temsil etmektedir. 260 sayıslı görüntüyü istenilen mesafede tutmak için deneysel belirlenen değerdir.



Şekil 7. İHA merkez ve hedef merkez gösterimi

Şekil 7’de görüldüğü üzere mavi çember ile İHA kamera merkezi işaretlenmiştir. Sarı daire ile tespit edilen dikdörtgen merkezi işaretlenmiştir. Mavi çember ile sarı daire arasındaki piksel cinsinden fark ile İHA Yaw eksenini düzeltmesi sağlanmaktadır. Şekil 7’de görüldüğü üzere kırmızı dikdörtgenin sol ve sağ kenarları kırmızı ile işaretlenmiştir. Bu kenarlar tespit edildikten sonra piksel cinsinden uzunluk hesabı yapılmıştır. Uzunluk farkı ile İHA Roll eksenini düzeltmesi sağlanmaktadır. İHA’nın dikdörtgenin içinden geçebilmesi için dikdörtgenin tam karşısında olması gerekmektedir. Bu nedenle Yaw ve Roll eksenleri düzeltmeleri sürekli olarak birlikte yapılmıştır. Şekil 7’de ki düzeltmede Roll düzeltmesi İHA’yı maksimum 0,1 komuta denk gelen -+1,5 derece olacak şekilde hareket ettirilmiştir. Bu hareket sırasında Yaw eksenini bozulur ve Yaw düzeltmesi İHA’yı merkezlemeye çalışmaktadır. Son olarak Yaw düzeltmesi İHA’yı merkezler ve İHA hem Yaw hem Roll eksenlerinde dikdörtgenin karşısına gelmektedir. İleri – Geri hizalanması içinse hesaplanan üst kenar uzunluğu piksel cinsinden kullanılmıştır.

3.1. Yaklaşım A: Görüntü Hatasına Doğrudan PID

İlk yaklaşımda kontrol girdisi doğrudan görüntü hatasından üretilir.

Yaw eksen için:

$$outYaw = KpYaw * ex + KiYaw * \int ex dt + KdYaw * d(ex)/dt \quad (7)$$

Gas eksen için:

$$outGas = KpGas * ey + KiGas * \int ey dt + KdGas * d(ey)/dt \quad (8)$$

Roll eksen için:

$$outRoll = KpRoll * kenarHata + KiRoll * \int kenarHata dt + KdRoll * d(kenarHata)/dt \quad (9)$$

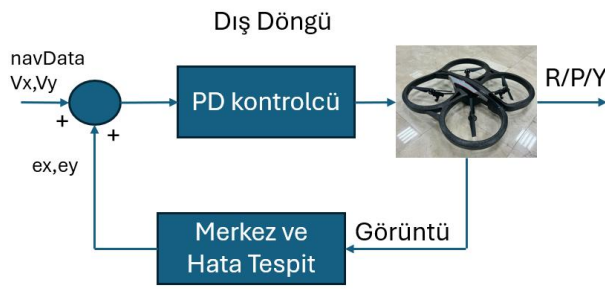
Pitch eksen için:

$$outPitch = KpPitch * ileriHata + KiPitch * \int ileriHata dt + KdPitch * d(ileriHata)/dt \quad (10)$$

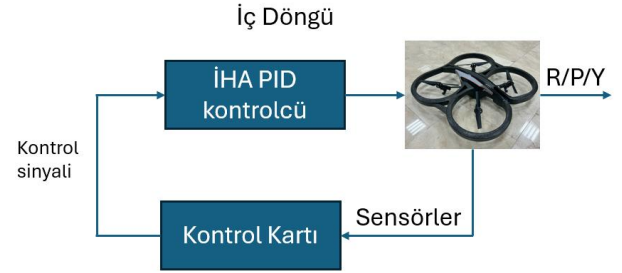
Burada dt, İHA'dan alınan iki görüntü arasında geçen süredir. 100 ms olarak kullanılmıştır. 100 ms'de bir görüntü alındığı durumda saniyede 10 görüntü elde edilmiştir. Bu değer ile yapılan denemelerin akıcı görüntü verdiği değerlendirilmiş, yapılan testlerde İHA'nın 100 ms'de bir verilen komutalara göre doğru hareket ettiği tespit edilmiştir.

3.2. Yaklaşım B: Vx-Vy Geri Beslemeli PID (Hız Döngüsü)

İkinci yaklaşımda, görüntü hatası önce istenen gövde hızlarına dönüştürülmüştür. (dış döngü) (Şekil 9.a), ardından navigasyon verisinden ölçülen hızlar (V_x , V_y) ile bir iç hız döngüsü (Şekil 9.b) kurulmuş olur.



Şekil 9.a) İHA Dış döngü blok diyagram



Şekil 9.b) İHA İç döngü blok diyagram

Şekil 9. Blok diyagramlar

Pitch ekseninde:

$$Vx_{error} = Vx_{ref} - Vx_{meas} \quad (11)$$

$$outPitch = KpPitch * Vx_{error} + KiPitch * \int Vx_{error} dt + KdPitch * d(Vx_{error})/dt \quad (12)$$

Vx_{ref} , kenar hatasından gelen değer katsayı ile çarpılması ile elde edilir.

Vx_{meas} , İHA navDatadan gelen Vx hızının deneysel elde edilmiş 0,1 değeriyle çarpılarak hataya dahil edilmiş halidir.

Benzer şekilde roll ekseninde:

$$Vy_{error} = Vy_{ref} - Vy_{meas} \quad (13)$$

$$outRoll = KpRoll * Vy_{error} + KiRoll * \int Vy_{error} dt + KdRoll * d(Vy_{error})/dt \quad (14)$$

Vy_{ref} , piksel hatasından gelen değer katsayı ile çarpılması ile elde edilmiştir.

Vy_{meas} , İHA navDatadan gelen Vy hızının deneysel elde edilmiş 0,05 değeriyle çarpılarak hataya dahil edilmiş halidir.

Bu yapıda kontrol, doğrudan piksel hatasına değil, hedeflenen ileri-geri ve yanal hıza ulaşmaya çalışılmıştır. Avantajı, İHA hızlandığında (Vx - Vy büyüdüğünde) kontrolün bunu algılayıp komutu azaltabilmesi ve aşımın azaltılmasıdır. Dezavantajı ise hız ölçümünün gürültülü olabilmesi ve ek filtreleme gerektirebilmesidir.

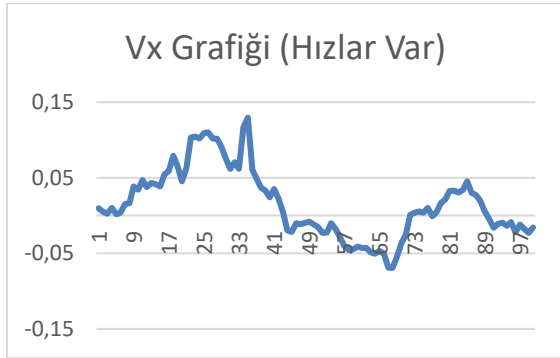
Her iki yöntem içinde aynı PID değerleri kullanılmış olup, I değeri 0 olarak belirlenmiştir. Sonuç olarak PD kontrolcü ile çalışma yapılmıştır.

3.3. Geçiş Fazı Yaklaşımı

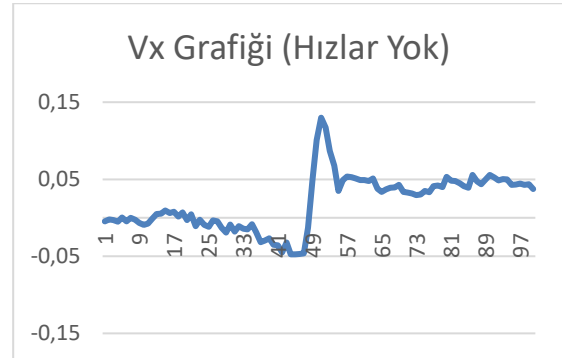
Hizalama fazı tamamlandığında (ör. $|e_x| < \epsilon_x$ ve $|e_y| < \epsilon_y$ koşulları 2 sn boyunca sağlandığında) İHA ileri yönde sabit bir pitch komutu ile hareket ettirilmiştir. Geçişin başarıyla gerçekleştiği; hedefin görüntüde hızla büyümesi, hedefin kaybolması veya hedef alanının belirli bir eşik üstüne çıkması gibi kriterlerle algılanabilir. İzleyen bölümde deneysel değerlendirme ve karşılaştırma grafikleri gösterilmiştir.

4. DENEYSEL DEĞERLENDİRME VE KARŞILAŞTIRMA

Bu bölümde, iki kontrol yaklaşımının (A: doğrudan PID, B: hız geri beslemeli PID) karşılaştırmalı olarak raporlanması için alınan grafikler gösterilmiştir. Denemeler formülde V_x ve V_y hızları varken ve yokken tekrarlanmıştır. Hedef ilerleme gerçekleşmeden önceki 100 görüntü karesi referans alınmıştır. 100 ms 'de bir görüntü alındığı için 10 sn'lik grafikler oluşmuştur. Grafiklerin y eksenini Şekil 10.a ve Şekil 10.b'de V_x hızlarını, Şekil 11.a ve Şekil 11.b'de V_y hızlarını göstermektedir.

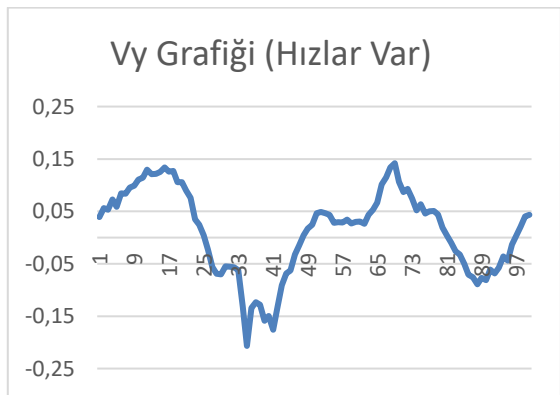


Şekil 10.a) Vx Grafiği (Formülde Hızlar Var)

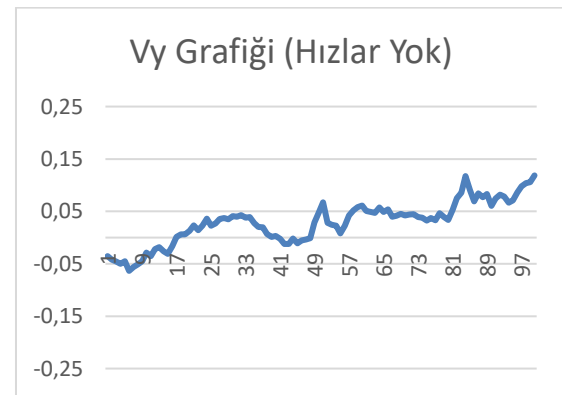


Şekil 10.b) Vx Grafiği (Formülde Hızlar Yok)

Şekil 10. Vx Grafikleri



Şekil 11.a) Vy Grafiği (Formülde Hızlar Var)



Şekil 11.b) Vy Grafiği (Formülde Hızlar Yok)

Şekil 11. Vy Grafikleri

Grafikler incelendiğinde, formülde hızlar var iken hatanın 0'a yaklaştığı, hızlar yok iken hatanın 0'dan uzak noktada salınım yaptığı görülmüştür. İzleyen bölümde sonuçlar açıklanmıştır.

5. SONUÇ

Bu bildiriye, AR.Drone 2.0 ön kamerası ile renk kodlu içi boş dikdörtgen hedefin tespiti ve otonom geçişi için görüntü işleme ve PID tabanlı kontrol yaklaşımı sunulmuştur. Balık gözü düzeltme sonrası Lab renk uzayında eşikleme ile yeşil ve kırmızı çerçeveler ayrıştırılmış, köşe noktaları üzerinden hedef doğrulaması yapılarak merkez hesaplanmıştır. Kontrol katmanında doğrudan görüntü hatasına dayalı PID ile, hız geri beslemeli (V_x-V_y) iç döngü yaklaşımı karşılaştırma çerçevesinde ele alınmıştır.

Beklenti, yaklaşım B'nin hız birikimini geri besleyerek özellikle hızlı manevralarda aşımı azaltması ve daha yumuşak komut üretmesidir. Ancak B yaklaşımı, hız ölçümlerindeki gürültü nedeniyle türev terimini hassaslaştırabilir; bu durumda düşük geçiren filtre (ör. üstel hareketli ortalama) veya türev terimini ölçülen hız üzerinden almak daha kararlı sonuçlar verebilir.

A yaklaşımı ise daha az parametreyle kurulabilir; ancak dış bozucular ve ataletsel etkiler altında salınım göstermesi muhtemeldir.

Gelecek çalışmalarda; kamera kalibrasyonunun hedefe özel yapılması, görüş kaybında güvenli davranışların tanımlanması ve daha ileri kontrol stratejilerinin denenmesi planlanmaktadır.

KAYNAKÇA

- [1] C. W. Chen, H. A. Hung, P. H. Yang, and T. H. Cheng, "Visual Servoing of a Moving Target by an Unmanned Aerial Vehicle," *Sensors*, 2021.
- [2] Z. Ceren and E. Altuğ, "Image Based and Hybrid Visual Servo Control of a Quadrotor UAV," *Journal of Intelligent & Robotic Systems*, 2012.
- [3] P. Durdevic and D. Arroyo, "A Deep Neural Network Sensor for Visual Servoing in 3D Spaces," *Sensors*, 2020.
- [4] G. Fu, H. Chu, L. Liu, L. Fang, and X. Zhu, "Deep Reinforcement Learning for the Visual Servoing Control of UAVs," *Drones*, 2023.
- [5] S. Li, M. Ozo, C. De Wagter, and G. De Croon, "Vision-Based Navigation and Control of a Quadrotor UAV," 2018.
- [6] OpenCV Documentation, "Color conversions (BGR \leftrightarrow Lab)"
- [7] Emgu CV Documentation, "Emgu CV: OpenCV in .NET".
- [8] Parrot AR.Drone 2.0 Developer Guide / SDK Documentation.
- [9] K. J. Åström ve R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

- [10] R. C. Gonzalez ve R. E. Woods, Digital Image Processing, 4th ed., Pearson, 2018.
- [11] Yury Rozhdestvensky , AR.Drone 2.0 controlling library for C#.NET
- [12] Hinz, S., & Topal, C. (2013) Camera calibration for mobile platforms using Parrot AR.Drone, Workshop on Computer Vision on Small Unmanned Aerial Vehicles (CV-SUAV)
- [13] Niemeyer, G., et al. (2015) Autonomous Indoor Navigation with Parrot AR.Drone 2.0, Stanford University, Mechanical Engineering Dept