



BİLECİK ŞEYH EDEBALI  
ÜNİVERSİTESİ

**BİLECİK  
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Ana Bilim Dalı**

**DÖNER KANATLI İHA OTONOM İNİŞ DENETLEME  
YAZILIMININ GELİŞTİRİLMESİ**

**Çağrı ÇİÇEKDEMİR  
Yüksek Lisans**

**Tez Danışmanı  
Doç. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2019  
Ref. No: 10277522**



BİLECİK ŞEYH EDEBALI  
ÜNİVERSİTESİ

**BİLECİK  
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Ana Bilim Dalı**

**DÖNER KANATLI İHA OTONOM İNİŞ DENETLEME  
YAZILIMININ GELİŞTİRİLMESİ**

**Çağrı ÇİÇEKDEMİR  
Yüksek Lisans**

**Tez Danışmanı  
Doç. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2019**



BİLECİK ŞEYH EDEBALI  
ÜNİVERSİTESİ

**BİLECİK**

**SEYH EDEBALI UNIVERSITY**

**Graduate School of Sciences  
Department of Computer Engineering**

**THE DEVELOPMENT OF ROTARY WING  
UAV LANDING CONTROL SOFTWARE**

**Çağrı ÇİÇEKDEMİR  
Master's Thesis**

**Thesis Advisor  
Assoc. Prof. Dr. Uğır YÜZGEÇ**

**BİLECİK, 2019**



**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS**  
**JÜRİ ONAY FORMU**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 26/06/2019 tarih ve 34/02 sayılı kararıyla oluşturulan jüri tarafından 11/07/2019 tarihinde tez savunma sınavı yapılan ÇAĞRI ÇİÇEKDEMİR 'in “DÖNER KANATLI İHA OTONOM DENETLEME YAZILIMININ GELİŞTİRİLMESİ” başlıklı tez çalışması BİLGİSAYAR MÜHENDİSLİĞİ Ana Bilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/ oy çokluğu ile kabul edilmiştir.

**JÜRİ**

**ÜYE : (TEZ DANIŞMANI) DOÇ. DR. UĞUR YÜZGEÇ**

**ÜYE : PROF. DR. CİHAN KARAKUZU**

**ÜYE : DR. ÖĞR. ÜYESİ NESİBE YALÇIN**

**ONAY**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun ...../...../..... tarih ve ...../..... sayılı kararı.

**İMZA/ MÜHÜR**

## TEŐEKKÜR

Tez alıőmam süresince gerek pratikte gerekse araőtırmalar konusunda yardımlarını benden esirgemeyen saygı deęer danıőman hocalarım Do. Dr. Metin KESLER ve Do. Dr. Uęur YÜZGE'e ok teőekkür ederim. Dięer taraftan tüm mekanik alıőmalar sırasında yanımda olan Arő. Gör. Hakan ÜĞÜN'e ve yazılım tarafında fikirleriyle yardımcı olan İbrahim KARATAŐ'a teőekkürlerimi sunarım.

Tüm bu alıőmalar süresince bana yardımcı olan, varlıkları ile destek ve moral veren eőime, aileme ve özellikle babam Sabri İEKDEMİR'e, beni her zaman ve her konuda destekledikleri için teőekkürlerimi sunuyorum.

Saygılarımla...

## **BEYANNAME**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kılavuzu'na uygun olarak hazırladığım bu tez çalışmada, tez içindeki tüm verileri akademik kurallar çerçevesinde elde ettiğimi, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun olarak sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu Üniversite veya başka bir üniversitede herhangi bir tez çalışmada kullanılmadığını beyan ederim.

...../...../ 2019

**Adı Soyadı)**

# DÖNER KANATLI İHA OTONOM İNİŞ DENETLEME YAZILIMININ GELİŞTİRİLMESİ

## ÖZET

İnsansız Hava Araçları (İHA), insan kontrolüne gerek duymadan otonom olarak uçabilen ya da bir yer istasyonu vasıtasıyla kontrol edilebilen hava araçlardır. Günümüzdeki teknolojik gelişmelere paralel olarak İHA'ların kullanımında da büyük bir artış gözlenmiştir. Bu kapsamda, İHA'ların yaygın kullanımıyla birlikte yeni çalışma alanlarına da ihtiyaç duyulmuştur. Yapılan araştırmalar sonucunda, otonom kontrol çalışmalarının diğer alanlara göre ön plana çıktığı görülmüştür. Bu çalışma kapsamında, İHA'lar için görüntü işleme tabanlı bir otonom iniş sistemi gerçekleştirilmiştir. İHA'nın alt kısmında bulunan yüksek çözünürlüklü ve geniş açılı kamera yardımıyla alınan görüntüler, Linux tabanlı işletim sistemi üzerinde görüntü işleme tekniklerine tabi tutularak iniş pistinin tespit edilmesi sağlanmaktadır. Tespit edilen iniş pisti, ortasında "H" harfi bulunan çember biçimindeki bir düzlem olarak tasarlanmıştır. Çemberin ve "H" harfinin rengi kırmızı seçilmiştir. Görüntü işleme teknikleri ile iniş pisti tespit edildikten sonra İHA, pist ile iz düşüm olarak üst üste geldiği anda havada askıda kalma moduna geçer. İHA'nın yüksekliği belirli bir oranda otonom olarak azaltılır. İHA, istenen yüksekliğe geldikten sonra, motorlar durdurularak iniş işlemi tamamlanır.

**Anahtar Kelimeler:** İnsansız Hava Aracı; Döner Kanat; İniş Pisti; Görüntü İşleme; Otonom

## **THE DEVELOPMENT OF ROTARY WING UAV LANDING CONTROL SOFTWARE**

### **ABSTRACT**

Unmanned Aerial Vehicles (UAVs), are air aircraft that can fly autonomously without human control or that can be controlled by a ground station. In parallel with the technological developments of the present day, it has been observed that the use of UAVs is also greatly increased. In this context, new working areas were needed together with the widespread use of UAVs. As a result of the researches, the autonomous control works were seen to become the foreground. Within the scope of this study, an autonomous landing system based on image processing has been realized for UAVs. The images taken with the help of a wide angle and high-resolution camera located at the bottom of the UAV are subjected to image processing techniques on the Linux based operating system to determine the landing station. The designed landing station is a circle with a "H" in the middle. The circle and letter "H" on the landing station are selected in red color. After determining landing station, the altitude of UAV is gradually decreased and the image acquisition and measurement process are repeated until it reaches a desired position. When arriving at the desired position, the UAV performs descent by stopping motors.

**Keywords:** Unmanned Aerial Vehicles; Rotary Wing; Landing Station; Image Processing; Autonomous

## İÇİNDEKİLER

|   | Sayfa No  |
|---|-----------|
| <b>TEŞEKKÜR</b> .....                                 |           |
| <b>BEYANNAME</b> .....                                |           |
| <b>ÖZET</b> .....                                     | I         |
| <b>ABSTRACT</b> .....                                 | II        |
| <b>ŞEKİLLER DİZİNİ</b> .....                          | IVI       |
| <b>ÇİZELGELER DİZİNİ</b> .....                        | IV        |
| <b>SİMGELER ve KISALTMALAR DİZİNİ</b> .....           | VI        |
| <b>1. GİRİŞ</b> .....                                 | 1         |
| <b>2. KULLANILAN DONANIM ve YAZILIM</b> .....         | 7         |
| 2.1. Döner Kanatlı İHA ve Donanımsal Özellikleri..... | 7         |
| 2.1.1. Çerçeve.....                                   | 9         |
| 2.1.2. Fırçasız DC Motor ve Pervane .....             | 10        |
| 2.1.3. Elektronik Hız Kontrolcüsü .....               | 11        |
| 2.1.4. Pil .....                                      | 12        |
| 2.1.5. Şarj Cihazı.....                               | 12        |
| 2.1.6. Uzaktan Kumanda ve Alıcısı.....                | 13        |
| 2.1.7. Ana Kontrol Kartı.....                         | 13        |
| 2.1.8. Raspberry Pi 3 Model B .....                   | 16        |
| 2.1.9. Kamera .....                                   | 18        |
| 2.2. Görüntü işleme yazılımı .....                    | 19        |
| 2.2.1. İniş düzleminin algılanması .....              | 20        |
| 2.3. Uçuş yazılımı.....                               | 23        |
| 2.4. Deneysel Sonuçlar.....                           | 24        |
| <b>3. SONUÇ</b> .....                                 | <b>31</b> |
| <b>KAYNAKLAR</b> .....                                | <b>33</b> |
| <b>EKLER</b> .....                                    | <b>36</b> |
| <b>ÖZ GEÇMİŞ</b> .....                                |           |

## ŞEKİLLER DİZİNİ

|  | <b>Sayfa No</b> |
|--|-----------------|
| Şekil 1.1. İHA tipleri.....  | 2               |
| Şekil 1.2. Kızılötesi kamera ile düzlem tespiti (Wenze, 2010).....                 | 4               |
| Şekil 1.3. Carreira çalışmasında kullanılan işaret düzlemi (Gomes, 2013). ....     | 4               |
| Şekil 1.4. Jung ve arkadaşlarının kullandıkları iniş düzlemi (Youeyun, 2015). .... | 5               |
| Şekil 1.5. Bruno ve arkadaşlarının RGB/HSV iniş düzlemi (Costa vd., 2015). ....    | 6               |
| Şekil 2.1. Hexacopter. ....  | 7               |
| Şekil 2.2. İHA yapılandırma tipleri.....   | 8               |
| Şekil 2.3. Özel firma çerçeveleri.....   | 9               |
| Şekil 2.4. Hexccopter çalışma prensibi.....  | 10              |
| Şekil 2.5. Kaldırma kuvveti ekipmanları. ....                                      | 11              |
| Şekil 2.6. Elektronik hız kontrolcüsü.....   | 12              |
| Şekil 2.7. LiPo pil (12V 3S 5000mAh). ....   | 12              |
| Şekil 2.8. LiPo şarj ve kontrol ekipmanları. ....                                  | 13              |
| Şekil 2.9. Uzaktan kontrol ekipmanları. ....                                       | 13              |
| Şekil 2.10. Çalışmada kullanılan açık kaynak kodlu autoquad uçuş kartı (URL8)..... | 14              |
| Şekil 2.11. Autoquad ve D-IMU (URL8). ....   | 15              |
| Şekil 2.12. Raspberry Pi 3 Model B (Smolov, 2017). ....                            | 17              |
| Şekil 2.13. Raspberry Pi 3 ve çevre birimleri (URL17). ....                        | 18              |
| Şekil 2.14. SJ5000x geniş açılı kamera. ....                                       | 18              |
| Şekil 2.15. Görüntü işleme ile çözülen yaygın problemler.....                      | 19              |
| Şekil 2.16. İniş düzlemi algılama adımları. ....                                   | 20              |
| Şekil 2.17. HSV renk uzayı.....  | 22              |
| Şekil 2.18. PID blok diyagram.....   | 23              |
| Şekil 2.19. Autoquad akış diyagramı. ....  | 24              |
| Şekil 2.20. Kullanılan sistemin bağlantı şeması. ....                              | 25              |
| Şekil 2.21. Rasbian üzerinden veri aktarımı. ....                                  | 26              |
| Şekil 2.22. Raspberry üzerinde çalışan uygulama adımları.....                      | 26              |
| Şekil 2.23. Raspberry üzerinde çalışan uygulamanın kaba kodu. ....                 | 27              |
| Şekil 2.24. Autoquad üzerinde çalışan uygulama adımları.....                       | 28              |
| Şekil 2.25. Autoquad üzerinde çalışan uygulamanın kaba kodu. ....                  | 29              |

|   |    |
|---|----|
| <b>Şekil 2.26.</b> Çalışmada kullanılan İHA. ....               | 30 |
| <b>Şekil 2.27.</b> İniş düzlemi algılama ortam testleri 2. .... | 36 |
| <b>Şekil 2.28.</b> İniş düzlemi algılama ortam testleri 3. .... | 37 |

**ÇİZELGELER DİZİNİ**

|  | <b>Sayfa No</b> |
|--|-----------------|
| <b>Çizelge 3.1.</b> Motor parametreleri. ....                    | 11              |
| <b>Çizelge 3.2.</b> Raspberry Pi modelleri karşılaştırması. .... | 16              |

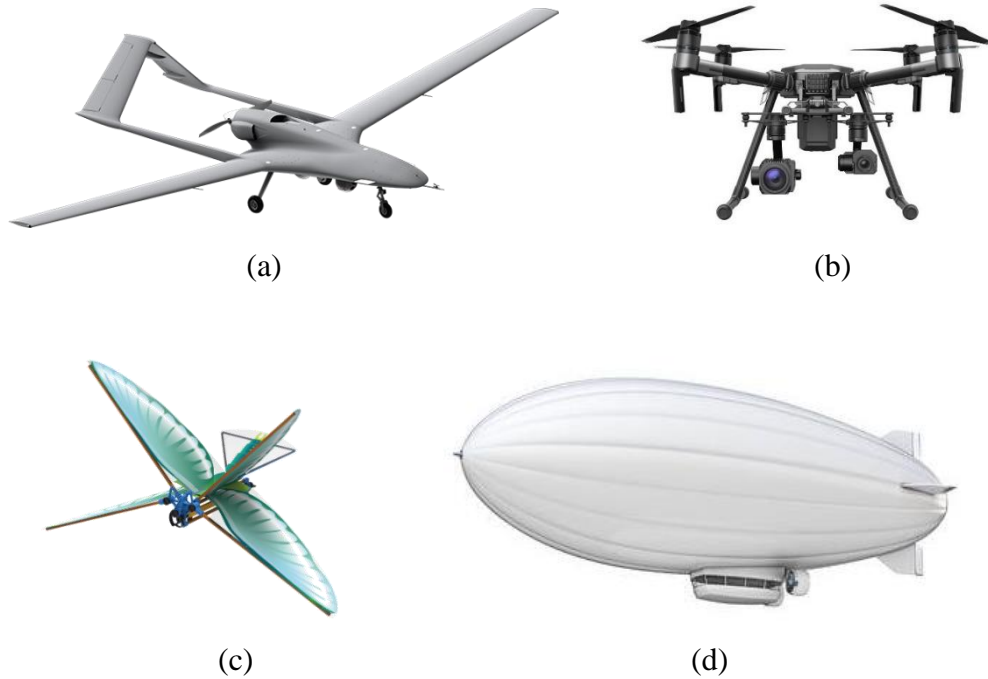
**SİMGELELER ve KISALTMALAR DİZİNİ**

|       |                                      |
|-------|--------------------------------------|
| İHA   | : İnsansız Hava Aracı                |
| UAV   | : Unmanned Aircraft Vehicle          |
| ILS   | : Instrument Landing System          |
| TLS   | : Transponder Landing System         |
| RGB   | : Red Green Blue                     |
| HSV   | : Hue Saturation Value               |
| GPS   | : Global Positioning System          |
| IMU   | : Inertial Measurement Unit          |
| LİPO  | : Lithium Polymer                    |
| ESC   | : Electronic Speed Controller        |
| PWM   | : Pulse Width Modulation             |
| GCS   | : Ground Control Station             |
| LIDAR | : Light Detection and Ranging        |
| VTOL  | : Vertical Take Off Landing          |
| BSD   | : Berkeley Software Distribution     |
| GUI   | : Graphical User Interface           |
| PID   | : Proportional, Integral, Derivative |
| USB   | : Universal Bus Line                 |
| CCW   | : Counter Clock Wise                 |
| CW    | : Clock Wise                         |

## 1. GİRİŞ

İnsansız hava araçlarının (İHA) gelişimi son yıllarda hızla artmaktadır. Sabit kanatlı ve döner kanatlı olarak iki ana sınıfa ayrılan İHA'lar askeri, ziraat, fotoğrafçılık, şehircilik, trafik izleme gibi farklı alanlarda kullanılmaktadır. Tam kontrollü insanlı bir hava aracı ile karşılaştırıldığında yüksek maliyetleri nedeniyle tehlikeli görevlerde daha çok tercih edilmektedir. Genellikle *drone* olarak adlandırılan İHA'ların üretilmesi ve kullanılması insanlı hava araçlarına göre daha ucuz olması nedeniyle popülerdirler. Birçok teknolojik gelişmede olduğu gibi İHA'larda ilk olarak askeri amaçla geliştirilmeye başlanmıştır. Ancak İHA'ların günümüzde sivil uygulamalara yönelik ihtiyaçları karşılayabilmesinden dolayı kullanımları giderek artmaktadır. Kullanım alanları olarak yangın söndürme, gözetleme, hava fotoğrafçılığı, haritalama, kimyasal ilaçlama gibi sıralanabilir.

İHA platformları Şekil 1.1.'de gösterildiği gibi uçuş özelliklerine göre dört kategoride sınıflandırılabilir; sabit kanatlılar, döner kanatlılar, çırpma kanatlılar ve hava dolumlulardır. Sabit kanatlılar, kalkış ve iniş için piste ihtiyaç duyarlar. Dikey kaldırma kuvvetini kanatlara çarpan rüzgâr sayesinde sağladığı için yüksek irtifalara çıkabilirler ve uzun uçuş sürelerine sahiptirler. Döner kanatlılar, dikey iniş kalkış yapabilme avantajına sahiptirler. Bunun yanında yüksek manevra kabiliyetleri bulunmakta olup, havada sabit durabilme özelliğine de sahiptirler. Hava dolumlu araçlar ise havadan daha az öz kütleli gazlar ile uçabilirler. Hız olarak yavaş ve hacim olarak büyük olmaları en önemli dezavantajlarıdır. Ornithopter olarak adlandırılan kanat çırpmalı araçlar; kuş, böcek benzeri uçabilen hayvanlardan ilham alınarak geliştirilmiştir. En önemli özellikleri ise çok küçük boyutlarda geliştirile bilinmesidir. Bunların dışında sabit kanatlı olarak görünse de rotorların dikey olarak yön değiştirmesi ile dikey kalkış iniş yapabilen hibrid tipli araçlarda vardır (Peddle, 2008).



**Şekil 1.1.** İHA tipleri.

(a) Sabit kanatlılar (b) Döner kanatlılar (c) Çırpma kanatlılar (d) Hava dolumlular

İHA'lar kontrol yöntemlerine göre uzaktan bir pilot yardımıyla kontrol edilebilenler ile önceden planlanmış uçuş programlarına dayanarak otonom olarak uçabilenler olmak üzere ikiye ayrılırlar. Uzaktan kumandalı olan araçlarda pilotlar için uzun süreli eğitimlere ihtiyaç duyulmaktadır. Diğer taraftan, otonom uçuşlar için karmaşık algoritmalar ve güçlü donanımlar gerekmektedir (Kırçalı, 2013).

İnsansız uçuş yapan hava araçlarının ana denetleyicilerinin görevleri kalkış, uçuş planlaması, iniş düzleminin tespiti ve otomatik iniş olarak dört ana grupta sınıflandırılabilir. Birçok araştırmacı, karmaşık uçuş planlaması görevlerine odaklanmıştır. Uçuş planlama görevinin sonunda, son varış noktasına ulaşıldıktan sonra, iniş için uygun bir yer bulmak önemlidir. Büyük araçlar için iniş bölgesi genellikle, ILS (*Instrument Landing System*) veya TLS (*Transponder Landing System*) gibi mevcut sistemlerin kullanılabileceği bir pisttir. Bu tür sistemlerin olmadığı yerlerde görüntü tabanlı metotlar kullanılabilir. Küçük boyutlu araçlar için ise bir yol ya da boş bir alan yeterlidir. Ancak dikey kalkış iniş yapabilen döner kanatlı araçların iniş bölgesi için piste ihtiyaç duyulmamaktadır. İniş bölgesi büyüklük olarak aracın boyutlarından biraz daha geniş olmalıdır. Bu sebeple, döner kanatlı araçların iniş bölgesinin tespiti, sabit kanatlı araçlara göre daha karmaşıktır.

Döner kanatlı İHA'lar, uçuş yapmak için gerekli olan enerjiyi aracın altına konumlandırılmış ağır ve limitli kapasiteye sahip Li-Po pillerden sağlamaktadır. Tam dolu pil kapasitesi ve uygun ortam koşullarında bile uçuş süresi 60 dakikadan daha azdır. Eğer kamera ya da başka aygıt taşınıyorsa, uçuş süresi bu süreden daha da az olacaktır. Bu nedenle pillerin sık sık şarj edilmesi gerekmektedir. Ancak günümüzde şarj işlemi, İHA iniş yaptıktan sonra pilot/kullanıcı tarafından başlatılabilmektedir (Çiçekdemir, 2012).

Li-Po pillerinde şarj süresi gücünün tükenme hızına göre daha çok zaman almaktadır. Ortalama özelliklere sahip bir Li-Po şarj cihazı ile iki saate yakın dolun işlemi yapılır. Ancak acil uçuş görevlerinde şarj işlemlerinin beklenmesi durumunda söz konusu görevler riske girecektir. Bu tarz sorunlar, İHA iniş yaptıktan sonra otomatik şarj sistemleri ile çözülebilir. Gelecek çalışmalarda otomatik şarj sistemlerine yoğunlaşacaktır.

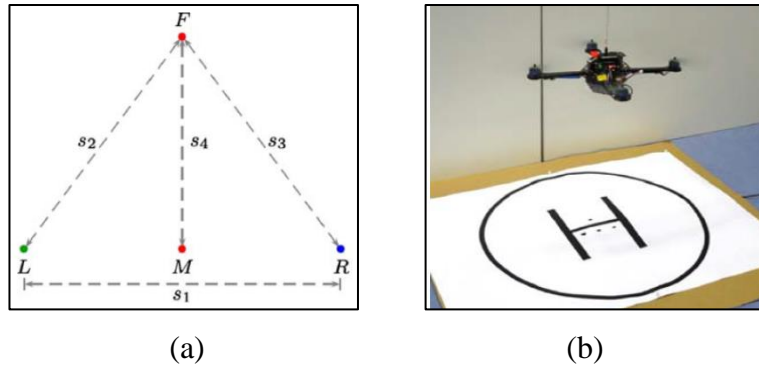
Literatürde İHA'ların otonom iniş sistemleri ile ilgili farklı çalışmalar yapılmıştır. Bu çalışmalardan aşağıda kısaca bahsedilmektedir.

Leong Woei ve arkadaşları, pilot tarafından gerçekleştirilen iniş verilerini analiz etmişler ve uçağın matematiksel modelini doğrulamak için sistem tanımlama tekniklerini kullanmışlardır. İniş verilerine dayanarak klasik PI yaklaşımı ile bir denetleyici tasarlamışlar. Uçağın mevcut konumuna bağlı olarak referans alınan merkez noktasının kayma eğilimindeki hatayı minimuma indirmişlerdir. İniş simülasyon sonuçları, uçağın simülasyonda inebileceğini ancak simüle edilmiş inişin doğruluğunu belirtmediğini göstermiştir (Leong, 2006).

Cho Am ve arkadaşları, İHA'nın tam otomatik uçuş, iniş ve kalış işlemlerini tek antene sahip bir GPS alıcısı ile gerçekleştirmişler. Yüksek çözünürlüklü GPS alıcısının yanında ek olarak hava hız sensörü kullanılmıştır. Otomatik iniş ve kalkış sisteminde doğrusal kuadratik regülatör (*LQR: Linear Quadratic Regulator*) kontrolcüsü kullanılmıştır. Test sonuçlarına ve simülasyon verilerine göre tek bir GPS antenini İHA için yedek veya düşük maliyetli bir kontrol sistemi olarak kullanılabilceğini belirlemişlerdir. Sistemin en büyük avantajı, pistten çok uzakta iniş faaliyetlerine başlanması göstermiştir (Cho, 2008).

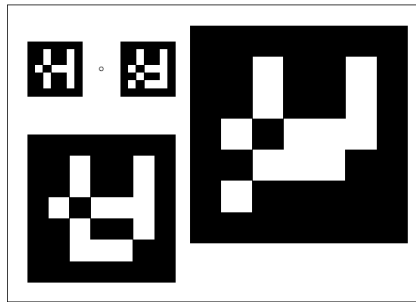
Karl E. Wenze ve arkadaşları İHA'ların otomatik iniş ve iniş düzlemi üzerinde dolaşma işlemlerinde düşük maliyetli teknolojiler kullanmaya odaklanmışlardır. Bu

kapsamda kızılötesi özellikli Wiimote (URL1) kamerasını kullanılmıştır. Kızılötesi özelliği ile günışığından etkilenme oranı minimuma indirilmiştir. Wiimote kamera ile Şekil 1.2.'de görüntü üzerindeki 4 noktanın piksel değerleri kontrol algoritmasına gönderilmiştir. Noktaların uzaklığına göre zemine olan yükseklik ve izdüşümü belirlenmiştir. Bu sayede kontrollü iniş sağlamıştır (Wenze, 2010).



**Şekil 1.2.** Kızılötesi kamera ile düzlem tespiti (Wenze, 2010).  
(a) Desen konfigürasyonu (b) Kamera görüşü

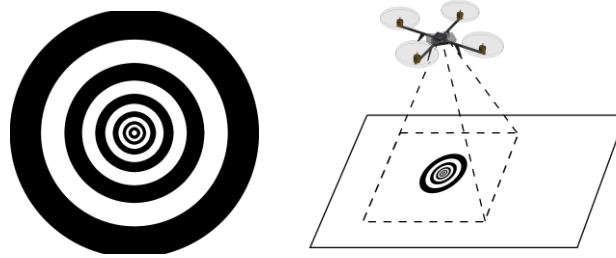
Tiago Gomes, çalışmasında İHA için uçuş sırasında özel bir kararlılık algoritması gerçekleştirerek aracın kontrollü inişini sağlamıştır. İniş işleminin sadece iç sensörler ve günümüzde yaygın olarak kullanılan bir kamera ile gerçekleştirilmesine önem vermiştir. Görüntü işleme tabanlı bir algoritma kullanılmıştır. Algoritma sayesinde, İHA'nın havadaki pozisyonu tahmin edilerek kontrollü inişi sağlanmaktadır. Geliştirilen algorithma Şekil 1.3'de gösterilen düzlemsel işaretleyici kullanılmıştır. İşaretleyiciler ile pozisyon tahmini yapılmıştır. Tüm çalışmalar gerçek bir İHA üzerinden başarılı sonuç vermese de simülasyon üzerinde başarıya ulaşmıştır. (Gomes, 2013).



**Şekil 1.3.** Carreira çalışmasında kullanılan işaret düzlemi (Gomes, 2013).

Lizarraga M., Amerikan Deniz Akademisinde yüksek lisans çalışmasında İHA'lar için otomatik iniş sistemi çalışmasında bulunmuştur. Çalışmasında klasik kontrol teorisine ve yavaş dış döngü (outer loop) kontrolcülerine dayalı hızlı iç döngü (inner loop) kontrolcüsü kullanmıştır. HIL (Hardware-in-the-loop simulation) simülasyonunda geliştirilen senaryoya göre, İHA çeşitli rüzgâr ve sensör gürültüsüne maruz bırakılmış. İHA, simülasyon üzerinde söz konusu gürültülüler altında 0,27 metre standart sapma ile hedefe inmiştir. Simülasyona göre geliştirilen metot, iç döngü kontrol teknikleri ile başarılı sonuç vermiştir (Lizarraga, 2014).

Youeyun Jung ve arkadaşları, iniş düzlemi olarak Şekil 1.4'te gösterilen arka planı beyaz ve üzerinde siyah renkte çember olan düzlem kullanmışlardır. Görüntüler PX4FLOW kameradan alınmıştır (URL2). Kamera dahili kızılötesi görüşe ve ultrasonik sensöre sahiptir. Zemine yaklaştıkça görüntüde oluşan bozuklukları, iç içe sürekli çember çizerek gidermeyi hedeflemişlerdir. Zemin tespitinde Hough dönüşümü (*Hough Transform*) (URL3) ve kenar tanımlama (*Edge Dedection*) (Dharampal ve Vikram, 2015) algoritmaları kullanılmıştır. Kontrol algoritmasına ek olarak barometre ile ultrasonik sensör verileri işlenerek yükseklikte değişimler tespit edilmiş ve başarılı iniş yapılmıştır (Youeyun, 2015).



**Şekil 1.4.** Jung ve arkadaşlarının kullandıkları iniş düzlemi (Youeyun, 2015).

Bruno Sielly Costa ve arkadaşları, RGB/HSV renk uzayını baz alarak farklı tasarımda iniş düzlemi geliştirmişlerdir. Şekil 1.5'te gösterildiği gibi ana renklerden oluşan 4 kare kutucuk ve ortasında çember olan bir düzlem kullanılmıştır. Kameradan gelen görüntüler dört aşamaya bölünmüş ve bir algoritmada sırasıyla ton filtreleme, kümeleme, eşleştirme ve merkezi konuma odaklama yöntemleri kullanılmıştır. Tüm işlemler android platformunda geliştirilmiştir. Sistemin en büyük avantajı, her bir resmin 777 milisaniye içinde işlene bilinmesidir (Costa vd., 2015).



**Şekil 1.5.** Bruno ve arkadaşlarının RGB/HSV iniş düzlemi (Costa vd., 2015).

Hadrien Beck ve arkadaşları sabit bir noktaya otomatik iniş için geri beslemeli kontrole dayalı metot geliştirmişlerdir. Bu yöntemde, İHA'nın durumunu ve hızını tahmin etmek için IMU ve GPS'den gelen veriler birleştirilerek, Kalman filtresinde kullanılmıştır. Ek olarak, iniş düzlemi ile İHA'nın mesafesini ölçmek için konumlandırma cihazı yerine Işık Algılama ve Değişme (LIDAR) sensörü kullanılmıştır. Elde edilen sonuçlara göre İHA'nın otomatik iniş yeteneğini kazandığı görülmüştür. Ancak, IMU ve GPS sensörlerinde oluşan hatalı verilerden dolayı, İHA'nın yüksekliğinin hesaplanmasında sorunlarla karşılaşmıştır (Beck vd., 2016).

Bu tez çalışması kapsamında, otonom uçuş özelliği olan döner kanatlı insansız bir hava aracının görsel kontrol yaparak önceden tasarlanmış bir piste iniş yapması gerçekleştirilmiştir. İniş pisti, beyaz zemin üzerinde kırmızı renkli çember içinde yine kırmızı renkli "H" sembolü olacak şekilde tasarlanmıştır. İniş işlemi, sembolün İHA'nın altına yerleştirilen geniş açılı kamera aracılığıyla çekilen görüntüler içinde bulunmasıyla başlar. İHA sabit hızla irtifasını azaltarak piste inişini tamamlar.

## 2. KULLANILAN DONANIM ve YAZILIM

Tez çalışması kapsamında hava aracı olarak döner kanatlı İHA modellerinden 6 rotorlu bir araç kullanılmıştır. İHA modeli ve donanımsal özellikleri ile ilgili ayrıntılı bilgiler alt bölümlerde verilmiştir. İHA'nın otonom uçuş ve iniş pistinin tespiti için birbirleri ile seri port üzerinden iletişim kuran iki ayrı yazılım bulunmaktadır. Otonom uçuş yazılımı için popüler ürünlerden olan açık kaynak kodlu *AutoQuad* platformu kullanılmıştır. İHA, havada AutoQuad kontrol kartı ile otonom uçuş gerçekleştirirken, yüksek çözünürlüklü kameradan gelen görüntüler, Raspberry Pi 3 mikro denetleyicisi tarafından görüntü işleme teknikleri kullanılarak iniş yerinin tespiti yapılmaktadır. Görüntü işleme metotları C++ programlama dili ve OpenCV kütüphanesi ile gerçekleştirilmiştir. Geliştirilen yazılımlarla ilgili ayrıntılı bilgiler alt bölümlerde verilmiştir.

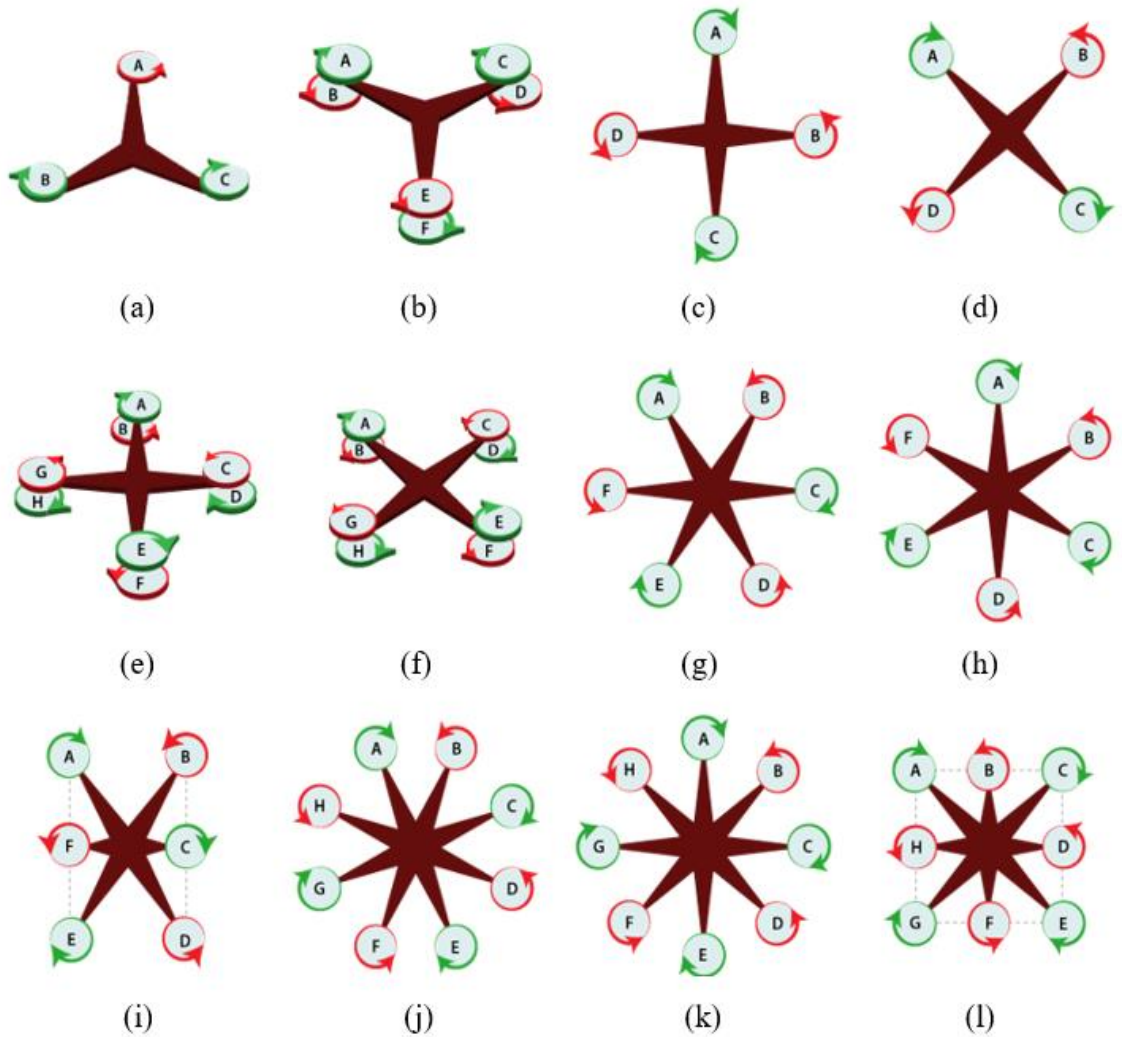
### 2.1. Döner Kanatlı İHA ve Donanımsal Özellikleri

ISO/TC20/SC16 insansız hava araçları standartlarına göre varsayılan versiyon olarak 4 rotorlu *QuadCopter* modeli kabul edilmektedir (URL4). Bu tez çalışması kapsamında kullanılan döner kanatlı İHA, QuadCopter'e göre fazladan 2 rotora sahip olan *Hexacopter* modelidir. Motor sayısının artması ve yeni kolların eklenmesi, İHA'nın maliyetini arttırmaktadır. Maliyetten yana oluşan olumsuz durum, motor sayısındaki artış ile olası motor bozulmalarında daha kontrollü bir şekilde iniş sağlayarak dengelenmiştir. Diğer taraftan taşıma kapasitesi de artmıştır. Çalışmada kullanılan, motorları ve pervaneleri takılmış, mikro denetleyicisi ve diğer sensörleri aktifleştirilmiş bir hexacopter sınıfı dönerkanatlı İHA modeli Şekil 2.1'de gösterilmektedir.



Şekil 2.1. Hexacopter.

İHA platformları, motor konumları bilindiği sürece farklı sayıda rotor ile istenen herhangi bir şekilde oluşturulabilir. Dikey kalkış yapan ortak İHA yapılandırmaları Şekil 2.2’de gösterilmiştir. Kontrol kartının yönünü değiştirerek farklı yapılandırmalar oluşturulabilir. Rotor sayısının dörtten fazla olması, uçuşun daha stabil olacağı anlamına gelmemekle beraber daha fazla kaldırma kapasitesi ve olası motor problemlerine karşı dayanıklılık kazandırır. Çalışmada, üzerinde ana kontrol kartı dışında görüntü işlemleri için ikinci bir kontrol kartı ve zemin tespiti için harici kamera kullanıldığı için kaldırma kapasitesi daha yüksek olan hexacopter modeli seçilmiştir.

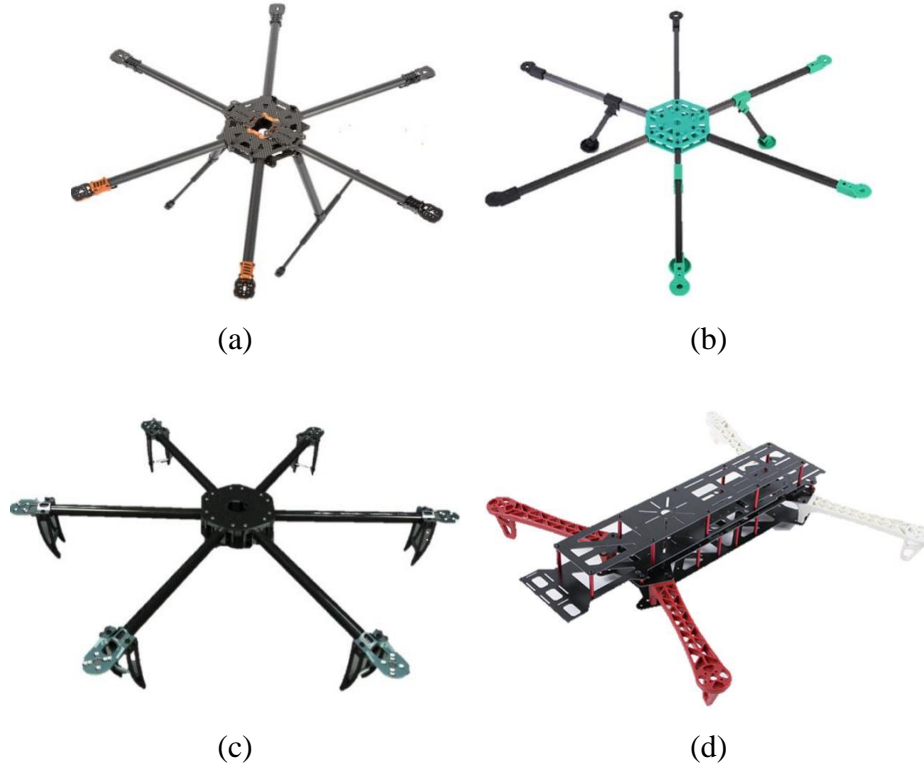


Şekil 2.2. İHA yapılandırma tipleri.

(a) Tricopter (b) Hex-Tricopter (c) Quadcopter Plus (d) Quadcopter X  
 (e) Qcto Quadcopter Plus (f) Qcto Quadcopter X (g) Hexacopter X (h) Hexacopter Plus  
 (i) Hexacopter H (j) Octocopter X (k) Octocopter Plus (l) Octocopter H

### 2.1.1. Çerçeve

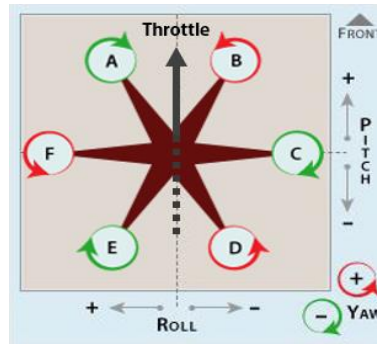
Günümüzde VTOL (*Dikey Kalkış İniş Yapma*) hava araçları için farklı tasarımda çerçeve (*frame*) üreten firmalar vardır. Şekil 2.3'te birkaç firma ve ürünleri gösterilmiştir. Bu çalışmada kendi tasarımımız olan 72 cm genişlikte bir çerçeve geliştirilmiştir. Çerçeve, 6 adet 1 mm kalınlıkta ve 30 cm uzunlukta alüminyum kare profilden oluşmaktadır. Kare profiller merkezde 32 cm genişlikte ve 3 mm kalınlıkta plexiglass ile birleştirilmiştir. Kullanılan malzemelerin hafifliği ile çerçevenin ham ağırlığı 350 gr olarak hesaplanmıştır. Motorların yüksek hızlarda dönerken oluşturdukları titreşimler alüminyum profiller ile tüm sisteme dağıtılır. Bu durum, uçuş güvenliği açısından sorun teşkil etmektedir. Merkezde kare profilleri birleştirmek için kullanılan plexiglass, tüm titreşimler emerek daha stabil uçuş sağlamıştır. Ayrıca titreşimlerin emilmesi sonucu merkeze konumlandırılan sensörlerden gelen verilerde oluşabilecek dalgalanmalar engellenmiştir (Kırçalı, 2013).



**Şekil 2.3.** Özel firma çerçeveleri.

(a) Tarot T960 (b) RotorBits HexCopter (c) Turnigy Talon (d) Hobbyking Super H600

Çalışmada tasarlanan ve Şekil 2.4’de gösterilen çerçevede tüm ekipmanlar X yapılandırma uçuş ilkelerine göre konumlandırılmıştır. Dikey kalkış (*throttle*), rotorların toplam kuvvetinden elde edilir. İleriye doğru hareket edebilmek (*Pitch* açısında değişim) için, sağ ve sol rotorlar sabit hızlarda iken, ön rotorlar yavaşlar, iki arka rotor hızlanır. Yanlara hareket etmek (*Roll* açısında değişim) için rotorlar istenen yönde yavaşlar ve karşı taraftakiler bir miktar hızlanır. Eğer hızlanma miktarı çok ani olursa, İHA’nın takla atma ihtimali oluşur. Bu gibi olumsuzluklarla karşılaşmamak için kontrol algoritmaları, pilotlardan gelen sağ/sol manevra sinyalleri belirli bir oranda hava aracına yansıtırlar. İHA’lar uçuş sırasında motorların dönüş yönünde oluşabilecek kuvvetleri engellemek için motorların dönüş yönleri sırasıyla saat yönünde (*Clock Wise*) ve saat yönünün tersi (*Counter Clock Wise*) olacak şekilde konumlandırılmıştır. Bu şekilde 3 motor saat yönünde diğer 3 motor ise saat yönünün tersi şeklinde dönmektedir. Eğer İHA, bulunduğu konumda yönünü değiştirmek (*Yaw* açısında değişim) istenirse, dönmek istediği yöne göre saat yönünde dönen 3’lü rotor grubu ile saat yönü tersinde dönen 3’lü rotor grubu arasında hız farklılıkları oluşturulur. Daha hızlı dönen grubun yönünde hareketlenme gerçekleşir (Gomes, 2013).



Şekil 2.4. Hexcopter çalışma prensibi.

### 2.1.2. Fırçasız DC Motor ve Pervane

İHA’nın uçması için gerekli olan kaldırma kuvveti 6 adet motor ile uygun pervane kombinasyonundan gelmektedir. Tasarlanan sistemde motor tipi olarak fırçasız DC motor tercih edilmiştir. İHA projelerinde genel olarak kullanılan fırçasız motorların en önemli avantajları; yüksek verim, fırça ve kolektör olmadığından dolayı kolay bakım, sessiz çalışma ve masrafsız olması sayılabilir (Aydoğdu, 2006). Çalışmada kullanılan motorlar Şekil 2.5’de gösterilen Turnigy firmasına ait D2830/11 modelidir. Çizelge 3.1’de üretici

firmanın yayınlamış olduğu batarya gücüne göre uygun pervane tipi ve kaldırma kuvveti bilgileri verilmiştir (URL5). Sistemde 3S batarya ile 9x6 boyutlarında pervaneler kullanılmıştır. Toplamda maksimum güçte 5340 gr'lık kaldırma kuvveti oluşmaktadır.



**Şekil 2.5.** Kaldırma kuvveti ekipmanları.  
(a) Fırçasız DC Motor (b) 9x6 Karbon Fiber Pervane

**Çizelge 3.1.** Motor parametreleri.

| Batarya   | KV (rpm/v) | Pervane | Kaldırma Kuvveti (gr) |
|-----------|------------|---------|-----------------------|
| 7.4V (2S) | 900        | 10 x 7  | 890                   |
| 11.1 (3S) | 1000       | 9 x 6   | 890                   |
| 14.8 (4S) | 1100       | 8 x 4   | 890                   |

### 2.1.3. Elektronik Hız Kontrolcüsü

Fırçasız motor, kabaca üç fazlı bir motordur ve bu nedenle fazları arasında doğru bir senkronizasyonla çalışmalıdır. Başka bir deyişle, fırçasız motorlarda, sadece akımı değiştirmek yeterli değildir. Hızı kontrol etmenin doğru yolu, akım değişiminin frekansını fazları arasında değiştirmektir. Bu nedenle, bu motor tipi, frekansını değiştirebilen ve motorun hızını kontrol edebilen bir sürücü gerektirir. Genellikle ESC (Electronic Speed Control) olarak adlandırılan hazır elektronik devreler, PWM (Pulse Width Modulation) sinyallerini giriş olarak kabul ederek uygun frekans değerlerinde çıkış verirler. Tasarlanan sistemde Şekil 2.6'da gösterilen 30 Amper ve 3S destekli ESC'ler kullanılmıştır (URL6).



Şekil 2.6. Elektronik hız kontrolcüsü.

#### 2.1.4. Pil

Pil seçimi, döner kanatlı İHA kurulumunun önemli yönlerinden birisidir. Lityum Polimer (LiPo) piller, bilinen tüm piller arasında en iyi enerji yoğunluğuna sahip olanlar oldukları için döner kanatlı İHA'larda yaygın olarak kullanılırlar (Bappy, 2007). Çalışmada Şekil 2.7'de gösterilen ortalama 12 Volt besleme gerilimli 3 hücreli bir LiPo pil tercih edilmiştir.



Şekil 2.7. LiPo pil (12V 3S 5000mAh).

#### 2.1.5. Şarj Cihazı

Sistem, kullanılan LiPo ile ortalama 20 dakikalık uçuş süresine sahiptir. Bu kapsamda LiPo'nun sürekli şarj edilmesi gerekir. Ancak LiPo pillerin şarj işlemleri dikkatli bir şekilde yapılmalıdır. Pil içerisindeki her hücrenin 3V'nin altına deşarj edilmesi bataryalara zarar verebilir ve bu seviyenin altında yüksek voltaj düşmesi oluşabilir. Hücrelerin dengesiz şarj edilmesi ya da toplam gerilimin 12.6 V üzerine çıkması durumunda da LiPo zarar görecektir. Bu gibi sorunlar ile karşılaşmamak adına kontrollü şarj işlemleri için Şekil 2.8'de gösterilen LiPo Balanslı Şarj (LiPo Balance Charger) devresi kullanılır. Diğer yandan yüksek irtifalarda uçuş sırasında, besleme gerilimindeki ani değişikliklerde motorlar doğrudan ya da ESC'lerin yumuşak geçiş (Soft CutOff) özelliği aktif ise yavaşlayarak duracaktır. Bu gibi durumlarda da İHA ciddi zararlar görecektir. Sistemde, pilotları LiPo'nun olası olumsuz durumunda yüksek sesle

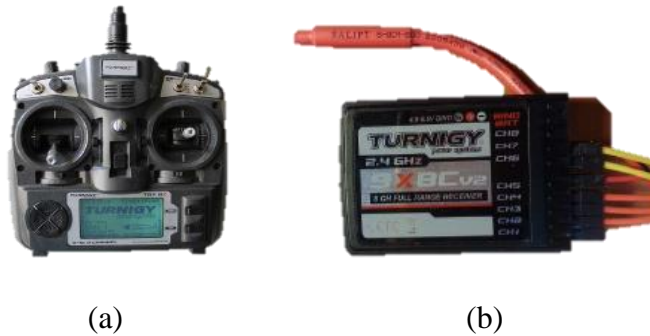
uyaran ve Şekil 2.8’de gösterilen LiPo Voltaj Alarmı (LiPo Voltage Alarm) kullanılmıştır (Kırcalı, 2013).



**Şekil 2.8.** LiPo şarj ve kontrol ekipmanları.  
(a) Turnigy accucel 6 balance charger (b) LiPo low voltage alarm

### 2.1.6. Uzaktan Kumanda ve Alıcısı

İHA’yı uzaktan kontrol etmek ve otonom modda güvenlik önlemlerini aktifleştirmek için uzaktan kumanda kullanılmaktadır. Kumandada dikey hareket, bulunduğu ekseninde dönme, ileri geri, sağa ve sola olmak üzere minimum 4 kanala ihtiyaç vardır. Çalışmada Şekil 2.9’da gösterilen 8 kanallı Turnigy kumanda ve alıcısı kullanılmıştır (Bappy, 2007).

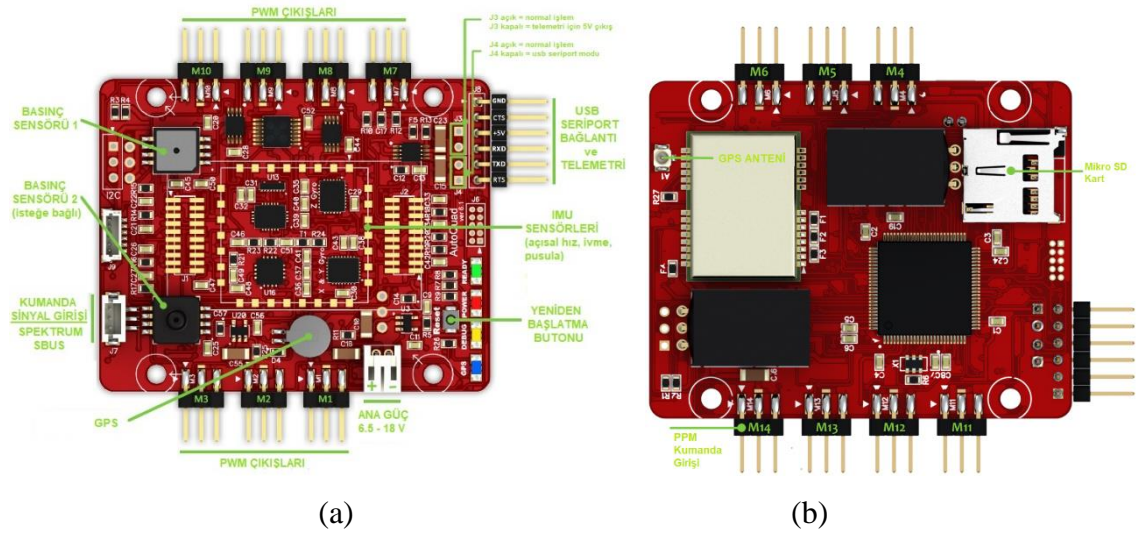


**Şekil 2.9.** Uzaktan kontrol ekipmanları.  
(a) 9 Kanallı uzaktan kumanda (b) Kumanda alıcısı

### 2.1.7. Ana Kontrol Kartı

Ana kontrol kartları İHA’lar için kuşkusuz en önemli kısımdır. Tüm sistemin beyni olarak kabul edilebilir. Tasarlanan sistemde Şekil 3.11’de gösterilen açık kaynak kodlu uçuş kartlarından *AutoQuad* kartı kullanılmıştır. AutoQuad, motorlar, servolar,

LED'ler, ses sinyalleri ve diğer kullanımlar için konfigüre edilebilen 14 giriş çıkış kanallına sahip geniş kontrol seçeneği sunabilen bir karttır. Temel olarak uçuş sırasında ivme, açısal hız, pusula ve barometre sensörlerinden gelen verileri analiz ederek STM32F407 işlemcisine iletir. Gelen veriler ile pilottan gelen sinyaller birleştirilerek ESC'lere iletilir ve motorların çalışması sağlanır. AutoQuad tüm sensör verilerini atalet ölçüm biriminden (*IMU- Inertial Measurement Unit*) elde eder. Kart üzerinde dahili olarak bulunan IMU, analog veri üretmektedir. Gelen verilerin çözünürlük değerleri kısıtlı olduğu için harici olarak üzerinde açısal hız, ivme, pusula ve barometre olmak üzere dört sensörden oluşan D-IMU (*Dijital IMU*) eklenmiştir. D-IMU montaj yapılışı AutoQuad üzerinde Şekil 2.10'da gösterilmiştir. Barometre sensörü dışındaki diğer sensörler x, y ve z olmak üzere her eksen için ayrı ayrı veri üretir. Toplamda IMU üzerinden 10 farklı veri elde edilir. Bu sebeple 10 DOF (*Degrees Of Freedom*) IMU olarak adlandırılır (URL7).



Şekil 2.10. Çalışmada kullanılan açık kaynak kodlu autoquad uçuş kartı (URL8)  
(a) Ön yüz (b) Arka yüz

AutoQuad uçuş kontrol kartının donanımsal özellikleri (URL8):

- 5,0 x 6,5 cm ölçülerinden standart kontrol boyutları ile 4,5 cm genişlikte montaj deliklerine sahiptir
- 6.5v ile 18v arasında giriş voltajı destekler
- STM32F407 32bit Cortex M4 işlemcisine sahiptir

- Diğer devre elemanları ve sensörler için yüksek verimli dahili 2 DC/DC dönüştürücüsüne sahiptir

- Standart bağlantı konektörlerine sahiptir
- 14 ayrı genel amaçlı PWM giriş çıkış pinlerine sahiptir
- 2.4Ghz frekanslı uzaktan kumanda sinyallerini algılayabilir
- 32 GB 'a kadar hafıza kartı destekler
- Dahili uBlox LEA-6T GPS modülüne sahiptir
- 6 pin FTDI bağlantısını destekler
- I2C ve SPI veri yolunu destekler
- Batarya voltaj takibi

Dahili IMU donanımsal özellikleri (URL8):

- İki eksenli dahili analog pusula sensörü (X, Y: HMC6042)
- Tek eksenli dahili analog pusula sensörü (Z: HMC1041Z)
- İki eksenli dahili analog açısal hız sensörü (X, Y: IDG500)
- Tek eksenli dahili analog açısal hız sensörü (Z: ISZ500)
- Üç eksenli dahili analog ivme sensörü (Z: ADXL325)



Şekil 2.11. Autoquad ve D-IMU (URL8).

Dijital IMU donanımsal özellikleri: (URL8)

- Yüksek çözünürlüklü veriler elde eder
- Dijital basınç sensörü (MP3H6115A)
- İkinci seri bağlantı konektörü

### 2.1.8. Raspberry Pi 3 Model B

Raspberry Pi gömülü sistem platformu İngiltere’de Raspberry Pi vakfı tarafından 2012 yılından itibaren geliştirilmeye başlanmıştır. Asıl amacı okullarda bilgisayar temelli eğitimi geliştirmek ve çocuklara bilgisayarı öğretmek olan mini bir bilgisayardır. Raspberry Pi kredi kartı büyüklüğünde olan içinde çeşitli Linux ve Android dağıtımları kurulabilen ARM mimarisine sahip bir mini bilgisayardır. Akıllı ev otomasyonu, kablosuz sensör ağı, taşınabilir e-sağlık izleme sistemi, yol işareti tanıma, SQL enjeksiyonunun önlenmesi ve tespiti gibi farklı sorunlara pratik çözümler sunmuştur.

Raspberry Pi 3 Model B, Raspberry Pi Vakfı'nın yeni bir ürünüdür. Şubat 2016'da ortaya çıkmıştır. Günümüzde ufak değişiklik ile versiyonu yükseltilerek B+ modeli yayınlanmıştır. Model 2’den model 3’e geçişte en önemli değişiklik işlemci mimarisinin 64 bite çıkartılmasıdır. Raspberry Pi modelleri karşılaştırılması Çizelge 3.2’de verilmiştir (Niemelä, 2016).

**Çizelge 3.2.** Raspberry Pi modelleri karşılaştırması.

|                                 | <b>Raspberry Pi</b>    | <b>Raspberry Pi 2</b> | <b>Raspberry Pi 3</b>     |
|---------------------------------|------------------------|-----------------------|---------------------------|
| <b>Yayınlanma Tarihi</b>        | Şubat 2012             | Şubat 2015            | Şubat 2016                |
| <b>İşlemci (CPU)</b>            | ARM1176JZF-S           | ARM Cortex-A7         | ARM Cortex-A53            |
| <b>İşlemci Hızı (CPU Speed)</b> | 700 MHz<br>single core | 900 MHz<br>quad core  | 1200 MHz<br>quad core     |
| <b>Geçici Bellek (RAM)</b>      | 512 MB                 | 1 GB                  | 1 GB                      |
| <b>Grafik Kartı (GPU)</b>       | Broadcom Videocore IV  | Broadcom Videocore IV | Broadcom Videocore IV     |
| <b>Depolama</b>                 | SDHC slot              | MicroSDHC slot        | MicroSDHC slot            |
| <b>USB Port</b>                 | 2                      | 4                     | 4                         |
| <b>Kablosuz Bağlantı</b>        | -                      | -                     | 802.11n and Bluetooth 4.1 |

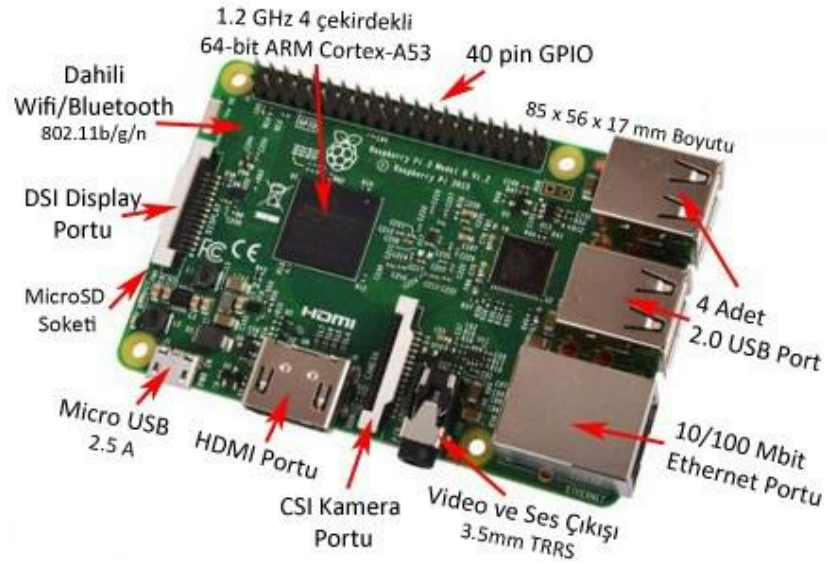
Çalışmada en son model olan Pi 3 Model B kullanılmıştır. Model B, 40 adet genel amaçlı giriş çıkış (GPIO) pinine sahiptir. Bağlantı pinleri GPIO, I2C, SPI, UART kısımlarından oluşmaktadır. Çevre birimlerinin bağlanabildiği USB çıkışları ve ethernet çıkışı bulunmaktadır (Smolov, 2017). Şekil 2.12’de Raspberry Pi 3 Model B gösterilmektedir.



Şekil 2.12. Raspberry Pi 3 Model B (Smolov, 2017).

Raspberry Pi 3 özellikleri ve çevre birimi çıkışlarının detayları aşağıda verilmiştir. Bu özelliklerin kart üzerinde dağılımı Şekil 2.13’de işaretler ile gösterilmiştir (URL16).

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1.2 GHz 4 çekirdekli 64-bit ARM Cortex-A53 işlemci
- 1 GB LPDDR2 bellek
- Dahili 802.11b/g/n destekli WiFi
- Dahili düşük enerji destekli Bluetooth
- 10/100 Mbit destekli Ethernet portu
- HDMI portu (HDMI 1.4 destekli)
- Kompozit video ve ses çıkışı için 3.5mm TRRS (4 uçlu) konektör
- 4 adet USB2.0 portu
- 40 adet GPIO pin
- CSI (kamera) ve DSI (ekran) konektörleri
- Mikro SD kart yuvası
- 85mm x 56mm x 21mm ölçülerinde



Şekil 2.13. Raspberry Pi 3 ve çevre birimleri (URL17).

Çalışma kapsamında işletim sistemi olarak Linux dağıtımlarından Debian tabanlı Rasbian sürümü tercih edilmiştir (URL15). Rasbian üzerinde geliştirilen grafik (GUI: Graphical User Interface) tabanlı uygulama ile görüntü işleme sonuçları AutoQuad kontrol kartına seri port üzerinden iletilmektedir.

### 2.1.9. Kamera

Çalışmada yüksek çözünürlüklü ve geniş açılı SJCAM firmasına ait Şekil 2.14'de gösterilen SJ5000x modeli kullanılmıştır. Raspberry Pi 3 kontrol kartına USB (*Evrensel Veri Yolu*) ile bağlıdır. 170 derecelik geniş açığa ve 4K görüntüler üretir. Ancak en önemli özelliği merceğinde dahili bulunan GYRO sensörü ile uçuş sırasında oluşan titreşimlerin engellenmesi sağlanmıştır.



Şekil 2.14. SJ5000x geniş açılı kamera.

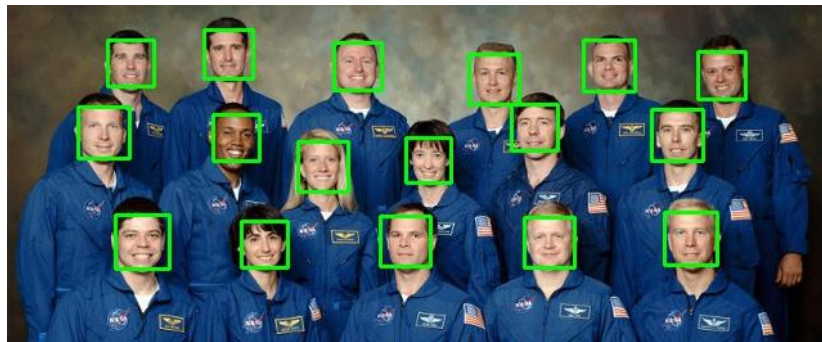
## 2.2. Görüntü işleme yazılımı

Intel firması tarafından 1999 yılında geliştirilmiş ve BSD lisansı ile lisanslanmış olan OpenCV, açık kaynaklı “Bilgisayarla Görme” kütüphanesidir. Resim veya video içerisinde bulunan anlamlı bilgilerin çıkarılıp işlenmesinde kullanılmaktadır. OpenCV, platformlardan bağımsız olarak geliştirilen bir kütüphane olduğu için Windows, Linux, MacOS, IOS, Android gibi ortamlarda çalışabilmektedir. C, C#, C++, Python, Java, Perl, Ruby gibi çok farklı programlama dilleri ile OpenCV uygulamaları geliştirilebilmektedir (URL9).

OpenCV kütüphanesi içerisinde görüntü işleme ve makine öğrenmesi için kullanılmak üzere pek çok algoritma bulunmaktadır. Bu algoritmalar ile Şekil 2.15’de gösterilen plaka tanıma, yüz tanıma, nesneleri sınıflandırma, nesneleri ayırt etme, görüntü karşılaştırma, insan hareketlerinin tespit edilmesi gibi pek çok işlem kolay bir şekilde yapılabilmektedir.



(a)



(b)



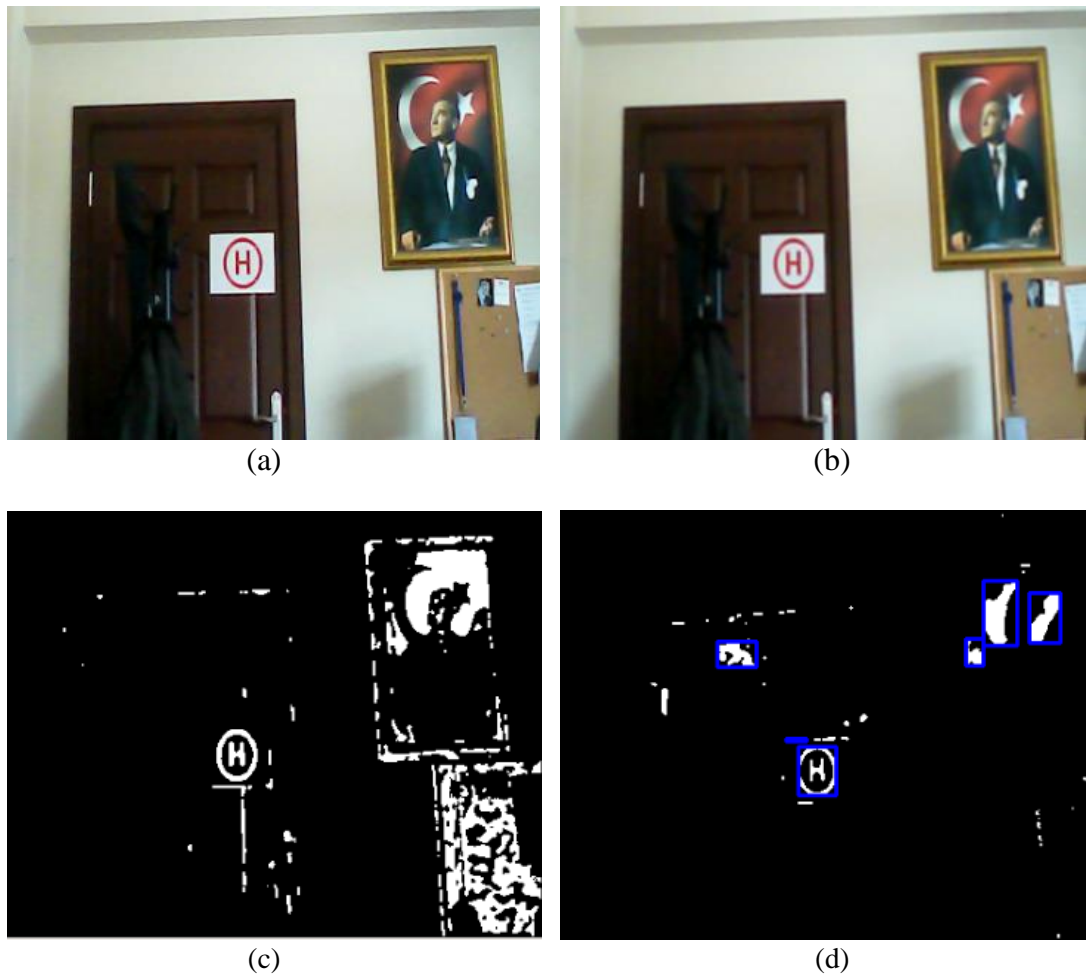
(c)

**Şekil 2.15.** Görüntü işleme ile çözülen yaygın problemler. (a) Plaka tanıma (b) Yüz tanıma (c) Görüntü karşılaştırma

Bu çalışmada kısaca kameradan alınan görüntüler önce bulanıklık metotlarına tabi tutulur. Görüntü, üzerindeki pürüzlerden kurtulduktan sonra çözünürlük değeri daha geniş olan renk uzayına geçiş yapılır. Renk filtresi uygulanarak istenmeyen renkler filtrelenir. Elde edilen görüntü üzerinde şekiller analiz edilerek hedef (iniş pistindeki H sembolü gibi) tespit edilir. Çalışmada tüm OpenCv işlemleri Raspberry Pi 3 mikro denetleyicisi üzerinde gerçekleştirilmiştir.

### 2.2.1. İniş düzleminin algılanması

Sistem, temel olarak kameradan alınan görüntülerin OpenCV kütüphanesi ve görüntü işleme filtreleri yardımıyla iniş pistinin konumunun belirlenmesine dayanmaktadır. Şekil 2.16’da verilen görsel, iniş pistinin nasıl tespit edildiğini, uygulanan filtreleri ve filtrelerin sonuçlarını göstermektedir.



Şekil 2.16. İniş düzlemi algılama adımları.

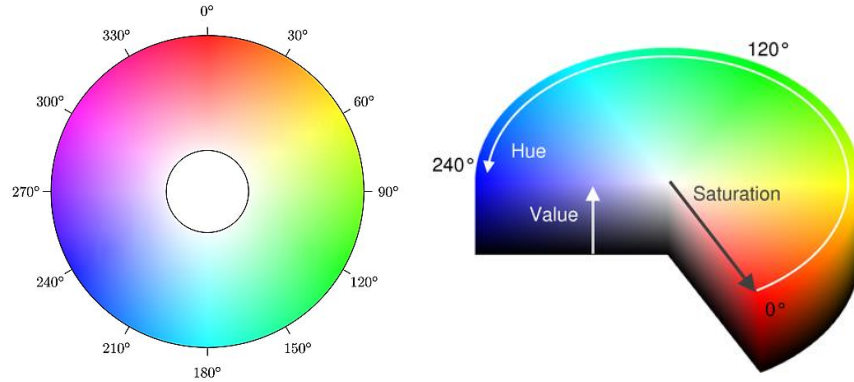
(a) Orijinal Resim (b) Bulanıklık Filtresi (c) Renk Tanımlama (d) Nesne Tanımlama

Kameradan alınan görüntüler mikro denetleyicinin takılması durumunda yaşanabilecek hatalardan dolayı döngü içerisinde değil de zamanlayıcı ile çağrılarak metot içerisinde işlemlere tabi tutulmuştur. Zamanlayıcı 200 milisaniye olacak şekilde ayarlanmıştır. Bu frekans ayarlarında saniyede beş resim işlenebilmektedir (Çiçekdemir, 2018). Kameradan gelen görüntüye yapılan işlemleri sırasıyla anlatmak gerekirse:

1. Görüntüler 640x480 boyutunda alınmaktadır. Görüntü, üzerinde daha hızlı işlem yapabilmek için 320x240 boyutuna dönüştürülür. Bu işlem için OpenCV kütüphanesinin *resize* metodundan yararlanılır (URL17).

2. Şekil 2.16 (b)'de *Average Blur* filtreli resim verilmiştir. Bu filtre, görüntü üzerindeki gürültüleri temizlemek için kullanılmıştır. Average Blur filtresi, görüntü üzerinde dolaşan 3x3 boyutlu bir matris ile tüm piksellerin ortalamasını alarak ortadaki piksel değerine eşitler. Bu yöntemle pikseller arası geçişler yumuşatılır (URL10).

3. Kameradan alınan görüntüler standart olarak RGB (Red Green Yellow) formatındadır. Her piksel değeri, 1x3 boyutlu matris yapısındadır. OpenCV kütüphanesinde RGB dışında HSV (Hue Saturation Value), YCbCr (Brightness, Blue, Red), LAB (Lightness, Red-Green, Blue-Yellow) gibi başka renk uzayları da bulunmaktadır (URL11). HSV renk uzayı, ana renklerin tespit edilmesinde RGB renk uzayına göre daha başarılıdır (Nursabillilah, 2013). Bu sebeple gelen görüntülerin renk uzayı HSV formatına dönüştürülmüştür. HSV renk uzayında da aynı RGB renk uzayında olduğu gibi her piksel değeri 1x3 boyutlu matris ile temsil edilir. Matrisin birinci elemanı H ana renk değerini, ikinci elemanı S doygunluk değerini ve üçüncü elemanı V parlaklık değerini temsil eder. Şekil 2.17'de gösterildiği gibi HSV renk uzayında H değeri 0 ile 360 arasında, S ve V değerleri 0 ile 255 arasındadır. Ancak OpenCV kütüphanesinde H değeri 8 bit'lik alanı aştığı için 0 ile 180 arasında ölçeklendirilir. S ve V değerleri yine 0 ile 255 arasındadır. Kırmızı renk ise H değeri için 0 ile 10 ve 170 ile 180 arasında tanımlanır.



Şekil 2.17. HSV renk uzayı.

4. Renk uzayı değiştirilen görüntü üzerindeki tüm piksellerin değerleri, kırmızı renk aralığında olup olmadıklarına göre kontrol edilir. Kırmızı renk aralığı dışında kalan pikseller beyaz renge, kırmızı renk aralığı içinde kalan pikseller ise siyah renge dönüştürülür. Bu işlem sonucunda her piksel değeri tek elemanlı matris ile temsil edilir. Bu sayede hafızadan kazanç sağlanır. Şekil 2.16 (c)'de renk filtresine tabi tutulan resim verilmiştir. OpenCV kütüphanesinde renk aralığına göre filtreleme işlemi *inRange* metodu ile yapılır (URL12).

5. Siyah beyaz formata dönüştürülen görüntüde birbirine bağlı pikseller analiz edilir. Her piksel, komşusu ile aynı değerde olup olmamasına göre numaralandırılır. Bu sayede görüntü üzerinden kaç adet şekil olduğu tespit edilir. Şekillerin büyüklükleri birbirlerinden farklı olabilir. Şekil 2.16 (d)'de görüntü üzerinden tespit edilen şekiller gösterilmiştir. OpenCV kütüphanesinde siyah beyaz resim üzerinden şekilleri tespit etmek için *findContours* metodu kullanılır (URL13). Bu metot 1985 yılında Satoshi Suzuki ve arkadaşları tarafından ikili resimler üzerinde geliştirilen algoritmaya dayalı çalışır (Suzuki,1985).

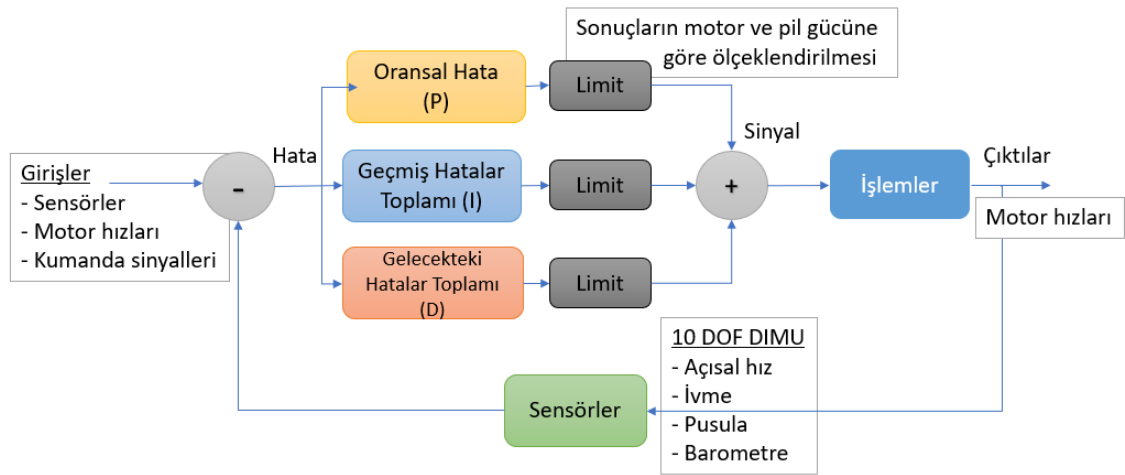
6. Tespit edilen şekiller boyutlarına göre büyükten küçüğe olacak şekilde bir dizide saklanır. Döngü içerisinde dizinin her elemanı, sistemde saklı tutulan iniş pisti görüntüsü ile karşılaştırılır. Karşılaştırma işlemi her pikselin aynı olmasına dayanmaktadır. Benzerlik oranı %75 ve üzerine ulaştığında iniş pisti tespit edilmiş olur.

7. Tespit edilen iniş pistinin görüntü üzerindeki koordinatları belirlenir. Koordinatlara göre İHA'nın hangi yönete gitmesi gerektiği hesaplanır ve seriport üzerinden AutoQuad kontrol kartına yeni motor hızları iletilir.

8. Yukarıda ayrıntıları verilen 7 adım, saniyede iki kere tekrar edilerek İHA'nın sürekli iniş pistinin ortasında kalması sağlanır. AutoQuad tarafından yükseklik, 25 cm veya altına inene kadar kademeli olarak azaltılır. Sonrasında ESC'nin yumuşak durma (Soft CutOff) özelliği ile motorlar durdurulur.

### 2.3. Uçuş yazılımı

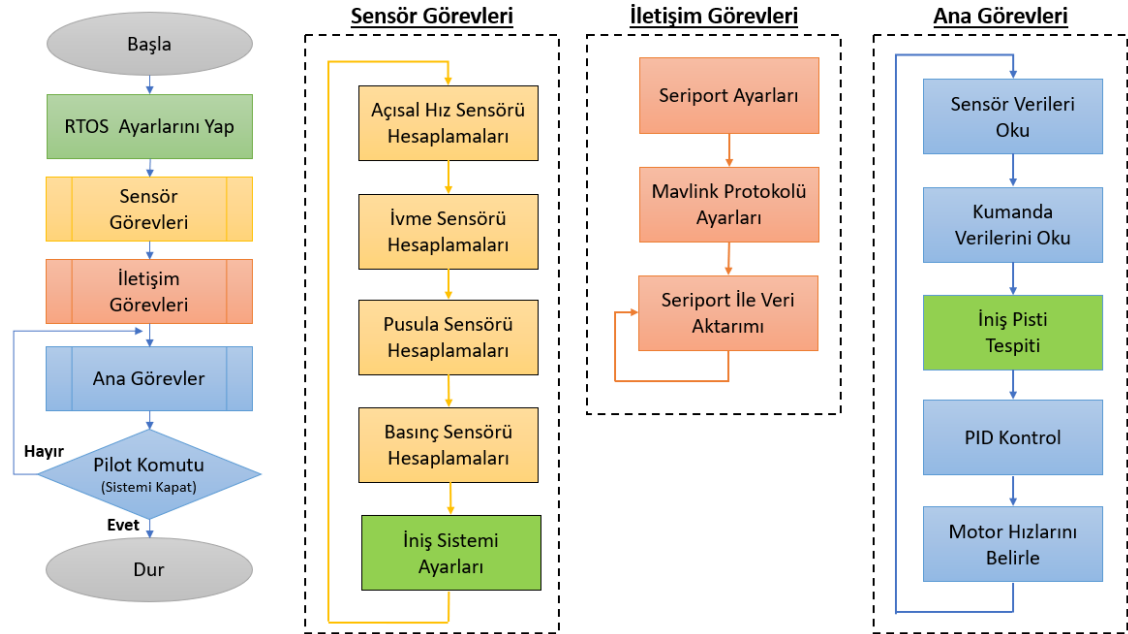
Açık kaynaklı bir proje olan AutoQuad, farklı rotor seçeneğine sahip çerçeveleri desteklemektedir. Mevcut yazılımın istikrarlı, akrobatik, irtifada tutma, eve dönüş dahil olmak üzere birçok uçuş modları vardır. Yazılım sürekli D-IMU sonuçlarını okur ve düzeltmeler yapar. Kontrol mekanizmasında, istenen açısal rotasyon oranı, iç PID (Oransal, İntegral, Türevsel denetleyici) döngüsündeki ham açısal hız çıktısı ile karşılaştırılır. Oluşan fark, Şekil 2.18'de gösterilen PID denetleyicisine geri besleme olarak iletilir ve motor hızlarının kontrolü yapılır. PID döngüsü temelde kullanıcı yani pilot girişi ve otonom uçuş verisi girişi olmak üzere iki parametresi vardır.



Şekil 2.18. PID blok diyagramı.

İHA otonom uçuş yapısının akış diyagramı Şekil 2.19'da verilmiştir. Tüm sistem RTOS (Real Time Operating System) üzerinde çalışmaktadır (Chandane, 2016). AutoQuad üzerinde günümüzde açık kaynak kodlu olarak sunulan gerçek zamanlı işletim sistemlerinden CoOS kullanılmıştır (URL14). CoOS sayesinde aynı anda birden fazla görev başlatılarak sistemin daha etkin çalışması sağlanmıştır. İlk olarak tüm sensörler aynı görev altında iş parçacıkları olarak ayarlanmıştır. Ana görev altında sürekli sensör

verileri ve pilottan gelen kumanda sinyalleri PID kontrol biriminden geçirilerek motor hızları belirlenmiştir. Eğer sisteme bir iletişim cihazı (bilgisayar, telefon) bağlı ise seri port konfigürasyonları da ayrı bir görev altında başlatılarak veri aktarımı sağlanmıştır.

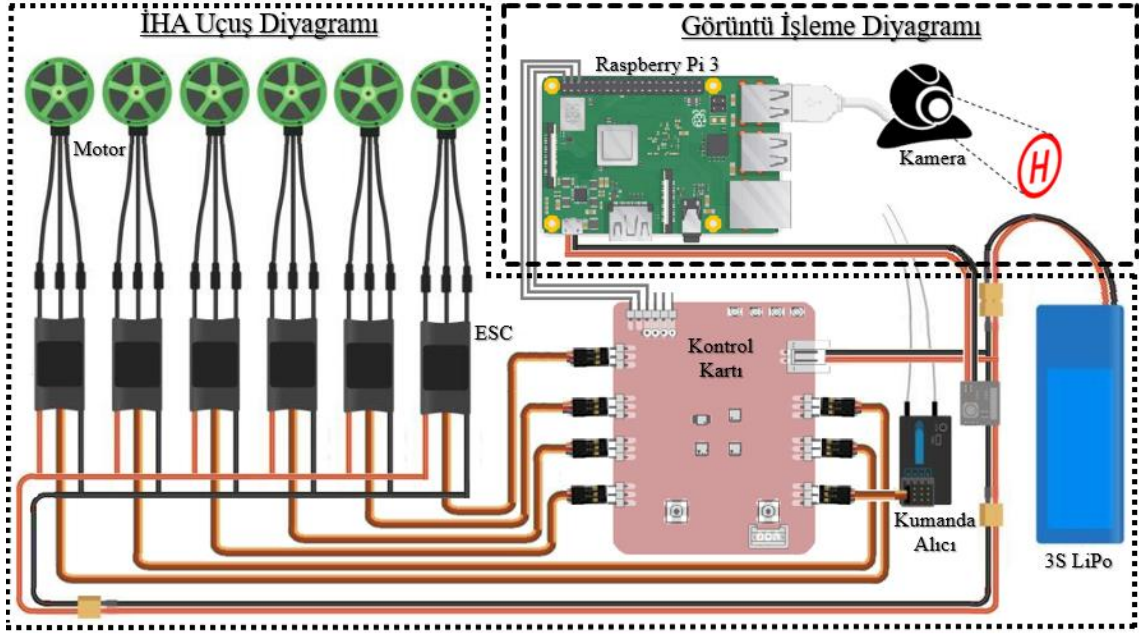


Şekil 2.19. Autoquad akış diyagramı.

## 2.4. Deneysel Sonuçlar

İnsansız hava araçlarının otonom uçuş özelliklerine ek olarak otonom iniş sistemi geliştirilmiştir. Sistemin bağlantı şeması Şekil 2.20’de gösterilmiştir. Sistemde kullanılan donanımsal ürünler genel olarak iki kategoriye ayrılmıştır. İlk kısımda otonom uçuş adımları, diğer kısımda ise görüntü işleme adımları gerçekleştirilmiştir.

Otonom uçuş için AutoQuad AQ6 model kontrol kartı kullanılmıştır. Kontrol kartına 6 adet hız kontrolcüsü ile 6 adet motor bağlıdır. Pilot tarafından 2.4 Ghz iletişim protokolüne sahip kumanda ile PPM formatında üretilen sinyaller kontrol kartına gönderilmektedir. Gerek anlık sensör verileri gerek motor hızları gerekse kumanda sinyalleri PID kontrol birimine giriş olarak kabul edilir. Veriler, kontrol filtrelerinden geçirilerek motorların yeni hızları belirlenir. Tüm adımlar sürekli tekrarlanarak daha kararlı bir uçuş gerçekleştirilir.

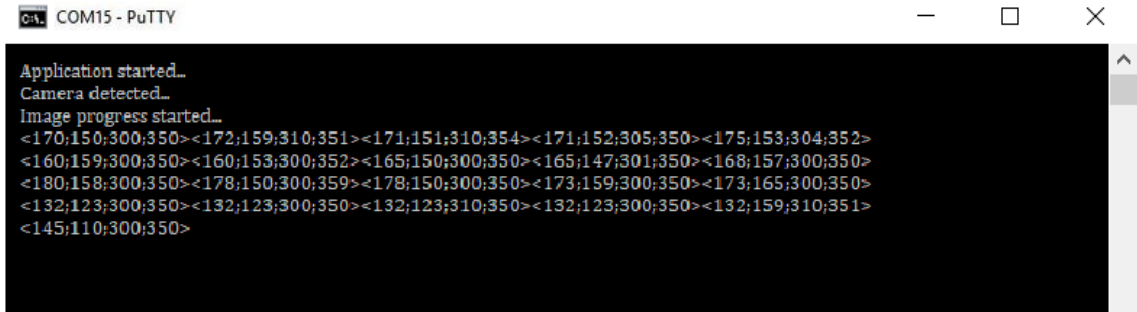


Şekil 2.20. Kullanılan sistemin bağlantı şeması.

Şekil 2.20'de gösterilen sistem şemasında görüntü işleme adımları sağ tarafta bulunan Raspberry kısmında gerçekleştirilmiştir. Raspberry Pi üzerinde linux dağıtımlarından debian tabanlı rasbian işletim sistemi çalışmaktadır. Görüntü işleme adımları uzun sürdüğü için geliştirilen uygulamada grafik ara yüzünden ziyade konsol yapısı kullanılmıştır. Rasbian işletim sistemi devreye girdikten sonra geliştirilen konsol uygulaması otomatik olarak çalışmaya başlar.

Kullanılan kamerada standart formatta yüksek çözünürlükte görüntü verdiği için ilk olarak görüntü üzerinde yeniden boyutlandırma yöntemi kullanılmıştır. Yeni görüntüler 640x480 boyutuna dönüştürülür. Güneşten veya ortanda bulunan ışık yoğunluklarından dolayı görüntü üzerinden pürüzler oluşur. Bu problem, pürüzlerin olduğu noktaları bulanıklaştırma yönetimi ile yumuşatılarak çözülür. Elde edilen görüntü, çözünürlüğü daha fazla olan HSV renk uzayına dönüştürülür. Yeni renk uzayına göre görüntüye renk filtresi uygulanır. Bu işlemlerinde sonunda tek boyutlu siyah beyaz renk elde edilir. Görüntü üzerinde belirlenen renkler beyaz ile diğer tüm renkler ise siyah ile temsil edilir. Beyaz alanların her birisi bir şekil olarak kabul edilerek görüntü üzerinden kaç adet şekil olduğu belirlenir. Her şekil kendi için boyutlarına göre sıralanır. Şekiller bir döngü içerisinde hedef görüntü ile karşılaştırılarak benzerlik oranı tespit edilir. Benzerlik oranının yüzde 75 'ten fazla olması durumunda şeklin merkez noktasının X ve

Y noktaları ile tahmini uzaklık değeri seriport üzerinden AutoQuad kontrol kartına gönderilir. Gönderilen verilerin konsol uygulaması üzerindeki örnek görüntüsü Şekil 2.21’de verilmiştir.



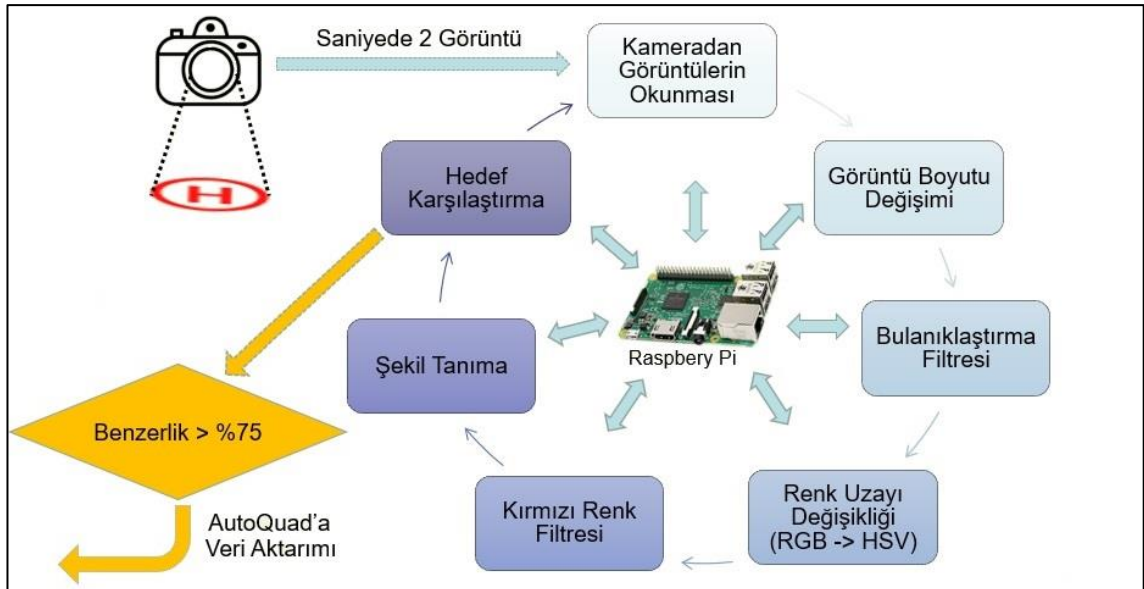
```

COM15 - PuTTY
Application started..
Camera detected..
Image progress started..
<170;150;300;350><172;159;310;351><171;151;310;354><171;152;305;350><175;153;304;352>
<160;159;300;350><160;153;300;352><165;150;300;350><165;147;301;350><168;157;300;350>
<180;158;300;350><178;150;300;359><178;150;300;350><173;159;300;350><173;165;300;350>
<132;123;300;350><132;123;300;350><132;123;310;350><132;123;300;350><132;159;310;351>
<145;110;300;350>

```

Şekil 2.21. Rasbian üzerinden veri aktarımı.

Görüntü işleme adımlarının tamamı 500 milisaniyelik periyotta ayarlanmıştır. Bu sayede saniyede iki görüntü işlenmektedir. İniş düzleminin tespiti için görüntü işleme tarafında geliştirilen uygulamanın görsel şeması Şekil 2.22’de verilmiştir.



Şekil 2.22. Raspberry üzerinde çalışan uygulama adımları.

Raspberry Pi mikro denetleyicisi üzerinde çalışan uygulamanın kaba kodu Şekil 2.23’te gösterilmiştir.

```

1. begin
2. set timer to 500 ms
3.   while timerTriger()
4.     l_image ← getFrame()
5.     s_image ← resize(l_image)
6.     s_image ← blur(s_image)
7.     s_image ← rgb2hsv(s_image)
8.     black_image ← inRange(s_image,RED)
9.     object() ← findContours(black_image)
10.    index=0
11.    while sizeof(object)
12.      rate ← checkSimilarity(object(index), target)
13.      if rate >= 75
14.        x,y ← getCenter()
15.        sendSerioPort(x,y)
16.      end
17.      index++
18.    end
19.    clear object(), rate, x, y
20.    clear black_image, s_image, l_image
21.  end
22. end

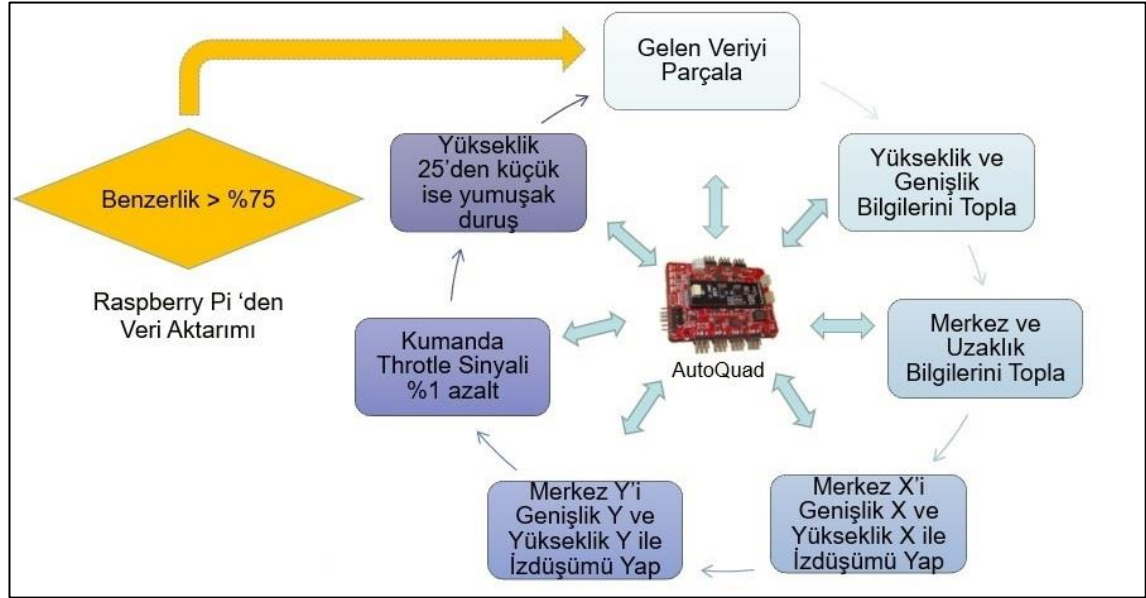
```

**Şekil 2.23.** Raspberry üzerinde çalışan uygulamanın kaba kodu.

Açık kaynak kodlu AutoQuad'ın Şekil 2.19'da gösterilen akış diyagramında sensör ve ana görevler altına yeşil renk ile belirtilen kısımlarda iniş işlemi için geliştirmeler yapılmıştır.

Algoritmada yapılan değişikliklerde ilk olarak iniş adımları için yeni bir görev başlatılır. Ana kontrol görevine paralel olarak devreye alınan iniş görevi, pilotun sistemi iniş moduna geçirmesini bekler. Pilot tarafından 8 kanallı kumandanın bir kanalına atanmış 3 seviyeli buton ile İHA havada askıda kalma (Altitude Hold) moduna alınır. Bu modda geçen İHA sabit hızda sabit irtifada uçuşuna devam eder. Aynı zamanda seri port ile Raspberry Pi iletişimi başlatılır. Kameradan gelen görüntüler Raspberry Pi tarafından işlenerek hedefin hangi yönde ve ne kadar yükseklikte olduğu tespit edilir. Tespit edilen veriler seriport ile AutoQuad'a gönderilir. Gelen verilerde ilk olarak çözümleme yapılır. Eğer veriler istenilen formatta değilse iniş işlemi başlatılmaz. Verilerin doğruluğu kesinleştikten sonra sistem iniş moduna geçer. İniş anında gerek verilerde bozulma gerekse dış etmenlerden kaynaklı sorunlar oluşabilme ihtimaline karşı ilk yükseklik bilgisi hafızaya alınır. Sistem seriport üzerinden sağlıklı veri geldikçe yüksekliği %2 oranında azaltarak havada asılı kalmaya devam eder. 25 cm yüksekliğe

inildikten sonra motorlar yumuşak kapanış (soft cut-off) özelliği ile durdurulur. İniş işlemi başladıktan sonra AutoQuad tarafında yapılan tüm işlemlerin görsel şeması Şekil 2.24’de verilmiştir.



Şekil 2.24. AutoQuad üzerinde çalışan uygulama adımları.

Otonom iniş özelliği kazandırmak için AutoQuad’a geliştirilen uygulamanın kaba kodu Şekil 2.25’de verilmiştir.

```
1. begin
2.  initilation landing task
3.  configure seriport2
4.  if motor != armed
5.    return
6.  if mode != AltitudeHold
7.    return
8.  set radio.enable = false
9.  set radio.roll = middle
10. set radio.pitch = middle
11. while mode == AltitudeHold
12.   landstr ← seriport2.read()
13.   if landstr.state != OK
14.     exit
15.   imgwith ← getWidth(landingstr)
16.   imgheight ← getHeight(landingstr)
17.   centerX ← getCenterX(landingstr)
18.   centerY ← getCenterY(landingstr)
19.   distance ← getDistance(landingstr)
20.   while ( centerX <= (imgwith/2)-20 )
21.     radio.roll += radio.roll*0.01
22.   while ( centerX >= (imgwith/2)+20 )
23.     radio.roll -= radio.roll*0.01
24.   while ( centerY <= (imgheight/2)-20 )
25.     radio.pitch += radio.pitch*0.01
26.   while ( centerY >= (imgheight/2)+20 )
27.     radio.pitch -= radio.pitch*0.01
28.   if distance > 25
29.     radio.throttle -= radio.throttle*0.01
30.   if distance <= 25
31.     set motor = disarmed
32. end
```

**Şekil 2.25.** Autoquad üzerinde çalışan uygulamanın kaba kodu.

Tüm sistemin tasarlanmış son hali Şekil 2.26'da verilmiştir.



**Şekil 2.26.** Çalışmada kullanılan İHA.

İniş pistinin tespiti için yapılan çalışmalar, öğle vakti gün ışığı yoğunluğuna sahip 3 farklı ortamda gerçekleştirilmiştir. Her ortamda ortalama 20'şer kez test yapılmış ve %90 oranında iniş pisti tespiti başarıyla yapılmıştır. Ortamlardan alınan görüntüler EK 1 ve EK 2'de verilmiştir.

### 3. SONUÇ

Bu tezin amacı otomatik iniş yapabilen özerk bir İHA geliştirmektir. Bu hedefe ulaşmak için iki temel görev vardır: Öncelikle kararlı uçuş gerçekleştirebilen hexacopter platformu, diğer taraftan iniş pistinin tespiti ve güvenli iniş koşullarının sağlanmasıdır.

Bu çalışmada, düşük maliyetli, kullanımı kolay ve istikrarlı bir platform oluşturmak için tasarlanmış hexacopter çerçevesi ve AutoQuad kontrol kartı kullanılmıştır. Açık kaynak kodlu AutoQuad AQ6 modeli ana uçuş kontrol kartı olarak tercih edilmiştir. Yazılımın komut alma ve önceden programlanmış güzergahta ilerleme kalkış, iniş, gezinme ve noktadan noktaya navigasyon yetenekleri vardır.

Sistemde ana uçuş kartına paralel görüntü işleme adımlarını gerçekleştiren ikinci bir kontrol kartı daha bulunur. Görüntü işleme adımları Raspberry Pi 3 Model B ile gerçekleştirilmiştir. Görüntüler her 500 ms bir kameradan okunarak çeşitli filtrelere (bulanıklaştırma, renk uzayı, şekil) tabi tutulur. Tüm işlemlerin sonunda iniş pisti tespit edilir. İlk görüntü ile bulunan iniş pisti görüntüsü karşılaştırılarak merkez koordinatları belirlenir. Koordinatlar seriport üzerinden AutoQuad ana kontrol kartına iletilir. Tüm işlemler saniyede iki kere gerçekleştirilir.

İniş pistinin kamera açısına göre hangi yönde olduğu bilgisini alan AutoQuad, motor hızlarında Şekil 9'da belirtilen çalışma prensiplerine göre değişiklik yapar. İniş pisti merkez noktaları ile kamera görüntüsü merkez notaları üst üste geldiği anda motor hızları dengeli bir şekilde azaltılır. Kameradan gelen görüntülerde zemine yaklaşıldıkça iniş pisti tespitinde pürüzler oluşur. 25 cm yüksekliğin altına inildiğinde hedef görüntü tespit edilemeyecek duruma gelir. Bu noktadan sonra ESC'ler ile yumuşak kapatma modunda tüm motorlar kapatılır.

Çalışma kapsamında tasarlanan sistemin gelecekte geliştirilmesi düşünülen yanları aşağıda özetlenmiştir.

- İlk olarak tüm sistemin tek bir kontrol kartı ile çalışması hedeflenmektedir. Bu sayede ikinci kontrol kartından gelen ağırlıklardan ve bağlantı kablolarından kutulunmuş olur. Hem ağırlığın hem de elektrik tüketiminin azalacağından dolayı uçuş süresinde bir miktar artış sağlanır.

- Sistemde kullanılan kameraya kızıl ötesi sensörler eklenerek gün ışığında dahi net ve pürüzsüz görüntüler elde edilmesi sağlanır. Diğer taraftan GPS kullanarak daha yüksek hassasiyet oranı ile iniş pisti tespiti sağlanabilir.

- Özellikle günümüzde İHA'ların uçuş süresinin kısa olmasından dolayı otomatik iniş sisteminin, otomatik şarj sistemi ile desteklenmesi hedeflenmektedir. İHA, iniş pistine indikten sonra gerek kablosuz gerekse mıknatıslar ile kutuplar birleştirilerek otomatik şarj işlem başlatılır. Bu şekilde İHA'lar, çok uzun süre görevlerine devam edebilir.

- Bazı durumlarda İHA ile iniş pisti arası uzaklık beklenende fazla olabilir. İHA'nın iniş pistine ulaşana kadar bataryanın bitme riski olur. Bu gibi durumlar tüm sistemi kurtarabilecek hareketli iniş pistleri tasarlanabilir. İHA ile hareketli iniş pisti benzer hızlarda aynı doğrultuda hareket ederek geliştirilecek özel algoritmalar ile piste iniş sağlanabilir.

## KAYNAKLAR

- Peddle K. (2008). *Acceleration Based Manoeuvre Flight Control System for Unmanned Aerial Vehicles*, Doktora Tezi, Stellenbosch University, Güney Afrika.
- Kırcalı D. (2013). *Ground Plane Detection Using An Rgb-D Camera For Quadcopter Landing*. Yüksek Lisans Tezi, Işık Üniversitesi, İstanbul.
- Çiçekdemir Ç. (2012). *ARM mikrodenetleyici ve 9 serbest dereceli sensör ile Bilkopter dengeleme ölçümü*. Lisans Tezi, Bilecik Şeyh Edebali Üniversitesi, Bilecik.
- Leong W. (2006). *Computer Aided Landing Control System Design For A Fixed Wing UAV*. Yüksek Lisans Tezi, National Cheng Kung University, Tayvan.
- Wenze K.E. Wenze. (2010). *J Intell Robot Syst. Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller*, 57:297–311.
- Gomes T. (2013). *Quadcopter Automatic Landing on a Docking Station*. Yüksek Lisans Tezi, Instituto Superior Técnico, Lizbon.
- Dharampal and Vikram M. (2015). *Methods of Image Edge Detection. A Review, Journal of Electrical & Electronic Systems*, New York.
- Youeyun J. (2015). *Robust Marker Tracking Algorithm for Precise VA V Vision-based Autonomous Landing. International Conference on Control Automation and Systems ICCAS*, Bremen, Almanya.
- Costa, B. S. J., Greati, V. R., Ribeiro, V. C. T., da Silva, C. S., and Vieira, I. F. (2015). *A visual protocol for autonomous landing of unmanned aerial vehicles based on fuzzy matching and evolving clustering. Fuzzy Systems*, İstanbul, Türkiye.
- Beck, H., Lesueur, J., Charland-Arcand, G., Akhrif, O., Gagnı, S., Gagnon, F., Couillard, D. (2016). *Autonomous takeoff and landing of a quadcopter. International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, USA.
- Aydođdu Ö. (2006) *Fırçasız Doğru Akım Motorlarının Genetik Tabanlı Bulanık Denetleyici ile Sensörsüz Kontrolü*. Doktora Tezi, Selçuk Üniversitesi, Konya.
- Bappy R. A. (2007). *Design and Development of Unmanned Aerial Vehicle (Drone) for Civil Applications*. Yüksek Lisans Tezi, BRAC Universty, Dhaka, Bangladeş.
- Niemelä H. (2016) *Robot Control With Raspberry Pi*”, Yüksek Lisans Tezi, Oulu University of Applied Sciences, Finlandiya.
- Smolov A. (2017). *Control Of Multitank System Using Raspberry PI*, Yüksek Lisans Tezi, Tallinn University Of Technology, Estonya
- Çiçekdemir Ç. (2018). *Designing Autonomous Landing System for Rotary Wing UAVs. International Conference on Advanced Technologies*, Karabük, Türkiye

### KAYNAKLAR (Devam Ediyor)

- Nursabillilah Mohd Ali, Nahrul Khair Alang Md Rashid and Yasir Mohd Mustafah. (2013). Performance Comparison between RGB and HSV Color Segmentations for Road Signs Detection. *Scientific.net/AMM*. 393.550
- Suzuki S. (1985). Topological Structural Analysis Of Digitized Binary Images By Border Following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46
- Lizarraga M.I.. (2014). *Autonomous Landing System of a UAV*. Yüksek Lisans Tezi, Naval Postgraduate School, Kaliforniya, Amerika
- Cho A. (2008). *Fully Automatic Taxiing, Takeoff and Landing of a UAV only with a Single-Antenna GPS Receiver*. The 21st International Technical Meeting of the Satellite Division of The Institute of Navigation. Savannah, Gürcistan
- M.P.Chandane. (2016). *Real Time Operating Systems*. International Journal of Electrical and Electronics Engeineers, India International Center, New Delhi
- URL1: <https://www.nintendo.com/wiiu/accessories>, (Erişim Tarihi: 10.01.2019).
- URL2: <https://docs.px4.io/en/sensor/px4flow.html>, (Erişim Tarihi: 10.01.2019).
- URL3: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>, (Erişim Tarihi: 10.01.2019).
- URL4: <https://www.iso.org/committee/5336224.html>, (Erişim Tarihi: 10.01.2019).
- URL5: <https://cdn-global-hk.hobbyking.com/media/file/1013956805X669553X14.pdf>, (Erişim Tarihi: 10.01.2019).
- URL6: <https://cdn-global-hk.hobbyking.com/media/file/1006513628X70599X0.pdf>, (Erişim Tarihi: 10.01.2019).
- URL7: <http://autoquad.org/wiki/wiki/autoquad-hardware-connections/>, (Erişim Tarihi: 10.01.2019).
- URL8: <http://autoquad.org/about/>, (Erişim Tarihi: 10.01.2019).
- URL9: <https://opencv.org/platforms/>, (Erişim Tarihi: 10.01.2019).
- URL10: [https://docs.opencv.org/3.1.0/d4/d13/tutorial\\_py\\_filtering.html](https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html), (Erişim Tarihi: 10.01.2019).
- URL11: <https://www.learnopencv.com/color-spaces-in-opencv-cpp-python/>, (Erişim Tarihi: 10.01.2019).
- URL12: [https://docs.opencv.org/3.2.0/d2/de8/group\\_\\_core\\_\\_array.html](https://docs.opencv.org/3.2.0/d2/de8/group__core__array.html), (Erişim Tarihi: 10.01.2019).

**KAYNAKLAR (Devam Ediyor)**

URL13: [https://docs.opencv.org/3.3.1/d3/dc0/group\\_\\_imgproc\\_\\_shape.html](https://docs.opencv.org/3.3.1/d3/dc0/group__imgproc__shape.html), (Eriřim Tarihi: 10.01.2019).

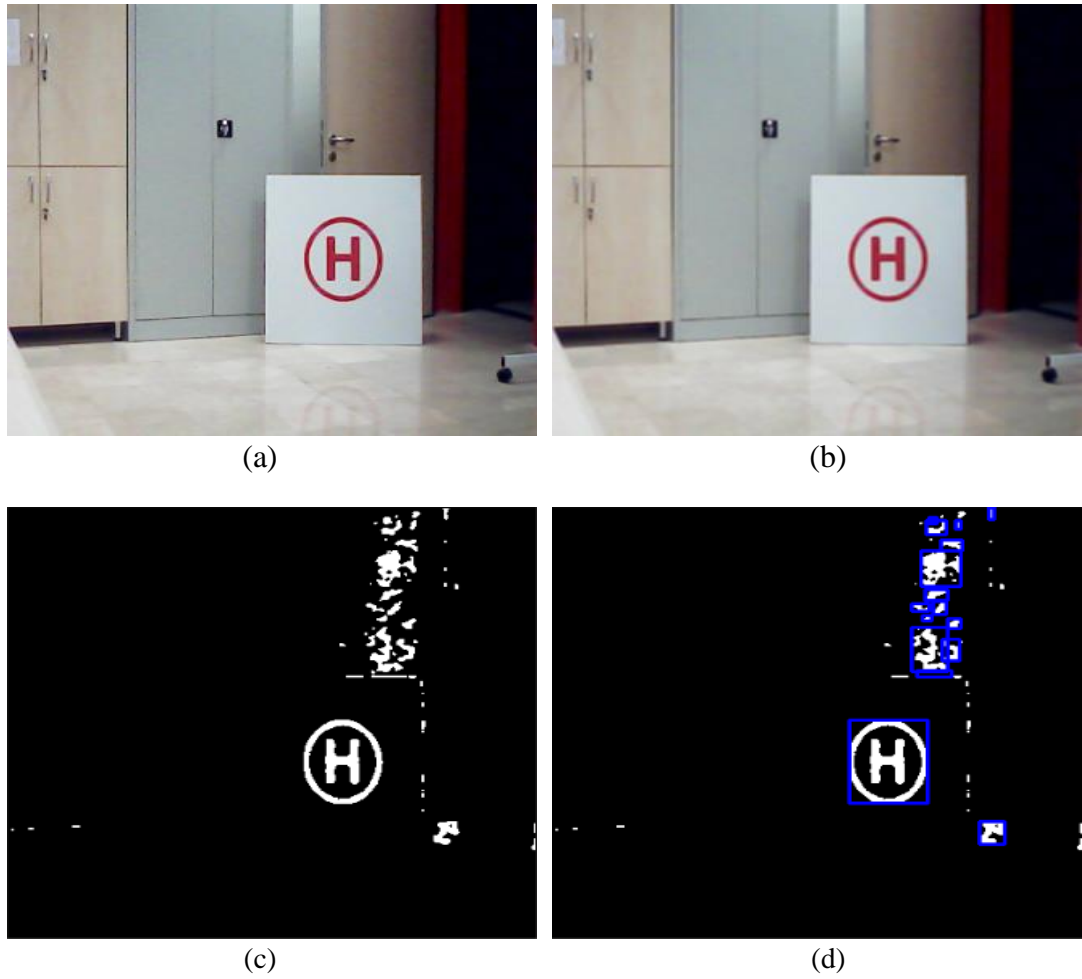
URL14: <https://github.com/coocox/CoOS>, (Eriřim Tarihi: 13.01.2019)

URL15: <https://www.raspbian.org/>, (Eriřim Tarihi: 12.01.2019)

URL16: <https://1volt1amper.com/raspberry-pi/raspberry-pi-3/> (Eriřim Tarihi: 10.01.2019).

URL17: [https://docs.opencv.org/3.0-eta/modules/imgproc/doc/geometric\\_transformations.html](https://docs.opencv.org/3.0-eta/modules/imgproc/doc/geometric_transformations.html), (Eriřim Tarihi: 10.01.2019).

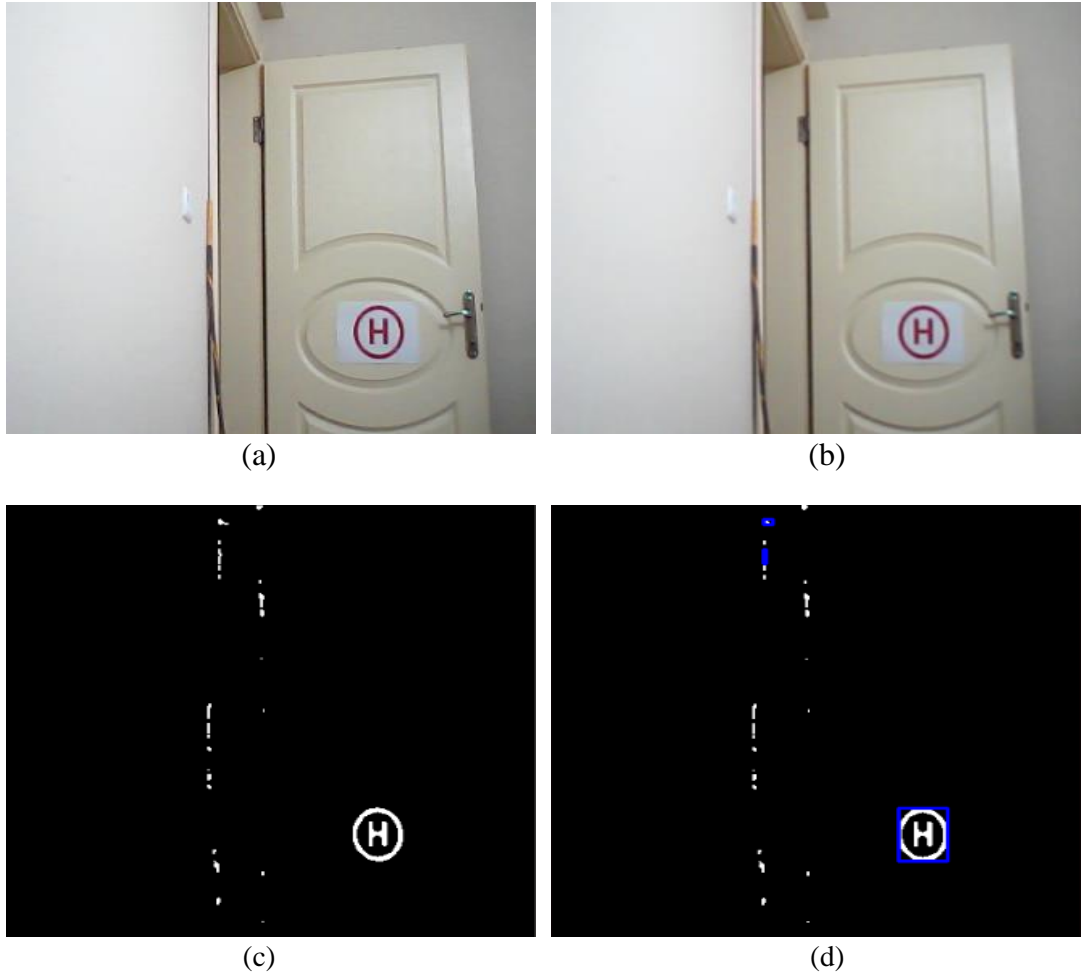
## EKLER (EK-1)



**Şekil 2.27.** İniş düzlemi algılama ortam testleri 2.  
 (a) Orijinal Resim (b) Bulanıklık Filtresi (c) Renk Tanımlama (d) Nesne Tanımlama

Laboratuvar ortamında normal ışık yoğunluğunun olduğu zaman diliminde 80x80 büyüklükte zemin üzerine 40x40 büyüklükte çember ve H harfli iniş pisti ile test yapılmıştır. Görüntüler saniyede iki kare olmak üzere ortalama 6 metre uzaklıktaki kamera ile çekilmiştir. Alınan görüntüler sırasıyla bulanıklaştırma, renk uzayı değişikliği, renk filtresi, şekil tanıma ve karşılaştırma yöntemleri kullanılmıştır. İlgili ortamda 15'den fazla yapılan testlerde %90 oranında hedef tespit edilmiştir.

## EKLER (EK-2)



**Şekil 2.28.** İniş düzlemi algılama ortam testleri 3.  
 (a) Orijinal Resim (b) Bulanıklık Filtresi (c) Renk Tanımlama (d) Nesne Tanımlama

Laboratuvar ortamında normal ışık yoğunluğunun olduğu zaman diliminde 40x40 büyüklükte zemin üzerine 20x20 büyüklükte çember ve H harfli iniş pisti ile test yapılmıştır. Görüntüler saniyede iki kare olmak üzere ortalama 3 metre uzaklıktaki kamera ile çekilmiştir. Alınan görüntüler sırasıyla bulanıklaştırma, renk uzayı değişikliği, renk filtresi, şekil tanıma ve karşılaştırma yöntemleri kullanılmıştır. İlgili ortamda 20'den fazla yapılan testlerde %90 oranında hedef tespit edilmiştir.

## ÖZ GEÇMİŞ



### Kişisel Bilgiler

Adı Soyadı : Çağrı Çiçekdemir  
Doğum Yeri ve Tarihi : Üsküdar 27/06/1989

### Eğitim Durumu

Lisans Öğrenimi : Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi,  
Bilgisayar Mühendisliği  
Bildiği Yabancı Diller : İngilizce

### İş Deneyimi

Stajlar : Nazilli Belediyesi Bilgi İşlem Daire Başkanlığı,  
BŞEÜ Mühendislik Laboratuvarı  
Projeler : İnsansız Hava Aracı Projeleri (Bilkopter)  
SAP İK Projeleri (Stm, Ford, Tofaş, Aktifbank)  
Çalıştığı Kurumlar : Asteroit Teknoloji, İksap Danışmanlık L.T.D.

### İletişim

Adres : Bahçelievler Mh. Sarmaşık Sk. No:6 Merkez, Bilecik  
E-Posta Adresi : cagricicekdemir@gmail.com

### Akademik Çalışmaları

- Çiçekdemir Çağrı, Üçgün Hakan, Yüzgeç Uğur, Kesler Metin  
Designing Autonomous Landing System For Rotary Wing Unmanned Air  
Vehicles (International Conference on Advanced Technologies, Computer  
Engineering and Science (ICATCES'18) (11.05.2018- 13.05.2018) İngilizce

**Tarih: 11/07/2019 .**