TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSIZ HAVA ARAÇLARINDA GÖRÜNTÜ İŞLEME TABANLI İNİŞ SİSTEMİ TASARIMI VE GERÇEKLEMESİ

Ali VURGUN

YÜKSEK LİSANS TEZİ
Mekatronik Mühendisliği Anabilim Dalı
Mekatronik Mühendisliği Programı

Danışman
Doç. Dr. Aydın YEŞİLDİREK

Temmuz, 2019

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSIZ HAVA ARAÇLARINDA GÖRÜNTÜ İŞLEME TABANLI İNİŞ SİSTEMİ TASARIMI VE GERÇEKLEMESİ

Ali VURGUN tarafından hazırlanan tez çalışması 10.07.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı Mekatronik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Aydın YEŞİLDİREK
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Aydın YEŞİLDİREK, Danışman
Yıldız Teknik Üniversitesi _____

Doç .Dr. Cüneyt Yılmaz, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Gökhan Erdemir, Üye
İstanbul Sabahattin Zaim Üniversitesi _____

Danışmanım Doç. Dr. Aydın YEŞİLDİREK sorumluluğunda tarafımca hazırlanan İNSANSIZ HAVA ARAÇLARINDA GÖRÜNTÜ İŞLEME TABANLI İNİŞ SİSTEMİ TASARIMI VE GERÇEKLEMESİ başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Ali VURGUN

İmza

Biricik eşime
ve
Aileme

TEŞEKKÜR

Bu çalışmada, iki yıl boyunca değerli bilgilerini ve tecrübesini benimle paylaşan saygıdeğer danışman hocam Doç. Dr. Aydın YEŞİLDİREK'e, benden bir an olsun yardımlarını esirgemeyen arkadaşlarıma, yüksek lisans eğitimimi destekleyen değerli yöneticilerime ve çalışma süresince tüm zorlukları benimle birlikte göğüsleyen sevgili eşim Elif BAĞATUR VURGUN'a sonsuz teşekkürlerimi sunarım.

Bu proje 2016-06-04-KAP01 nolu BAP projesi çerçevesinde sağlanan platform ile gerçekleştirilmiştir.

Ali VURGUN

İÇİNDEKİLER

SİMGE LİSTESİ viii

KISALTMA LİSTESİ ix

ŞEKİL LİSTESİ xi

TABLO LİSTESİ xiv

ÖZET xv

ABSTRACT xvii

1 Giriş 1

1.1 Literatür Özeti . . . 3
1.2 Tezin Amacı . . . 6
1.3 Orijinal Katkı . . . 6
1.3.1 Yönlendirme ve Kontrol . . . 6
1.3.2 Görüntü İşleme . . . 7
1.3.3 Test . . . 7

2 İnsansız Hava Araçları 8

2.1 Tanım . . . 8
2.2 İHA'ların Tarihi Gelişimi . . . 9
2.3 İHA'ların Kullanım Alanları . . . 10
2.3.1 Sivil Kullanım . . . 10
2.3.2 Askeri Kullanım . . . 11
2.4 İHA'ların Sınıflandırılması . . . 13
2.4.1 Sabit Kanatlı İHA'lar . . . 13
2.4.2 Döner Kanatlı İHA'lar . . . 13

3 Çalışmada Kullanılan Donanım ve Yazılımlar 17

3.1 Kullanılan Donanımlar . . . 17
3.1.1 İnsansız Hava Aracı - 3DR Solo . . . 17
3.1.2 Yer İstasyonu . . . 19

3.2 Kullanılan Yazılımlar 19
3.2.1 Python 19
3.2.2 Dronekit 20
3.2.3 Matlab 21
3.2.4 OpenCV 21
3.2.5 PyCharm 22

4 Haberleşme Altyapısı ve Görüntü Transferi **24**
4.1 İHA ve Yer istasyonu Arasındaki Haberleşme Altyapısı 24
4.2 İHA'dan Yer İstasyonuna Görüntü Aktarımı 26
4.2.1 GoPro Hero4 Kamerasının Kablosuz Ağı ile Yer İstasyonuna Görüntü Aktarımı 27
4.2.2 İHA'dan Yer İstasyonuna Görüntü Aktarımı 27

5 Hedef Platform Tespiti **34**
5.1 İniş Yapılacak Hedef Platform - Helipad 35
5.2 Haar Cascade Sınıflandırma Yöntemi ile Helipad Tespiti 35
5.2.1 Haar Cascade Öznitelikleri 37
5.2.2 Integral Image 37
5.2.3 Adaboost Algoritması 38
5.2.4 Kaskad Yöntemi 38
5.2.5 Helipad Bulma Algoritmasının Matlab Ortamında Eğitilmesi . . 39
5.3 Harris Köşe Bulma Algoritması 41

6 İHA Yükseklik Hesabı **44**
6.1 İHA Yükseklik Hesabı 44

7 Görüntü İşleme İle İniş Kontrolü **48**
7.1 İHA Hareket Kontrolü 48
7.1.1 X ve Y Eksenlerinde Helipad Üzerine Hareket Kontrolü 50
7.1.2 Z Ekseninde İniş Kontrolü 52
7.2 Rota Açısına Göre Hız Kontrolü 52
7.3 Görüntü İşleme İle İniş Kontrolü 53

8 Kullanıcı Arayüz Çalışmaları **56**
8.1 İHA ile Bağlantı Kurma ve Uçuş Modu Değişikliği 57
8.2 Otonom İnişin Başlatılması ve Kontrol Parametrelerin Ayarlanması . . . 58
8.3 Uçuş Verilerinin Gözlemlenmesi 58
8.4 Tanımlanmış Hareket Fonksiyonları 59
8.5 Kamera Görüntülerinin Ekranda Gösterilmesi ve Görüntü Ayarlarının Yapılması 59

8.6 İHA'nın X, Y ve Z Eksenlerinde Doğrusal Hareketi 60

9 Sonuçlar ve Öneriler **61**

9.1 Test ve Ölçüm Çalışmaları 61

9.1.1 Test Alanı 61

9.1.2 Uçuş ve İniş Testi Veri Kayıtları 62

9.1.3 İniş Testi Videoları 64

9.2 Sonuç ve Ölçüm Grafikleri 64

9.2.1 Görüntü Alma ve Görüntü İşleme Performansı 64

9.2.2 Görüntü İşleme ile Hesaplanan Yükseklik Verisi Doğruluğu . . . 66

9.2.3 Görüntü İşleme Tabanlı İniş Testi Verileri 66

9.2.4 Değerlendirme 68

9.3 Öneriler 72

9.3.1 Hareket Eden Platformlara İniş Kabiliyeti 72

9.3.2 Kamera Bakış Açısının Değişimine Göre İniş Algoritmasının Düzenlenmesi 73

9.3.3 İniş Platformu Değişikliği 73

9.3.4 Kapalı Ortamlarda Test Kabiliyeti 75

9.3.5 Pozisyon Referans Komutu İyileştirilmesi 75

Referanslar **77**

Tezden Üretilmiş Yayınlar **80**

SİMGE LİSTESİ

ΔL_i	H Harfinde Yer Alan Köşe Noktaların Orta Noktaya Olan Uzaklığı
h	İHA'nın Hedeften Yüksekliği
I_x, I_y	Piksellerin X ve Y Yönlerindeki Türevi
$P_h ys$	Histerezis Bölge Fonksiyonu
R	Harris Algoritması Köşe Noktası Cevabı
R_p	mm/piksel Oranı
S_{xx}, S_{yy}, S_{xy}	Image Structure Tensor Matrisi
v_x, v_y	İHA'nın X ve Y Yönlerindeki Hızı
x_m, y_m	H Harfi Orta Nokta Piksel Koordinatı

KISALTMA LİSTESİ

ABD	Amerika Birleşik Devletleri
AVP	Audio/Video Profile
BLDC	Brushless Direct Current
CM	Santimetre
CSV	Comma Seperated Values
CV	Computer Vision
ESC	Electronic Speed Control
FPS	Frame Per Second
GB	Gigabyte
GHz	Gigahertz
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
HDMI	High Definition Multimedia Interface
Hz	Hertz
IDE	Integrated Development Environment
IP	Internet Protocol
İHA	İnsansız Hava Aracı
kg	Kilogram
kV	KiloVolt
m	Metre
mAh	Mili Amper Saat
MAVLINK	Micro Aerial Vehicle Link

MEMS	Mikro Elektro-Mekanik Sistemler
MLL	Machine Learning Library
MM	Milimetre
MPEG	Moving Picture Experts Group
ms	milisaniye
NED	North-East-Down
PC	Personal Computer
PI	Proportional-Integral
RAM	Random Access Memory
RGB	Red - Green - Blue
ROI	Region of Interest
RPM	Revolution Per Minute
RPV	Remotely Piloted Vehicle
RTP	Real-Time Transport Protocol
s	saniye
SD	Secure Digital
SDP	Session Description Protocol
TB	Terabyte
VFR	Visual Flight Rules
VTOL	Vertical Take-Off and Landing
WiFi	Wireless Fidelty
xml	Extensible Markup Language

ŞEKİL LİSTESİ

Şekil 2.1 İnsansız Hava Aracı 8
Şekil 2.2 İlk İHA, Hewitt-Sperry Automatic Airplane 9
Şekil 2.3 Sabit Kanatlı İHA 14
Şekil 2.4 Döner Kanatlı; 3, 4, 6 ve 8 Rotorlu İHA 15
Şekil 2.5 İHA, Dönüş Yönlerine Göre Hareket Düzlemi 15
Şekil 2.6 İHA, İleri-Geri ve Sağ-Sol Hareket 16
Şekil 3.1 3DR Solo İnsansız Hava Aracı 17
Şekil 3.2 Pixhawk 2 Black Otopilot Kartı 18
Şekil 3.3 Matlab-OpenCV 22
Şekil 3.4 PyCharm 23
Şekil 4.1 İHA Basitleştirilmiş Sistem Diyagramı 25
Şekil 4.2 İHA-Yer İstasyonu Haberleşme Yapısı 26
Şekil 4.3 GoPro Hero4 Black Kamera 27
Şekil 4.4 GoPro Wifi Üzerinden Alınan Görüntü 28
Şekil 4.5 NetCat Aracı ile Yer İstasyonundan İHA'ya Bağlantı 29
Şekil 4.6 İHA Kumandası Bağlantı Sağlandı Uyarısı 29
Şekil 4.7 İHA Kumandası Bağlantı Kesildi Uyarısı 29
Şekil 4.8 VLC Multimedya Programı Üzerinden Alınan Görüntü 31
Şekil 4.9 FFMPEG Multimedya Programı Üzerinden Alınan Görüntü 32
Şekil 4.10 GStreamer Multimedya Programı Üzerinden Alınan Görüntü 32
Şekil 5.1 Görüntü İşleme Algoritması Akış Diyagramı 34
Şekil 5.2 İniş Yapılacak Hedef Platform - Helipad 35
Şekil 5.3 Helipad Ölçüleri 36
Şekil 5.4 Haar Cascade Sınıflandırıcı Çalışma Prensibi 36
Şekil 5.5 Haar Kaskad Öznitelikleri 37
Şekil 5.6 Integral Image 38
Şekil 5.7 Haar Kaskad 39
Şekil 5.8 Matlab Image Labeler Uygulaması Arayüzü 40
Şekil 5.9 Matlab Çalışma Ortamına Aktarılmış Helipad ROI Bilgileri 40
Şekil 5.10 Haar Cascade Sınıflandırıcı ile Helipad Tespiti 41
Şekil 5.11 Harris Köşe Bulma Algoritması Çalışma Yöntemi 42

Şekil 5.12 Haar Cascade ve Harris Algoritmaları ile Helipad Tespiti 43
Şekil 6.1 İHA Yükseklik Görüntüsü 45
Şekil 6.2 H Harfi Orta Nokta 46
Şekil 6.3 H Harfi Köşe Uzaklık 46
Şekil 6.4 GoPro Hero4 Görüş Alanı 47
Şekil 7.1 Kontrol Algoritması Blok Diyagramı 48
Şekil 7.2 İHA Motor Control 49
Şekil 7.3 PI Kontrolcü Yapısı 49
Şekil 7.4 Görüntüdeki Bölgeler ve Köşe Nokta Piksel Verileri 50
Şekil 7.5 Yüksekliğe Bağlı Histeresis Bölgesi (Piksel) 51
Şekil 7.6 Yüksekliğe Bağlı Histeresis Bölgesi (Santimetre) 51
Şekil 7.7 20 cm Yükselikten Alınan Görüntü 52
Şekil 7.8 İHA Rota Açısı 53
Şekil 7.9 Otonom İniş Algoritma Akış Diyagramı 54
Şekil 7.10 Hedef Arama Modu 55
Şekil 7.11 Hedef Takip Modu 55
Şekil 8.1 Kullanıcı Arayüzü 56
Şekil 8.2 İHA Bağlantı Bölümü 57
Şekil 8.3 Uçuş Modu Seçimi Bölümü 57
Şekil 8.4 Otonom İniş ve Kontrol Parametreleri Ayarı 58
Şekil 8.5 Uçuş Hız ve Pozisyon Verileri 58
Şekil 8.6 Tanımlanmış Hareket Noktaları 59
Şekil 8.7 İHA Görüntü Kontrastı ve Parlaklığı Ayarı 60
Şekil 8.8 X, Y ve Z Eksenlerinde İHA Hareketi 60
Şekil 9.1 Test Alanı Google Maps 61
Şekil 9.2 Test Ortamından Görüntü - 1 62
Şekil 9.3 Test Ortamından Görüntü - 2 63
Şekil 9.4 Test-1 İHA Videosu 64
Şekil 9.5 Test-1 Yer İstasyonu Videosu 64
Şekil 9.6 Test-2 İHA Videosu 64
Şekil 9.7 Test-1 Yer İstasyonu Videosu 64
Şekil 9.8 İHA'dan 1 Saniyede Alınan Görüntü Sayısı 65
Şekil 9.9 İHA'dan 1 Saniyede Alınan ve İşlenen Görüntü Sayısı 66
Şekil 9.10 Görüntü İşleme ile Bulunan İHA Yükseklik Bilgisi Doğrulaması . . . 67
Şekil 9.11 İHA'nın Helipad'e İnişi Esnasında X, Y ve Z Eksen Pozisyonları 68
Şekil 9.12 X ve Y Eksenleri Pozisyon Kontrol Grafikleri 69
Şekil 9.13 İHA NED'e Göre X Y ve Z Eksen Hız Kontrol Grafiği 69
Şekil 9.14 Rota Açısına Göre X ve Y Eksen Hız Kontrol Grafikleri 70
Şekil 9.15 Görüntü İşleme İle Hesaplanan Zamana Bağlı Yükseklik Grafiği . . . 70

Şekil 9.16 Rp(mm/piksel) Oranı Grafiği 71
Şekil 9.17 İHA'nın Havaya ve Yere Göre Hızı 71
Şekil 9.18 İHA'nın Uçuşu Sırasında Yunuslama, Sapma ve Yuvarlanma Açıları . 72
Şekil 9.19 İHA'nın Hareket Eden Platformlara İnişi 73
Şekil 9.20 Kamera Açısı Değişimi 74
Şekil 9.21 Kullanılabilecek Muhtemel İniş Platformu 74
Şekil 9.22 Adım Girişi Rampa Çevrimi 76

TABLO LİSTESİ

Tablo 3.1 3DR Solo İHA Özellikleri 18
Tablo 3.2 Yer İstasyonu Özellikleri 19
Tablo 4.1 GoPro Hero4 Black Çözünürlüğü 28
Tablo 4.2 Multimedya Çerçeve Yazılımları Karşılatırması 33

ÖZET

İNSANSIZ HAVA ARAÇLARINDA GÖRÜNTÜ İŞLEME TABANLI İNİŞ SİSTEMİ TASARIMI VE GERÇEKLEMESİ

Ali VURGUN

Mekatronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Aydın YEŞİLDİREK

Döner kanatlı İnsansız Hava Araçları (İHA); manevra ve operasyon kabiliyetlerinin yüksek olması, insan müdahalesinin mümkün olmadığı durumlarda kullanılabilir olması, içerisinde herhangi bir yaşamsal faaliyetin bulunmaması ve bakım idame masraflarının daha düşük olması gibi sebeplerden dolayı 21. yüz yıl itibari ile çok farklı uygulama alanlarında yaygın olarak kullanılmaya başlanmıştır. İHA'ların görevlerini veya uçuş sürelerini tamamlamasından sonra, önceden belirlenmiş bir hedef konuma güvenli bir şekilde iniş yapabilmesi önemli ve kritik bir görevdir. Pilot dikkatsizliği, ölçüm hataları veya çevresel etkilerden kaynaklı nedenlerle İHA'ların inişini güvenli ve yüksek doğrulukta gerçekleştirilememesi; genellikle görevin başarısızlığıyla, maddi hasarlarla, çevre kirliliğiyle veya hayati tehlikelerle sonuçlanabilmektedir. Bu çalışmada, dikey iniş-kalkış yapabilen döner kanatlı İHA'lar için görüntü işleme tabanlı hedef tespiti yapılmış, bu hedefe hassas bir şekilde iniş yapabilmek için kontrol algoritmaları geliştirilmiş ve geliştirilen tüm algoritmalar bir quadcopter platformunda gerçeklenmiştir. İHA'nın iniş yapacağı hedef platform olan Helipad'in konumunu algılamak için herhangi bir özel sensör kullanılmamış olup, sadece İHA'da bulunan kameradan alınan görüntüler gerçek zamanlı işlenerek bu işlem istenilen doğrulukta sağlanmıştır. Bu sayede, düşük maliyetli MEMS tabanlı ataletsel sensörlerin pozisyon ve hız ölçüm doğruluklarından kaynaklı hatalı konum geri beslemesinin önüne geçilmiş ve İHA'nın Helipad'e çok daha hassas iniş yapması sağlanmıştır. Helipad'in görüntü işleme yöntemiyle bulunmasında Haar-Cascade nesne bulma yöntemi ve Harris köşe bulma algoritmaları baz alınarak bir rota planlaması yapılmıştır. Helipad üzerinde tanımlanan özellik noktaları lokal hareket sensörleriyle birleştirilerek İHA'nın

Helipad'ten olan yüksekliği bulunmuş ve iniş için gerekli hassas pozisyon bilgileri elde edilmiştir. Bu algoritmaların gerçeklenmesi için açık kaynak kodlu PX4 otopilot yazılımı kullanılmış ve testleri 3DR Solo quadcopter üzerinde yapılmıştır. Geliştirilen tüm algoritmalar bir yer istasyonunda çalıştırılmış ve İHA'nın tüm kontrol komutları bu yer istasyonundan İHA'ya gönderilmiştir. İHA'nın yer istasyonundan kontrolünün daha rahat gerçekleştirilebilmesi ve uçuş esnasında uçuş verilerin anlık takip edilebilmesi için kullanıcı arayüzü tasarlanmıştır. Uçuş testleri dış ortamda gerçekleştirilmiş ve geliştirilen tüm algoritmaların doğrulaması yapılmıştır.

Anahtar Kelimeler: İnsansız Hava Aracı, Görüntü İşleme, Otonom İniş, Nesne Tespiti

ABSTRACT

DESIGN AND REALIZATION OF VISION BASED LANDING ALGORITHM FOR UNMANNED AERIAL VEHICLE

Ali VURGUN

Department of Mechatronics Engineering
Master of Science Thesis

Advisor: Assoc. Prof. Dr. Aydın YEŞİLDİREK

Rotary-wing Unmanned Aerial Vehicles (UAV) have widely been used in the 21th century because of some advantages such as high maneuver and operation capabilities, usability without human interventions, unmanned usability and low maintenance and operation costs. The safe landing of UAVs to a predetermined target location, after completion of their missions, have been a challenging task. Failure to perform safe landing of a UAV caused by pilot carelessness, sensor errors or environmental factors are likely to be followed by severe material damages, environmental pollutions and life threating accidents. In this study, a vision based landing algorithm is designed for UAV that can vertical-takeoff-and-land (VTOL). The target platform used for landing is selected as Helipad. No sensors other than the camera have been used to detect the location of the Helipad for landing. Helipad is detected using real-time image processing algorithms. Haar-Cascade object detection and Harris corner detection algorithms are employed to find it. Features defined on the Helipad image fused by the local navigation sensory data to obtain precise pose information needed in an autonomous vertical-landing operation. In this way, incorrect position information due to position measurement accuracy of the sensors is prevented and a much more accurate landing is made to the target platform. All algorithms are developed and integrated in to a ground station, and tested on 3DR Solo quadcopter that has open-source PX4 autopilot. Also, UAV's control commands are sent from the ground station back to the UAV. A user interface is designed to easly control the UAV from ground station and to track flight data instantly during flight. Flight tests are

performed in outdoor environment and developed algorithms are validated.

Keywords: Rotary-wing Unmanned Aerial Vehicle, Vision Based Autonomous Landing,Object Detection

1
Giriş

İnsansız Hava Araçları, uzaktan kontrol edilebilen veya otonom olarak çalışabilen insansız uçaklardır. Bu uçaklar, insan müdahalesinin mümkün olmadığı durumlarda rahatlıkla kullanılabilir olduğundan sivil ve askeri alanda yaygın olarak kullanılmaktadır. İHA'ların görevlerini yeterine getirdikten sonra belirlenen hedefe güvenli bir şekilde indirilmesi gerekmektedir. İHA'nın önceden belirlenmiş bir konuma olabildiğince sapma yapmadan indirilebilmesi, bu araçları yönlendiren operatörlerin dikkat ve kabiliyetine bağlıdır. Operatörün dikkatsizliğinden veya çevresel birtakım sorunlardan kaynaklı İHA'nın inişinin güvenli bir şekilde sağlayamaması çoğu zaman maddi hasarlarla sonuçlanmaktadır. Var olan otomatik iniş sistemlerinde yüksek doğruluklu sensörler kullanılması gerektiğinden maliyetler çok yüksek olabilmektedir. Düşük maliyetli sensörlerin kullanıldığı uygulamalarda ise sensör doğruluğundan kaynaklı hedefe güvenli iniş sağlanamamaktadır. Bu çalışmada, İHA'lardaki bu gibi problemlere karşılık maliyet avantajlı, hassas ve farklı yüksekliklerdeki hedef noktalara iniş yapabilen görüntü işleme tabanlı otonom iniş çözümü sunulmuştur.

Çalışma boyunca dikey iniş kalkış yapabilen dört rotorlu 3DR Solo marka İnsansız Hava Aracı kullanılmıştır. Bu İHA'da, otopilot algoritmasının koştuğu Pixhawk 2 Black otopilot donanımı yer almaktadır. Bu donanım, üzerinde bulunan I2C, SPI ve Seri haberleşme modülleri ile dış ortamla haberleşebilmektedir.

İniş yapılacak hedef platform, üzerinde H harfi bulunan minyatürize edilmiş helikopter iniş pisti şeklinde tanımlanan Helipad seçilmiştir. Bu Helipad, sabit bir zeminde olacak şekilde iniş testleri gerçekleştirilmiştir.

Helipad'in bulunması ve İHA'nın Helipad'ten yüksekliğinin hesaplanması için görüntü işleme yöntemi kullanılmıştır. Helipad'in bulunması için Haar Cascade sınıflandırıcısı ve Harris köşe bulma algoritmaları kullanılmıştır. İHA'nın Helipad'ten yüksekliğini hesaplarken, Helipad içerisinde yer alan H harfinin köşe noktaları ve İHA'da bulunan kameranın görüş açısı verisi kullanılmıştır.

İHA'nın otonom inişi için tasarlanan görüntü işleme ve kontrol algoritmaları bir yer istasyonunda çalışmaktadır. İHA'da bulunan kameradan alınan görüntüler anlık olarak kablosuz haberleşme ağı üzerinden yer istasyonuna gönderilmektedir. Bu görüntüler yer istasyonunda işlenerek İHA'ya gitmesi gereken yön ve hız bilgisi gönderilmektedir. Yer istasyonunda İHA'nın üç eksende hareketini kontrol eden üç adet pozisyon kontrol çevrimi çalışmaktadır. Pozisyon kontrolünde PI kontrolcüler kullanılmıştır. Pozisyon kontrolcülerinin çıkış değerleri, İHA'nın ilgili eksendeki hız referansını oluşturmaktadır. Yer istasyonundan İHA'ya gönderilen referans hız komutları, İHA içerisinde yer alan otopilot donanımında işlenerek İHA'nın hareketi sağlanmaktadır.

İHA seyir halindeyken yer istasyonundaki Kullanıcı Arayüzü'nden otomatik inişin aktif etmesiyle birlikte İHA, bulunduğu konumda Hedef Arama Modu'na sokulmaktadır. Hedef Arama Modu'nda İHA, yüksekliğini sabit tutarak daha önceden belirlenen çaptaki bir alanda seyir yapmaktadır. Helipad'in görüntü işleme yöntemiyle bulunması ile birlikte, yine yerden olan yükseklik sabit tutularak Hedef Takip Modu aktif edilmekte ve İHA Helipad'in üst noktasına gönderilmektedir. İHA, belirli toleranslar dahilinde Helipad'in üst noktasına geldiği anda İniş Modu aktif edilmekte ve iniş işlemi başlamaktadır. İnişin kontrollü bir şekilde gerçekleştirilebilmesi için iniş esnasında yanal pozisyonalar da kontrol edilmektedir.

İHA'nın tüm hareketlerini kontrol etmek ve aynı ekrandan uçuş verilerini takip etme amacıyla Kullanıcı Arayüzü geliştirilmiştir. Bu arayüz üzerinden; İHA ile bağlantı kurulabilmekte, İHA istenilen herhangi bir konuma gönderilebilmekte, İHA'nın x, y ve z eksenlerinde doğrusal pozisyon ve hız kontrolü yapılabilmekte, İHA'dan alınan görüntüler izlenebilmekte ve otonom iniş başlatılabilmektedir.

Geliştirilen tüm algoritmaların altsistem ve sistem seviyesi testleri yapılmıştır. Bu algoritmalarla çok sayıda iniş testi yapılarak, otonom inişin hassas, kararlı ve başarılı bir şekilde gerçekleştirilebilmesi için kontrol parametreleri optimize edilmiştir. Tüm uçuş testleri dış ortamda gerçekleştirilmiştir.

Sivil havacılıkta kullanılan sensörlerin pahalı ve yedekli olmasından dolayı otomatik iniş sistemlerinin uçak başına maliyeti oldukça yüksektir. Otomatik iniş teknolojisinin yerlileştirilmesinin ilk adımı olabilecek ve malzemede dışa bağımlılığı azaltıcı etkilerinin olmasından dolayı projemizin önem seviyesi ülke kazanımı açısından yüksek olduğu düşünülmektedir.

1.1 Literatür Özeti

İnsansız Hava Araçları'nın kullanımının yaygınlaşması ile birlikte bu alandaki ihtiyaçlar artmıştır. İhtiyaçların artmasından dolayı bu alada yapılan çalışmalarda zaman geçtikçe artmaktadır.

İHA'ların belirli bir hedef noktaya hassas şekilde indilerilebilmesi de bu ihtiyaçların başında gelmektedir. İHA'ların belirlenen hedef noktaya iniş yapabilmesi için öncelikle iniş yapılacak hedef platformun pozisyon değerlerini yüksek doğruluklu tespit etmek gerekmektedir. Günümüz teknolojisindeki GPS konumlandırma sensörlerinin pozisyon ölçüm hassasiyeti metreler mertebesinde olabilmektedir. Ölçüm hassasiyeti yüksek olan GNSS modüllerinin ise fiyatları oldukça yüksektir. İHA'nın iniş yapabileceği tüm hedef platformlarda bu sensörlerin kullanılması durumunda maliyet anlamında oldukça dezevantajlı durum ortaya çıkmaktadır. Ayrıca, iniş yapılacak olan hedef platformların GPS gibi konumlandırma sensörleri ile tespit edilmesi durumunda, hedefe hassas iniş yapması istenilen tüm İHA'ların donanımsal ve yazılımsal olarak bu altyapıya uygun olması gerekmektedir. Aynı zamanda, İHA'lar ile iniş yapılacak olan hedef platformların haberleşebiliyor olması da gerekmektedir [1].

Yukarıda bahsedilen birtakım sebeplerden dolayı, İHA'ların hedef platformlara hassas iniş yapabilmesi için görüntü işleme tabanlı çözümler ele alınmıştır. Bu çalışmaların bir kısmı halen geliştirme aşamasında olamakla birlikte askeri ve sivil alanlarda ürün haline dönüşen çalışmalarda bulunmaktadır.

Görüntü işleme tabanlı iniş sistemi tasarımında donanımsal olarak iki farklı yaklaşım bulunmaktadır. Bunlar; tüm kontrol algoritmalarının bir yer istasyonunda çalıştırıldığı yaklaşım ve bu algoritmaların İHA üzerinde bulunan donanımlarda (onboard) çalıştırıldığı yaklaşımdır [2][3].

Yer istasyonunun kullanıldığı çalışmalarda tüm görüntü işleme ve kontrol algoritmaları yer istasyonunda (genellikle kişisel bilgisayarlarda) çalışmaktadır. İHA'da bulunan kameradan alınan görüntüler yer istasyonuna gönderilmektedir. Yer istasyonunda yapılan hesaplamalar sonrasında İHA'nın hareket komutları da aynı şekide yer istasyonundan İHA'ya gönderilmektedir. Yer istasyonu ve İHA arasında kablosuz haberleşme bağlantısı ile veri transferi yapılmaktadır. Bu sayede kullanıcının denetiminde güvenli iniş gerçekleştirilebilmektedir.

Donanımların İHA üzerinde yer aldığı durumda ise tüm görüntü işleme ve kontrol algoritmaları harici donanımlarda çalışmaktadır. İHA ile harici donanım arasındaki haberleşme kablolar vasıtasıyla sağlanmaktadır. Harici donanımların enerji ihtiyacı İHA'nın bataryasından karşılanmaktadır. Harici donanımların İHA

üzerinde bulunması, İHA için ekstra ağırlığa sebep olamaktadır. Aynı zamanda, herhangi bir veri kayıt işlemi yapılmadığı sürece uçuş ve iniş esanasındaki veriler kullanıcı tarafından takip edilememektedir. Bu da geliştirme aşamasında problem çözüm sürelerinin uzamasına ve hatta İHA'nın hatalı hareket etmesinden dolayı yıkıcı sonuçlar doğmasına sebep olabilmektedir [4].

Görüntü işleme tabanlı geliştirilen algoritmalarda kameralardan alınan görüntüler kullanılmaktadır. Kameradan RGB formatında alınan görüntü 3 boyutlu bir matrisi ifade etmektedir. Bu matrisler bütünüyle işlem yapmak standart işlemci mimarileri ile çok kolay değildir. Bu sebeple, kullanılacak donanımlarla birlikte, bu donanımlarda çalıştırılacak görüntü işleme algoritmaları birlikte değerlendirilmektedir. Her zaman en yüksek performansa erişme hedefi olduğundan dolayı, en uygun görüntü işleme yöntemini kullanmak önemlidir. Eğer uygulamada kullanılacak görüntü işeleme yöntemine karar verilmişse, bu yönteme en uygun donanım altyapısını kullanmak gerekmektedir. Bu bakış açısıyla değerlendirildiğinde, iniş yapılacak hedef platformun görüntü işleme yönemi ile tespit edilmesinde farklı görüntü işleme teknikleri kullanılabilmektedir [5][6]. Bunlardan bazıları aşağıdaki gibi sınıflandırılmıştır:

- Nesnenin rengine göre hedef tespiti (dış ortamın renginden farklı renkte seçilen hedef platformlarda uygulanmaktadır)
- Nesne şekline göre hedef tespiti (üçgen, çember, dikdörtgen vb. şekillerin tespiti)
- Nesne sınıflandırma yöntemi uygulanarak (hedef platformun makine öğrenmesi tabalı sınıflandırma algoritmaları kullanılarak)
- Hedef platform üzerindeki köşe noktaların tespiti
- Hedef platform üzerindeki kenar özelliklerin tespiti
- Derin öğrenme algoritmaları kullanarak

Her bir görüntü işleme tekniğinin avantajlı ve dezavantajlı yönleri bulunmaktadır. Görüntü işleme algoritmasının çalıştırıldığı donanım aynı tutulduğu durumda, nesne şekline göre hedef tespiti yapılan bir algoritmada görüntü işleme süresi derin öğrenme yöntemlerine göre çok daha kısadır. Fakat görüntü işleme performansı daha düşüktür. İniş yapılacak hedef platformun tespiti için ilgili isterler ve teknik kısıtlar göz önüne alındığında en uygun yöntemin kullanılması iniş başarımını doğrudan etkilemektedir.

Hedef platforma inişin başarılı bir şekilde gerçekleştirilmesinin yanında, İHA'nın hedefe göre oryantasyonuda ayarlanabilmektedir. Helipad üzerinde

belirlenen referans açı değerlerine göre İHA'nın rota açısı değiştirilerek bu işlem gerçekleştirilebilmektedir [2].

Görüntü işleme tabanlı iniş sistemlerinde, iniş yapılacak olan hedef platformun büyüklüğü, kullanılan kameranın çözünürlüğü ve İHA'nın iniş yapmaya başlayacağı maksimum yükseklik gibi parametreler birbirleri arasında bağıntılıdır. Hedef platformun görüntü işleme yöntemiyle tespit edilebilmesi için kameradan alınan görüntüdeki hedef platform büyüklüğü, görüntü işleme yapabilecek seviyede olması gerekmektedir. Hedef platformun çok küçük ve İHA'nın çok yüksekte olması durumunda, hedefin tespit edilebilemsi için kamera çözünürlüğünün çok yüksek olması gerekmektedir. Kamera çözünürlüğünün çok düşük olması durumunda ise hedef platformun tespit edilebilmesi için İHA'nın inişe başlayacağı maksimum yükseklik değeri sınırlanmaktadır. Hedef platformun yüksek irtifalarda İHA tarafından görülebilmesi için çok büyük seçilmesi durumuna ise, alçak irtifada hedef platformun kamera görüş açısına sığmaması gibi problemler çıkabilmektedir. Bu soruna çözüm olarak tek bir hedef platformda farklı büyüklüklerde nesnelerin yer aldığı çalışmalar yapılmıştır. Bu sayede İHA, yüksekliğine bağlı olarak hedef platformdaki nesneleri sırasıya bularak iniş işlemini gerçekleştirebilmektedir [5][6].

İniş yapılacak hedef platform sabit veya hareketli olabilmektedir. Sabit duran hedef platformlara iniş, hareketli hedef platformlara göre daha kolaydır. Hareketli platformlara inişin gerçekleştirilebilmesi için dinamik değişikliklere daha hızlı cevap verebilen yapının ortaya koyulması gerekmektedir [7][6].

İniş esnasında İHA'nın hedeften yüksekliğini tespit etmek için farklı yöntemler kullanılabilmektedir. Hedef platform zeminde yer alıyorsa eğer, İHA'da bulunan GPS modülünden alınan veriler kullanılabilmektedir. Kapalı ortamlarda çalışabilen bir sistem tasarımı yapıldığı durumda, GPS verileri kullanılamayacağı için barometre sensörleri kullanılabilmektedir. Barometre ile ölçülen basınç verisi yükseklik bilgisine çevrilerek kontrol çevrimlerinde geribesleme verisi olarak kullanılabilmektedir [2]. Hedefe inişte yüksek doğruluklu pozisyonlama gerektiren durumlarda sonar sensörü kullanılabilmektedir. Sonar'ın mesafe ölçüm aralığı havada kısıtlı olduğundan dolayı, hedef platforma inişe birkaç metre kalana kadar GPS verilerinden faydalanılırken, belirlenen limit değerin altına inildiğinde sonar sensöründen alınan veriler kullanılarak hassas iniş gerçekleştirilebilmektedir. Bunların dışında, hedef platformda tespit edilebilen bir uzunluk verisi ve görüntü alınan kameranın görüş açısı bilgileri kullanılarak yine görüntü işleme yöntemi ile İHA'nın hedeften yüksekliği hesaplanabilmektedir.

1.2 Tezin Amacı

Hava araçlarında güvenlik, güvenli iniş sayısının kalkış sayısına eşit olduğu durumda sağlanmaktadır. İHA'ların güvenli inişinin yanında, tam olarak hedeflenen noktalara güvenli iniş yapmaları beklenmektedir. Hali hazırdaki sistemlerde iniş yapılacak olan hedef platformların konumlarını tespit etme amacıyla düşük maliyetli GPS modülleri kullanılmaktadır. Bu GPS modüllerinden elde edilen pozisyon verisinin doğruluğu birkaç metreler mertebesinde olabilmektedir. İHA'lar, bu GPS modüllerinden elde edilen konum verileri ile belirlenen hedef platformlara hassas bir şekilde iniş gerçekleştirememektedir.

1.3 Orijinal Katkı

İHA'ların yakın gelecekte günlük hayatta daha falza kullanılanılacağı öngörülmektedir. Bu sebeple, İHA'lar için ileri düzey yönlendirme sistemlerinin geliştirilmesi, kesintisiz bir şekilde hareket sağlayabilmesi ve güvenli iniş kalkış yapabilmesi çok büyük önem arz etmektedir. Bu projede, İHA'ların belirlenen hedef noktalara daha hassas ve güvenli bir şekilde inişi için görüntü işleme tabanlı çözüm sunulmaktadır.

Bu çalışmanın temel amacı, harici sensörlerin getirdikleri dezavantajları ortadan kaldırıp, İHA'nın otonom uçuşu esnasında iniş yapacağı hedef platformu görüntü işleme tekniklerini kullanarak tespit etmesi ve bu hedef platforma olabildiğince sapma yapmadan güvenli bir şekilde iniş yapmasını sağlamaktır.

1.3.1 Yönlendirme ve Kontrol

İHA'nın uçuşu esanasında dengede kalabilmesi ve belirlenen noktalara hareketini sağlayabilmesi için uygun motor hızları otopilot yazılımı tarafından ayarlanmaktadır. Otopilot, İHA'nın belirlenen hareketini sağlayabilmek için uyun motor hızlarını ayarlarken GPS ve Jiroskop verilerini kullanmaktadır. Otopilot yazılımı İHA'nın içerisinde yer alan mikrodenetleyiciler üzerinde çalışmaktadır. Döner ve sabit kanatlı İHA'lar için açık kaynak kodlu otopilot yazılımları bulunmaktadır. Ardupilot bunlardan birtanesidir. Bu çalışmada, ArduPilot Copter v1.5.4 otopilot yazılımı kullanılmıştır.

İHA'nın yönledirme komutları, İHA'nın uzaktan kumandası yada bir yer istasyonundan gönderilebilmektedir. Bu komutların uzak bir göndericiden İHA'ya iletilebilmesi için gönderici ve alıcı arasında haberleşme altyapısı kurulması gerekmektedir. Bu çalışmada, İHA ve yer istasyonu arasında haberleşmenin sağlanabilmesi için kablosuz haberleşme ağı üzerinden SoloLink ile haberleşme altyapısı kurulmuştur. İHA kamerasından alınan görüntüler ve İHA'nın uçuş verileri İHA'dan yer istasyonuna

gönderilirken, yer istasyonundan İHA'ya hareket komutları gönderilmektedir.

Yer istasyonunda çalışan, İHA'nın pozisyonunu kontrol eden algoritma içerisinde PI kontolcüler kullanılmaktadır. PI kontrol, oransal ve integratör terimlerden oluşan geri beslemeli kontrol tekniğidir. Referans pozisyon değerinden ölçülen pozisyon değeri çıkartılarak hata değeri bulunmaktadır. Bu hata değeri, PI kontrolcüye giriş olarak beslenerek çıkışta İHA'nın ulaşması istenilen çizgisel hız referansı üretilmektedir.

1.3.2 Görüntü İşleme

İHA'nın iniş yapacağı hedef platformu tespit edebilmesi için birden fazla yöntem bulunmaktadır. Konum bilgisi geri beslemeli sensörlerin hatalarından kaçınma amacıyla bu çalışmada görüntü işleme tabanlı çözümler ele alınmıştır. İHA'dan alınan görüntünün işlenmesinde Haar Cascade nesne sınıflandırma algoritması ve Harris köşe bulma yöntemi kullanılmıştır. Haar Cascade sınıflandırma algoritması makine öğrenmesi tabanlı bir algoritmadır. Algoritmanın eğitim işlemi MATLAB programında gerçekleştirilmiştir. Harris köşe bulma yöntemi için ise OpenCV fonksiyonlarından faydalanılmıştır. Yer istasyonunda çalışan tüm görüntü işleme fonksiyonları Python dilinde yazılmıştır.

1.3.3 Test

Uçuş ve iniş testleri, geliştirilen algoritmaların denenmesi, optimize edilmesi, uçuş esnasında verilerin toplanması ve bu verilerin değerlendrilmesi sürecinin tamamıdır. Algoritma geliştirme süreçlerinin en kısa sürede tamamlanması, tasarım problemlerinin ortaya çıkartılması ve en optimum tasarımın ortaya koyulabilmesi adına çok sayıda test yapılması önemli bir iş kalemidir. Bu çalışmada uçuş ve iniş testleri dış ortamda gerçekleştirilmiştir. Gerçekleştirilen testlerinin tamamında uçuş verileri kayıt altına alınmıştır. Testlerin icra edilmesinde kullanılan donanımlar Başlık 3'da detaylı aktarılmıştır. Uçuş testleri, bulunulan bölgedeki sivil havacılık kurallarına uygun olacak şekilde icra edilmiştir. Hava alanı limitleri göz önünde bulundurularak İHA'nın yükseklik değeri limitlendirilmiş ve testler kontrollü olarak gerçekleştirilmiştir.

2
İnsansız Hava Araçları

2.1 Tanım

İnsansız Hava Aracı, uzak bir yer istasyonundan kontrol edilebilen veya otonom olarak çalışabilen insansız uçaklara verilen addır [8]. Genellikle askeri alanlarda keşif ve saldırı amacıyla kullanılmaktadır. Bunun yanında sivil amaçlı kullanımı oldukça yaygındır. İşleyiş olarak uzaktan kumanda edilebilmekte veya öntanımlı bir uçuş planı ile otonom uçuş gerçekleştirebilmektelerdir. Bu araçlar genellikle normal uçaklar için zor, kirli ve tehlikeli olan görevlerde kullanılmaktadır.



Şekil 2.1 İnsansız Hava Aracı

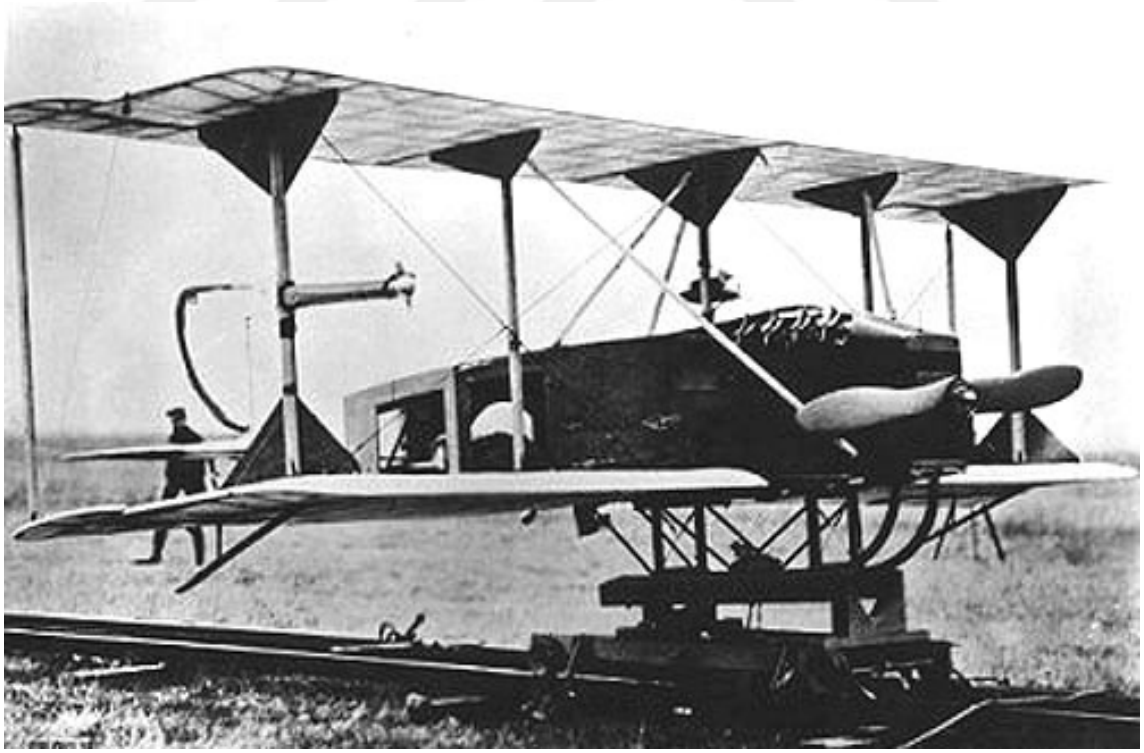
İHA'lar temel olarak üç bileşenden oluşmaktadır. Bunlar; uçağın kendisi, uçaktaki faydalı yük (payload) ve yer kontrol istasyonudur. Uçağın kendisi; uçak gövdesi, uçak motoru (elektrikli veya yakıtlı), uçuş planlama ve kontrol sistemi (PC), navigasyon sistemi (GNSS) ve çarpışma güvenlik sisteminden oluşmaktadır. Uçak faydalı yükü olarak; kameralar, fotoğraf makineleri ve lazer tarayıcılar, termal görüntüleme sistemleri, radarlar, meteorolojik sensörler ve silah sistemleri olabilmektedir [9].

İHA'ların insanlı uçaklara göre avantajları şu şekilde sıralanabilmektedir :

- Otonom ya da bir yer istasyonu aracılığıyla kontrol edilebiliyor olması
- Yaşamsal sistemelere ihtiyaç olmaması
- Kokpit içinde ihtiyaç duyulan hacmin az olması ve mürettebatın getirdiği ağırlık yükünün olmaması
- Maliyetinin az olması
- Manevra ve operasyon kabiliyetinin insan limitleri ile sınırlanmaması
- Düşman tarafından fark edilme ve vurulma ihtimalinin düşük olması

2.2 İHA'ların Tarihi Gelişimi

İHA'ların gelişimi 1909 yılında Elmer Sperry'in cyroskopu icat etmesi birlikte başlamıştır. İlk İHA, Elmer Sperry ve Peter Cooper Hewitt tarafından 1917 yılında tasarlanmış ve adına Hewitt-Sperry Automatic Airplane verilmiştir. Takip eden yıllarda bu uçaktan 7 tanesi prototip olmak üzere toplam 13 adet üretilmiştir. Hewitt-Sperry adlı otomatik uçak, Birinci Dünya Savaşı'nda mühimmat taşıma amaçlı kullanılmıştır.



Şekil 2.2 İlk İHA, Hewitt-Sperry Automatic Airplane

Amerikalı bilim adamı Charles Kettering tarafından geliştirilen ve 'Kettering Bug' adı verilen İHA, ilk uçuşunu 1918 yılında yapmıştır. Kettering Bug, önceden

ayarlanabilen bir cayroskop ve barometresi sayesinde belirli bir rotada uçabilmektedir. Yapılan ilk uçuş testlerinde, başlangıç noktasından 121 kilometre uzağa gidebildiği gözlemlenmiştir.

1935 yılında bir film yıldızı ve model uçak tasarımcısı olan Reginald Denny, ilk ölçekli uzaktan komutalı araç olan RPV(Remote Piloted Vehicle) modelini geliştirmiştir [10].

1939 yılında ise Alman Dr. Fritz Gosslau, biri pilotlu diğeri pilotsuz iki uçaktan oluşan sistem üzerine çalışmalar yapmıştır. Pilotsuz radyo kontrollü İHA, pilotlu uçaktan kontrol edilmektedir. Fritz'in geliştirdiği bu insansız uçak, 1 tona kadar mühimmat taşıyabilmektedir.

1942 yılında TDN-1 adında İHA'ların üretimine başlanmıştır. Bu İHA'lar gerektiğinde güdümlü mermi şeklinde çalışabilmekte, gerektiğinde ise mühimmatı hedefe bırakıp kalış yaptığı üsse geri dönüş sağlayabilmektedir.

12 Ağustos 1944'de B-24 uçağının pilotları ile birlikte Manş Denizi üzerinde inflak etmesi sonucu (pilotsuz uçağı yönlendiren uçaktaki pilotlardan biri, dönemin amerikan başkanı John F. Kennedy'nin büyük oğlu Joseph Kennedy idi) ABD İHA'lar ile ilgili çalışmalarını belirli bir süre rafa kaldırmıştır.

İHA'lar ilk olarak ABD ve İsrail'in Vietnam ve Lübnan Savaşı'larında istihbarat, gözetleme ve keşif amaçları ile kullanılmıştır [11]. 2000 li yılların başında, ABD keşif için kullanıdığı İHA'lara güdümlü tank savar füzleri takarak avcı katil adında taarruz drone konsepti geliştirmiş ve ilk denemesini 2002 Kasım'da Cibuti'de Elkaide üzerinde yapmıştır. İsrail ise 2002'de Filistin'le olan çatışmasında ilk defa silahlı İHA kullanmıştır [12]. 2002 yılında ABD, kabul edilen savaş bölgeleri dışındaki saldırılar için silahlı İHA kullanmaya karar vermiştir [13].

2.3 İHA'ların Kullanım Alanları

İHA'lar sivil ve askeri alanlarda yaygın olarak kullanılmaktadır. Sivil ve askeri alanlarda kullanım yerleri ve amaçları Başlık 2.3.1 ve Başlık 2.3.2'deki gibi aktarılmıştır.

2.3.1 Sivil Kullanım

İHA'lar sivil alanda bir çok amaç için kullanılmaktadır. Bunlar aşağıdaki gibi sıralanabilmektedir:

- Hava fotoğrafçılığı (Film, video ve fotoğraf çekimleri)

- Tarımsal alanlar (Mahsul izleme ve ilaçlama)
- Taşıma (Düşük ağırlığa sahip eşyaların taşınması)
- Sahil Güvenlik (Arama ve kurtarma, sahil ve deniz izleme)
- Çevresel Koruma (Kirlilik takibi ve arazi izleme)
- Trafik ajansları (Karayolu trafiğinin izlenmesi ve kontrolü)
- Gümrük ve vergilendirme (Yasadışı ithalat için gözetleme)
- Elektrik şirketleri (Elektrik şebekelerinin muayenesi)
- Yangın Hizmetleri ve Ormancılık (Yangın algılama ve doğal afet kontrolü)
- Balıkçılık (Balıkçılık koruması)
- Gaz ve petrol tedarik şirketleri (Arazi etüdü ve boru hattı güvenliği)
- Bilgi servisleri (Haber bilgilendirmeleri, anlık resimler ve video paylaşım)
- Cankurtaran kurumları (Olay soruşturması, rehberlik ve kontrol)
- Yerel yönetimler (Araştırma ve kurtarma, teftiş ve kontrol)
- Meteorolojik hizmetler (Hava durumu tahminleri için atmosferin örneklenmesi ve analizi)
- Petrol şirketleri (Boru hattı güvenliğinin sağlanması)
- Polis makamları (Kayıp kişilerin bulunması, olay gözetimi ve güvenliğin sağlanması)
- Nehir yetkilileri (Su sahası ve seviye izleme, sel ve kirlilik kontrolü)
- Araştırma kuruluşları (Coğrafi, jeolojik ve arkeolojik araştırmalar)
- Su panoları (Rezervuar ve boru hattı izleme)

2.3.2 Askeri Kullanım

İHA'lar hava kuvvetleri, deniz kuvvetleri ve kara kuvvetleri olmak üzere tüm askeri bölgelerde yaygın olarak kullanılmaktadır. Bu araçlar genellikle normal savaş uçakları için zor, kirli ve tehlikeli olan görevlerde kullanılırmaktadır.

2.3.2.1 Deniz Kuvvetlerinde Kullanım Alanları

İHA'ların deniz kuvvetlerinde başlıca kullanım alanları aşağıdaki gibidir:

- Düşman filolarını gözetleme
- Yapay izlerle düşman füzelerini aldatmak

- Elektronik zeka
- Radyo sinyallerinin transferi
- Limanların deniz saldırılarına karşı korunması
- Sonar şamandıraları gibi insan ihtiyacı duyulan kompanentlerin yerleştirilmesi ve izlenmesi

2.3.2.2 Kara Kuvvetlerinde Kullanım Alanları

İHA'ların kara kuvvetlerinde başlıca kullanım alanları aşağıdaki gibidir:

- Keşif ve gözetim (düşmana ait cephe bilgilerini toplayan araçlar)
- Düşman faaliyetlerinin gözetimi
- Nükleer, biyolojik veya kimyasal kirlenmenin izlenmesi
- Elektronik zeka
- Atış yapılacak hedefin belirlemesi ve izleme
- Kara mayınlarının yerinin tespiti ve imhası

2.3.2.3 Hava Kuvvetlerinde Kullanım Alanları

İHA'ların hava kuvvetlerinde başlıca kullanım alanları aşağıdaki gibidir:

- Çatışma (Yüksek riskli görevlerde kullanılan saldırı kapasitesine sahip araçlar vasıtasıyla)
- Hedef ve yem (düşman hava savunma veya savaş uçaklarına karşı yem olarak kullanılarak hedef belirlemede yardımcı olan araçlardır)
- Lojistik (Kargo ve lojistik destek amaçlı araçlar araştırma ve geliştirme)
- Uzun menzilli, yüksek irtifa gözetimi
- Radar sistemlerinde frekans bozma ve imhası
- Elektronik zeka
- Havaalanı üssü güvenliği
- Havaalanı hasar değerlendirmesi
- Patlamamış bombaların imha edilmesi

2.4 İHA'ların Sınıflandırılması

İHA'lar uçuş menzili ve irtifalarına göre sınıflandırılmaktadır. Bu sınıflar aşağıdaki gibidir:

- Handheld sınıfı (Elle taşınabilen, 600 metre irtifa ve ortalama 2 km menzil)
- Close sınıfı (1500 metre irtifa ve ortalama 10 km menzil)
- NATO sınıfı (3000 metre irtifa ve ortalama 50 km menzil)
- Taktiksel (5500 metre irtifa ve ortalama 160 km menzil)
- MALE sınıfı (9000 metre irtifa ve ortalama 200 km menzil)
- HALE sınıfı (9000 metre irtifa ve belli olmayan menzil)
- HYPERSONIC sınıfı (1-5 Mach arası hız, 15000 metre irtifa ve ortalama 200 km menzil)
- ORBITAL sınıfı (Alçak dünya yörüngesinde 25 Mach hıza ulaşabilmektedir)
- CIS Lunar sınıfı (Dünya-Ay uçuşu)

İHA'lar aynı zamanda kanat yapılarına görede kendi içerisinde Sabit Kanatlı ve Döner Kanatlı İHA'lar olarak ikiye ayrılmaktadır.

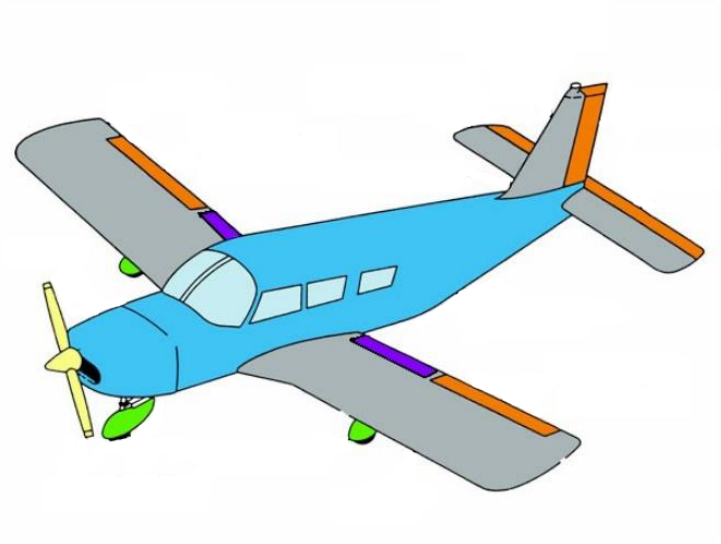
2.4.1 Sabit Kanatlı İHA'lar

Sabit kanatlı İHA'lar, sabit kanata sahip olan ve İHA'nın ileri hava hızının oluşturduğu yükselme kuvvetiyle uçuş yapabilen uçaklara verilen addır. Bu hava hızı, genellikle bir içten yanmalı motor veya elektrik motoru tarafından döndürülen bir pervane vasıtasıyla üretilir.

Sabit kanatlı İHA'lar, iniş ve kalkış için piste ihtiyaç duymaktadır. Bu tür İHA'lar genellikle yüksek dayanıklılığa sahiptir ve yüksek hızlarda uçabilmektedirler Şekil 2.1.

2.4.2 Döner Kanatlı İHA'lar

Döner kanatlı İHA'lar, dikey bir şafta bağlı olarak dönen pallerin (kanatların) ürettiği kaldırma kuvvetiyle yerçekimini yenen ve uçuşunu gerçekleştiren hava araçlarıdır. Bu tür İHA'lar dikey iniş kalkış(VTOL) yapabilmektedir. Bu sayede iniş ve kalkış için genellikle büyük bir piste ihtiyaç duymazlar. Bu yeteneğinden dolayı döner kanatlı İHA'lar özellikle sivil uygulamalarda birçok robotik görevde rol almaktadır [14].



Şekil 2.3 Sabit Kanatlı İHA

Döner kanatlı İHA'lar; küçük ebatları, manevra kabiliyetlerinin yüksek olması, iç ve dış mekanlarda uçabilir olması gibi sebeplerden dolayı günümüzde yaygın olarak kullanılmaktadır. Ayrıca, elektronik alanındaki gelişmelerin hızla artmasıya İHA alt bileşenlerinin üretimi kolaylaştırmış ve sektörün büyümesine katkı sağlamıştır [15]. Döner kanatlı İHA'ların 2, 3, 4, 6 ve 8 rotorlu modelleri yaygın olarak kullanılmaktadır. Bu çalışmada dört rotorlu döner kanatlı İHA kullanılmıştır.

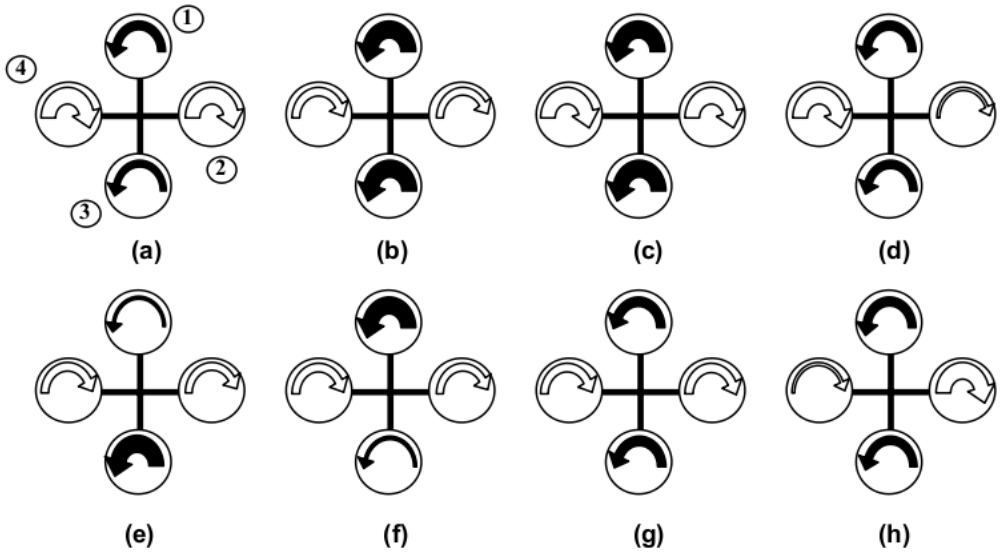
Dört rotorlu sabit kanatlı İHA'larda çarpraz lokalize edilmiş dört adet motor ve bu motorlara entegre edilmiş pervaneleri bulunmaktadır. BLDC motorlar; verimlerinin yüksek olması, hacim ve ağırlıklarının düşük olması, yüksek tork elde edilebilmesi ve sessiz çalışmaları gibi avantajlarından dolayı bu İHA'larda yaygın olarak kullanılmaktadır [16]. Uçuş esanasındaki motor hızları, motorun güç isterine ve motor tasarımına bağlı olarak değişmekte ve 10350 RPM ile 41400 RPM aralığında yer almaktadır [17]. İHA üzerindeki motorların hızlarını kontrol etmek için her bir motora ait ESC (elektronik hız kontrolcü) donanımları bulunmaktadır. Bu ESC'ler üzerinde, motor hızını ve akımını kontrol etmek için bir adet işlemci; gerekli gerilimi motor fazlarına uygulamak için ise güç devresi yer almaktadır [18].

Döner kanatlı İHA'ların uçuş esnasında aerodinamik etkilere karşı dengede kalabilmesi ve hareketini kararlı şekilde devam ettirebilmesi için ön ve arka motorları saat yönünün tersine dönmekte, sağ ve sol motorları ise saat yönünde dönmektedir. Cayroskopik etkilerin ihmal edildiği ve tüm motorların rotor hızları aynı RPM de döndüğü durumda İHA havalanmaktadır. Havalanmanın düzgün bir şekilde gerçekleşebilmesi için tüm motorlardan üretilen itki kuvveti aynı olmalıdır. Motorlardan herhangi birinin itkisindeki en ufak bir değişim hava aracının ani bir



Şekil 2.4 Döner Kanatlı; 3, 4, 6 ve 8 Rotorlu İHA

şekilde ters düz olmasına sebep olabilir. Tek bir motorun hızını değiştirerek taşıma kuvveti değiştirilebilmekte; yatay ve dikey hareketler bu şekilde sağlanmaktadır [19].



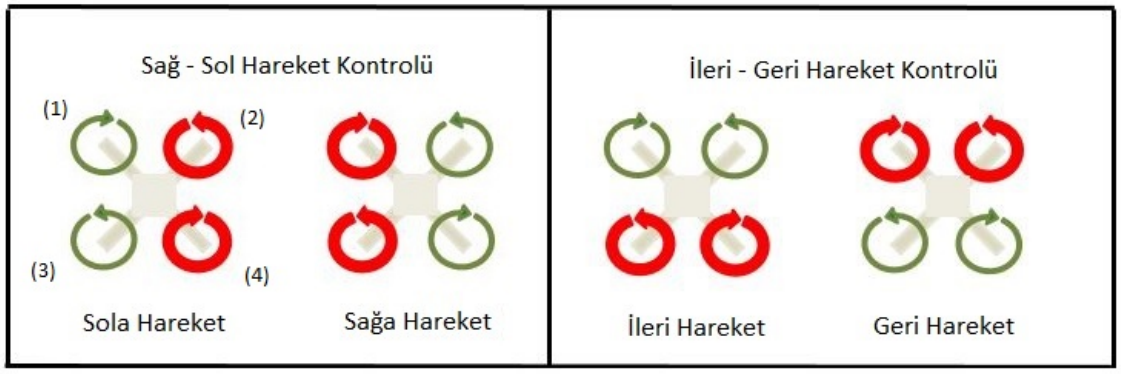
Şekil 2.5 İHA, Dönüş Yönlerine Göre Hareket Düzlemi

Şekil 2.5'de İHA'nın motor dönüş yönleri ve hızları gösterilmektedir. Ön ve arka motorlar (1 ve 3) saat yönünün tersine, sağ ve sol motorlar (2 ve 4) saat yönünde

dönmektedir. Kalın oklar hızlı dönüşü, ince oklar yavaş dönüşü temsil etmektedir.

Şekil 2.5'de gösterilen 8 farklı dönüş konfigürasyonu için İHA'nın hareket durumu aşağıdaki gibi aktarılmaktadır:

- (a) İHA, kendi ekseni etrafında saat yönünün tersi yönde döner
- (b) İHA, kendi ekseni etrafında saat yönünde döner
- (c) İHA havalanır (take-off)
- (d) İHA, sapma ekseni etrafında saat yönünde döner
- (e) İHA, yunuslama ekseni etrafında saat yönünün tersi yönde döner
- (f) İHA, yunuslama ekseni etrafında saat yönünde döner
- (g) İHA, alçalır (land)
- (h) İHA, sapma ekseni etrafında saat yönünün tersi yönde döner



Şekil 2.6 İHA, İleri-Geri ve Sağ-Sol Hareket

İHA'nın doğrusal hareketini sağlaması için aynı taraftaki motorların eşlenik çalışması gerekmektedir. Şekil 2.6'de görüldüğü gibi; 2 ve 4 numarlı motorlar daha hızlı döndüğünde İHA sol yönde, 1 ve 3 numarlı motorlar daha hızlı döndüğünde İHA sağ yönde hareket etmektedir. Aynı şekilde 3 ve 4 numaralı motorlar daha hızlı döndüğünde İHA ileri yönde, 1 ve 3 numaralı motorlar daha hızlı döndüğünde İHA geri yönde hareket etmektedir.

3
Çalışmada Kullanılan Donanım ve Yazılımlar

3.1 Kullanılan Donanımlar

3.1.1 İnsansız Hava Aracı - 3DR Solo

Algoritma testleri için Şekil 3.1'deki 3D Robotics firmasının ürünü olan döner kanatlı ve dört rotorlu 3DR Solo marka İHA kullanılmıştır. Bu İHA'nın özellikleri Tablo 3.1'deki gibi sunulmuştur [20]. Hem İHA'da hemde uzaktan kumandasında ARM tabanlı işlemci bulunmaktadır ve bu işlemcilerde Linux tabanlı gerçek zamanlı işletim sistemi çalışmaktadır. İHA'da, üzerinde yer alan kameranın 3 eksende hareket kontrolünü sağlama amacıyla gimbal (kamera tutacağı) bulunmaktadır. Bu gimbalin 3 eksende yaptığı hareketler yer istasyonundan kontrol edilebilmektedir.



Şekil 3.1 3DR Solo İnsansız Hava Aracı

3D Robotics firmasının ürettiği İHA'larda yer alan Pixhawk 2 otopilot kartının piyasada Green Cube ve Black Cube olmak üzere iki farklı versiyonu bulunmaktadır. Projede kullanılan İHA'nın 2016 yılı üretimi olmasından dolayı içerisinde eski versiyon olan Black Cube (Şekil 3.2) otopilot kartı yer almaktadır. Bu Otopilot kartı içersinde ArduPilot Copter v1.5.4 otopilot yazılımı çalışmaktadır. ArduPilot Copter, açık

Tablo 3.1 3DR Solo İHA Özellikleri

Komponent	**Açıklama**
Boyutlar	45.7 x 45.7 x 25.4 mm
Ağırlık	1.5 kg
Yük Kapasitesi	0.42 kg
Otopiloy Donanımı	3DR Pixhawk 2 Black
Maksimum Hız	25.5 m/s
Maksimum Yükseklik	100 m
Menzil	800 m
Uçuş Süresi	20-25 dk
Batarya	Lityum Polimer, 5200 mAh, 14.8V
Haberleşme	Wifi, 2.4GHz
Motor	BLDC Motor, 880 kV
Çalışma Sıcaklığı	0 °C - 45 °C
Kamera	GoPro HERO4 ve Gimbal

kaynak kodlu bir otopilot yazılımıdır. PX4 tabanlı otopilot kartlarında bu yazılım kullanılabilmetedir.



Şekil 3.2 Pixhawk 2 Black Otopilot Kartı

ArduPilot Copter v1.5.4, Black Cube otopilot kartı ile kullanılabilir olan en yeni versiyon otopilot yazılımdır. Bu yazılımın İHA'ya yüklenebilmesi için hem İHA tarafında hemde uzaktan kumanda tarafında gerekli arayüz güncellemeleri yapılmış ve tüm foksiyonlar kullanabilir hale getirilmiştir. 3DR Solo'nun yazlım güncellemesi aşağıda belirtilen 5 farklı şekilde yapılabilmektedir:

- Bilgisayar ile İHA'ya root üzerinden bağlanarak
- MissionPlanner yazılımı üzerinden
- QgroundControl yazılımı üzerinden

- 3DR Solo IOS uygulaması üzerinden
- Solex Android uygulaması üzerinden

Bilgisayar ile root üzerinden bağlantı kurularak yazılımın güncellenmesi önerilmemektedir.

3.1.2 Yer İstasyonu

Görüntü işleme algoritmasının koştuğu ve İHA'nın hareket kontrolünün yapıldığı yer istasyonu, DELL marka bir dizüstü bilgisayardır. Bu bilgisayar üzerinde açık kaynak kodlu işletim sistemi olan Ubuntu 16.04 bulunmaktadır. Yer istasyonu olarak kullanılan dizüstü bilgisayarın özellikleri Tablo 3.2'deki gibi sunulmuştur:

Tablo 3.2 Yer İstasyonu Özellikleri

Komponent	**Açıklama**
İşletim Sistemi	Ubuntu 16.04
İşlemci	Intel Core İ7 - 8550U
sistem Belleği (RAM)	8 GB
Çekirdek Sayısı	4 Çekirdek
Ekran Kartı	Radeon R5 M530
Hafıza Alanı	1 TB
Çalışma Sıcaklığı	−40 °C - 70 °C

3.2 Kullanılan Yazılımlar

Bu çalışma içerisinde geliştirilen tüm algoritmalar Python dilinde yazılmıştır. Yazılım geliştirme IDE'si olarak PyCharm yazlımı kullanılmıştır. İniş yapılacak olan hedef platformun görüntü işleme yöntemi ile bulunmasında OpenCV ve MATLAB yazılımları kullanılmıştır. Yer istasyonu ile İHA arasında veri alışverişi MAVLink haberleşme protokolü üzerinden sağlanmıştır. MAVLink haberleşme komutları ise Dronekit kütüphanesi kullanılarak gönderilmiştir.

3.2.1 Python

Python, ilk sürümü Guido van Rossum tarafından 1991'de ortaya konulmuş; nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir [21]. Yorumlanan ve dinamik bir dil olan Python, esas olarak nesne tabanlı programlama yaklaşımlarını ve belli bir oranda da fonksiyonel programlamayı desteklemektedir.

Python Yazılım Vakfı, Python'ın ana gerçekleşimi olan C dili gerçekleşimini özgür ve açık kaynak kod mantığı altında yürütmektedir ve Python'ın fikirsel haklarını korumaktadır. Günümüzde Python, görece kolaylığı ve sahip olduğu geniş standart kütüphane sayesinde oldukça popülerleşmiş ve büyük kurumların da arasında olduğu yaygın bir kullanıcı kitlesine ulaşmıştır [22].

Python; Unix , Linux, Mac, Windows, Amiga, Symbian gibi işletim sistemlerinde rahatlıkla kullanılabilmektedir. Sistem programlama, kullanıcı arayüzü oluşturma, ağ programlama, web programlama, uygulama ve veritabanı yazılımı programlama gibi birçok alanda yazılım geliştirilebilmektedir.

Bu çalışmada, diğer yazılım platformları ile etkileşimin versiyon açısından uygunluğu amaçlanarak Python 3.5 kullanılmıştır. Tüm algoritma geliştirme çalışmaları ve kullanıcı arayüz çalışmaları Python dili ile gerçekleştirilmiştir.

3.2.2 Dronekit

Dronekit, 3D Robotics firmasının geliştirdiği ve ArduPilot yazılımlarına sahip cihazlarla düşük gecikmeli ve kolay haberleşmeye olanak sağlayan Python ve Android tabanlı bir yazılımdır. İHA ile haberleşme MAVLink haberleşme protokolü üzerinden yapılmaktadır. Dronekit'te MAVLink haberleşme protokolü üzerinden İHA ile iletişim kurmakta, durum ve parametre bilgilerine programlı erişim sağlamakta ve hem görev yönetimi hem de araç hareketi işlemleri üzerinde doğrudan kontrol sağlamaktadır [23]. Linux, Mac OS veya Windows işletim sistemleri üzerinde çalışmaktadır.

Dronekit yazılımı ile aşağıdaki işlemler yapılabilmektedir:

- Bir veya birden fazla İHA ile bağlantı kurulabilmektedir
- Uçuş esnasında İHA'dan uçuş bilgileri alabilmekte ve İHA'ya kontrol komutları gönderilebilmektedir
- İHA, belirli bir konuma gönderilebilmektedir
- Uzaktan kumanda ayarları etkisizleştirilebilmektedir
- Geliştirilen kodları test etmek için multicopter simülatörleri (SITL) üzerinden testler yapılabilmektedir
- Bir yer istasyonunda veya araç üstü bir bilgisayarda çalışabilmektedir

Bu çalışmada, İHA ile haberleşmenin sağlanması ve veri alışverişinin etkin bir şekilde sağlanması için Dronekit V 2.9.0 yazılımı kullanılmıştır. Dronekit'in desteklemediği

haberleşme komutları direkt olarak MAVLink haberleşme protokolü üzerinden Python kütüphanesi kullanılarak gönderilmiştir.

3.2.3 Matlab

MATLAB (MATrix LABratory), New Mexico Üniversitesi'de matematik profesörü olan Cleve Moler tarafından geliştirilmiştir. Çok paradigmalı sayısal hesaplama yazılımı ve dördüncü nesil teknik hesaplama dilidir. Fortran ve C dillerinde yazılmaya başlanmıştır. Aynı zamanda birden fazla programa dili ile birlikte etkileşimli çalışabilmektedir [24].

Matlab'in uygulama alanları aşağıdaki gibidir [25]:

- Teknik Hesaplama
- Görüntü İşleme
- Sinyal İşleme ve Haberleşme
- Modelleme ve Kontrol
- Gömülü Yazılım
- Test ve Ölçüm
- Otonom Sistemler
- Hesaplamalı Finans
- Otomotiv ve Havacılık

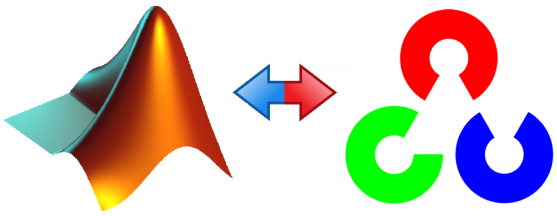
Matlab, içerisinde yer alan 90'dan fazla araç kutusu ile farklı çalışma alanlarında kullanılmak üzere hazır fonksiyonlar sunmaktadır. Bunlardan birtanesi de Computer Vision (CV) araç kutusudur.

Bu çalışmada, hedef platformun görüntü işleme ile tespit edilmesinde kullanılan Haar Cascade sınıflandırma algoritmanın eğitimi için Matlab programı kullanılmıştır. Matlab içerisinde yer alan Computer Vision araç kutusu ve Image Processing araç kutusu fonksiyonlarından faydalanılmıştır. Matlab'in R2018b sürümü içerisinde yer alan Matlab v9.5, Computer Vision araç kutusu v8.2 ve Image Processing araç kutusu v10.3 versiyonları kullanılmıştır.

3.2.4 OpenCV

OpenCV (Open Source Computer Vision), 47.000'den fazla kullanıcı tarafından yaygın olarak kullanılan ve 14 milyonu aşkın indirme sayısına sahip, açık kaynak

kodlu 'Bilgisayarlı Görü' kütüphanesidir [26]. 1999 yılında İntel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmiştir [27]. İlk olarak C programlama dili ile geliştirilmeye başlanmış, ardından çok sayıda OpenCV algoritması C++ dili ile geliştirilmiştir. OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir [26]. OpenCV, karmaşık görüntü işleme algoritmalarının geliştirme sürecini kolaylaştırmakta ve kullanımı kolay bir görüntü işleme altyapısı sağlamaktadır.



Şekil 3.3 Matlab-OpenCV

OpenCV kütüphanesi; tıbbi görüntüleme, fabrika ürün incelemesi, güvenlik, kamera kalibrasyonu ve robotik gibi bir çok alanda kullanılabilir 1000'den fazla fonksiyon içermektedir. Görüntü işleme ve makine öğrenmesi iç içe konular olmasından dolayı genel amaçlı bir Makine Öğrenimi Kütüphanesi (MLL) içermektedir [26].

OpenCV aynı zamanda Matlab programı ile etkileşimli olarak kullanılabilmektedir. Bu etkileşim Matlab-OpenCV Interface modülü ile birlikte kurulabilmekte ve Matlab ortamında OpenCV fonksiyonları çalıştırılabilmektedir. Aynı zamanda Matlab dilinde oluşturulan görüntü işleme algoritmalarıda C++ diline çevirilerek OpenCV kütüphanesi ile birlikte kullanılabilmektedir.

Bu çalışmada, görüntü işleme yöntemi ile hedef platform tespiti için Python dilinde OpenCV fonksiyonlarından ve OpenCV altyapısından faydalanılmıştır. OpenCV 3.4.0 versiyonu kullanılmış olup, Matlab ortamında eğitilmiş görüntü işleme algoritmaları OpenCV fonksiyonları ile birlikte kullanılmıştır.

3.2.5 PyCharm

PyCharm, Jetbrains firması tarafından geliştirilen, birden fazla işletim sisteminde çalışabilme özelliğine sahip bir Python geliştirme ortamıdır (IDE). Kod analizleri, grafiksel hata ayıklamacısı ve web arayüzleri geliştirmeye olanak sağlamaktadır. Windows, OS X ve Linux işletim sistemleri üzerinde çalışabilmektdir [28].

Bu çalışmada, Python dili ile geliştirilen tüm algoritmalar PyCharm IDE'si üzerinde gerçekleştirilmiştir.



Şekil 3.4 PyCharm

4
Haberleşme Altyapısı ve Görüntü Transferi

4.1 İHA ve Yer istasyonu Arasındaki Haberleşme Altyapısı

Bu çalışmada; İHA'nın iniş yapacağı hedef platformu tespit eden görüntü işleme algoritmalarının çalıştırıldığı, İHA'nın uçuş verilerinin gözlemlendiği ve İHA'ya hareket komutlarının gönderildiği bir yer istasyonu kullanılmaktadır. Bu yer istasyonu, tipik olarak İHA ile kablosuz telemetri üzerinden haberleşebilen ve bu haberleşme için içerisinde gerekli yazılımları bulunduran bir yer bilgisayarıdır. Bu yer istasyonu, İHA'nın uçuş verileri ve performans parametrelerini gerçek zamanlı olarak takip etmeye olanak sağladığından dolayı sanal kokpit işlevi görmektedir.

Yer istasyonu ile 3DR Solo İHA arasında haberleşme sağlayabilecek çok sayıda açık kaynak kodlu yazılım bulunmaktadır. Bilgisayar tabanlı yazılımlar şu şekilde sıralanabilmektedir:

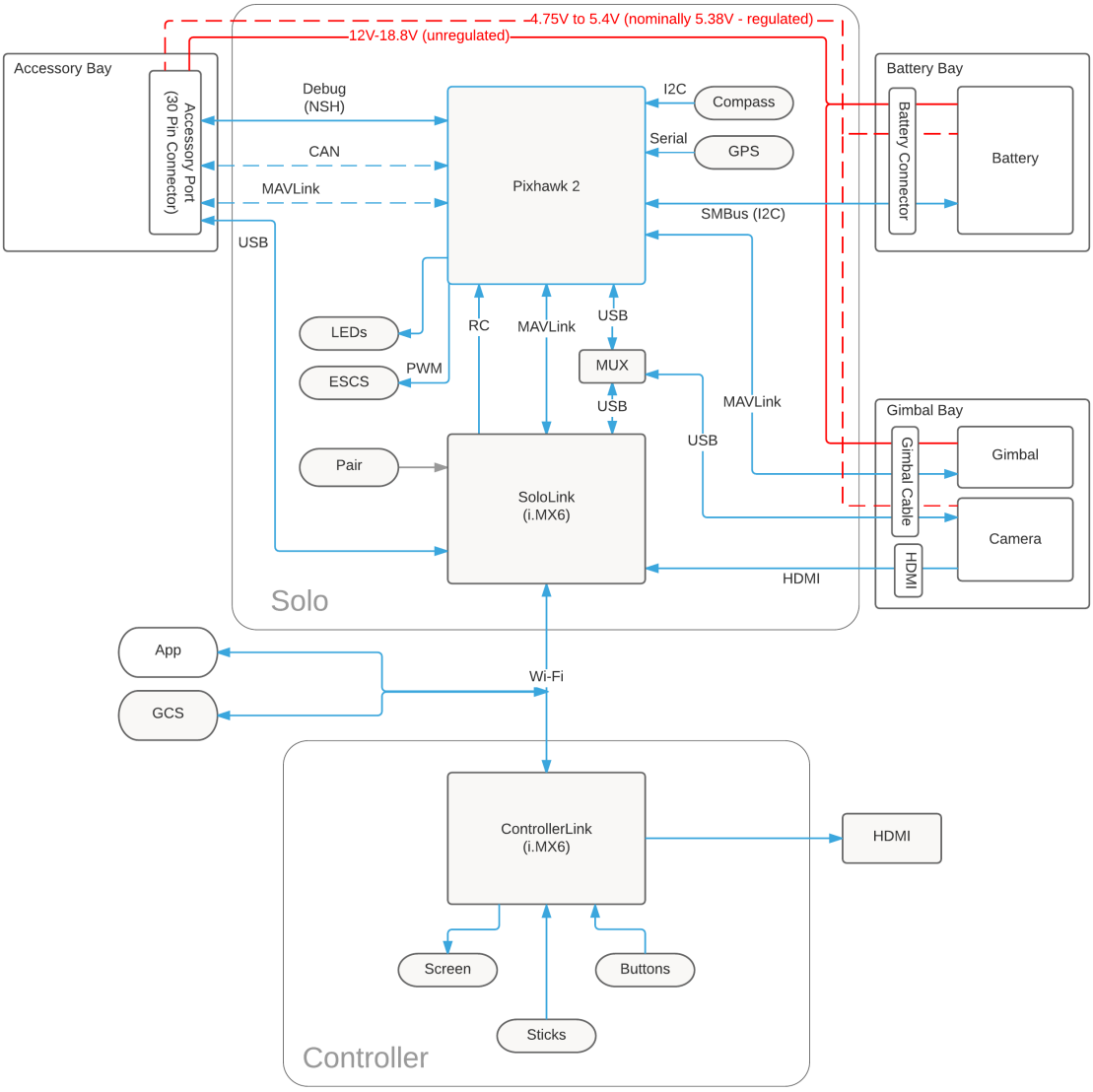
- Mission Planner
- APM Planner 2
- MAVProxy
- QGroundControl
- UgCS

Telefon ve tablet uygulamaları ise aşağıdaki gibidir:

- Tower (DroidPlanner 3)
- MAVPilot
- AndroPilot
- SidePilot
- 3DR Solo App
- Solex

Bu çalımada İHA ile haberleşme sağlamak için yukarıda aktıralan yazılım veya uygulamalar kulanılmamış olup Başlık 3.2.2'de açıklanan Dronekit kütüphanesi ve Python programlama dili ile uygulama özelinde arayüz geliştirilmiş ve bu arayüz üzerinden işlemler gerçekleştirilmiştir.

Şekil 4.1'de gösterildiği gibi İHA kontrolcüsü olan Pixhawk, MAVLink telemetri protokolü üzerinden hem içerisideki Linux işletim sistemine sahip anabilgisayarla hemde uzaktan kumada ve yer istasyonu yazılımlarıyla haberleşmektedir. İHA'daki Linux tabalı işlemci, yüksek seviye (high-level) işlemleri kontrol etmektedir.



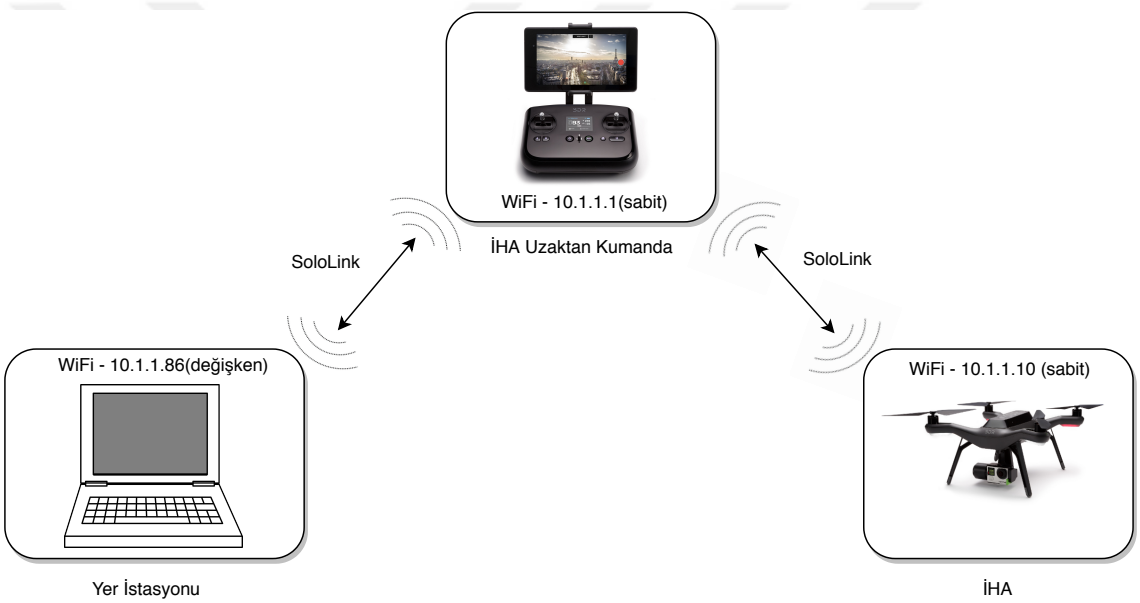
Şekil 4.1 İHA Basitleştirilmiş Sistem Diyagramı

MAVLink, 2009'dan beri küçük boyutlu İHA'larda yer istasyonu ile haberleşmede kullanılan bir haberleşme protokolüdür. MAVLink ile veriler belirli bir yapıya uygun olarak paketlenmekte ve yer istasyonuna gönderilmektedir. Bu veri paketinin

uzunluğu 17 byte tır. 17 byte'lık veri paketinin 6 byte'ı header, 9 byte'ı veri ve 2 byte'ı ise doğrulama(checksum) verisidir. MAVlink; Python, C, C++, Java gibi yaygın kullanılan programlama dillerine destek vermektedir.[29]

MAVLink telemetri protokolü (İHA ile bağlantı kurmak için kullanılan) 14550 numaralı UDP portunu kullanmaktadır. Yer istasyonunda yazılan kod ile Wifi ağına bağlı herhangi bir harici aygıt, UDP istemcisi olarak udpin:0.0.0.0:14550'e veya İHA'nın terminalinden İHA'ya bağlanabilmektedir. Böylece, WiFi üzerinden İHA ile yer istasyonu arasında haberleşme başlatılmaktadır.[20].

Şekil 4.2'de gösterildiği gibi bu bağlantı şemasında sunucu (server) uzaktan kumandadır. Yer istasyonu ve İHA, uzaktan kumandaya istemci (client) olarak bağlanmaktadır.



Şekil 4.2 İHA-Yer İstasyonu Haberleşme Yapısı

4.2 İHA'dan Yer İstasyonuna Görüntü Aktarımı

Kullanılan 3DR Solo marka İHA'nın üzerinde Şekil 4.3'deki GoPro Hero4 kamerası bulunmaktadır. Görüntü işleme için bu kameradan alınan veriler kullanılmıştır. GoPro Hero4 kamerasının çözünürlükleri aşağıdaki Tablo 4.1'deki gibidir [30]. Tablo 4.1'deki çözünürlük değerleri, kameranın aldığı görüntü ve videoların kamera dahilinde bulunan SD Kart'a kayıt edildiği durum için geçerlidir. Görüntülerin gerçek zamanlı transfer edilmesi durumunda haberleşme hızlarından kaynaklı görüntü çözünürlükleri değişebilmektedir.

GoPro Hero4 kameradan yer istasyonuna görüntü aktarımı için iki farklı seçenek



Şekil 4.3 GoPro Hero4 Black Kamera

bulunmaktadır. Bu yöntemlerden ilki, GoPro Hero4 kameranın kendi WiFi ağı üzerinden yer istasyonuna görüntüyü aktarmaktır. İkinci yöntem ise HDMI üzerinden İHA bilgisayarına gönderilen görüntüyü yine İHA'nın kablosuz bağlantısı üzerinden yer istasyonuna göndermektir. Bu yöntemlerin detayları Başlık 4.2.1 ve Başlık 4.2.2'de aktarılmıştır.

4.2.1 GoPro Hero4 Kamerasının Kablosuz Ağı ile Yer İstasyonuna Görüntü Aktarımı

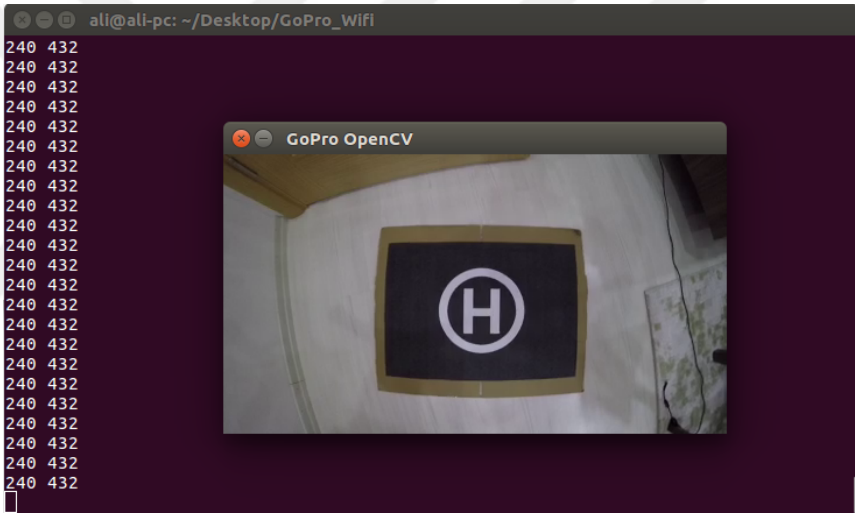
GoPro Hero4 kameranın kablosuz ağı üzerinden görüntü transferi için gerekli altapı hazırlanarak ilgili testler gerçekleştirilmiştir. Kablosuz ağ üzerinden görüntü transferi yapıldığı durumda yer istasyonuna gönderilen görüntünün 432x240 piksel olduğu görülmüştür. Bu çözünürlük değeri, hedeflenen yüksekliklerde iniş yapılacak olan platformun bulunması için yeterli değerlidir. Ayrıca bu yöntemle, kameradan görüntü almak ve İHA'yı kontrol edebilmek için yer istasyonundan iki ayrı WiFi ağ bağlantısı kurmak gerekmektedir. Tek bir bilgisayar üzerinden bu işlemin gerçekleştirilebilmesi için ekstra donanımsal ihtiyaçlar söz konusu olmaktadır. Belirtilen sebeplerden dolayı bu yöntem kullanılmamıştır.

4.2.2 İHA'dan Yer İstasyonuna Görüntü Aktarımı

İkinci yöntem olarak GoPro Hero4 kameradan İHA bilgisayarına gönderilen görüntüyü yine İHA'nın kablosuz bağlantısı üzerinden yer istasyonuna göndermek için gerkli altyapı kurulmuş ve görüntü aktarımı başarılı bir şekilde gerçekleştirilmiştir. Bu yöntemle İHA'dan yer istasyonuna gönderilen görüntüler 1280x720 piksel boyutunda

Tablo 4.1 GoPro Hero4 Black Çözünürlüğü

Video Çözünürlüğü	**Frame Per Second**	**Görüş açısı**	**Ekran Çözünürlüğü**
4K	30,25,24	Geniş Görüş	3840x2160
2.7K	60, 50, 48, 30, 25, 24	Geniş, Orta ve Doğrusal Görüş	2704x1520
2.7K 4:3	30, 25	Geniş Görüş	2704x1520
1440p	80, 60, 50, 48, 30, 25, 24	Geniş Görüş	1920x1440
1080p	120, 90, 60, 50, 48, 30, 24, 24	Geniş, Orta ve Doğrusal Görüş	1920x1080
720p	240, 120, 60, 50, 30, 25	Geniş, Orta ve Doğrusal Görüş	1280x720

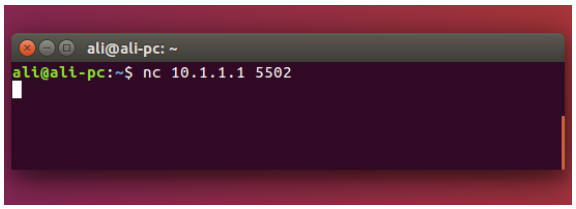


Şekil 4.4 GoPro Wifi Üzerinden Alınan Görüntü

olmaktadır.

İHA kontrolcüsü, sadece bağlı olduğu bir cihaz olduğunda ilgili port üzerinden görüntü transferi yapmaktadır. Yer istasyonundan İHA ile bağlantı kurulduğu anda, İHA'dan yer istasyonuna görüntü aktarımı başlamaktadır. Bu sebeple öncelikli olarak yer istasyonundaki terminal üzerinden İHA ile bağlantı kurulmuştur. Bu bağlantının sağlanması için NetCat aracı kullanılmıştır. NetCat, TCP/IP üzerinden veri gönderme veya almaya yarayan açık kaynak kodlu bir araçtır. Bu araç sayesinde herhangi bir port dinlenebilmekte, port tarama yapılabilmekte, bağlantı oluşturulabilmekte ve dosya transferi yapılabilmektedir [31].

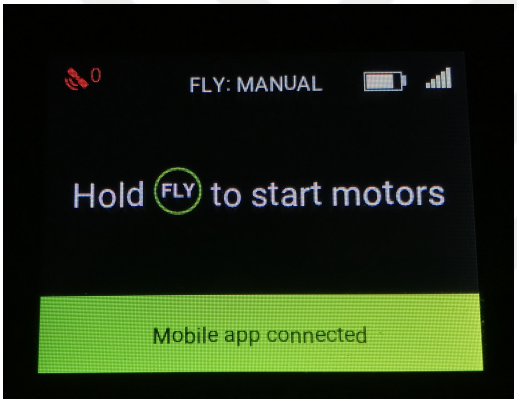
Şekil 4.2'da gösterildiği gibi sistemde sunucu (master) İHA'nın uzaktan kumandasıdır.



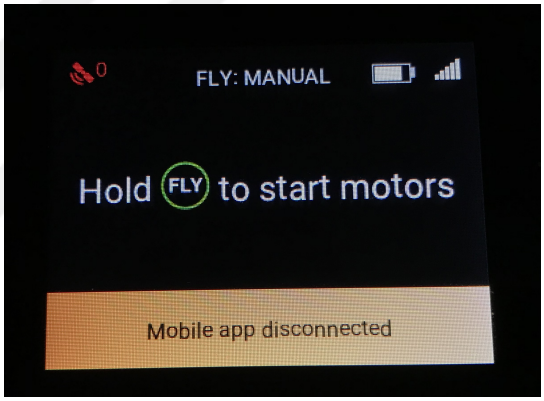
Şekil 4.5 NetCat Aracı ile Yer İstasyonundan İHA'ya Bağlantı

Yer istasyonu ve İHA, bu sunucuya wifi üzerinden istemci olarak bağlanmaktadır. İHA uzaktan kumandasının IP adresi 10.1.1.1'dir. Yer istasyonu üzeriden 5502 portu kullanılarak bu IP ile bağlantı başlatılabilmektedir. Aynı zamanda, 10.1.1.10 IP adresi kullanılarak yer istasyonundan direkt İHA ile bağlantı kurulabilmektedir [32].

NetCat ile bağlantı başlatıldığında, İHA uzaktan kumandası üzerinde Şekil 4.6'deki gibi bağlantının sağlandığını gösteren uyarı çıkmaktadır. Bağlantı kesindiğinde ise Şekil 4.7'deki gibi bağlantının kesildiğini gösteren uyarı çıkmaktadır.



Şekil 4.6 İHA Kumandası Bağlantı Sağlandı Uyarısı



Şekil 4.7 İHA Kumandası Bağlantı Kesildi Uyarısı

Yer istasyonu ile İHA kumandası arasında bağlantı kurulduğu andan itibaren İHA'dan yer istasyonuna 5600 portu üzerinden H264 formatında sıkıştırılmış görüntüler gönderilmektedir. Gönderilen bu görüntünlerin yer istasyonundan alınabilmesi için çoklu ortam haberleşme oturumunun başlatılması ve görüntü akışının sağlanması gerekmektedir. Yer istasyonu üzerinden çoklu ortam haberleşme oturumunun başlatılması için SDP protokolü kullanılmıştır. SDP; oturum duyurusu, daveti ve çoklu ortam haberleşme oturumlarının başlatılması amacıyla tasarlanmış bir protokoldür. SDP protokolünün kullanımı için aşağıdaki tanımlamaların yapılması gerekmektedir:

- İletim ortamı tipi (video, ses, vb.)
- Ulaşım katmanı protokolü tipi (RTP/UDP, H320, vb.)

- İletim formatı (H264 video, MPEG video, vb.)
- İletilen mesajı alabilmek için gerekli bilgiler (IP adresi, port, format, vb.)

İHA ile yer istasyonu arasındaki haberleşme UDP üzerinden kurulduğu için SDP protokolü içerisinde tanımlanan ulaşım katmanı protokolü RTP seçilmiştir.

SDP protokolü için kullanılan örnek kod dizini aşağıdaki gibidir:

```
c=IN IP4 10.1.1.1
m=video 5600 RTP/AVP 96
a=rtpmap:96 H264/90000
t=0 0
```

SDP protokolü ile görüntünün yer istasyonuna alınması ve görsellenebilmesi için bir multimedya çerçeve yazılımı kullanılması gerekmektedir. Bu multimedya çerçeve yazılımları, SDP dosyasını kullanarak görüntü alma işlemini gerçekleştirmekte ve görsellemektedir. Alınan görüntünün kalitesi ve görüntünün alınmasından itibaren yer istasyonuna gönderilmesine kadar geçen süre, kullanılan multimedya çerçeve programının performansına göre değişmektedir. Bu çalışmada, VLC, FFMPEG ve GStreamer multimedya çerçeve yazılımlarının performansları karşılaştırılmış ve aralarından en iyi performasa sahip olanı kullanılmıştır.

4.2.2.1 VLC

VLC media player, açık kaynak kodlu bir video oynatıcı yazılımdır. Bünyesinde barındırdığı çok sayıda çözücü(codec) sayesinde ek bir paket yükleme gerektirmeksizin çok sayıda video ve ses dosya türünü oynatmaya izin vermektedir [33].

İHA'dan görüntü almak için hazırlanan SDP dosyası VLC'ye tanıtılarak görüntü alma işlemi gerçekleştirilmiştir. VLC ile bir saniyede alınan görüntü sayısı yaklaşık olarak 25-30 adet olarak hesaplanmıştır. Alınan görüntülerin kalitesi kullanılabilir seviyede olduğu tespit edilmiştir. İHA'dan görüntünün yer istasyonuna gönderilmesi ile aynı görüntünün yer istasyonunda görsellenmesi arasında geçen süre yaklaşık olarak 1.7 saniye olarak ölçülmüştür.

4.2.2.2 FFMPEG

FFMPEG, bir çok platformda kullanılabilen, ses ve görüntü kaydı dönüştürme yapılabilen, yayın işlemlerinin gerçekleştirmesini sağlayan açık kaynak kodlu bir



Şekil 4.8 VLC Multimedya Programı Üzerinden Alınan Görüntü

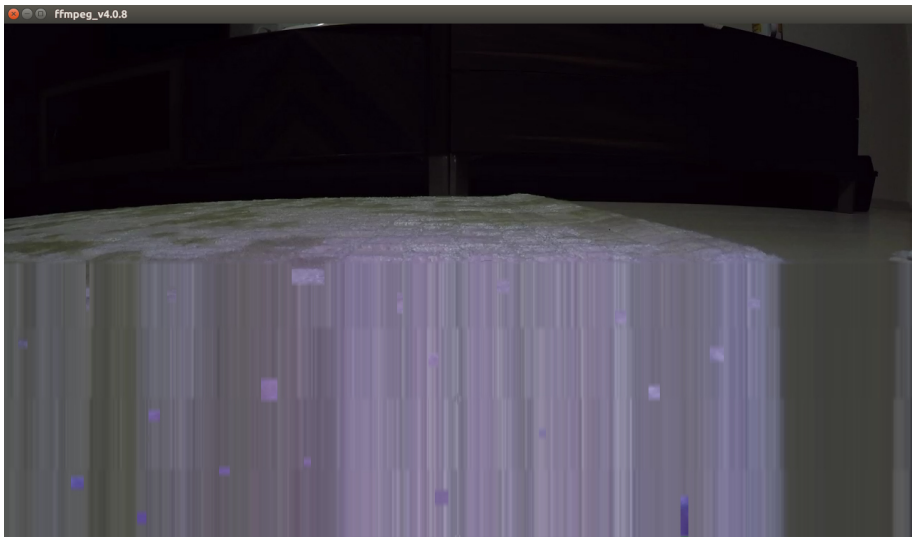
multimedya uygulamasıdır. Decode, encode, transcode, mux, demux, stream, filtreleme ve oynatma başta olmak üzere bir çok işlem gerçekleştirmeye olanak sunan FFmpeg; libavcodec, libavutil, libavformat, libavfilter, libavdevice, libswscale ve libswresample gibi paketleri de işlenebilir olarak kendi içerisinde barındırmaktadır. Bu çalışmada FFMPEG 4.0.8 versiyonu kullanılmıştır.

İHA'dan görüntü almak için hazırlanan SDP dosyası FFMPEG'e tanıtılarak görüntü alma işlemi gerçekleştirilmiştir. Bir saniyede alınan görüntü sayısı yaklaşık olarak 15-20 adettir. Görüntü alma hızı iyi seviyededir, fakat görüntü kalitesi kötüdür. FFMPEG'in ara belleğinin(buffer) çok düşük olmasından dolayı görüntü akışının 10. saniyesinden sonra görüntü donmakta ve hata vermektedir. Bu yüzden bu çalışma için FFMPEG multimedya çerçeve yazılımının uygun olmadığı tespit edilmiştir.

4.2.2.3 GStreamer

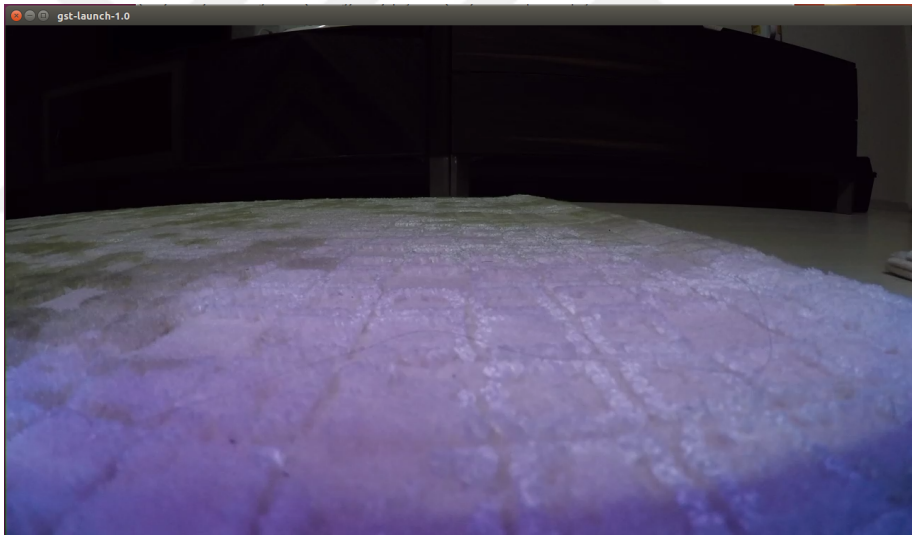
GStreamer, GObject yazı sistemine dayanarak C programlama dilinde yazılan ve grafik arayüz dışında ihtiyaç duyulan her türlü yeteneği sağlayabilen özgür bir multimedya çerçeve yazılımıdır. Gerçek zamanlı medya yayını yapan uygulamalar ve medya oynatıcıları altında çalışan bir uygulama olarak kullanılmaktadır. Bu çalışmada için OpenCV ile uyumlu olmasından dolayı GStreamer 1.0 versiyonu kullanılmıştır.

SDP dosyası içerisindeki veriler kullanılarak GStreamer üzerinden yer istasyonuna görüntü alımı gerçekleştirilmiştir. Bir saniyede alınan görüntü sayısı yaklaşık olarak 25-30 adet olarak hesaplanmıştır. Alınan görüntülerin kalitesi çok iyi seviyededir. İHA'nın hareketinden kaynaklı dahi görüntü akışı bozulmamaktadır. İHA'dan



Şekil 4.9 FFMPEG Multimedya Programı Üzerinden Alınan Görüntü

görüntünün yer istasyonuna gönderilmesi ile aynı görüntünün yer istasyonunda görsellenmesi arasında geçen süre yaklaşık olarak 0.5-0.6 saniye olarak ölçülmüştür.



Şekil 4.10 GStreamer Multimedya Programı Üzerinden Alınan Görüntü

İHA'dan gönderilen görüntülerin yer istasyonundan alınmasında kullanılan FFMPEG, VLC ve GStreamer multimedya çerçeve yazılımlarının performansları karşılaştırılarak sonuçlar Tablo 4.2'deki gibi aktarılmıştır. Yapılan testler sonucunda görüntü aktarımı sırasında görüntü kalitesi en iyi olan ve zaman gecikmesi en düşük olan multimedya çerçeve yazılımının GStreamer olduğu tespit edilmiştir. Bu sebeple, bu çalışmada yer istasyonundan görüntü alımı için GStreamer multimedya çerçeve yazılımı kullanılmıştır.

Yer istasyonuna alınan görüntüler, Python ve OpenCV fonksiyonları ile birlikte kullanılabilmesi için NumPy kütüphanesi kullanılarak OpenCV formatına

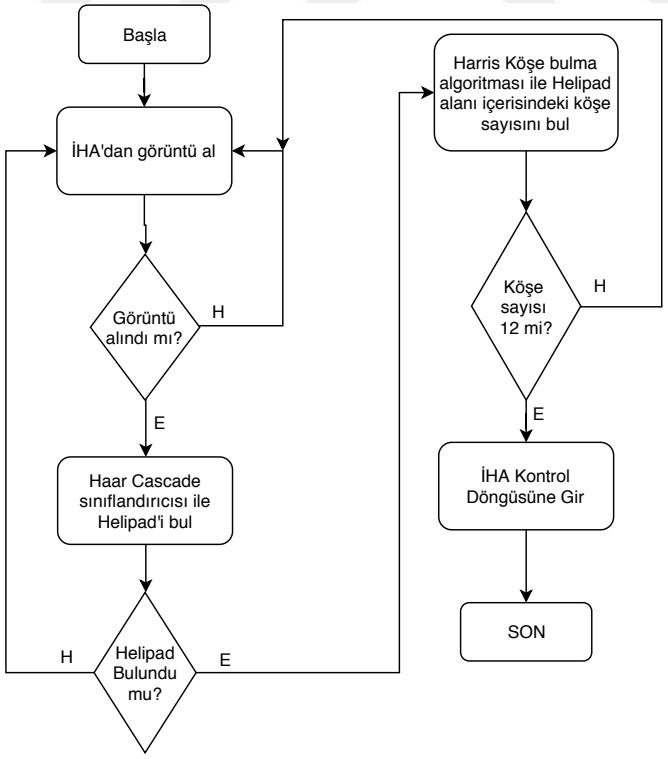
Tablo 4.2 Multimedya Çerçeve Yazılımları Karşılatırması

Multimedya Yazılımı	**Zaman Gecikmesi(s)**	**Görüntü Kalitesi**	**FPS Oranı**
FFMPEG	0.8	Kötü	15-20
VLC	1.7	Orta	25-30
GStreamer	0.6	İyi	25-30

dönüştürülmüş ve işlenmeye hazır hale getirilmiştir.

5
Hedef Platform Tespiti

İHA'nın iniş yapacağı hedef platform olan Helipad'in tespiti için görüntü işleme tabanlı çözümler ele alınmıştır. Öncelikli olarak Haar cascade sınıfladırma yöntemi ile Helipad bulunmuştur. Görüntü işleme algoritmasının daha kararlı çalışmasını destekleme amaçlı olarak Haar cascade sınıflandırma yöntemi ile bulunan Helipad alanı içerisinde Harris köşe bulma algoritmasıyla köşe noktalar aranmıştır. Helipad'in bulunduğu alan içerisinde köşe noktalar sadece H harfi üzerinde yer almakta ve sayısı en fazla 12 olmaktadır. Bu sebeple, Helipad'in bulunduğu alan içerisinde köşe nokta sayısı 12'ye eşit olan görüntüler kontrol açısından değerlendirmeye alınmıştır. Görüntü işleme algoritmasının akış diyagramı Şekil 5.1'deki gibi verilmiştir.



Şekil 5.1 Görüntü İşleme Algoritması Akış Diyagramı

Helipad'in Haar cascade sınıflandırma yöntemi ile bulunması Başlık 5.2'de, Helipad alanı içerisindeki köşe noktaların bulunması ise Başlık 5.3'de detaylı aktarılmıştır.

5.1 İniş Yapılacak Hedef Platform - Helipad

İniş yapılacak hedef platform Helipad olarak seçilmiştir (Şekil 5.2). Helipad; helikopterleri kullanan pilotların aletsiz ve görerek uçuş kuralları (VFR) dahilinde iniş-kalkış yapabildikleri, basit teknik yapılar dışında önemli üstyapı tesisleri bulunmayan heliport olarak açıklanmaktadır [34].

Helipad, H harfi ve etrafında bulunan çemberden oluşmaktadır. Dış zemini siyah renkli, H harfi ve çember ise beyaz renklidir. Helipad'in alınan görüntülerde arka plan şekillerinden ve renklerinden kolaylıkla ayırt edilebileceği öngörülmüştür. Bu sebeple iniş platformu olarak Helipad seçilmiştir.

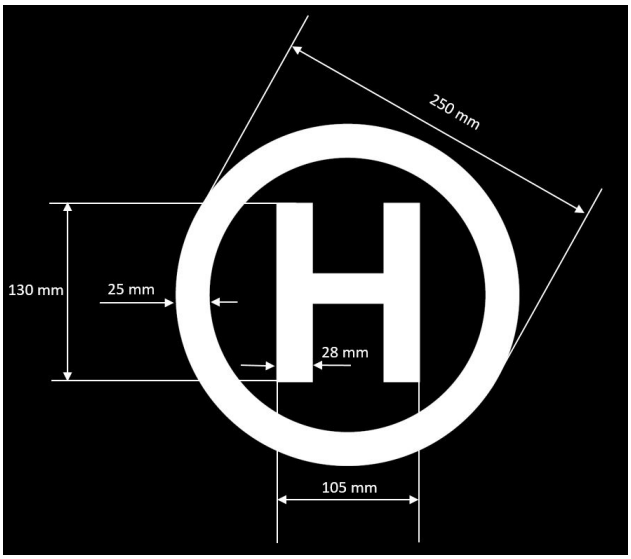


Şekil 5.2 İniş Yapılacak Hedef Platform - Helipad

Bu çalışmada kullanılan Helipad'in detaylı ölçüleri Şekil 5.3'deki gibidir. Helipad ölçüleri, İHA ile testlerin en falza 5 metre yükseklikten yapılacağı göz önününde bulundurularak belirlenmiştir. Ölçüler ayarlanırken, *Standards for Offshore Helicopter Landing Areas* dokümanından faydalanılmıştır [35].

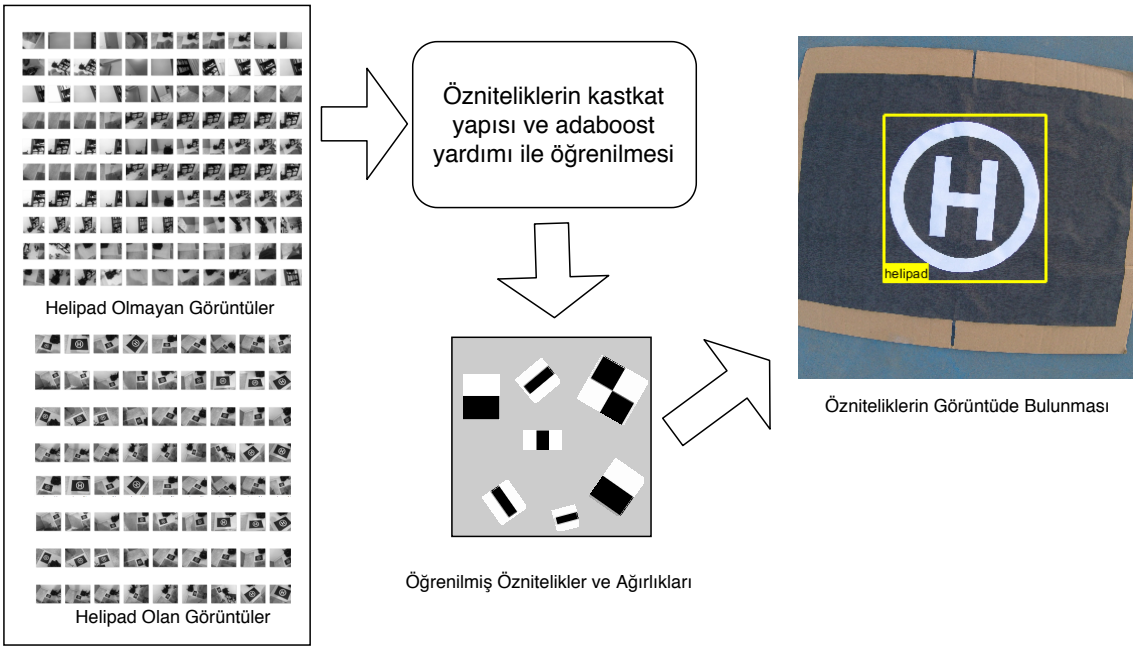
5.2 Haar Cascade Sınıflandırma Yöntemi ile Helipad Tespiti

İHA'dan alınan görüntülerde Helipad'in varlığını tespit etmek ve Helipad'in görüntü içerisindeki yerini bulmak için Haar Cascade sınflandırma yöntemi kullanılmıştır. Haar Cascade sınıflandırıcısı, görüntüleri normal yöntemlere göre çok daha hızlı şekilde işleyebilen ve yüksek doğruluk oranına sahip olan bir sınıflandırma yöntemidir [36]. Bu yöntemi normal nümerik yöntemden ayıran üç farklı temel özellik bulunmaktadır. Bunlar sırasıyla integral image yöntemi, adaboost algoritması



Şekil 5.3 Helipad Ölçüleri

ve kaskat yapısıdır. Integral image, Haar Cascade algoritmasının çalışması için gerekli olan işlem kapasitesini azaltmayı sağlamaktadır. Görüntü üzerinde bulunan bütün öznitelikler, Integral Image yöntemi kullanılarak belirlenmektedir. Adaboost yardımıyla bu özniteliklere nesne belirlemedeki önem derecesine göre ağırlık değerleri atanmaktadır. Kaskat yapısı ise adaboost tarafından atanan yüksek ağırlıklı öznitelikleri aramaya başlamakta ve bu yapı sayesinde ağırlığı yüksek öznitelikler kullanılarak hızlıca yanlış görüntüler ayrıştırılmaktadır. Bu sayede, ağırlığı yüksek öznitelik içeren görüntüler üzerinde detaylı algoritmalar çalıştırmak için zaman yaratılmaktadır.

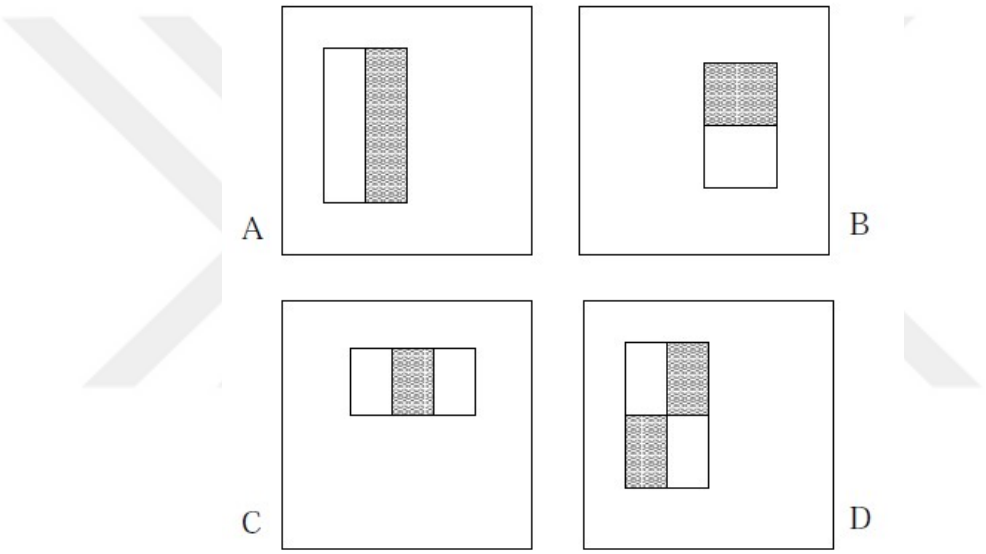


Şekil 5.4 Haar Cascade Sınıflandırıcı Çalışma Prensibi

5.2.1 Haar Cascade Öznitelikleri

Haar Cascade sınıflandırıcısının Şekil 5.5'de gösterildiği gibi dört farklı özniteliği bulunmaktadır. A, B, C ve D öznitelikleri, doğrudan piksel şeklinde karşılastırma yapmak yerine belirlenen en küçük ve en büyük boyutlarda piksel alanlarının karşılaştırması yapılarak incelenmektedir. Bu şekilde benzetme algoritmasının çalışması sağlanmaktadır [37].

Gri skala görüntülerde renk değeri 0-255 arasında değişmektedir. Haar Cascade sınıflandırma algoritmasında ise bu aralık, 0-1 aralığına denk gelecek şekilde tekrar düzenlenmektedir. Görüntüde bulunan her bir özniteliğin siyah ve beyaz bölgelerinin ortalamasına göre 0 ile 1 değeri arasında bir ağırlık değeri üretilmektedir. Bu değer, nesne bulma algoritmasında kullanılmaktadır.

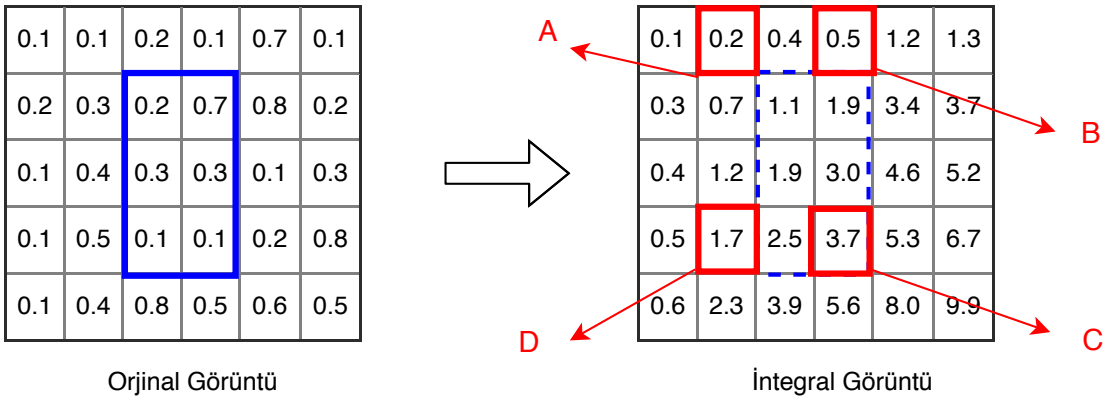


Şekil 5.5 Haar Kaskad Öznitelikleri

Şekil 5.5'deki A ve B; Kenar özniteliği, C; Çizgi özniteliği, D ise Dört-kare özelliği temsil etmektedir.

5.2.2 Integral Image

Çeşitli boyutlardaki özniteliklerin aranması için görüntü üzerindeki bütün piksellerin farklı kombinasyonlarda toplamına ihtiyaç duyulmaktadır. Bu işlem çok fazla zaman aldığından dolayı integral image yöntemi kullanılmaktadır. Integral image, rekursif tanımlanmış olup ilgili görüntüde bulunan piksellerin solunda ve üstünde bulunan piksel değerlerinin toplamı alınarak hesaplanmaktadır. Bu yöntem sayesinde, ilgili aralıkta bulunan piksel toplamları tek tek tüm pikselleri toplayarak değilde aynı aralığın uç noktalarında yer alan piksel toplamları kullanılarak hesaplanabilmektedir. Bu şekilde piksel toplamları çok daha hızlı hesaplanabilmektedir.



Şekil 5.6 Integral Image

Haar Cascade yönteminde, Şekil 5.6'de yer alan Orijinal Görüntü'de mavi dikdörtgen içerisindeki piksel değerlerinin toplamının bulunması hedeflenmektedir. Bu toplam, İntegral Görüntü'de gösterildiği gibi dikdörtgenin köşelerindeki piksel değerleri yardımıyla zaman kaybetmeden bulunabilmektedir.

$$\Sigma(Piksel) = (A + C) - (B + D) \tag{5.1}$$

Şekil 5.6'de gösterilen örnekte 6 adet piksel değerinin toplanması yerine 4 adet piksel değerinin toplanması işlem hızı anlamında çok fazla önem arz etmemektedir. Ancak, 500 adet pikselin yer aldığı bir dikdörtgen için aynı işlemin yapılması gerektiğinde, Integral image yöntemini kullanmak çok daha hızlı ve faydalı olmaktadır.

5.2.3 Adaboost Algoritması

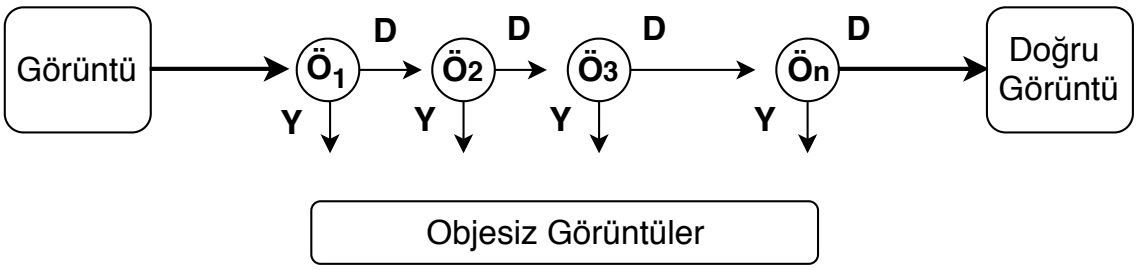
Adaboost algoritması, görüntü içerisinde var olan özniteliklerin tamamına ağırlık ataması yapmak için kullanılmaktadır. Nesneyi belirlemek için bu algoritma tarafından özniteliklerin önem derecesine göre ağırlık ataması yapılmaktadır. Bu sayede, haar kaskad yöntemiyle ağırlığı yüksek olan öznitelikler görüntülerde taranmaktadır.

Adaboost algoritması sayesinde önem derecesine göre her bir özniteliğe bir ağırlık atanmış olmakta ve bu ağırlık değerleri sayesinde seçim yapılmaktadır.

5.2.4 Kaskad Yöntemi

Kaskad yöntemi, görüntüde yer alan nesnenin tespit edilebilme oranını arttırmakta ve bu işlem için harcanan zamanı ciddi oranda azaltmaktadır. Görüntüler üzerinde ilk olarak ağırlık oranı yüksek olan öznitelikler aratılmaktadır. Bu sayede, negatif

görüntüler ile pozitif görüntüler yüksek olasılıkla birbirlerinden ayrılmaktadır. Düşük ağırlıklı öznitelikler, kaskat yapısının son katmanlarına yerleştirilmektedir. Negatif görüntüler ise ilk katmanlarda hızlıca ayrıştırılmaktadır, Geriye kalan görüntülerde de ağırlığı düşük öznitelikler aratılarak objenin bulunması sağlanmaktadır. Kaskat yöntemi sayesinde gereksiz işlem yükünden kaçınılmakta ve hesaplama zamanları minimum seviyede tutulmaktadır.



Şekil 5.7 Haar Kaskad

Ağırlığı yüksek olan öznitelikler ilk sıralardaki katmanlara yerleştirilmiştir. Nesnenin bulunmadığı negatif görüntülerin büyük çoğunluğu ilk katmanlarda elenmektedir. Kalan pozitif olma olasılığı yüksek görüntülerde düşük ağırlıklı öznitelikler aranmakta ve doğru görüntü bulunmaktadır.

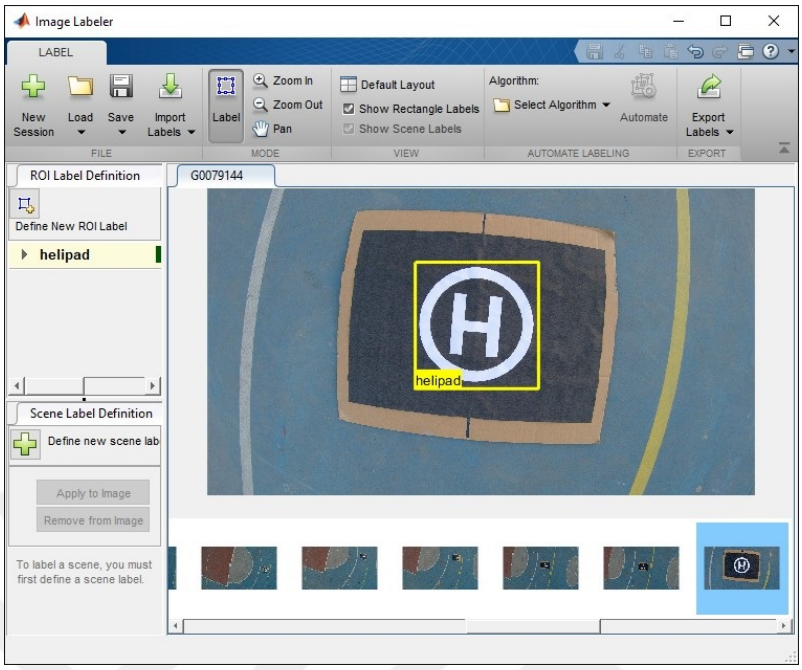
5.2.5 Helipad Bulma Algoritmasının Matlab Ortamında Eğitilmesi

Helipad bulma algoritması, Haar Cascade yöntemi kullanılarak Matlab ortamında eğitilmiştir. Haar Cascade algoritması ile eğtimin gerçekleşebilmesi için Helipad'in bulunduğu pozitif görüntüler ve Helipad'in bulunmadığı negatif görüntüler toplanmıştır. Bu görüntüler RGB formatından Gri formata dönüştürülmüştür. Pozitif görüntülerdeki Helipad'in bulunduğu alanlar Matlab Image Labeler modülü kullanılarak etiketlenmiştir. Son olarak, Matlab'de yer alan "trainCascadeObjectDetector" fonksiyonu kullanarak algoritmanın eğitme işlemi gerçekleştirilmiştir. Bu eğitme işleminin tamammasından sonra içerisinde özniteliklerin ve bu öz niteliklerin ağırlık değerlerinin yer aldığı .xml formatında veri dosyası elde edilmiştir. Bu veri dosyası Python ortamına taşınarak Helipad'in görüntü işleme ile bulunmasında kullanılmıştır.

5.2.5.1 Helipad İçeren Görüntülerin Matlab Ortamında Etiketlenmesi

Image Labeler, Matlab'deki Image Processing araç kutusu altında yer alan bir uygulamadır. Bu uygulama sayesinde herhangi bir görüntü üzerinde yer alan ROI (Region of Interest) bölgeleri kolaylıkla etiketlenebilmektedir. Şekil 5.8'deki arayüz

üzerinden pozitif görüntüler içeriye aktarılarılmış ve tüm görüntüler için Helipad alanları etikenlenmiştir.



Şekil 5.8 Matlab Image Labeler Uygulaması Arayüzü

Etiketleme işlemi tamamlandıktan sonra arayüz üzerindeki "Export Labels" kısmından Helipad'i içeren görüntülerin adres bilgileri ve aynı görüntü içerisinde Helipad'in bulunduğu piksel koordinat bilgileri tablo formatında Matlab çalışma ortamına aktarılmıştır. Matlab çalışma ortamına aktarılmış tablo Şekil 5.9'daki gibidir. Bu tablodaki ilk sütun görüntünün bulunduğu adres bilgisini, ikinci sütunu ise görüntüde etiketlenen Helipad'in piksel olarak koordinatlarını belirtmektedir.

pos_roi

47x2 table

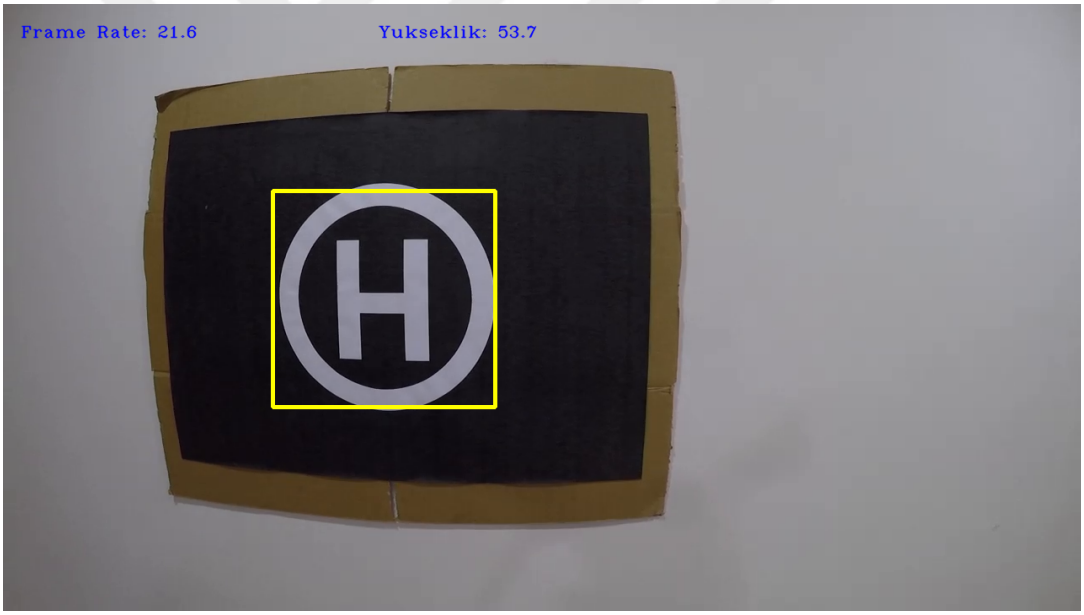
	1 imageFilename	2 helipad
1	'C:\Users\ali_v\Desktop\Gör...	[238,235,130,119]
2	'C:\Users\ali_v\Desktop\Gör...	[276,137,222,224]
3	'C:\Users\ali_v\Desktop\Gör...	[348,226,106,93]
4	'C:\Users\ali_v\Desktop\Gör...	[237,121,200,197]

Şekil 5.9 Matlab Çalışma Ortamına Aktarılmış Helipad ROI Bilgileri

5.2.5.2 Matlab Ortamında Algoritma Eğitimi

Helipad bulma algoritmasının Haar Cascade sınıflandırma yöntemi ile eğitimi Matlab ortamında gerçekleştirilmiştir. Eğitme işleminde "trainCascadeObjectDetector" fonksiyonu kullanılmıştır. Bu fonksiyonda giriş verisi olarak Başık 5.2.5.1'de aktarılan pozitif görüntü bilgilerinin yer aldığı tablo değerleri, negatif görüntülerin adres bilgisi ve eğitim sonucu hedeflenen hata oranı bilgisi kullanıllanılmış ve algoritma eğitim işlemi tamamlanmıştır. "trainCascadeObjectDetector" fonksiyonu, algoritma eğitiminin gerçekleştirilmesinden sonra .xml formatında veri dosyası oluşturmaktadır. Matlab R2018b sürümünden elde edilen veri dosyası, OpenCV 3.4.0 ve öncesi versiyonları desteklemektedir. Bu sayede, Matlab'de üretilen veri dosyası Python ortamında taşınmış ve OpenCV foksiyonları ile birlikte kullanılmıştır.

Görüntü işleme algoritmasının Matlab ortamında eğitilmesinin ardından yer testleri yapılmıştır. Bu testlerde Şekil 5.10'da görüldüğü gibi Helipad başarılı şekilde bulunmuştur.



Şekil 5.10 Haar Cascade Sınıflandırıcı ile Helipad Tespiti

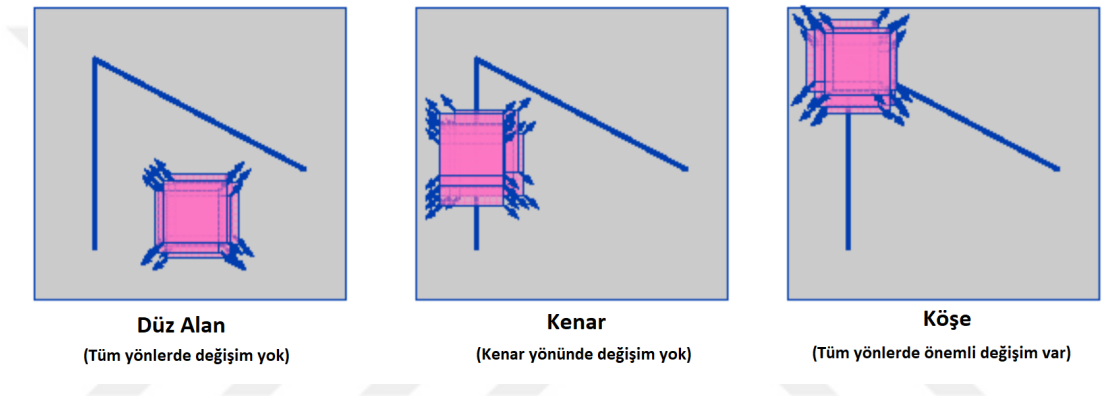
5.3 Harris Köşe Bulma Algoritması

Harris köşe bulma algoritması, genellikle görüntü işleme algoritmalarında köşe noktaları bulmak ve görüntü özelliklerini ortaya çıkarmak için kullanılmaktadır. İlk olarak 1988 yılında Chris Harris ve Mike Stephens tarafından nokta özniteliği bulma algoritması olarak tanıtılmıştır [38].

Harris aynı zamanda sliding window algoritmasıdır. Görüntü içerisinde bir yerel algılama penceresi oluşturulmakta ve bu pencere görüntünün üzerinden geçirilerek

görüntünün x ve y yönlerinde türevi hesaplanmaktadır. Bu türev değerlerinin kareleri ve iki yöndeki türev çarpımları kullanılarak Image Structure Tensor matrisi oluşturulmaktır. Bu matirisin determinantı ve izi bulunarak Köşe Noktası Cevabı (R) değeri hesaplanmaktadır [39]. R değeri, önceden belirlenmiş bir eşik değer ile karşılaştırılarak işlem yapılan piksel bölgesinde köşe olup olmadığı tespit edilmektedir.

Yerel algılama penceresi görüntü üzerinde her kaydırıldığında, tüm yönlerdeki R değeri hesaplanmaktadır. Eğer tüm yönlerdeki R değeri belirlenen eşik değerinin altında ise bulunulan bölge düz bölgedir. R değeri tek bir yönde ve negatif değere sahipse, bu yöne dik kenar özellik olduğu söylenmektedir. Eğer R değeri tüm yönlerde eşik değerinin üzerinde kalıyorsa, bulunulan bölgede köşe nokta var denmektedir (Şekil 5.11).



Şekil 5.11 Harris Köşe Bulma Algoritması Çalışma Yöntemi

Harris köşe bulma algoritması için kullanılan denklemler aşamalar şeklinde aşağıdaki gibi aktarılmıştır:

- RGB görüntünün Gri skalaya dönüştürülmesi

$$Gri = 0.299 * R + 0.587 * G + 0.114 * B \tag{5.2}$$

- X ve Y yönlerinde türev alınması:

$$I_x = G_\sigma^x * I \ , \ I_y = G_\sigma^y * I \tag{5.3}$$

- Image Structure Tensor matrisi hesabı:

$$\begin{aligned} S_{xx} &= G_{\sigma'} * (I_x)^2 \\ S_{yy} &= G_{\sigma'} * (I_y)^2 \\ S_{xy} &= G_{\sigma'} * I_x * I_y \end{aligned} \tag{5.4}$$

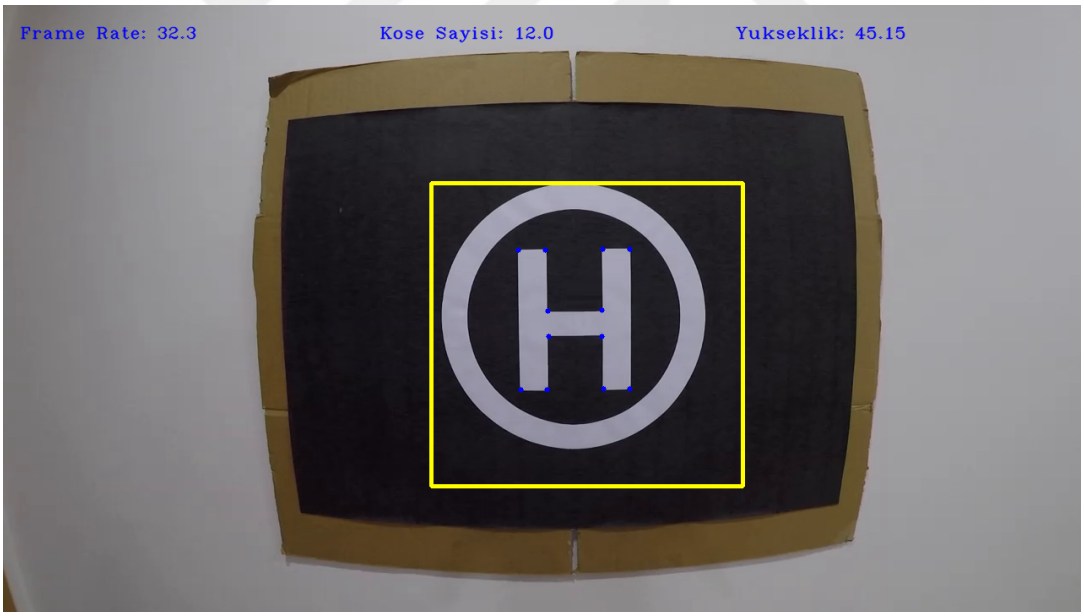
$$H(x,y) = \begin{bmatrix} S_{xx}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{yy}(x,y) \end{bmatrix} \tag{5.5}$$

- R - köşe noktası cevabı değeri:

$$R = Det(H) - k * (Trace(H))^2 \tag{5.6}$$

Harris algoritması, Python dili ile OpenCV fonksiyonları kullanılarak gerçekleştirilmiştir. Görüntü üzerinde kaydırılan pencerenin büyüklüğü 5x5 olarak seçilmiştir.

Harris algoritması ile bulunan köşe noktalar, sadece Haar Cascade sınıflandırma yöntemi ile bulunan Helipad alanı içerisinde aranmıştır. Belirlenen alan içerisinde köşe sayısı toplam 12 adet olan görüntüler değerlendirmeye alınmış olup, diğer görüntüler kontrol açısından değerlerdirilmemiştir. Şekil 5.12'de gösterildiği gibi Helipad'in görüntü işleme yöntemi ile tespiti başarılı bir şelkilde gerçekleştirilmiştir.



Şekil 5.12 Haar Cascade ve Harris Algoritmaları ile Helipad Tespiti

6
İHA Yükseklik Hesabı

6.1 İHA Yükseklik Hesabı

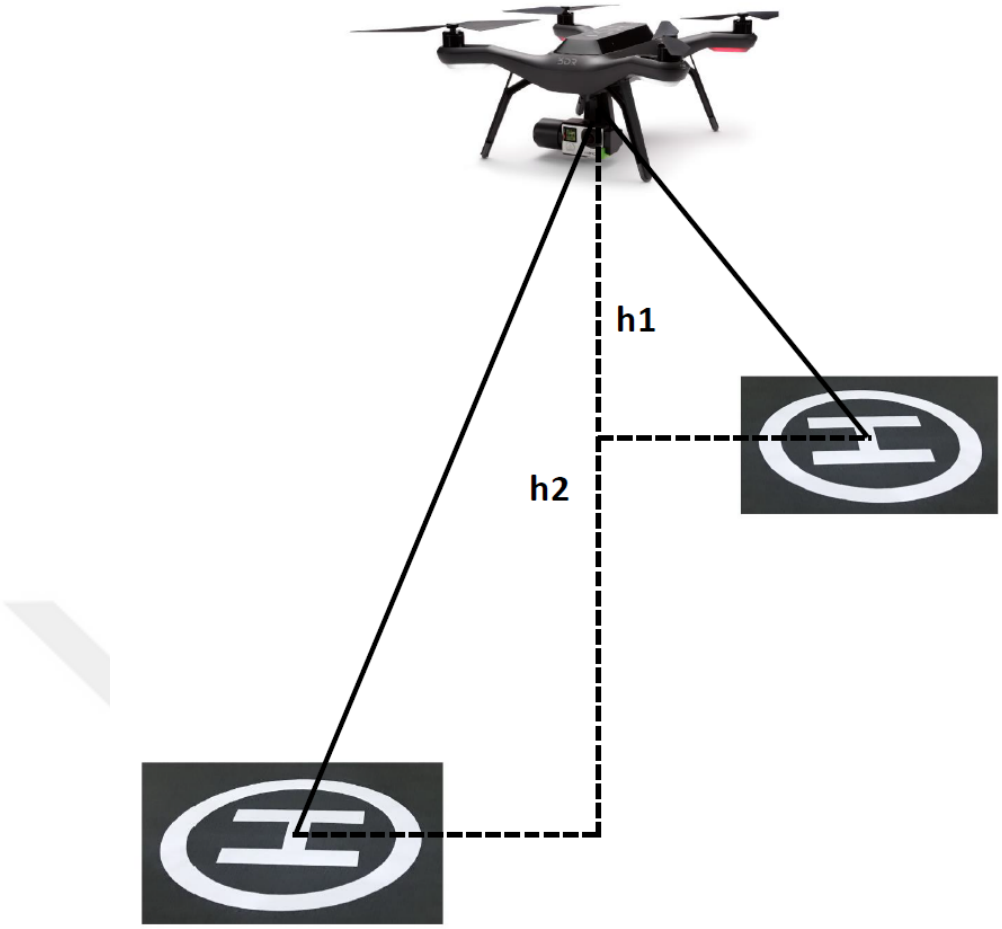
3DR Solo İHA'da enlem, boylam ve yükseklik verilerini ölçen NEO-7N üretici kodlu GPS modül bulunmaktadır. Bu GPS modül sayesinde İHA'nın uçuş esnasında bulunduğu konum verileri ölçülebilmektedir. NEO-7N GPS modülü 2.5 metre doğrulukla pozisyon bilgisi üretmektedir [40], yapılan uçuş testleri ile bu değer doğrulanmıştır. Bu çalışmada yapılan tüm iniş testleri en fazla 5 metre yükseklik seviyelerinde gerçekleştirilmiştir. Bu sebeple, kapalı çevrim yükseklik kontrolünde doğruluk değeri 2.5 metre olan sensör verilerinin bu çalışmada kullanılması uygun bulunmamıştır.

GPS verisi, İHA'nın yüksekliğini her zaman zemine göre vermektedir. Helipad'in zeminden yüksek bir konumda olduğu durumda, GPS verileri ile tasarlanmış bir iniş kontrol sistemi ile İHA'nın Helipad'e güvenli bir şekilde iniş yapması mümkün olamayacaktır.

Görüntü işleme ile tasarlanan iniş kontrol sisteminde kontrol çevrim süreleri 25-30 Hz bandındadır. Yukarıda bahsedilen GPS modlünün veri güncelleme hızı yaklaşık 5-10 Hz olduğundan dolayı, bu GPS verisinin kullanıldığı yükseklik kontrol çevrimlerinde yaklaşık her 3 çevrimde aynı yükseklik verisinin kullanılması gerekecektir. Bu durum kontrol performansı açısından kabul edilebilir değildir.

İHA'nın Helipad'e olan dikey mesafesi yani yüksekliği, yukarıda aktarılan sebeplerden dolayı GPS modulü yerine görüntü işleme yöntemi kullanılarak hesaplanmıştır. Görüntü işleme yöntemiyle İHA'nın Helipad'ten yüksekliğini hesaplayabilmek için görüntü alınan kameranın görüş açısı ve görüntüdeki bir pikselin gerçekteki milimetre karşılığının bilinmesi gerekmektedir. Bu yöntemle Şekil 6.1'deki gibi İHA'nın Helipad'ten yüksekliği (h1 ve h2 yükseklikleri) hesaplanmıştır.

R_p değeri bir pikselin milimetre karşılığıdır. Bu oranın hesaplanabilmesi için Helipad üzerinde gerçek ölçüsü bilinen bir uzunluğun ve bu uzunluğun görüntüdeki



Şekil 6.1 İHA Yükseklik Görüntüsü

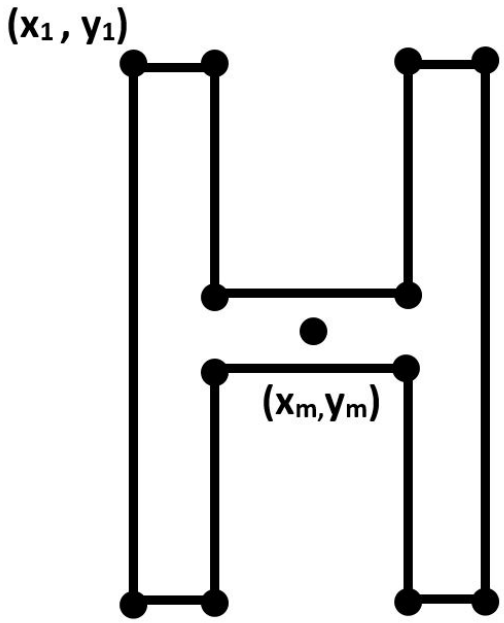
piksel değerinin bulunması gerekmektedir. Bu değerler orantılanarak R_p değeri hesaplanmaktadır. Helipad'te yer alan H harfinin, ölçüleri ve şekli gereği bu hesaplama için uygun olduğu değerlendirilmiştir.

H harfinin tam orta noktasının en uzak dört köşesinden birine olan mesafesi sabit ve 66.5mm dir. Bu mesafenin piksel değeri, Denklem (6.1), Denklem (6.3) ve Denklem (6.4)'deki formüller ile hesaplanmıştır.

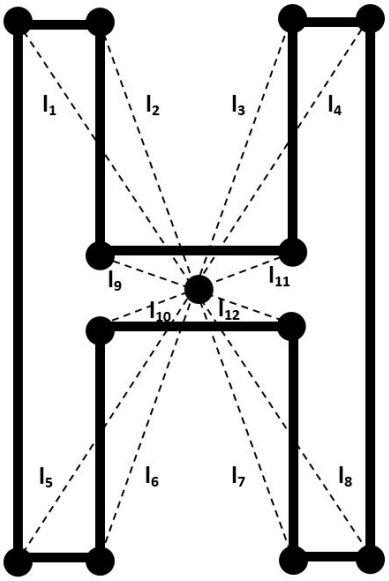
Denklem (6.1)'de H harfinin tüm köşe noktalarının aritmetik ortalamaları alınarak orta nokta piksel değerinin koordinatları bulunmuştur.

$$x_m = \frac{\Sigma(x_i)}{12} \ , \ y_m = \frac{\Sigma(y_i)}{12} \tag{6.1}$$

H harfi üzerindeki 12 köşe noktanın orta noktaya olan uzunlukları hesaplanarak bu küme içerisinden en büyük olanı bulunmuştur (Denklem (6.3) ve Denklem (6.4)). Bulunan en büyük uzunluk değeri her zaman l_1, l_4, l_6 veya l_8'den bir tanesi olmaktadır



Şekil 6.2 H Harfi Orta Nokta



Şekil 6.3 H Harfi Köşe Uzaklık

ve bu değerler birbirine eşittir (Denklem (6.2)).

$$l_1 = l_4 = l_6 = l_8 \tag{6.2}$$

$$\Delta l_i = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} \tag{6.3}$$

$$l_{max} = max(\Delta l_i) \tag{6.4}$$

l_{max}'ın piksel değeri ile gerçek ölçüsü olan 66.5 mm değeri Denklem (6.5)'deki gibi orantılanarak bir pikselin kaç mm olduğu hesaplanmıştır.

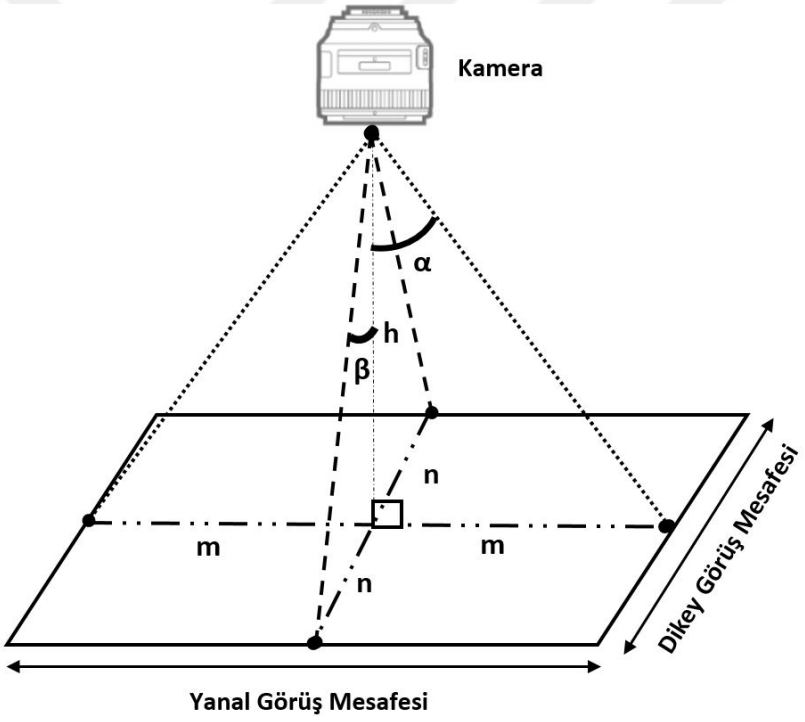
$$R_p = \frac{l_{max}}{66.5} \tag{6.5}$$

İHA'da bulunan GoPro Hero4 Black model kameranın dikey($2*\beta$) ve yanal($2*\alpha$) görüş açıları sırasıyla 69.5 ve 118.2 derecedir [41]. Şekil 6.4'de gösterilen α açısı 59,1 ve β açısı 34,75 derece olarak hesaplanmaktadır. Bu açı değerlerinin tanjantı, İHA'nın görüş mesafesinin yüksekliğe oranıdır ve Denklem 6.6'daki gibi hesaplanmaktadır.

$$tan(\alpha) = \frac{m}{h}\ ,\ tan(\beta) = \frac{n}{h} \tag{6.6}$$

Denklem 6.6'deki m ve n değerleri, İHA'nın bulunduğu yükseklikten görebildiği yanal ve düzlemsel mesafenin yarısına eşittir. İHA'dan alınan görüntüler 1280x720 piksel ve bir pikselin milimetre karşılığı Denklem 6.5'deki gibi hesaplandığından dolayı m ve n değerleri $m = 640 * R_p$, $n = 360 * R_p$ şeklinde hesaplanabilmektedir. Bu hesaplamalar sonucunda m değeri; R_p değeri ve α açısı kullanılarak İHA'nın yüksekliği Denklem 6.7'deki gibi hesaplanabilmektedir.

$$h = \frac{m}{tan(\alpha)} = \frac{R_p * 640}{tan(\alpha)} \tag{6.7}$$

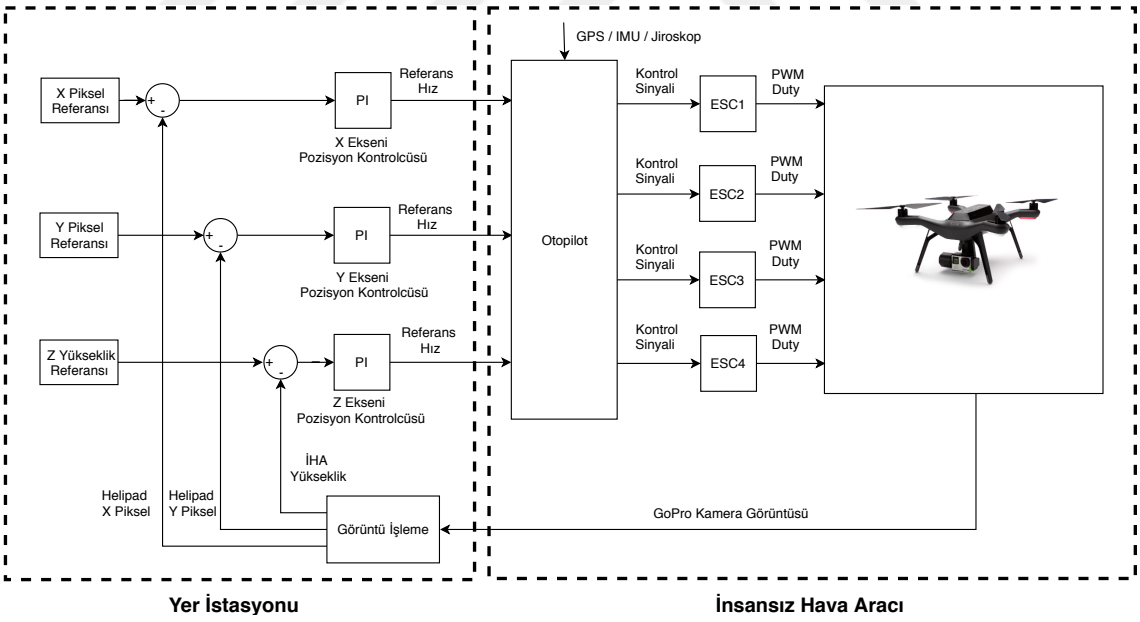


Şekil 6.4 GoPro Hero4 Görüş Alanı

7
Görüntü İşleme İle İniş Kontrolü

7.1 İHA Hareket Kontrolü

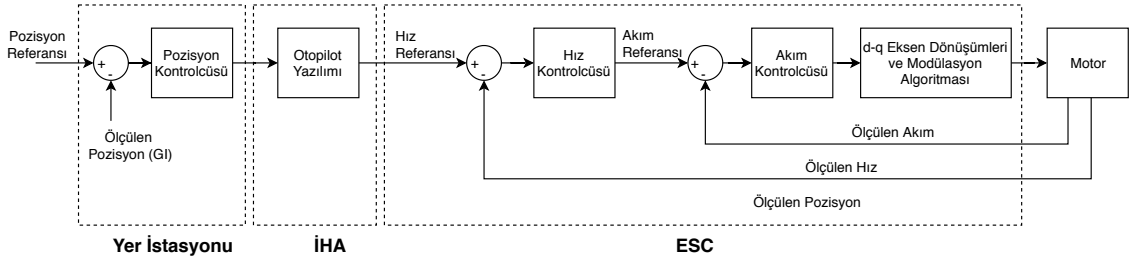
İHA'nın x, y ve z eksenlerindeki hareket kontrolü Şekil 7.1'de gösterilen şekilde yapılmaktadır. Yer istasyonunda üç hareket ekseni için görüntü işleme ile geri besleme alan pozisyon kontrolcüleri çalışmaktadır. İHA'nın yerleşik donanımlarında ise iç içe çalışan akım-hız kontrol döngüleri ve otopilot yazılımı çalışmaktadır. x, y ve z eksen kontrolcüleri, yine bu eksenler için İHA'nın referans çizgisel hız komutlarını üretmektedir. Bu hız komutları yer istasyonundan İHA'ya gönderilerek İHA'nın istenilen doğrultuda hareket etmesi sağlanmaktadır.



Şekil 7.1 Kontrol Algoritması Blok Diyagramı

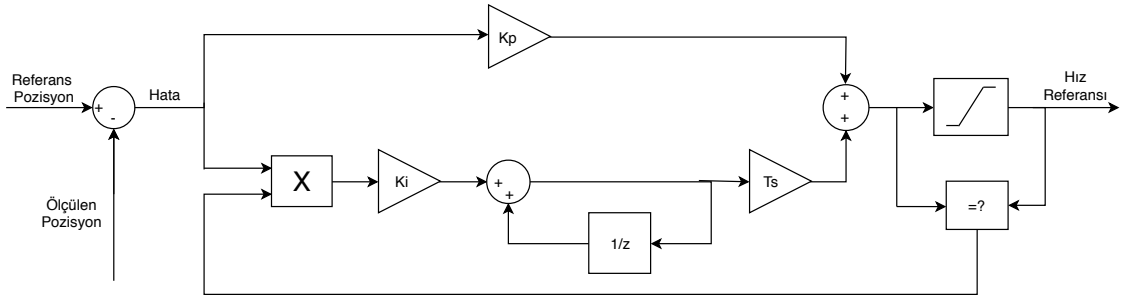
Otopilot algoritması, yer istasyonundan İHA'ya gelen hız komutlarını işleyerek İHA'nın belirtilen doğrultuda ve belirtilen hızda hareketini sağlayabilmesi için üzerinde bulunan 4 motor sürücüsüne (ESC) referans motor dönüş hızlarını göndermektedir. ESC'ler, otopilot algoritmasından gelen bu komutlara göre kendi içerisinde akım ve motor dönüş hızı kontrolü yaparak İHA'nın hareketini sağlamaktadırlar. ESC'lerin

akım kontrolü için kullanıdıkları akım geri besleme değerleri motor fazlarında yer alan akım sensörlerinden ölçülmektedir. Hız kontrol döngüsündeki geri besleme hız bilgisi için herhangi bir pozisyon sensörü kullanılmaktadır. Hız bilgisi, faz gerilimleri faz akımlarından çıkartılmaktadır. İHA'nın kontrolü için Şekil 7.2'deki gibi gösterilen iç içe döngülerin bulunduğu kontrol yapısına kaskad kontrol topolojisi adı verilmektedir [20].



Şekil 7.2 İHA Motor Control

Yer istasyonunda pozisyon kontrolü için PI kontrolcüler kullanılmaktadır. Bu PI kontrolcüleri, referans ile ölçülen pozisyon değerleri arasında farkı hesaplayıp bu farkı minimize edecek hız çıkışlarını ayarlamaktadır. Kontrolde oluşabilecek bir takım sıkıntılardan dolayı İHA'yı belirsiz noktalara göndermemek için bu kontrolcü çıkışları belirli bir değere limitlendirilmiştir. Bu limit sayesinde İHA en fazla 2,5 m/s hıza ulaşabilmekte ve bu değerin üzerinde olan hız referansları İHA'ya gönderilmemektedir.



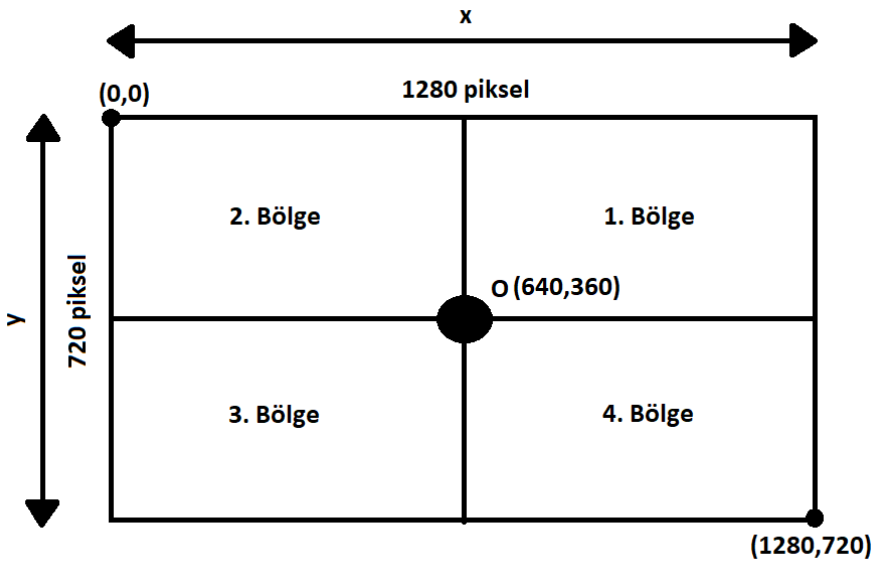
Şekil 7.3 PI Kontrolcü Yapısı

PI kontrolcü içerisinde yer alan integratör, her kontrol adımında önceki değer ile o anki değeri toplamaktadır. PI çıkışının limitlendiği durumlarda integratör çalışmasını devam ettirdiğinden dolayı integratör üzerinde hata birikmesi meydana gelmektedir. Kontrolcü performansının eniyileştirilmesi ve integratördeki hata birikmesinin önüne geçmek için PI kontrolcülere hata birikmesi engelleme (anti-windup) özelliği eklenmiştir. Bu özellik sayesinde, PI çıkışında limit durumu algılanıp integral terimi sıflanmaktadır.

7.1.1 X ve Y Eksenlerinde Helipad Üzerine Hareket Kontrolü

İHA'nın bulunduğu noktadan Helipad'in üst noktasına gelmesi için Şekil 7.1'de gösterildiği gibi x ve y eksenlerinde pozisyon kontrolü yapılmaktadır.

İHA, Helipad'in tam olarak üst noktasında olduğu durumda Helipad'te bulunan H harfinin orta noktası, kameradan alınan görütünün tam orta pikseline denk gelmektedir. Bu sebeple, İHA'yı hedefin üst noktasına gönderebilmek için Şekil 7.4'de belirtilen x ve y eksenlerinin referans piksel değerleri 640 ve 360 piksel olarak girilmiştir. Böylece İHA, yüksekliği sabit kalacak x ve y eksenlerinde pozisyon kontrolü yapılarak hedefin üst noktasına gönderilmektedir. Pozisyon kontrolünde kullanılan x ve y eksenlerinin geri besleme değerleri, Bölüm 5'de aktarıldığı gibi görüntü işleme ile bulunan ve Helipad içerisinde yer alan H harfinin orta nokta piksel değerleridir. Geri besleme piksel değerleri Denklem 6.1'deki gibi hesaplanmaktadır.

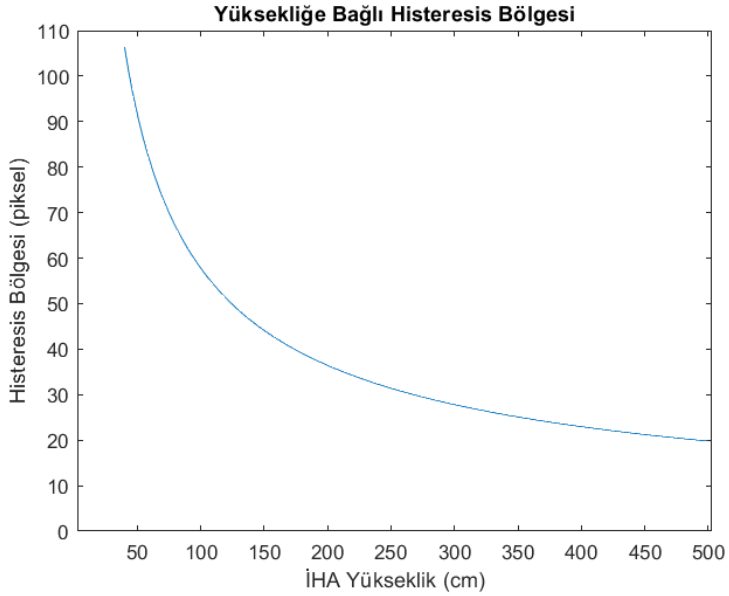


Şekil 7.4 Görüntüdeki Bölgeler ve Köşe Nokta Piksel Verileri

Başlık 7.3'de belirtildiği gibi, inişe başlanabilmesi için İHA'nın Helipad'in tam olarak üst noktasına gelmesi gerekmektedir. Kontrol sisteminde oluşan kalıcı durum hatalarından dolayı İHA, hiç bir zaman tam olarak (640,360) pikselinin üzerine gelememektedir. Bu sebeple, İHA'nın Helipad'in tam olarak üst noksana geldiğini belirleyecek ve piksel değerine bağlı olan bir histerezis bölgesi tamlanımlanmıştır. Bu histerezis bölgesi piksel değerine bağlı olduğu için yüksekliğe göre de değişmesi gerekmektedir. Bu sebeple, İHA ile yapılan testlerden veriler toplanarak Denklem 7.1'deki gibi bir eğri fonksiyonu oluşturulmuştur.

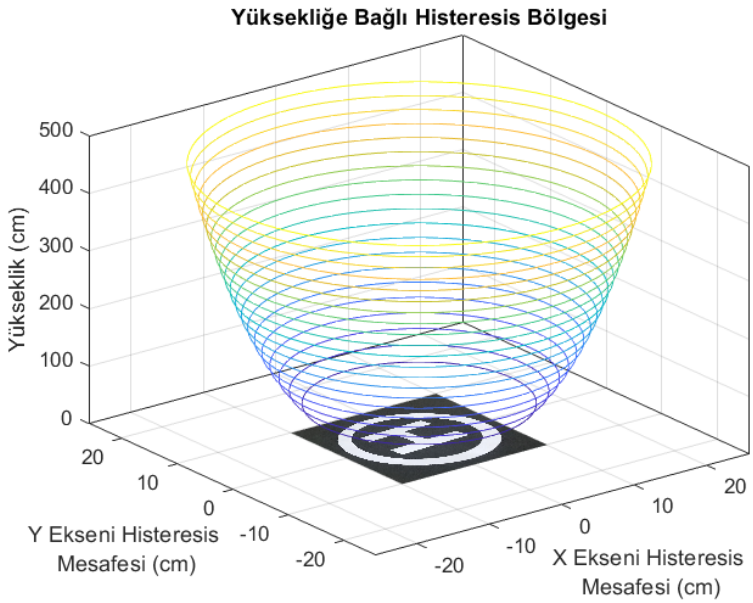
$$P_{hys} = 1240,529 * h^{-0,665861} \tag{7.1}$$

Denklem 7.1'deki P_{hys}, histerizis bölgesinin yüksekliğe bağlı piksel değerini, h ise İHA'nın yüksekliğini ifade etmektedir. Yüksekliği 40 cm ile 500 cm arasında değişmesi durumunda histeresis bölgesi piksel olarak Şekil 7.5'deki gibi oluşmaktadır.



Şekil 7.5 Yüksekliğe Bağlı Histeresis Bölgesi (Piksel)

Denklem 7.1'deki P_{hys} değerinin R_p değeri ile çarpılması sonucunda Şekil 7.6'daki gibi yüksekliğe bağlı histerezis bölgesi santimetre değeri elde edilmektedir.



Şekil 7.6 Yüksekliğe Bağlı Histeresis Bölgesi (Santimetre)

7.1.2 Z Ekseninde İniş Kontrolü

Şekil 7.1'da belirtilen Z ekseni, İHA'nın yüksekliğini kontrol eden eksendir. Bu eksende de pozisyon kontrolü için PI kontrolcüsü kullanılmıştır.

Helipad'in büyüklüğü ve İHA'da bulunan GoPro Hero4 kameranın çözünürlüğünden kaynaklı olarak Helipad'in kamera ile görülebildiği minimum yüksek limiti bulunmaktadır. Kamera, Helipad'in tam orta noktasında yer aldığı durumda bu yükseklik değeri 30 cm dir. İHA'nın Helipad'ten yüksekliği 30 cm'nin altına düştüğü durumlarda Helipad tam olarak görülemediği için alınan görüntüler işlenebilir durumda olmamaktadır. İHA'nın uçuş esnaasındaki salınımları da göz önünde bulundurularak iniş referans değeri herzaman sabit 45 cm olarak girilmiştir. Başlık 7.3'de aktarıldığı gibi, İHA bu yüksekliğe ulaştığında uçuş modu "Land" e çekilerek iniş sonlandırılmaktadır.

Helipad'ten 20 cm yükseklikte alınan görüntüler Şekil 7.7'deki gibidir.



Şekil 7.7 20 cm Yükselikten Alınan Görüntü

İHA'nın yer istasyonundaki yükseklik pozisyon kontrolü sadece Başlık 7.3'de aktarılan İniş Modu aktif olduğu durumda yapılmaktadır. Hedef Arama Modu ve Hedef Takip Modu içerisinde yükseklik kontorlü otopilot tarafından GPS verileri kullanılarak yapılmaktadır.

7.2 Rota Açısına Göre Hız Kontrolü

Dronekit yazılımı ve Python programlama dili ile geliştirilmiş haberleşme komutlarıyla İHA'dan anlık pozisyon, hız ve açı bilgileri alınmaktadır. Aynı zamanda NED(North, East, Down)'e göre pozisyon referansı ve hız referansı ayarlanabilmektedir.

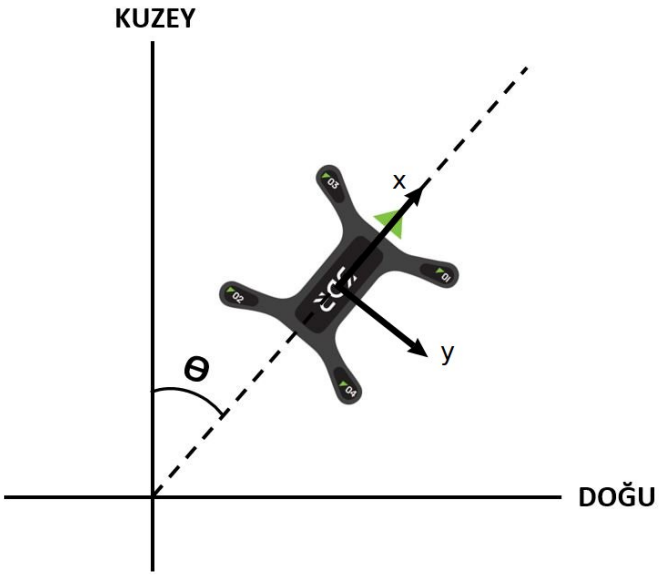
Yer istasyonundaki pozisyon kontrolcülerin çıkışları, İHA'nın rota açısına göre hız referansı üretmektedir. Fakat, yer istasyonundan İHA'ya sadece NED'e göre düzlemsel hız referansı gönderilebilmektedir [20]. Bu sebeple, x ve y eksenlerinin pozisyon kontrolcü çıkışlarındaki değerlerin İHA'nın rota açısı kullanılarak NED'e göre

ayarlanması ve İHA'ya bu şekilde gönderilmesi gerekmektedir.

Denklem 7.2'deki formül kullanılarak İHA'nın rota açısına göre üretilmiş hız değeri NED'e göre ayarlanmakta ve İHA'ya hız referansları bu şekilde gönderilmektedir.

$$\begin{bmatrix} v_{xr} \\ v_{yr} \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} * \begin{bmatrix} v^*_{xr} \\ v^*_{yr} \end{bmatrix} \tag{7.2}$$

Rota açısı Şekil 7.8'de gösterildiği gibi İHA'nın x ekseni ile kuzey yönü arasındaki açıdır[42]. Bu değer İHA'da bulunan magnetometre ile ölçülmekte ve yer istasyonuna gönderilmektedir.



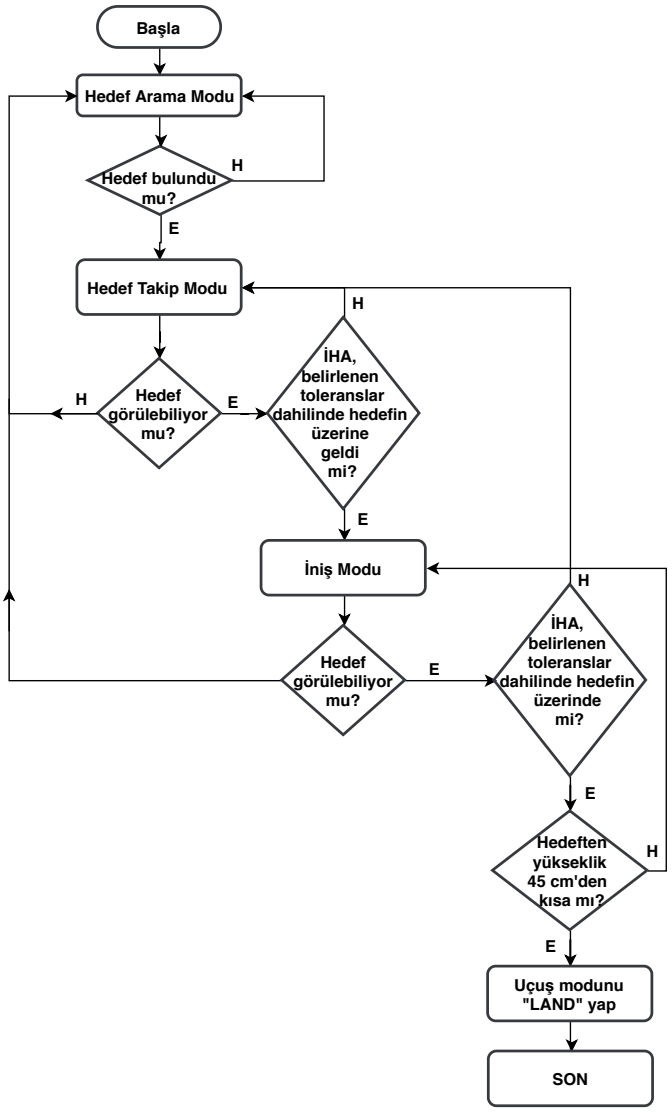
Şekil 7.8 İHA Rota Açısı

7.3 Görüntü İşleme İle İniş Kontrolü

İHA'nın otonom inişi için Şekil 7.9'deki akış diyagramda belirtilen adımlar takip edilmiştir. Bu akış diyaragmı içerisinde İHA'nın takip ettiği üç ana mod bulunmaktadır. Bu modlar aşağıdaki gibidir:

- Hedef Arama Modu
- Hedef Takip Modu
- İniş Modu

İHA'nın aktif uçuşu esanasında Kullanıcı Arayüzü'nden iniş komutunun verilmesi ile birlikte Hedef Arama Modu aktif olmaktadır. Hedef Arama Modu'nda, Bölüm



Şekil 7.9 Otonom İniş Algoritma Akış Diyagramı

5'de aktıralan yöntemler izlenerek Helipad aranmaktadır. İHA, bu mod içerisinde yüksekliğini sabit tutarak, Şekil 7.10'daki gibi başlangıç noktasını merkez kabul edecek şekilde dairesel hareket etmekte, Helipad'i aramaktadır. Şekil 7.10'nda belirtilen dairenin yarı çap değeri, uçuş testlerinde 3 metre olarak belirlenmiştir. Bu değer uçuş yapılan ortama ve Helipad'in büyüklüğüne göre değişiklik gösterebilmektedir.

Hedef Arama Modu'nda Helipad'in bulunmasıyla birlikte Hedef Takip Modu aktif edilmektedir. Bu mod içersinde İHA, buluduğu konumdan yüksekliğini sabit tutarak Helipad'in üst noktasına gönderilmektedir.

İHA belirlenen limitler dahilinde hedefin üst noktasına geldiğinde İniş Modu aktif edilmektedir. İniş Modu için tasarlanmış PI kontrolcü, sadece İniş Modu aktif olduğu durumda çalıştırılmaktadır. İniş Modu, Helipad'te bulunan H harfi orta noktasının Şekil 7.6'deki yüksekliğe bağlı verilen alan içerisinde olduğu durumda



Şekil 7.10 Hedef Arama Modu

aktif olmaktadır. İHA'nın uçuşu esnasında H harfi orta noktasının bu alan içerisinden çıkması durumunda iniş durdurulmakta ve İHA'nın bu alan içerisine girmesi beklenmektedir.

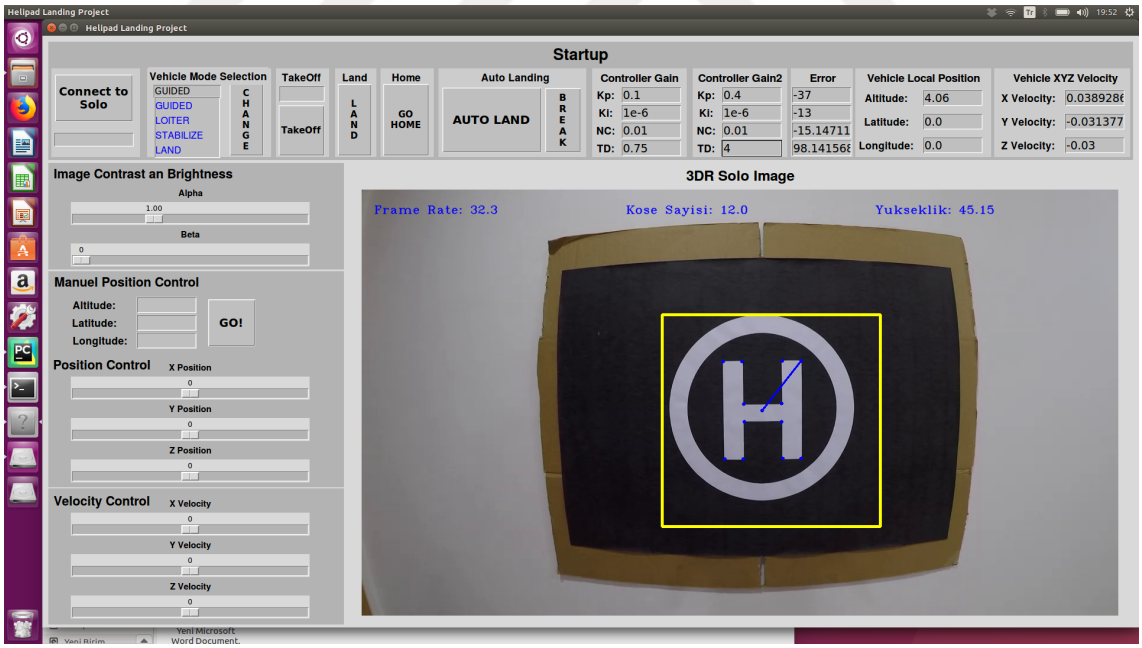
İniş Modu içerisinde z ekseninde yükseklik kontrolü yapılırkan aynı zamanda x ve y eksenlerindeki yanal ve düzlemsel pozisyon kontrolü de yapılmaktadır.



Şekil 7.11 Hedef Takip Modu

8
Kullanıcı Arayüz Çalışmaları

İHA'nın tüm uçuş hareketlerini kontrol edebilmek, uçuş esnasında İHA'ya müdahalede bulunabilmek, kontrol parametrelerini ayarlayabilmek ve uçuş verilerini anlık olarak takip edebilmek amacıyla Kullanıcı Arayüzü tasarımı yapılmıştır. Bu arayüz, Python dili ile Tkinter kütüphanesi kullanılarak geliştirilmiştir. Tkinter, Python kurulumu ile birlikte gelen ve pencereli-menülü modern arayüzler geliştirmeyi sağlayan GUI (grafiksel kullanıcı arayüzü) geliştirme kütüphanesidir. Diğer arayüz kütüphanelerine göre kullanımının daha kolay ve fonksiyonel olmasından dolayı Kullanıcı Arayüzü geliştirmede Tkinter kütüphanesinin kullanımı tercih edilmiştir.



Şekil 8.1 Kullanıcı Arayüzü

Geliştrilen yazılımın Pycharm IDE'si üzerinden başlatılması ile Şekil 8.1'deki Kullanıcı Arayüzü açılmaktadır. Kullanıcı Arayüzü fonksiyonları aşağıdaki altbaşlıklarda açıklanmıştır.

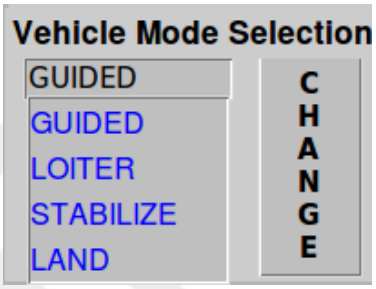
8.1 İHA ile Bağlantı Kurma ve Uçuş Modu Değişikliği

İHA ile yer istasyonu arasında veri alışverişinin başlatılabilmesi için yer istasyonundan İHA ile bağlantı kurulması gerekmektedir. Bu işlem için öncelikle yer istasyonundan İHA'nın Wi-Fi ağına bağlanılması gerekmekte ve ardından Kullanıcı Arayüzü üzerinden "Connect to Solo" butonuna basılması gerekmektedir. Bu buton sayesinde yer istasyonu ile İHA arasında haberleşme başlatılabilmektedir. Kullanıcı Arayüzü'nün kapatılması durumunda kurulan bağlantı kesilmektedir.

Yer istasyonundan İHA'ya bağlanılan port numarası sabittir. Bu port numarasının kullanıcı tarafından değiştirilmesine izin vermeme adına port numarası kod tarafında sabit olarak girilmiştir.



Şekil 8.2 İHA Bağlantı Bölümü



Şekil 8.3 Uçuş Modu Seçimi Bölümü

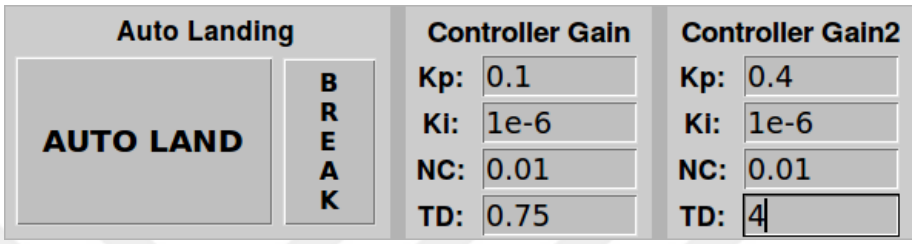
3DR Solo İHA'nın 5 farklı uçuş modu bulunmaktadır. Uçuş testleri esanasında bu uçuş modları arasında geçişlerin yapılması gerekebilmektedir. Bu sebeple, Kullanıcı Arayüzü üzerinde uçuş modlarının değiştirilebilmesi için Şekil 8.3'deki gibi bir bölüm oluşturulmuştur. Uçuş modları arasında geçiş yapabilmek için Şekil 8.3'de yer alan liste kutusu üzerinden geçiş yapılmak istenen uçuş modunun seçilmesi ve ardından "CHANGE" butonuna basılması gerekmektedir. İHA'nın destek verdiği uçuş modları aşağıdaki gibidir:

- GUIDED
- LOITER
- STABILIZE
- RTL
- SPORT

Uçuş modunun herhangi bir döngü içerisinde değiştirilmesi durumunda, Kullanıcı Arayüzü'nün bu bölümünde yer alan aktif uçuş modu bilgisi güncellenmektedir.

8.2 Otonom İnişin Başlatılması ve Kontrol Parametrelerin Ayarlanması

Otonom inişin Kullanıcı Arayüzü üzerinden başlatılabilmesi için arayüz üzerinde Şekil 8.4'deki gibi bir bölüm oluşturulmuştur. Bu bölümde bulunan "AUTO LAND" butonuna basılması ile birlikte otonom iniş başlatılabilmektedir. Otonom inişin herhangi bir anda durdurulması istenildiğinde, "BREAK" butonuna basılarak otonom iniş durdurulabilmektedir. "BREAK" butonuna basıldığı anda İHA'nın x, y ve z eksenlerindeki hızları sıfırlanarak uçuş modu "LOITER" yapılmaktadır.

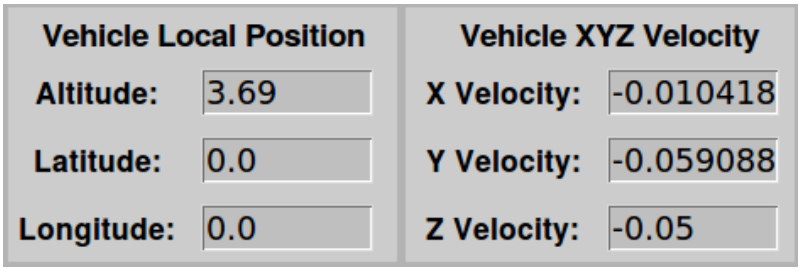


Şekil 8.4 Otonom İniş ve Kontrol Parametreleri Ayarı

İHA'nın otonom inişi esnasında kullandığı PI kontrolcü parametreleri Şekil 8.4'deki bölümden değiştirilebilmektedir. Bu parametreler, otonom inişin gerçekleştiği esnada değiştirilebilir bir yapıya sahiptir. Bu bölmdeki Kp, Ki, NC ve TD katsayıları PI kontrolcülerin sırasıyla oransal, integral, normalizasyon ve zaman katsayılarıdır. "Controller Gain" bölümündeki parametreler x ve y eksen kontrolcülerinde kullanılmakta olup, "Controller Gain2" bölümündeki parametreler z ekseni kontrolcüsünde kullanılmaktadır.

8.3 Uçuş Verilerinin Gözlemlenmesi

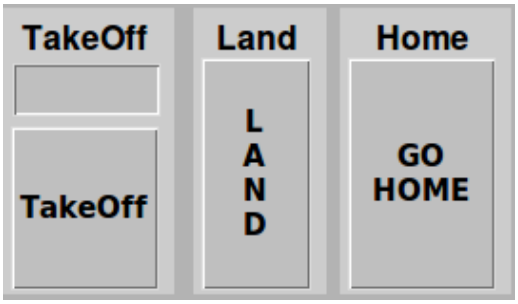
Yer istasyonu ile İHA arasındaki haberleşmenin başlatılması ile birlikte İHA'nın x, y ve z eksenlerindeki anlık pozisyon ve hız verileri alınmakta, Şekil 8.5'deki gibi bu veriler Kullanıcı Arayüzü'nde gösterilmektedir. Pozisyon verilerinin birimi metre, hız verilerinin birimi ise m/s'dir.



Şekil 8.5 Uçuş Hız ve Pozisyon Verileri

8.4 Tanımlanmış Hareket Fonksiyonları

İHA'nın Kullanıcı Arayüzü üzerinden kalkışının ve inişinin yapılabilmesi için önceden tanımlanış hareket fonksiyonları oluşturulmuştur. Bu bölümde Şekil 8.6'de gösterildiği gibi TakeOff, Land ve Home olmak üzere 3 farklı fonksiyon bulunmaktadır.



Şekil 8.6 Tanımlanmış Hareket Noktaları

TakeOff: İHA'nın belirli bir yüksekliğe kalkışı için "TakeOff" bölümünden istenilen yükseklik değerinin girilmesi ve "TakeOff" butonuna basılması gerekmektedir. İHA bu fonksiyon sayesinde belirtilen yükseliğe kalkışını yapmaktadır.

LAND: Uçuşun herhangi bir anında "LAND" butonuna basılması durumunda İHA bulunduğu yükseklikten dikey iniş yapmakta ve uçuşu sonlandırmaktadır.

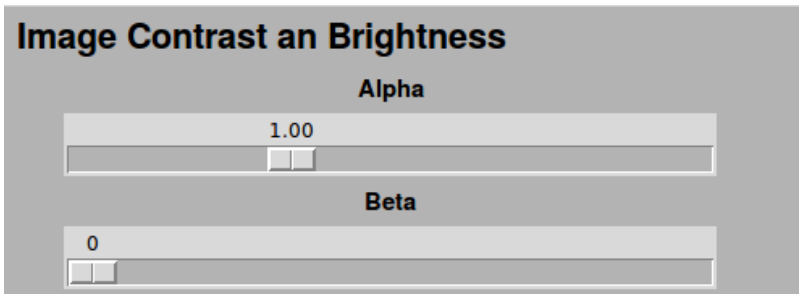
GO HOME: İHA, kalkış yaptığı pozisyonu her zaman ev pozisyonu olarak hafızasında tutmaktadır. "GO HOME" butonuna basılması durumunda İHA, ilk kalkış noktasına inişini yaparak uçuşunu sonlandırmaktadır.

8.5 Kamera Görüntülerinin Ekranda Gösterilmesi ve Görüntü Ayarlarının Yapılması

İHA'nın GoPro Hero4 kamerasından işlenmek üzere alınan görüntüler Şekil 8.1'de gösterildiği şekilde Kullanıcı Arayüzü'nde gösterilmektedir. Otonom inişin aktif olmadığı durumda alınan ham görüntüler ekranda gösterilirken, otonom inişin aktif olduğu durumlarda işlenmiş görüntüler aynı ekranda gösterilmektedir.

İşlenmek üzere alınan görüntülerin ortam ışık şiddetine göre parlaklıklarının ayarlanabilmesi ve yüksekliğe göre görüntü kontrast ayarının yapılabilmesi için Kullanıcı Arayüzü'nde Şekil 8.7'deki gibi ayrı bölüm oluşturulmuştur.

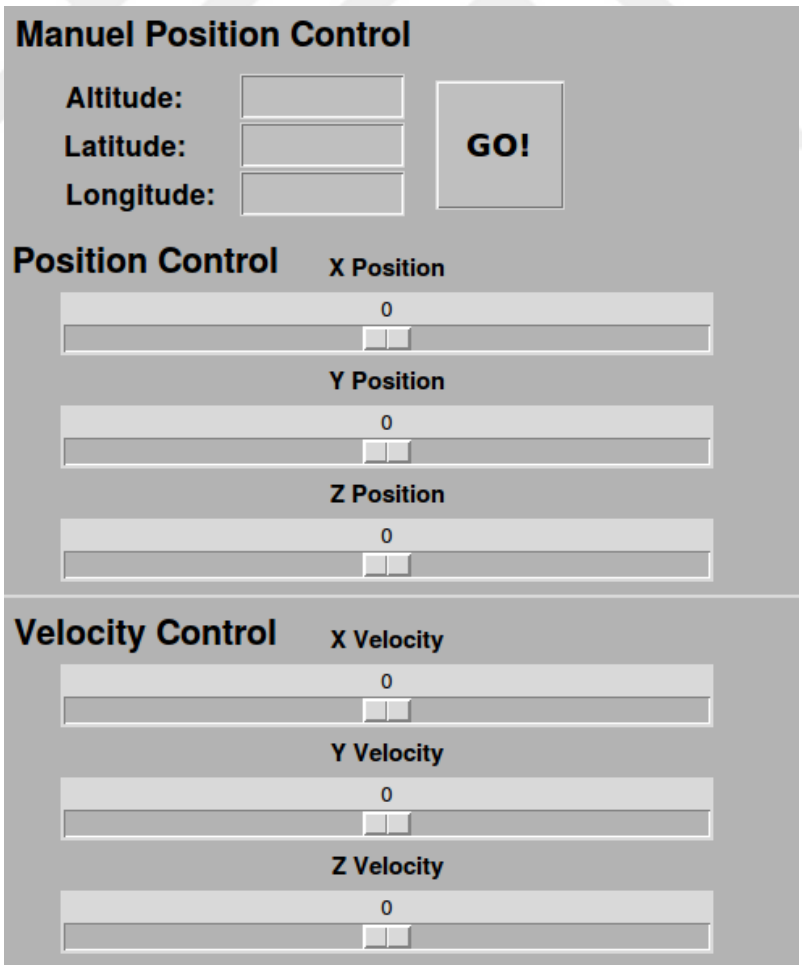
Şekil 8.7'deki Alpha kaydıracı ile görüntü parlaklığı ayarlanmaktadır. Beta kaydıracı ile ise kontrast ayarı yapılmaktadır.



Şekil 8.7 İHA Görüntü Kontrastı ve Parlaklığı Ayarı

8.6 İHA'nın X, Y ve Z Eksenlerinde Doğrusal Hareketi

İHA'nın uzaktan kumanda ile hareket ettirildiği şekilde arayüz üzerinden de hareketinin sağlanabilmesi için Kullanıcı Arayüzü'ne Şekil 8.8'deki gibi pozisyon ve hız referanslarının ayarlanabildiği bir bölüm eklenmiştir. Bu bölümde bulunan kaydıraçlar kullanılarak, İHA'nın NED'e göre x, y ve z eksenlerinde doğrusal pozisyon ve hız referansı ayarlanabilmektedir. Ayrıca, Şekil 8.8'de yer alan "Manuel Position Control" bölümünden NED'e göre koordinatlar girilerek İHA bu koordinatlara gönderilebilmektedir.



Şekil 8.8 X, Y ve Z Eksenlerinde İHA Hareketi

9
Sonuçlar ve Öneriler

9.1 Test ve Ölçüm Çalışmaları

9.1.1 Test Alanı

İHA'nın zeminden kalkışını kararlı bir şekilde yapabilmesi ve uçuş süresi boyunca kontrollü bir şekilde hareket edebilmesi için GPS verilerine ihtyacı bulunmaktadır. Kapalı ortamda GPS modulü yeterli sayıda uydu bulamadığı için testler açık ortamda gerçekleştirilmiştir. İHA ile yapılan uçuş ve iniş testleri, 40.868483-29.331674 koordinatlarında olan Şekil 9.1'deki alanda gerçekleştirilmiştir.



Şekil 9.1 Test Alanı Google Maps

İHA'nın uçuş testleri esanasında İHA'yı kaybetmeye yol açabilecek etkilere karşı tüm yazılımsal önlemler alınmıştır. Aynı zamanda, İHA'nın harketini kolaylıkla durdurabilecek şekilde Kullanıcı Arayüzü'nde bölümlere yer verilmiştir. Ayrıca, ilk iniş testlerinde oluşabilecek her türlü olumsuz durumun önüne geçme amaçlı olarak

İHA bir ip vasıtasıyla bağlanarak zeminden yönetilebilir hale getirilmiş ve ilk testler bu şekilde yapılmıştır.



Şekil 9.2 Test Ortamından Görüntü - 1

İHA'nın algoritma enyileştirme süreçlerinde çok sayıda test yapılmıştır. Yapılan testler esnasında yer istasyonunun ekran görüntüleri kayıt edilmiştir. Aynı zamanda, İHA'nın uçuş verileri ve kontrol algoritmasında bir çok parametre zaman bilgisi ile birlikte bir .CSV dosyasına kayıt edilmiştir.

9.1.2 Uçuş ve İniş Testi Veri Kayıtları

İniş algoritmasının eniyileştirilmesi aşamasında kontrolde kullanılan veriler başta olmak üzere birçok verinin işlem sonrası incelenmesi gerekmektedir. Bu sebeple, İHA'nın uçuş verileri ve kontrol algorimasındaki öncelikli veriler zaman bilgisi ile birlikte .CSV dosya formatında kayıt edilmiştir. Bu kayıt için Python dili içerisinde bulunan csv kütüphanesi kullanılmıştır.

Testler esnasında kayıt altına alınan veriler aşağıdaki gibidir:

- Helipad, H harfi orta piksel konumu (x ve y eksen indeksi)
- Rota açısına göre x ve y ekseni referans ve ölçülen hız değerleri

- NED'e göre x, y ve z ekseni referans ve ölçülen hız değerleri
- x, y ve z eksenleri pozisyon bilgisi
- Rota Açısı
- İHA'nın havaya ve yere göre hızları
- Görüntü işleme ile hesaplanan İHA yükseklik bilgisi
- Yüksekliğe göre histeresis bölgesi (piksel)
- R_p (mm/piksel) bilgisi
- İHA'nın yunuslama, sapma ve yuvarlanma açıları
- İHA'da bulunan kamera tutacağının(Gimbal) yunuslama, sapma ve yuvarlanma açıları
- Bulunan uydu sayısı
- Batarya gerilimi, akımı ve doluluk seviyesi
- Uçuş modu bilgisi



Şekil 9.3 Test Ortamından Görüntü - 2

9.1.3 İniş Testi Videoları

İHA'nın dış ortamdaki iniş testi videolarına aşağıdaki linklerden veya karekodlardan ulaşabilirsiniz:



Şekil 9.4 Test-1 İHA Videosu



Şekil 9.5 Test-1 Yer İstasyonu Videosu



Şekil 9.6 Test-2 İHA Videosu



Şekil 9.7 Test-1 Yer İstasyonu Videosu

Linkler:

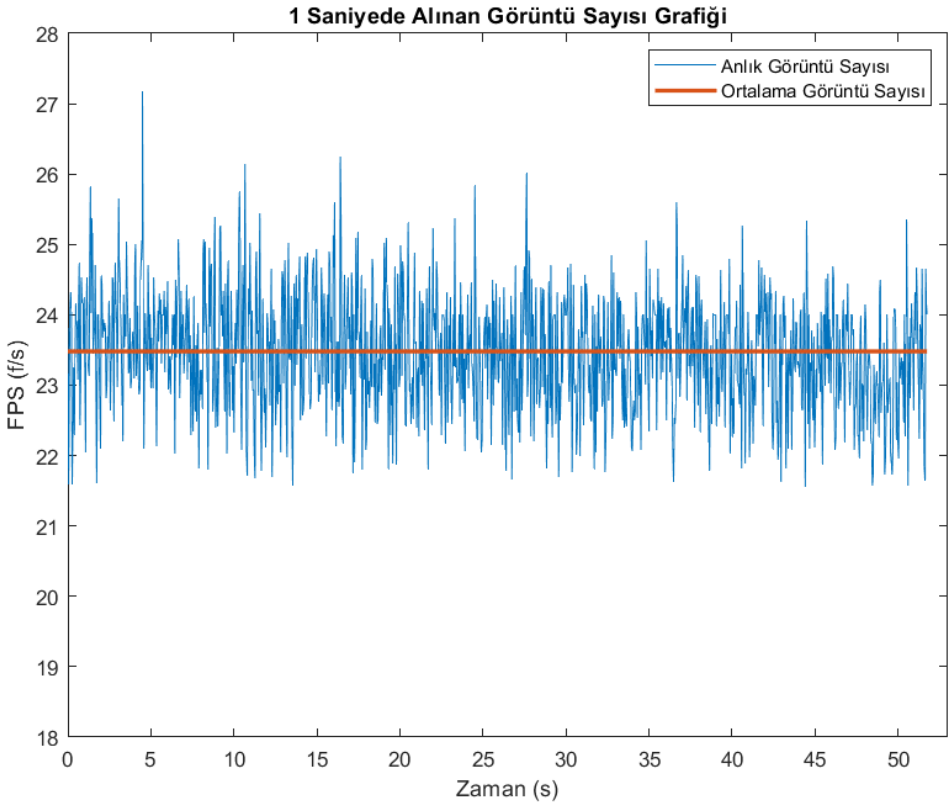
- Test-1 İHA Videosu : https://www.youtube.com/watch?v=UfAQ42hvVtw
- Test-1 Yer İstasyonu Videosu : https://www.youtube.com/watch?v=GB042CspOqI
- Test-2 İHA Videosu : https://www.youtube.com/watch?v=D-mAcJn9D4Y
- Test-2 Yer İstasyonu Videosu : https://www.youtube.com/watch?v=4by6qELCFY

9.2 Sonuç ve Ölçüm Grafikleri

9.2.1 Görüntü Alma ve Görüntü İşleme Performansı

Bölüm 4.2'te anlatıldığı şekilde İHA'dan yer istasyonuna görüntü aktarımı başarıyla sağlanmıştır. İHA'dan 1 saniyede alınan görüntü sayısını bulabilmek için yaklaşık 50

saniyelik test yapılmıştır. Test süresi boyunca alınan her bir görüntünün süresi kayıt edilmiş ve bu süreden saniyede alınan görüntü sayısı hesaplanarak Şekil 9.8'deki gibi grafik halinde sunulmuştur.

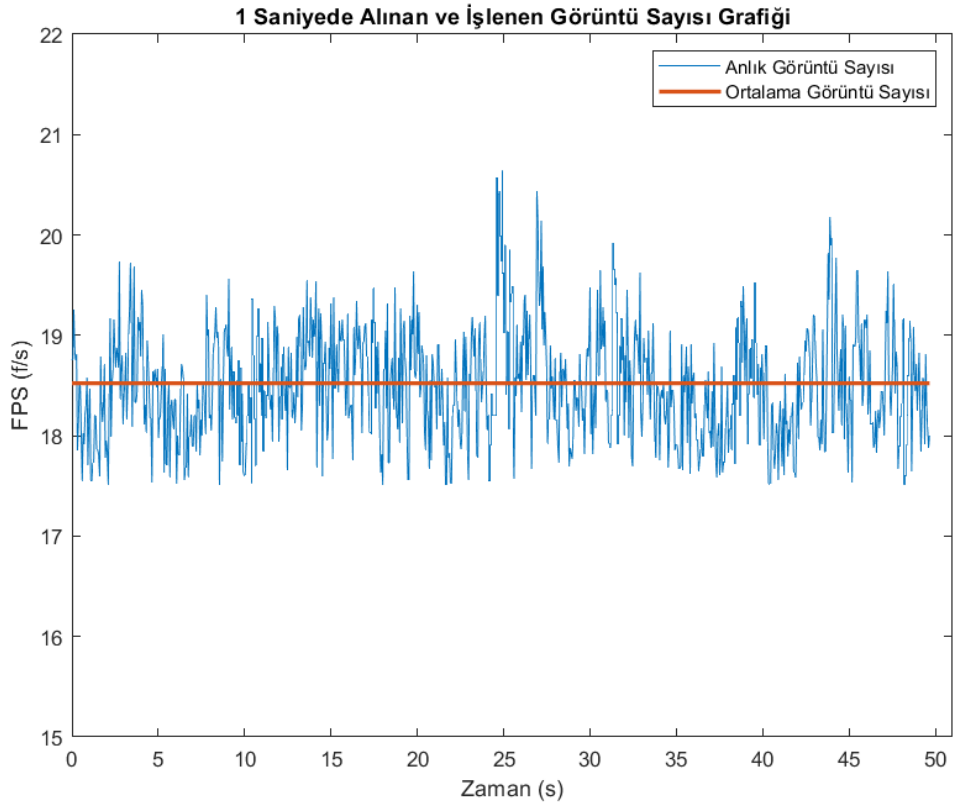


Şekil 9.8 İHA'dan 1 Saniyede Alınan Görüntü Sayısı

Herhangi bir görüntü işleme yapmaksızın İHA'dan görüntünün alması ve bu görüntünün Kullanıcı Arayüzü'nde göstermesi sırasında geçen süre yaklaşık 42,5 ms olarak ölçülmüştür. Bu değer, bir saniyede yaklaşık olarak 23-24 tane görüntü alınıdığını göstermektedir.

İHA'dan alınan görüntülerin işlenmesi, kontrol döngülerinin çalıştırılması, İHA'ya kontrol komutlarının gönderilmesi, uçuş verilerinin kayıt edilmesi ve Kullanıcı Arayüzü'nde bu verilerin paylaşılması durumunda görüntü başına harcanan süreyi tespit edebilmek için ayrı bir test yapılmıştır. Bu test sonucunda tek bir çevrimin tamamlanması için geçen süre yaklaşık olarak 54 ms olarak ölçülmüştür. Bu değer, bir saniyede yaklaşık olarak 18-19 adet görüntünün işlendiğini göstermektedir. Aynı zamanda, iniş için tasarlanan kontrol çevrimlerinin de yaklaşık olarak 20 Hz frekansta çalıştığını göstermektedir.

Bir saniye içerisinde İHA'dan alınan ve işlenen görüntü sayısını gösteren grafik Şekil 9.9'deki gibi sunulmuştur.



Şekil 9.9 İHA'dan 1 Saniyede Alınan ve İşlenen Görüntü Sayısı

9.2.2 Görüntü İşleme ile Hesaplanan Yükseklik Verisi Doğruluğu

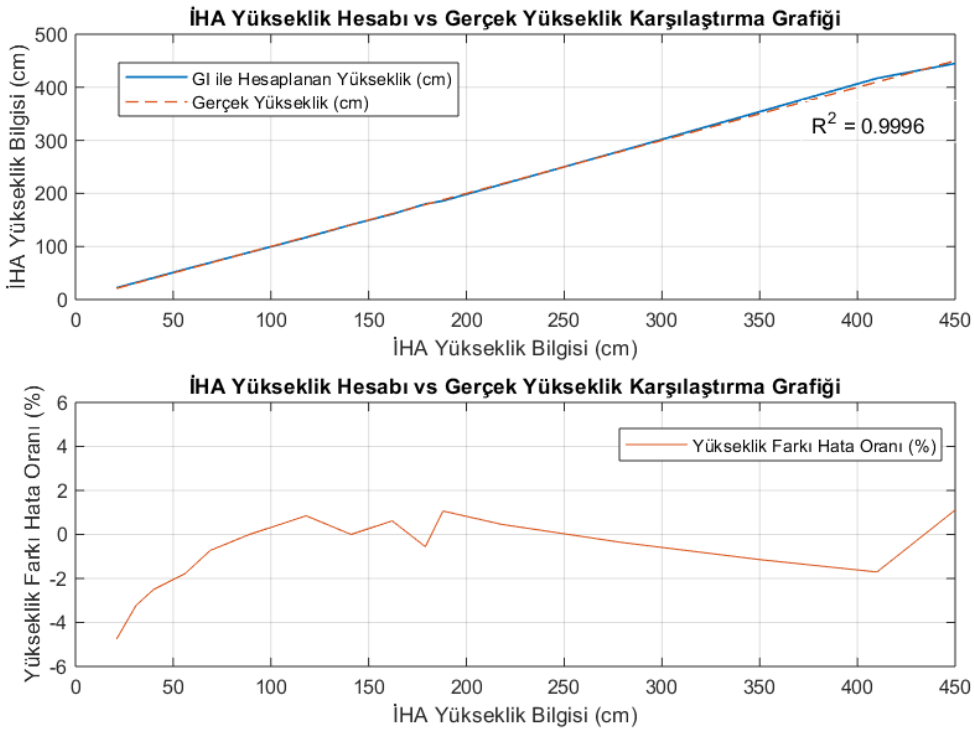
İHA'nın görüntü işleme ile bulunan yükseklik değeri ile gerçek yüksekliği karşılaştırılarak Şekil 9.10'deki grafik oluşturulmuştur. 300 cm'e kadar olan bölgede yükseklik farkının çok düşük olduğu gözlemlenmiştir. Bu farkın 300 cm sonrasında bir miktar arttığı fakat bu artış durumunda dahi %1 civarında seyrettiği gözlemlenmiştir. Helipad büyüklüğünün arttırılması ya da daha yüksek çözünürlüklü kameranın kullanılması durumunda daha yüksek mesafelerde yapılan testlerde bu farkın minimum olacağı öngörülmektedir.

9.2.3 Görüntü İşleme Tabanlı İniş Testi Verileri

İHA ile yapılan uçuş testleri esnasında ölçülen ve kayıt edilen verilerin grafikleri aşağıdaki gibi aktarılmıştır.

İHA'nın görüntü işleme ile Helipad'e inişi esanasında kayıt edilen GPS verilerinin grafiği Şekil 9.11'deki gibi sunulmuştur.

İHA'nın Helipad'e inişi esnasında X ve Y eksenlerinde piksel olarak pozisyon kontrol grafiği Şekil 9.12'deki gibi sunulmuştur. İHA, Helipad'in tam olarak üzerinde olduğu



Şekil 9.10 Görüntü İşleme ile Bulunan İHA Yükseklik Bilgisi Doğrulaması

durumda, Helipad'te yer alan H harfinin orta noktası kameradan alınan görüntünün orta noktasında denk gelmektedir. İHA'dan alınan görüntünün tam orta noktasındaki piksel değeri 640 ve 360'dır. Bu sebeple, X ekseni pozisyon kontrolcüsünün referans piksel değeri 640, Y ekseni pozisyon kontrolcüsünün referans piksel değeri 360 olarak girilmiştir. Bu şekilde piksel üzerinden pozisyon kontrolü yapılarak İHA Helipad'in üst noktasına gönderilmiştir.

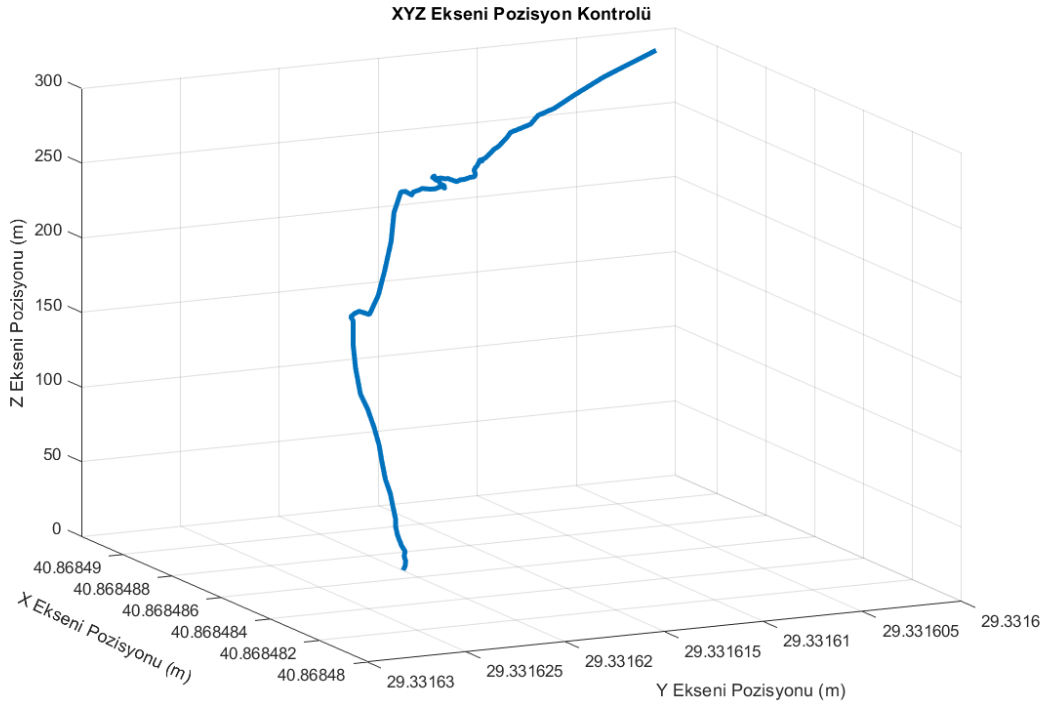
İHA'nın Helipad'e inişi esnasında X, Y ve Z eksenlerinin NED'e göre hız kontrolünde kullanılan referans ve ölçülen hız değerlerinin zamana bağlı grafikleri Şekil 9.13'deki gibi sunulmuştur.

İHA'nın Helipad'e inişi esnasında X ve Y eksenlerinin rota açısına göre hız kontrolünde kullanılan referans ve ölçülen hız değerleri Şekil 9.14'deki gibi sunulmuştur.

İHA'nın Helipad'e inişi esanasında görüntü işleme yönemi ile hesaplanan İHA yükseklik verisinin zamana bağlı grafiği Şekil 9.15'deki gibi sunulmuştur.

İHA'nın Helipad'e inişi esanasında yüksekliğe göre değişen mm/piksel oranı grafiği grafiği Şekil 9.16'deki gibi sunulmuştur.

İHA'nın Helipad'e inişi esanasında havaya ve yere göre hızlarının zamana bağlı grafiği Şekil 9.17'deki gibi sunulmuştur.



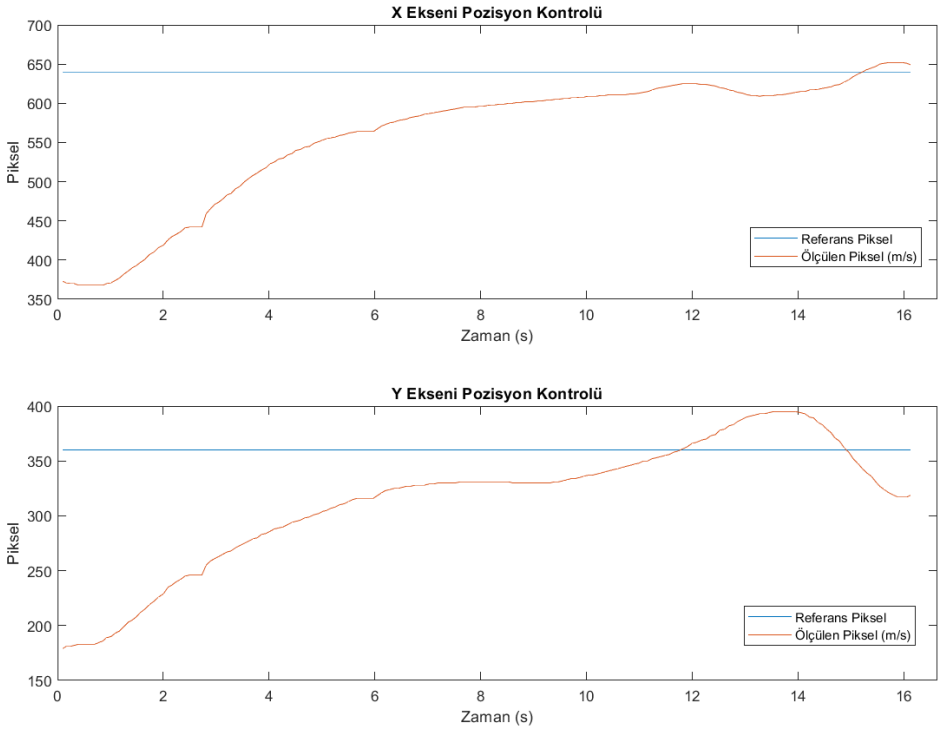
Şekil 9.11 İHA'nın Helipad'e İnişi Esnasında X, Y ve Z Eksen Pozisyonları

İHA'nın Helipad'e inişi esanasında yunuslama, sapma ve yuvarlanma açılarının zamana bağlı grafiği Şekil 9.18'deki gibi sunulmuştur.

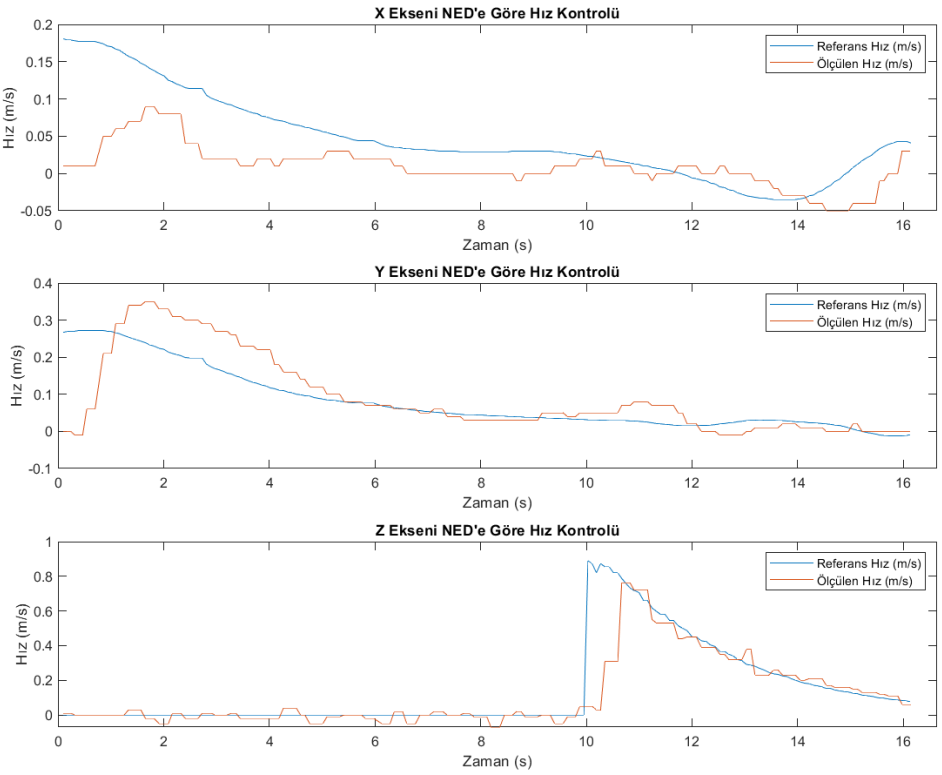
9.2.4 Değerlendirme

Bu proje kapsamında iligli başlıklarda belirtilen çalışmaların tamamı gerçekleştirilmiş olup çıktılar bu bölümün içerisinde paylaşılmıştır. Görüntü işleme, İHA ile haberleşme, kontrol algoritma geliştirme, test ve uygulama gibi ana çalışmalar düzenli ve sistematik bir şekilde gerçekleştirilmiştir. Çalışma içerisinde geliştirilen tüm algoritmaların sistem seviyesi ve altsistem seviyesi etkileri değerlendirilmiş olup, bu algoritmalar yapılan uçuş ve iniş testleriyle birlikte doğrulanmıştır.

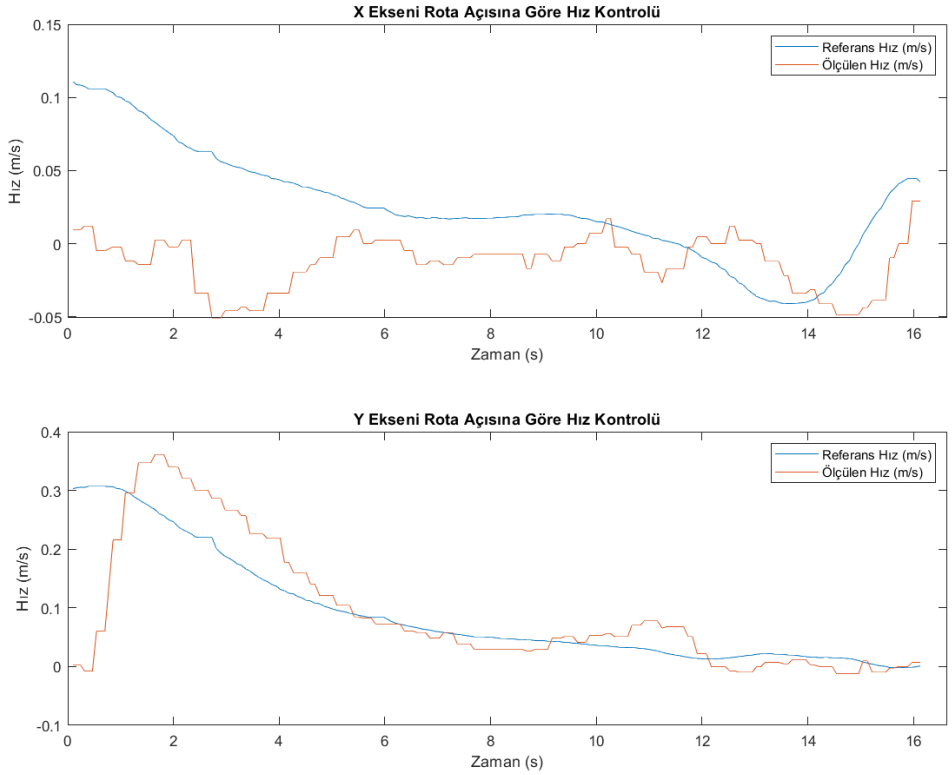
Bu çalışma sonucunda, düşük maliyetli hareket sensörleri ve kameradan alınan görüntüler birleştirilerek İHA'nın Helipad üzerine çok hasssas ve güvenli bir şekilde iniş yapması mümkün kılınmıştır. Bu çalışma sayesinde, MEMS tabanlı sensörlerle geliştirilen otonom iniş sistemleride yaşanılan performans problemlerinin önüne geçilmiş, İHA'nın çok daha düşük maliyetli çözümlerle hedefe iniş yapması sağlanmıştır. Aynı zamanda, İHA'nın iniş güvenliğinin sağlanamamasından kaynaklı oluşabilecek görev başarısızlığı, çevre kirliliği, maddi hasarlar ve hayati tehlikelerin önüne geçilmiştir.



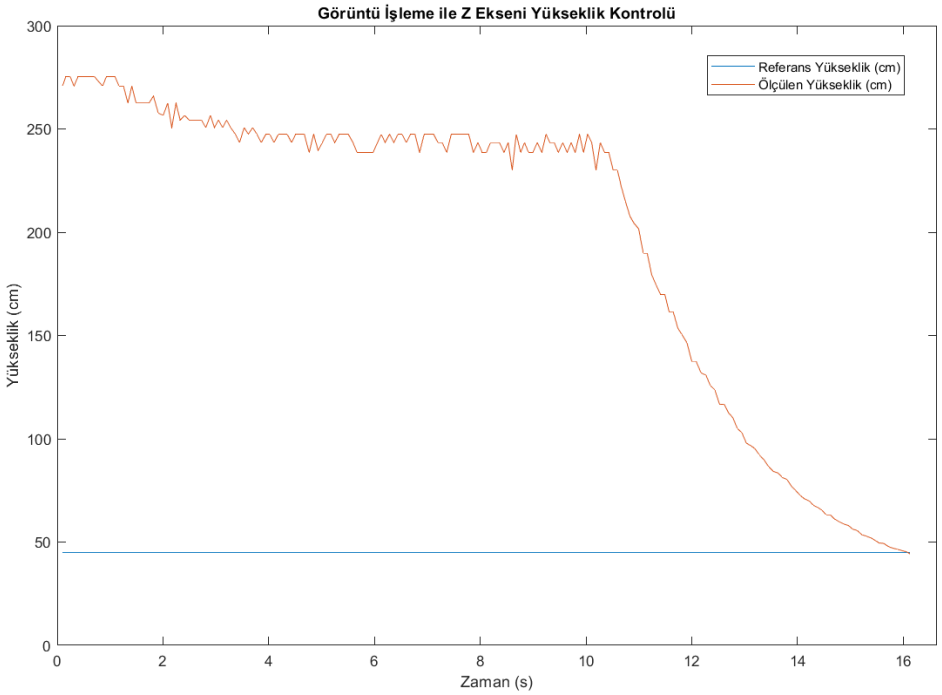
Şekil 9.12 X ve Y Eksenleri Pozisyon Kontrol Grafikleri



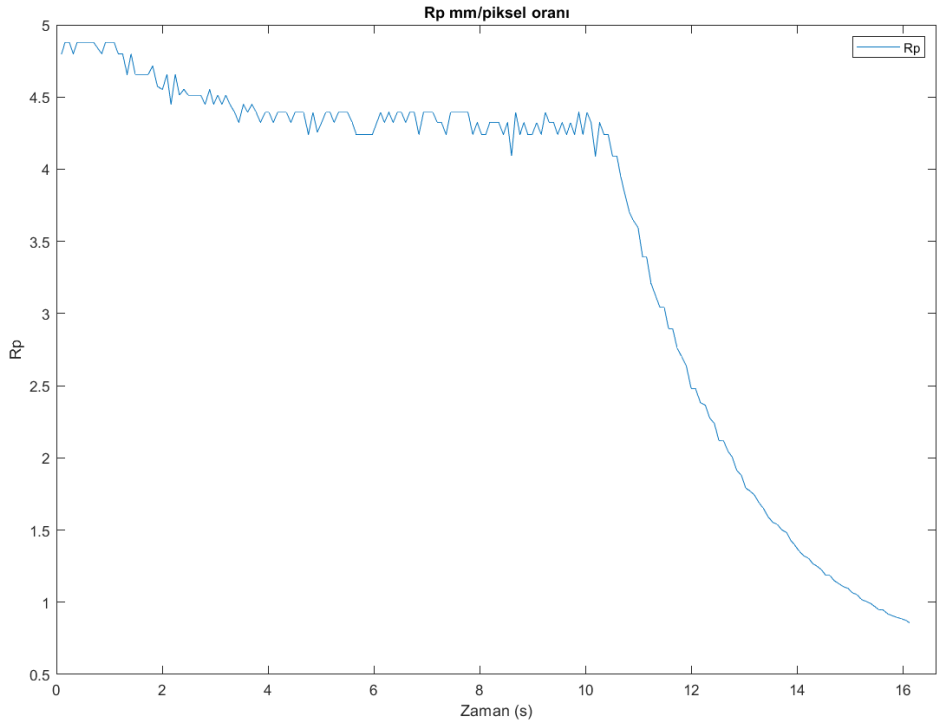
Şekil 9.13 İHA NED'e Göre X Y ve Z Eksen Hız Kontrol Grafiği



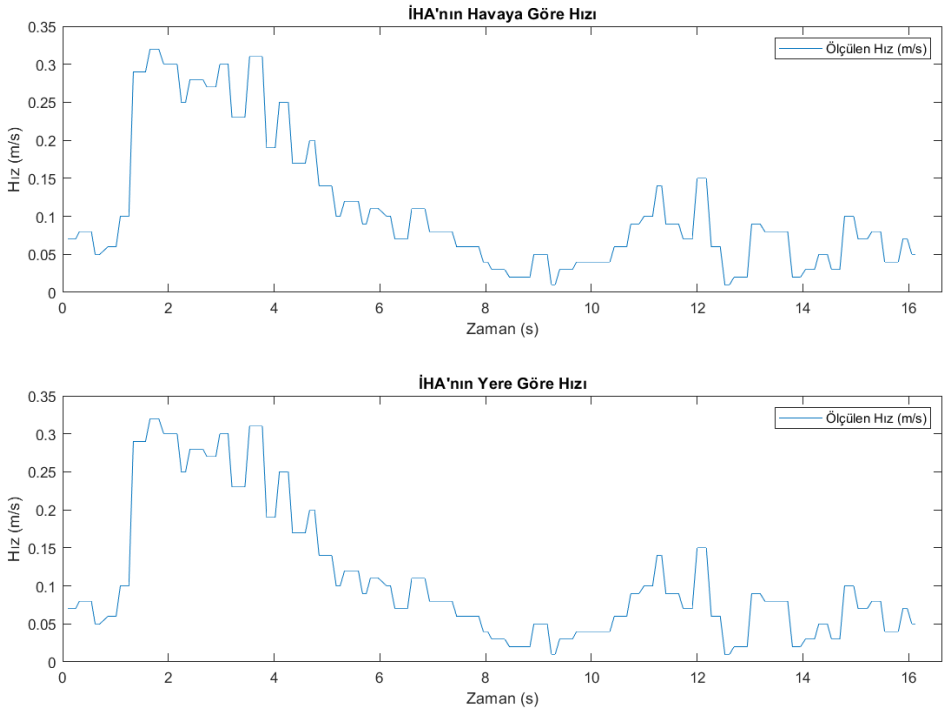
Şekil 9.14 Rota Açısına Göre X ve Y Eksen Hız Kontrol Grafikleri



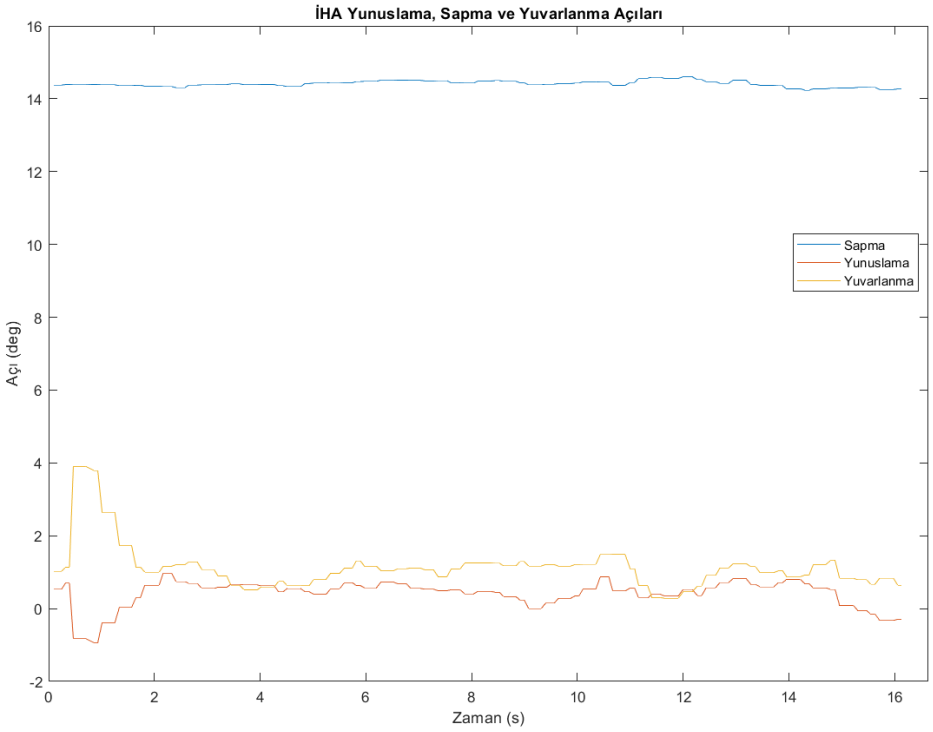
Şekil 9.15 Görüntü İşleme İle Hesaplanan Zamana Bağlı Yükseklik Grafiği



Şekil 9.16 Rp(mm/piksel) Oranı Grafiği



Şekil 9.17 İHA'nın Havaya ve Yere Göre Hızı



Şekil 9.18 İHA'nın Uçuşu Sırasında Yunuslama, Sapma ve Yuvarlanma Açıları

Bu çalışmanın, otonom iniş sistemlerinin yerlileştirilmesinde başlangıç olabilecek seviyede olması ve dışa bağımlılığı azaltıcı olumlu etkilerinin olmasından dolayı ülke kazanımı açısından önem seviyesi yüksektir.

9.3 Öneriler

İHA'nın görüntü işleme tabanlı otonom inişini güvenli bir şekilde gerçekleştirebilmesi için gerekli tasarım ve testler yapılmıştır. Bu tez çalışması kapsamında olmayıp, gelecekte yapılması planlanan çalışmalar aşağıdaki başlıklarda aktarılmıştır.

9.3.1 Hareket Eden Platformlara İniş Kabiliyeti

Şuana kadar yapılan çalışmalarda İHA, herzaman sabit duran Helipad'e inişini gerçekleştirmektedir. İHA'nın hareketli platformlara iniş yapılabilmesi için çalışma performansının iyileştirilmesi gerektiği gözlemlenmiştir. Bu sebeple, ilerleyen çalışmalarda İHA'nın çalışma performansı iyileştirilip İHA'ya hareket eden platformlara iniş yapabilme yetkinliği kazandırılacaktır.



Şekil 9.19 İHA'nın Hareket Eden Platformlara İnişi

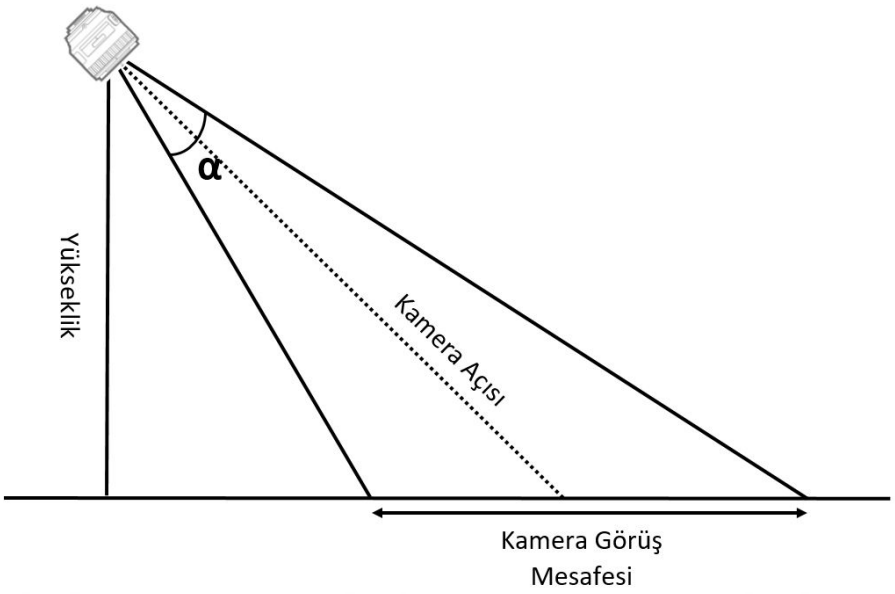
9.3.2 Kamera Bakış Açısının Değişimine Göre İniş Algoritmasının Düzenlenmesi

İHA'da bulunan kamera, hali hazırdaki çalışmada her zaman zemine paralel olacak şekilde tutulmaktadır. Kamera açısının zemine paralel tutulduğu durumda İHA'nın bulunduğu yükseklikten görebildiği alan sınırlı olmaktadır. Bu sebeple İHA, iniş yapacağı hedef platformu bulmakta zorlanmaktadır. Gelecek dönemlerde yapılacak çalışmalarla İHA'nın kamera açısı ayarlanabilir hale getirilecek olup, İHA'nın çok daha uzak mesafelerde bulunan hedeflere iniş yapılabilmesi sağlanacaktır.

İHA yüksekliği Şekil 9.20'de gösterildiği gibi hesaplanacaktır. Aynı zamanda, İHA'yı Helipad'in üst noktasına gönderirken kamera açısının Helipad'in konumuna göre ayarlanması planlanmaktadır.

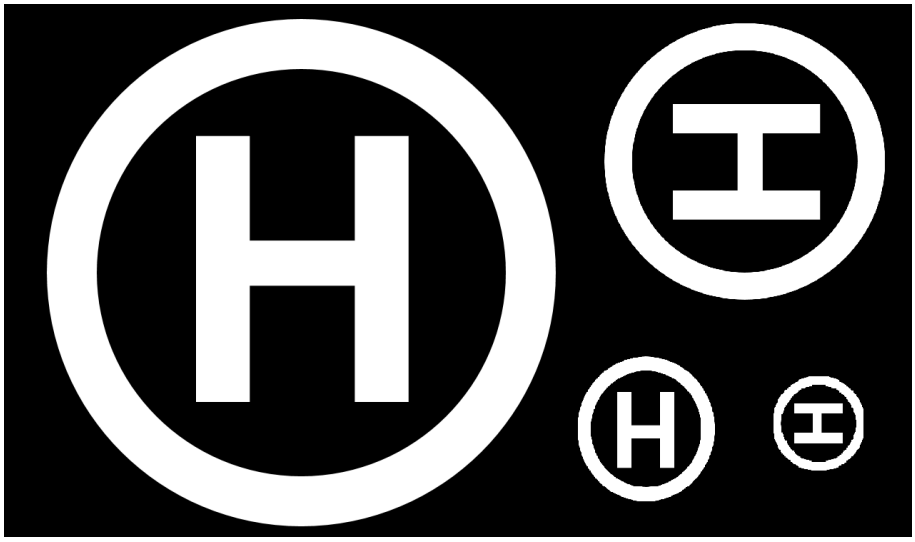
9.3.3 İniş Platformu Değişikliği

İHA, Başlık 5'de aktarılan ölçülerdeki Helipad'e iniş yapmaktadır. İHA'dan yer istasyonuna alınan görüntülerin çözünürlüğü ve Helipad'in büyüklüğü, maksimum



Şekil 9.20 Kamera Açısı Değişimi

inişe başlanacak yükseği etkileyen en önemli parametrelerdir. Bu yüksekliği arttırabilmek için kamera çözünürlüğünü değiştirmek çok etkili ve kolay bir yöntem değildir. Bu yüzden, Helipad'in büyüklüğü değiştirilerek Helipad'in tespit edilebildiği maksimum yükseklik seviyesi arttırılabilmektedir. Helipad'in büyüklüğü arttırıldıkça İHA tarafından daha yüksek irtifalardan görünebilir olacaktır fakat tam tersi olarak kamera açısına girmeyeceği için düşük irtifalarda da görülemeyecektir. Bu gibi sebeplerden dolayı iniş yapılacak platformda Şekil 9.21'deki gibi farklı büyüklüklerde Helipad bulundurulması planlanmaktadır. Bu sayede farklı yükseklik seviyerlerinde Helipad görülebilir olacak ve İHA inişini yapabilecektir.



Şekil 9.21 Kullanılabilecek Muhtemel İniş Platformu

9.3.4 Kapalı Ortamlarda Test Kabiliyeti

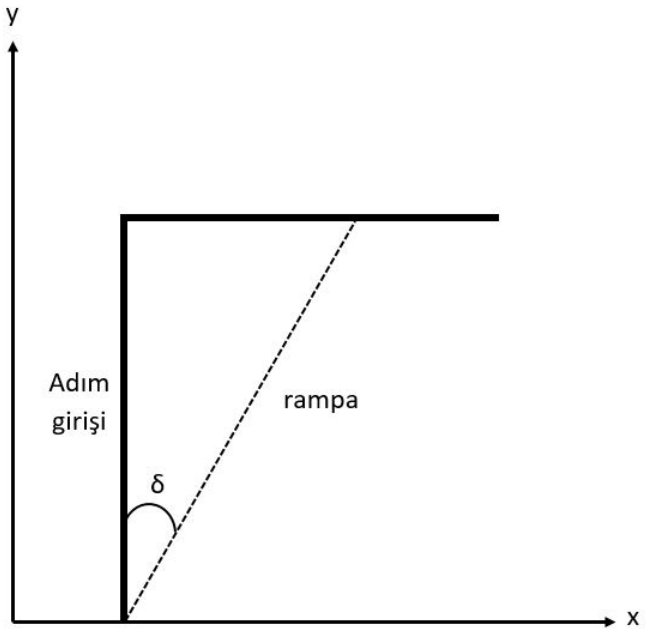
İHA'larda kullanılan GPS modülleri kapalı ortamlarda kararlı bir şekilde çalışmamaktadır. Bu nedenle, GPS verilerini kullanarak uçuş yapan İHA'lar kapalı ortamlarda kullanılamamaktadır. Geliştirilen iniş algoritmalarının kapalı ortamlarda çalışan İHA'lar için kullanılabilir hale getirilmesi önemlidir. Bu sebeple, ilerleyen aşamalarda İHA'ya GPS verisi kulanmadan uçuş ve iniş yapabilme kabiliyeti kazandırılması planlanmaktadır.

3DR Solo İHA, halihazırda kullanılan "GUIDED" modunda GPS verilerini kullanarak uçuş yaptmakta ve bahsedilen sebeplerden dolayı "GUIDED" modunda kapalı alanlarda kullanılamamaktadır. Bu İHA'nın GPS verisi kullanmadan uçuş yapabildiği "STABILIZE" ve "GUIDEDNoGPS" modu bulunmaktadır. Yer istasyondan sadece "GUIDEDNoGPS" modunda kontrol komutları gönderilebildiği için ilerleyen aşamalarda "GUIDEDNoGPS" modu ile İHA'ya iniş yaptırılması planlanmaktadır.

İHA'nın z ekseninde yükseklik kontrolünün sağlanabilmesi için Başlık 6'de aktarıldığı şekilde yükseklik verisi görüntü işleme yöntemiyle hesaplanabilmektedir. Bu sebeple, hedefe inişte kullanılan yükseklik verisi için GPS verilerinin olması zorunlu değildir. Aynı şekilde x ve y eksenlerinde pozisyon kontrolü piksel üzerinden yapıldığı için bu bölümde de GPS verilerinin kullanımı zorunlu değildir. Sonuç olarak, geliştirilen iniş kontrol algoritması, kapalı ortamlarda çalışabilen İHA'lar için tekrar düzenlenerek ve gerekli iyileştirmeler yapılararak uygulanabilir durumdadır. Gelecek çalışmalarda bahsedilen doğrultuda değişiklikler yapılarak İHA'ya kapalı ortamlarda otonom bir şekilde iniş yaptırılması planlanmaktadır.

9.3.5 Pozisyon Referans Komutu İyileştirilmesi

İHA'nın uçan bir araç olması sebebiyle pozisyon kontrol bant genişlikleri düşüktür. Bu nedenle, İHA'nın adım(step) girişi şeklinde uygulanan referanslara verdiği cevaplara bakıldığında oturma sürelerinin uzun olduğu görülmüştür. İHA'nın pozisyon kontrol cevaplarını iyileştirme ve geçici(transient) durumlarda daha yumşak hareket etmesini sağlama amacıyla halihazırda adım şeklinde uygulanan referansların rampa şeklinde uygulanması planlanmaktadır. Adım şeklinde gelen referans pozisyon değerinin bir rampa fonksiyonuyla rampa şekline dönüştürülmesi planlanmaktadır.



Şekil 9.22 Adım Girişi Rampa Çevrimi

Referanslar

[1] S. Saripalli, J. Montgomery, and G. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," *Proceedings 2002 IEEE International Conference on Robotics and Automation*, 2002.

[2] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying uav," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. 12317917, 2011.

[3] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous landing of an unmanned aerial vehicle," *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2017.

[4] H. Lee, S. Jung, and D. H. Shim, "Vision-based uav landing on the moving vehicle," *International Conference on Unmanned Aircraft Systems (ICUAS)*, no. 16123969, 2016.

[5] C. Patruno, M. Nitti, E. Stella, and T. Dorazio, "Helipad detection for accurate uav pose estimation by means of a visual sensor," *International Journal of Advanced Robotic Systems*, pp. 1–15, 2017.

[6] T. Nakamura, S. Haviland, D. Bershadsky, and E. Johnson, "Vision based optimal landing on a moving platform," *72nd American Helicopter Society International Annual Forum*, vol. 4, no. 11, pp. 3331–3341, 2016.

[7] D. Lee, T. Ryan, and J. Kim, "Autonomous landing of a vtol uav on a moving platform using ımage-based visual servoing," *2012 IEEE International Conference on Robotics and Automation*, pp. 971–976, 2012.

[8] C. Karaağaç, "Geleceğin insansız hava aracı sistemlerinde uçuş emniyeti," *Havacılık Emniyeti Yönetim Sistemi Sempozyumu*, pp. 41–66, 2018.

[9] M. Kahveci and N. Can, "İnsansız hava araçları: Tarihçesi, tanımı, dünyada ve türkiye'deki yasal durumu," *Selçuk Üniversirtesi Mühendislik Bilim ve Teknoloji Dergisi*, vol. 5, no. 4, pp. 511–535, 2017.

[10] T. John W. K., "Jane's pocket book of remotely piloted vehicles," 1977.

[11] G. R. I. Shaw, "'scorched atmospheres: The violent geographies of the vietnam war and the rise of drone warfare," *Annals of the American Association of Geographers*, vol. 106, no. 3, pp. 688–704, 2016.

[12] R. Sanders, "The drone revolution: Less than meets the eye," *Joint Forces Quarterly*, 2002.

[13] L. Freedman, "The drone revolution: Less than meets the eye," *Foreign Policy*, vol. 95, no. 6, pp. 153–158, 2016.

[14] N. Kenzo, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, "Autonomous flying robots," Springer, 2010.

[15] G. Hoffmann, D. Rajnarayan, S. Waslander, D. Dostal, J. Jang, and C. Tomlin, "'the stanford testbed of autonomous rotorcraft for multi agent control," *The 23rd Digital Avionics Systems Conference*, pp. 2–10, 2004.

[16] J. Blazek, "Leakage magnetıc fields of the electrıc bldc motors of small uav," *Journal of ELECTRICAL ENGINEERING*, vol. 66, no. 7, pp. 14–17, 2015.

[17] L. Brown, *Drone motors: Choose the best motors for your quadcopter*, https://filmora.wondershare.com/drones/drone-motors.html, [Online; accessed 01-January-2019], 2017.

[18] M. G. Yoon, "On driving signal of electronic speed controller for small multi-rotor helicopter," *International Journal of Engineering Research and Technology*, vol. 4, no. 11, pp. 456–459, 2015.

[19] D. C. TOSUN, "Dört motorlu bir insansız hava aracı için adaptif kontrol sistem tasarımı," Master's thesis, Anadolu Üniversitesi, 2015.

[20] 3DRobotics, *Solo user manuel*, English, version Version 8, 3DR, 81 pp., 2015.

[21] Python, *General python faq*, https://docs.python.org/3/faq/general.html, [Online; accessed 05-January-2019],

[22] B. Kepil, *Python programlama nedir? ne işe yarar? nerelerde kullanılır?* https://teknolojiprojeleri.com/programlar/python-nedir-ne-ise-yarar-nerelerde-kullanilirl, [Online; accessed 03-March-2019], 2019.

[23] 3DRobotics, *About dronekit*, http://python.dronekit.io/about/overview.html/open-source-community, [Online; accessed 01-March-2019], 2016.

[24] S. Nagar, "Introduction to matlab for engineers and scientists," in *Solutions for Numerical Computation and Modeling*, 2017, pp. 1–217.

[25] Mathworks, *Products and services*, https://www.mathworks.com/products.html?s_tid=gn_ps, [Online; accessed 05-March-2019], 2019.

[26] G. Bradski and A. Kaehler, "Learning opencv," in *Computer Vision with the OpenCV Library*, OREILLY, 2008, pp. 1–571.

[27] O. Team, *About opencv*, https://opencv.org/about.html, [Online; accessed 04-March-2019],

[28] E. Gizem, *Pycharm nedir? nasıl kurulur?* http://www.elifelse.com/python/pycharm-nedir-nasil-kurulur, [Online; accessed 19-March-2019], 2018.

[29] MAVLINK, *Mavlınk haberleşme protokolü*, https://mavlink.io/en/, [Online; accessed 04-March-2019], 2019.

[30] GoPro, *Available video resolutions for hero4 black and silver*, https://gopro.com/help/articles/block/Available-Video-Resolutions-for-HERO4-Black-and-Silver, [Online; accessed 07-March-2019], 2018.

[31] İ. Uçar, *Linux sistemlerde netcat (nc) aracının kullanımı*, http://ucribrahim.blogspot.com/2015/03/linux-sistemlerde-netcat-nc-aracnn.html, [Online; accessed 10-March-2019], 2015.

[32] 3. Solo, *Accessing solo*, https://dev.3dr.com/starting-network.html, [Online; accessed 10-March-2019], 2015.

[33] V. Organization, *Vlc features*, https://www.videolan.org/vlc/features.html, [Online; accessed 11-March-2019], 2017.

[34] D. H. M. İşletmesi, *Helipad*, https://www.dhmi.gov.tr/Sayfalar/terimler.aspx?Tip=EN-TR&A=helipad, [Online; accessed 12-March-2019],

[35] "Standards for offshore helicopter landing areas," Civil Aviation Authority, Standard, 2018.

[36] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simplefeatures," *Computer Vision and Pattern Recognition Conference*, pp. 1–9, 2001.

[37] C. Papageorgiou, M. Oren, and T. Poggio, "General framework for object detection," *International Computer Vision Conference*, no. 555-562, 1998.

[38] C. Harris and M. Stephens, "A combıned corner and edge detector," *Plessey Research Roke Manor*, vol. 10, no. 2, pp. 147–152, 1988.

[39] T. L. Chao and K. H. Wong, "An efficient fpga implementation of the harris corner feature detector," *14th IAPR International Conference on Machine Vision Applications (MVA)*, vol. 18, no. 22, pp. 89–93, 2015.

[40] *Neo-7 series gnss module*, NEO-7, Rev. 7, UBlox, 2015.

[41] GoPro, *Hero4 field of view (fov) ınformation*, https://gopro.com/help/articles/Question-Answer/HERO4-Field-of-View-FOV-Information, [Online; accessed 01-January-2019],

[42] P. Kaniewski and J. Kazubek, "Integrated system for heading determination," *Optical and Acoustical Methods in Science and Technology*, vol. 116, no. 3, pp. 325–330, 2009.

Tezden Üretilmiş Yayınlar

İletişim Bilgileri: alivurgun92@gmail.com

Konferans Bildirisi

1. A. Vurgun and A. Yeşildirek, "İnsansız Hava Araçlarında Görüntü İşleme Tabanlı İniş Sistemi Tasarımı ve Gerçeklemesi", 4. Uluslararası Bilimsel Araştırmalar Kongresi, pp. 251-261, 2019