

T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

DIKEY İNİŞ KALKIŞLI İNSANSIZ HAVA ARAÇLARI İÇİN  
GÖRSEL İŞARETLEYİCİ SİSTEMLER İLE GÖRÜNTÜ  
TABANLI KONUM KESTİRİMİ

Mücteba UZUNOĞLU

YÜKSEK LİSANS TEZİ  
Kontrol ve Otomasyon Mühendisliği Anabilim Dalı  
Kontrol ve Otomasyon Mühendisliği Programı

Danışman  
Dr. Öğr. Üyesi Muharrem MERCİMEK

Temmuz, 2022

**T.C.**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**DİKEY İNİŞ KALKIŞLI İNSANSIZ HAVA ARAÇLARI İÇİN GÖRSEL**  
**İŞARETLEYİCİ SİSTEMLER İLE GÖRÜNTÜ TABANLI KONUM**  
**KESTİRİMİ**

Mücteba UZUNOĞLU tarafından hazırlanan tez çalışması 05.07.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Kontrol ve Otomasyon Mühendisliği Anabilim Dalı Kontrol ve Otomasyon Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Muharrem MERCİMEK  
Yıldız Teknik Üniversitesi  
Danışman

**Jüri Üyeleri**

Dr. Öğr. Üyesi Muharrem MERCİMEK, Danışman  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Prof. Dr. Şeref Naci ENGİN, Üye  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

Dr. Öğr. Üyesi Nurullah ÇALIK, Üye  
İstanbul Medeniyet Üniversitesi

\_\_\_\_\_

Danışmanım Dr. Öğr. Üyesi Muharrem MERCİMEK sorumluluğunda tarafımda hazırlanan Dikey İniş Kalkışlı İnsansız Hava Araçları İçin Görsel İşaretleyici Sistemler ile Görüntü Tabanlı Konum Kestirimi 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.

Mücteba UZUNOĞLU

İmza

*Kıymetli aileme  
ve arkadaşlarıma*



## TEŐEKKÜR

---

Bu alıőmada, tecrube ve bilgisi ile bana yol gosterip araőtırmaya teővik eden deęerli danıőman hocam Dr. Öğr. Üyesi Muharrem MERCİMEK'e teőekkürü bor bilirim. Araőtırma sürecinde, mekanik ve kontrol alanındaki alıőmalarımda yardımlarını esirgemeyen arkadaşlarım Recep Behlül ŐAHİN ve Enes EMİR'e teőekkür ederim. Yüksek lisans eęitimim ve tez sürecinde varlıklarıyla manevi desteklerini esirgemeyen, zorluklarda yanımda olan aileme, yöneticilerime ve arkadaşlarıma teőekkürlerimi sunarım.

Mücteba UZUNOęLU

# İÇİNDEKİLER

<b>KISALTMA LİSTESİ</b>	<b>vii</b>
<b>ŞEKİL LİSTESİ</b>	<b>viii</b>
<b>TABLO LİSTESİ</b>	<b>ix</b>
<b>ÖZET</b>	<b>x</b>
<b>ABSTRACT</b>	<b>xii</b>
<b>1 GİRİŞ</b>	<b>1</b>
1.1 Literatür Özeti . . . . .	2
1.2 Tezin Amacı . . . . .	4
1.3 Orijinal Katkı . . . . .	4
<b>2 SİSTEM MİMARİSİ</b>	<b>6</b>
2.1 İnsansız Hava Araçları . . . . .	6
2.2 Quadrotor Bileşenleri . . . . .	9
2.2.1 Gövde . . . . .	9
2.2.2 Otopilot . . . . .	9
2.2.3 Yardımcı Bilgisayar . . . . .	10
2.2.4 Kamera . . . . .	10
2.2.5 İtki Sistemi . . . . .	11
2.2.6 Dronekit Kütüphanesi . . . . .	11
2.2.7 Gazebo Ortamı . . . . .	12
2.3 Quadrotor Sistemi Modellemesi . . . . .	13
2.3.1 Eksen Takımları . . . . .	13
2.3.2 Eksen Takımları Arasındaki Dönüşümler . . . . .	15
2.3.3 Açısal Hız Hesapları için Kinematik Denklemler . . . . .	16
2.3.4 Lineer Olmayan Dinamik Model . . . . .	17
2.3.5 Ötelemenin Hareket Denklemleri . . . . .	20
2.3.6 Motor Modeli . . . . .	20
<b>3 GÖRÜNTÜ İŞLEME</b>	<b>22</b>

3.1	Görsel Referans İşaretçi Sistemleri . . . . .	22
3.2	Geliştirilen Algoritma . . . . .	23
3.3	Kalman Filtresi Uygulaması . . . . .	31
<b>4</b>	<b>SONUÇ VE ÖNERİLER</b>	<b>35</b>
4.1	Simülasyon Ortamı . . . . .	35
4.2	Test Düzenegi Ortamı . . . . .	36
4.3	Analizler . . . . .	36
4.4	Sonuç . . . . .	40
	<b>KAYNAKÇA</b>	<b>42</b>
	<b>TEZDEN ÜRETİLMİŞ YAYINLAR</b>	<b>45</b>



## KISALTMA LİSTESİ

---

DİK	Dikey İniş Kalkışlı
ESC	Electronic Speed Controller
FPS	Frame Per Second
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GET	Gövde Eksen Takımı
İHA	İnsansız Hava Aracı
KF	Kalman Filtersi
NED	North East Down
PID	Proportional Integral Derivative
ROS	Robot Operating System
RPM	Revolution Per Minute
SLAM	Simultaneous Localization and Mapping
VO	Visual Odometry
VTOL	Vertical Takeoff and Landing
YET	Yer Eksen Takımı

## ŞEKİL LİSTESİ

Şekil 1.1	Slam algoritması . . . . .	3
Şekil 1.2	MoCap sistemi . . . . .	3
Şekil 2.1	Sabit kanatlı İHA - Bayraktar TB2 . . . . .	7
Şekil 2.2	Döner kanatlı İHA - Quadrotor . . . . .	8
Şekil 2.3	VTOL İHA . . . . .	8
Şekil 2.4	Otopilot donanımı . . . . .	9
Şekil 2.5	Görüntü işleme birimi . . . . .	10
Şekil 2.6	Kamera modülü . . . . .	11
Şekil 2.7	İtki sistemi, Motor-Pervane . . . . .	12
Şekil 2.8	Gazebo simülasyon ortamı ve yer kontrol arayüzü . . . . .	13
Şekil 2.9	Yer eksen takımı . . . . .	14
Şekil 2.10	Gövde eksen takımı . . . . .	15
Şekil 3.1	ArTag, AprilTag, ArUco ve CALTag etiket sistemlerinden örnekler . . . . .	24
Şekil 3.2	AprilTag'in robotik araçlar için seyrüsefer uygulaması . . . . .	24
Şekil 3.3	Algoritma adımları . . . . .	25
Şekil 3.4	Etiket köşe tespit algoritması . . . . .	27
Şekil 3.5	Gradyan, blur ve bazı filtreleme işlemleri sonrası çıktı . . . . .	28
Şekil 3.6	Tespit adımları . . . . .	29
Şekil 3.7	Farklı aileye ait etiketlerin aynı anda tespiti . . . . .	32
Şekil 3.8	Kalman filtresi blok diyagramı . . . . .	33
Şekil 4.1	Simülasyon ortamı blok diyagramı . . . . .	36
Şekil 4.2	Test ortamı . . . . .	37
Şekil 4.3	X eksen konum karşılaştırması . . . . .	38
Şekil 4.4	Y eksen konum karşılaştırması . . . . .	38
Şekil 4.5	Z eksen konum karşılaştırması . . . . .	39
Şekil 4.6	Test ortamında ölçümlerin kıyaslanması . . . . .	39
Şekil 4.7	Kalman kazancı . . . . .	40
Şekil 4.8	Kestirim belirsizliği . . . . .	40

## TABLO LİSTESİ

---

<b>Tablo 3.1</b>	Kalman Filtresi Katsayı Tablosu . . . . .	34
------------------	---	----



# Dikey İniş Kalkışlı İnsansız Hava Araçları İçin Görsel İşaretleyici Sistemler ile Görüntü Tabanlı Konum Kestirimi

Mücteba UZUNOĞLU

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı  
Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Muharrem MERCİMEK

Çoğunlukla İnsansız Hava Aracı (İHA) sistemleri küresel konum bilgisini GNSS'den alır. GNSS'nin olmadığı durumlarda, otonom uçuş görevi ve özellikle iniş görevleri gerçekleştirmek için hava aracının konum bilgisinin kontrolcü tarafından bilinmesi gerekmektedir. Bu çalışmada, Dikey İniş Kalkışlı (VTOL) İHA sistemleri için bir kamera ve görsel referans işaretçi sistemleri yardımıyla görüntü tabanlı bir lokal konum kestirim algoritması geliştirilmiştir. Görsel referans işaretçi sistemlerin diğer bir adıyla etiketlerin tespiti için etiketin köşe noktaları, kameradan alınan görüntülerdeki şekillerin kontur bilgilerinden elde edilmektedir. Ardından, etiketlerin taslak görüntülerinin kameradan alınan görüntülerdeki etiket ile eşleşmesi durumunda aracın etikete göre konumu hesaplanır. Görüntü tabanlı algoritma kullanılarak lokal konum hesaplandıktan sonra bozulmalara ve yanlış ölçümlere karşı tek boyutlu Kalman filtresi uygulanır. Kestirilen lokal konum bilgisi otonom iniş yapmak için kullanılır. Ayrıca geliştirilen algoritma farklı etiket sistemleri ile de çalışabilmektedir. Bunun için kullanılan etiketin taslak görüntüsünün olması yeterlidir. İHA'nın hasas konum kestirimi ve kararlı konum kontrolü için uygulamanın geliştirme ve doğrulaması simülasyon ortamında ve döner kanat quadrotor İHA sisteminden oluşan gerçek test ortamında yapılmıştır. Quadrotor dinamiği, bir simülasyon ortamında matematiksel olarak modellenmiştir. Geliştirilen görüntü tabanlı konum kestirim algoritmasının doğruluğu, simülasyon ortamındaki quadrotor hava aracının konumu ve deney düzeneğindeki manuel ölçümler ile karşılaştırılmıştır. Sonuçlar, İHA konum kestiriminin kabul edilebilir bir seviyede olduğunu göstermiştir.

**Anahtar Kelimeler:** İnsansız hava aracı, konum kestirimi, bilgisayarlı görü, görsel işaretleyiciler, kalman filtresi



---

YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

# Vision Based Position Estimation with Markers for Vertical Takeoff and Landing Unmanned Air Vehicles

Mücteba UZUNOĞLU

Department of Control and Automation Engineering  
Master of Science Thesis

Supervisor: Asst. Prof. Muharrem MERCİMEK

In general, the Unmanned Air Vehicle (UAV) systems take the global position information from Global Navigation Satellite System (GNSS). In the GNSS-denied environment, to perform autonomous flight mission and especially landing missions, it is necessary to know the position information of the unmanned air vehicle. In this study, a vision-based local positioning algorithm for Vertical Takeoff and Landing (VTOL) UAV systems is developed with a camera and fiducial markers. The algorithm has the three main parts that are detection, position estimation and Kalman filter implementation. Firstly, the corner points are obtained from the contour information of the markers in the images taken from the camera. Then, the position of the vehicle relative to the marker is calculated if the template images match with marker in the images from camera. After calculating the local position using vision-based positioning algorithm, a one-dimensional Kalman filter is applied. The estimated local position is used for autonomous landing. In addition, the local positioning algorithm can be worked with different fiducial marker systems. The analysis for the application is performed in the simulation environment. Also, the experimental studies are conducted for accurate position estimation and stable control of the UAV. Quadcopter physics is modeled mathematically in a simulation environment. The accuracy of the proposed vision-based position estimation algorithm is compared with the positions of the quadcopter in the simulation environment and the manual measurements in the experimental setup. The results demonstrated that the UAV position estimation is in an acceptable range.

**Keywords:** Unmanned aerial vehicle, position estimation, computer vision, fiducial

markers, kalman filter



---

**YILDIZ TECHNICAL UNIVERSITY  
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING**

# 1 GİRİŞ

---

Günümüzde İnsansız Hava Aracı (İHA) sistemleri, keşif, gözetleme, istihbarat, arama, kurtarma gibi askeri ve sivil amaçlarla aktif olarak kullanılmaktadır.[1] İHA'nın otonom sürüşü, seyrüsefer sensörleri ve uçuş kontrol sistemleri ile sağlanmaktadır. İHA'ların navigasyonu için GNSS, ana konum sensörüdür. Önceden belirlenmiş rotada hareket eden İHA görevini bitirdikten sonra, belirlenen noktaya iniş gerçekleştirir. İniş sırasında operatörün dikkat ve yeteneği önem arz etmektedir. İniş bölgesinde İHA'nın GNSS hataları ve hassasiyet düşüklüğünden ötürü kayma veya sapma durumu, iniş bölgesinin sadece harita üzerinden koordinat olarak genelde gözle görülmeden belirtilmesinden kaynaklı riskli olabilecek durumlar meydana gelebilir. Diğer taraftan inişte kullanılan lidar, barometre gibi sensörlerden kaynaklı hatalar da iniş görevini riskli duruma sokabilmektedir.[2] Bu durumlarda otonom iniş sırasında kaza riskinin yüksek olması nedeniyle İHA sistemlerinde iniş aşaması en önemli sorunlardan biri olmuştur. Bu nedenle otonom iniş görevi hassas konum bilgisi gerektirir. Maliyeti ucuz olan sensörlerin kullanılması kaza riskini artırmakla beraber maliyeti yüksek cihazların kullanılması da nihai bir ürün olan İHA sistemlerinde maliyet etkin durumundan çıkarabilmektedir.

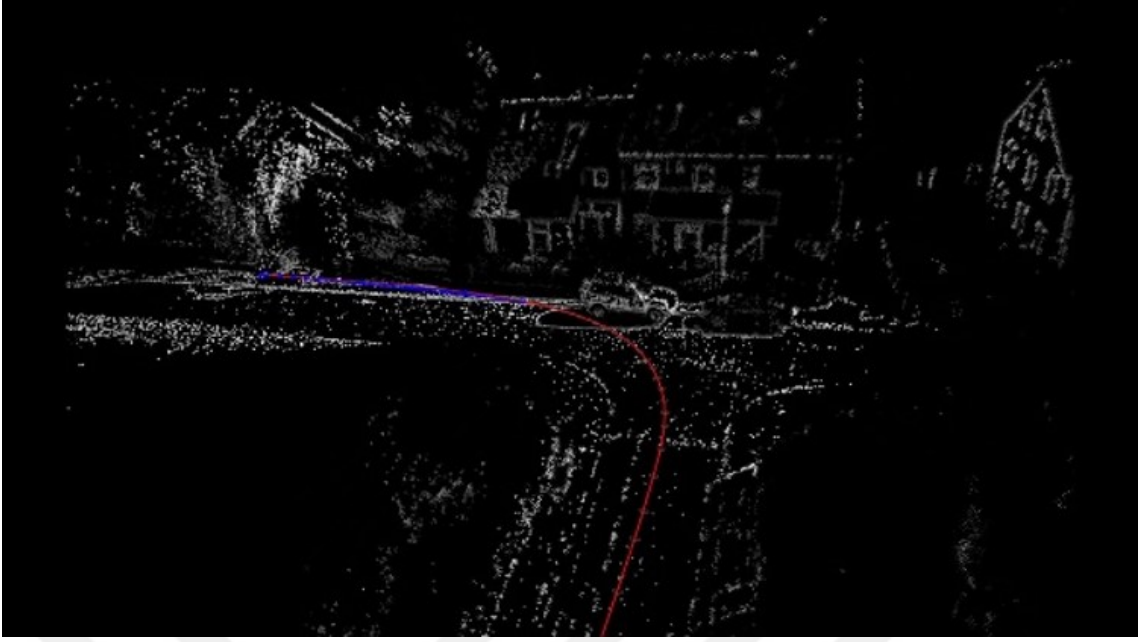
Bu çalışmada düşük maliyetli yüksek doğruluk oranına sahip bir konum kestirimi üzerinde durulmuştur. Çalışmada algoritma geliştirme ve doğrulama için riski ve zaman maliyetini azaltmak amacıyla ilk aşamada simülasyon ortamı tercih edilmiştir. Simülasyon ortamında döner kanat quadrotor İHA'nın hem dinamikleri yüksek doğrulukta modellenmiş, hem de kamera çıktısı olarak gerçeğe yakın sonuçlar üreten bir ortam hazırlanmıştır. Böylelikle simülasyon ortamında geliştirilen algoritma gerçek kamera ile de kameraya özel birkaç parametre değiştirilerek başarılı şekilde çalıştırılmıştır. Literatürde hem yedeklilik hem de yardım amaçlı konum sensörleri ile ilgili birçok çalışma bulunmaktadır. Lidar, ultrasonik ve IR sensörler bahsi geçen problem için sıklıkla kullanılmaktadır ve literatürde mevcuttur. Etkili sonuçlar elde edilmesine rağmen, bu yaklaşımlar gerçek zamanlı dış mekan uygulamaları için yeterli değildir.[3, 4] Bu gibi durumlarda, bilgisayarlı görme algoritmaları ile

yapılan çalışmalar, GNSS'in olmadığı durumlarda ve kentsel alanlarda konum ve duruş kestirimi için etkili bir çözüm sunmaktadır. Görüntü tabanlı konumlandırma sisteminin düşük maliyeti ve güvenilir sonuçları nedeniyle İHA sistemlerinde otonom iniş için önemli bir yardımcı sensör olduğu düşünülmektedir. Ayrıca gemi gibi hareketli platformlarda iniş destek sistemi olarak kullanılabilir. Çalışmanın yapısı aşağıdaki gibi organize edilmiştir. Bölüm 2'de quadrotor sistemi, alt bileşenleri, ve simülasyon ortamına uygulanmış matematiksel modeli anlatılmıştır. Bölüm 3'te etiketin tespiti, homografi kullanılarak konum kestirimi ve kalman filtresi uygulaması anlatılmıştır. Bölüm 4'te ise algoritmanın geliştirildiği ve doğrulamasının yapıldığı simülasyon ortamı ve test düzeneği anlatılıp, sonuçları analiz edilmiştir.

## 1.1 Literatür Özeti

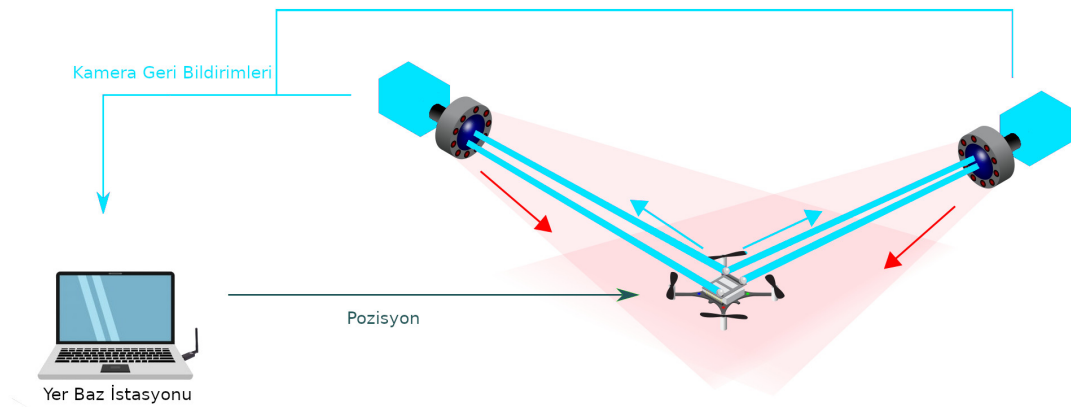
İHA sistemlerinde güvenli iniş literatürde oldukça yaygın çalışılan bir konudur. Özellikle piste iniş kalkış yapan sabit kanatlı İHA'lar aletli iniş sistemi denilen ILS sistemlerini kullanmaktadırlar. Bu sistemde yerden İHA'ya piste göre olan konum düzeltilmesi verilmektedir. İHA'larda GNSS'e alternatif diğer bir çözüm ise Beacon sistemleridir. Bu sistemde ise İHA'nın çevresinde konumlandırılmış baz cihazlar İHA'ya RF sinyaller aracılığıyla konum bilgisinde kullanacağı sinyaller göndermektedirler. Bu sistemler genelde kapalı mekanlarda konumlandırma için kullanılmaktadır. Bu sistemlerde baz olarak kullanılan cihazlar İHA'nın çevresinde genelde en az dört adet olacak şekilde konumlandırılırlar. Literatürde mevcut olan diğer bir çözüm ise optik akış sensörü ve algoritmalarıdır. [5, 6]

İHA üzerinde bulunan optik akış sensörü veya kamera ile aracın yere göre hızı kestirilmeye çalışılır. Bu çalışmalarda görüntüler arasındaki piksel farklarından yer hızı bulunup sonrasında hızın integrali alınarak konum bilgisi elde edilir. Bir sensörden okunan değerın integrasyon işleminde genelde kayma ve rastgele yürüme olacağından hassas bir hesap gerektirir. Yine çoğunlukla kapalı ortamlarda çalışan bir diğer çözüm MoCap olarak adlandırılan hareket algılama sistemidir. Beacon sistemine benzer çalışan sistem İHA'nın çevresinde konumlandırılmış kameralar ile baz bir hesaplama istasyonuna görüntü bilgilerini aktarır. Sonrasında baz istasyonundan İHA sistemine konum bilgisi gönderilir. İHA'lar üzerinde bulunan kameralar kullanılarak iniş için konumlandırmaya yönelik bilgisayarlı görü çalışmaları da literatürde yaygın olarak kullanılmaktadır. [7, 8]. [9]'da döner kanat quadrotor hava aracı kullanılmış, iniş bölgesinde bulunan helipadin tespiti sonrasında helipad kameranın x ve y ekseninde ortalanacak şekilde piksel farkına göre kontrol edilerek iniş çalışması gerçekleştirilmiştir. Literatürde Görsel Odometri (VO), Eş Zamanlı Konumlandırma ve Haritalama (SLAM) veya görsel referans işaretçileri gibi yöntemler



Şekil 1.1 Slam algoritması

sıklıkla kullanılmakta ve önerilmektedir. VO ve SLAM yöntemlerinin iyi çalıştığı ortamlar olsa da görsel özellikleri ve aydınlatma koşullarının zayıf olduğu ortamlarda genellikle başarısız olurlar. [10, 11] Bu gibi durumlarda, görsel referans işaretçileri genellikle İHA platformunun konumlandırma kabiliyetini artırmak için kullanılabilir. Görsel referans işaretçileri, literatürde etiket olarak bilinir ve tespit için her etiket sistemi kendi algoritmalarına sahiptir. Ayrıca güçlü görsel özelliklere sahip oldukça ayırt edilebilir bir model sunar.



Şekil 1.2 MoCap sistemi

Bu çalışmada, bir İHA'nın iniş görevi için herhangi bir etiket sistemi ile çalışabilen görüntü tabanlı konum kestirim algoritması geliştirilmiştir. İHA'nın altında aşağı doğru

bakacak şekilde takılı monoküler kamera ile görüntüler alınır. Etiket'in kontur bilgileri [12] monoküler kameradan alınan görüntülerden elde edilir. Daha sonra etiket'in köşe koordinat bilgisi çıkarılır. Etiket'in taslak görüntüsü kameradan alınan görüntü ile eşleştigi durumda etiket algılanır. Daha sonra homografi hesabı ve kamera matrisi ile konum kestirimi yapılır. Görüntüdeki gürültü ve bozulmaları en aza indirmek için hesaplanan konum sonuçlarına tek boyutlu Kalman Filtresi uygulanır. Stabil bir otonom iniş gerçekleştirmek için, algoritmanın etiket tanıma ve konum tahmini simülasyonda ve gerçek ortamda test edilmiştir. Bu konumlar, aracın etikete göre görelî, lokal konumunu gösterir.

## 1.2 Tezin Amacı

Çalışmanın amacı VTOL İHA'larda konum sensörünün aktif olmadığı veya hassas çalışmadığı durumlarda iniş sağlığını ve güvenliğini artırmak için görüntü tabanlı konum kestirim çözümü sunmaktır. Önerilen metotta iniş yapılan yerde bulunan etiket olarak da adlandırılan görsel işaretçi sistemler kullanılarak iniş merkezine göre görelî konum kestirimi yapılmaktadır. Düşük maliyetli bu çözümün hem simülasyon ortamında hem de gerçek ortamda doğrulamalarının yapıp iniş görevinde kullanılması amaçlanmıştır.

## 1.3 Orijinal Katkı

Literatürde bulunan görüntü tabanlı seyrüsefer çalışmalarına bakıldığında kamera ile iniş bölgesi tespiti ve inişe yardımcı çalışmalarda genelde kamera koordinat sistemine göre piksel cinsinden koordinatlar ile İHA'nın uçuş kontrol sistemi beslenmektedir. Piksel cinsinden koordinatlarla kontrol sistemi çalıştırmanın problemi, tespit sırasında İHA'nın dolayısıyla kameranin oryantasyonunun değişmesi durumunda bakış açısındaki perspektif durumundan ötürü aynı nokta için farklı piksel çıktıları alınmaktadır. Bu da hassas kestirim ve kararlı kontrolün önüne geçmektedir. Bu çalışmada İHA'nın iniş bölgesine göre metre cinsinden üç boyutta konum bilgisi bulunarak ve oluşabilecek sensör bazlı hatalara karşı Kalman filtresi kullanılarak daha kararlı bir kontrol yapısı ile İHA'nın iniş sistemi geliştirilmiştir.

Bu yapı aynı zamanda hareketli platformlara iniş için de kullanılabilir şekilde oluşturulmuştur. Literatürde bulunan görsel işaretçi etiket sistemleri kendi bünyelerinde kendi ailesi ile çalışabilecek şekilde etiket yapılandırma, etiket tespiti, poz hesaplama gibi özelliklere sahiptir. Bu sistemler birbiri aralarında kıyaslanmaktadır. Bu çalışmanın katkılarında biri de herhangi bir aileye ait bir etiketi tespit edebilen, konum hesaplayabilen jenerik bir algoritma geliştirmekle birlikte,

otonom bir iniş görevi için yardımcı bir sensör gibi çalışabilen bir sistem, bu konum bilgisini kullanarak seyrüsefer ve iniş yapacak bir hava aracı sistemini hem simülasyon ortamında hem de gerçekte ortaya çıkarmaktır.



## 2 SİSTEM MİMARİSİ

---

Bu bölümde görüntü tabanlı konum kestirimi geliştirmelerinin yapıldığı platform olan quadrotor insansız hava aracının bileşenleri ve bu sistemin modellenmesi anlatılmıştır. Geliştirilen çalışma öncelikle simülasyon ortamında test edilmiştir. Doğrulama çalışması için simülasyon ortamında quadrotor hava aracı dinamikleri matematiksel olarak modellenmiştir. Böylelikle matematiksel model çıktısı konum bilgileri, doğrulama için referans değer olarak kullanılmıştır. Hava aracı sisteminin modellenmesi ne kadar doğru olursa geliştirilen algoritmanın karşılaştırması da o kadar doğru olacaktır. Daha sonra ise aşağıda bileşenleri anlatılan quadrotor sisteminde geliştirme çalışmaları yapılmıştır.

### 2.1 İnsansız Hava Araçları

İnsansız hava araçları, belirli bir görev için tasarlanmış, otomatik seyrüsefer kabiliyeti olan, genelde bir yer istasyonundan kontrol edilen, uçan robotik sistemler olarak tanımlanmaktadır. Son yıllarda İHA sistemlerinin gelişimi ve kullanımı dikkate değer şekilde artmaktadır. İHA sistemleri, platform üzerinde görevli pilot olmaması sebebiyle insanlı hava araçlarına göre daha düşük maliyetle geliştirilmekte, ayrıca risk faktörünü düşürmesi sebebiyle tehlikeli görevlerde tercih edilmektedir. İHA sistemi denince akla ilk önce uçak veya hava aracı gelse de bu sistemler yer istasyonu ve hava aracı olarak iki temel yapıdadır. Hava aracında motor ve pervaneden oluşan itki sistemi, otopilot sistemi, uçuş kontrol sistemi ve sensör sistemi gibi alt sistemlerden oluşmaktadır. Yer istasyonunda ise yer haberleşme anteni, yer kontrol bilgisayar, kumanda kolu gibi sistemler mevcuttur. İHA'lar genelde bir faydalı yük taşınması ve onun kullanılması amacıyla geliştirilmektedir. Bu faydalı yüklere elektro-optik kameralar, termal kameralar, lidar, radar sensörler ve mühimmatlar örnek olarak gösterilebilir. 1900'lü yıllarda geliştirilen Kettering Bug isimli ilk İHA cayroskop ve barometre yardımıyla belirlenen rotada gitme özelliğine sahiptir. Sonrasında 1. ve 2. Dünya Savaşında geliştirilen İHA'lar keşif, gözetleme ve bombardıman amacıyla kullanılmıştır. İlk kullanımları askeri alanda olan bu sistemler, artık günümüzde askeri

ve sivil alanlarda gün geçtikçe yaygınlaşmaktadır.

Askeri alanda, keşif ve gözetleme, sinyal istihbaratı, mühimmat ile düşman unsurların etkisiz hale getirilmesi, meskun mahal devriye faaliyetleri, lojistik, hasar değerlendirilmesi ve bazı imha faaliyetleri gibi kullanım alanları bulunmaktadır. Sivil alanlarda ise havadan fotoğrafçılık, tarımsal alanların izlenmesi, kimyasal ilaçlama, haritalama, yük taşımacılığı, trafik kontrolü, yangın algılama ve doğal afetlerde görüntüleme, meteorolojik hizmetlerin yapılması, petrol boru hatlarının güvenliği gibi durumlarda sık sık kullanılmaktadır.



**Şekil 2.1** Sabit kanatlı İHA - Bayraktar TB2

İHA'lar genelde yapısal olarak sabit kanatlı, döner kanatlı ve hibrit yapıda olabilirler. Sabit kanatlı hava araçları kalkış ve iniş ihtiyacı duymakla beraber, taşıma üreten kanatları sayesinde uzun süre havada kalma kapasitesine sahiptirler. Döner kanatlı araçlar ise aracın yatay düzeleminde bulunan itki sistemi ile pist ihtiyacı duymadan dikey iniş-kalkış ve asılı kalma özelliğine sahiptir. Fakat bu durumda havada kalma süresi sabit kanatlı araçlara kıyasla çok daha az olmaktadır. Literatürde VTOL olarak geçen, genelde İngilizce isminin kısaltması ile bilinen (Vertical Takeoff and Landing) hibrit hava araçları döner kanat yapısı ile dikey kalkış özelliğine sahip olmakla beraber sabit kanatlı yapısı ile de geleneksel uçak gibi uçabilmektedirler. Bu hibrit sistemler tilt motorlu, kuyruğa oturtmalı olarak da tasarlanmaktadır.

İHA'lar belirlenmiş uçuş planlarını GNSS gibi konum sensörlerinden aldığı bilgiye göre sürdürmektedirler. Tanımlanmış görev sonrası küresel koordinatları belirtilmiş noktaya iniş gerçekleştirirler. Büyük uçaklar için iniş destek sistemi olarak Aletli İniş



Şekil 2.2 Döner kanatlı İHA - Quadrotor



Şekil 2.3 VTOL İHA

Sistemi (Instrument Landing System - ILS) gibi yardımcı sistemler kullanılmaktadır. Dikey iniş kalkışlı araçlar için pist ihtiyacı olmamakla birlikte aracın boyutunun 2 katı kadar düz bir alan iniş ve kalkış için yeterli olmaktadır. Gemiye konuşlu araçlar

ve hareketli platforma iniş yapacak araçlar için görüntü tabanlı çözümler üzerinde literatürde çalışmalar mevcuttur.

## 2.2 Quadrotor Bileşenleri

Çalışma kapsamında geliştirilen yazılımın entegre edileceği platform döner kanat dört rotorlu hava aracı sistemi oluşturulmuştur. İHA sisteminin donanımsal özellikleri bu bölümde anlatılmıştır. İHA üstünde uçuş kontrol yazılımının çalıştırıldığı bir bilgisayar ile görüntü tabanlı konum kestiriminin yapıldığı bilgisayar olmak üzere 2 ana bilgisayar bulunmaktadır. Yazılım geliştirmesi aşamasında zaman ve maliyet kazancı için aracın matematiksel modeli bilgisayar ortamında oluşturulmuş olup, yazılım geliştirmeleri burada tamamlanmıştır.

### 2.2.1 Gövde

Geliştirilecek algoritma döner kanat quadrotor İHA üstünde test edilecektir. Döner kanatlı quadrotor sistemi Şekil 2.2'de görüldüğü üzere motor ve pervaneler dikey ekseninde itki üretecek şekilde yerleştirilmiştir.

### 2.2.2 Otopilot

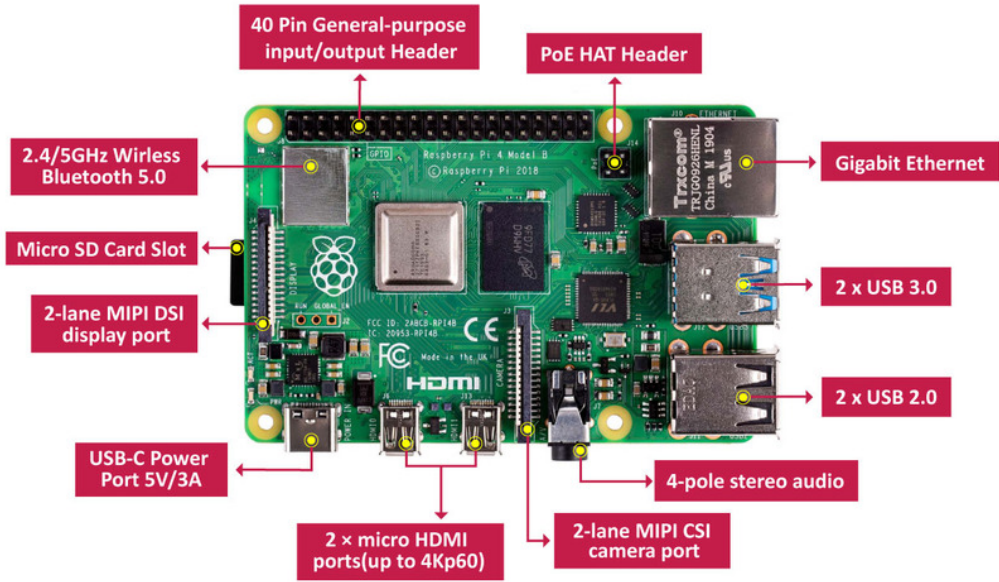
Quadrotor sisteminde otopilot olarak açık kaynak bir devre olan Pixhawk Cube Orange donanımı kullanılmıştır. Dahil IMU, barometre, manyetometre gibi sensör birimleri ve açık kaynak yazılım kullanılabilmesi dolayısıyla konum kestirim ve otonom iniş kodunun geliştirilmesine kolaylık sağlaması sebebiyle seçilmiştir.



Şekil 2.4 Otopilot donanımı

### 2.2.3 Yardımcı Bilgisayar

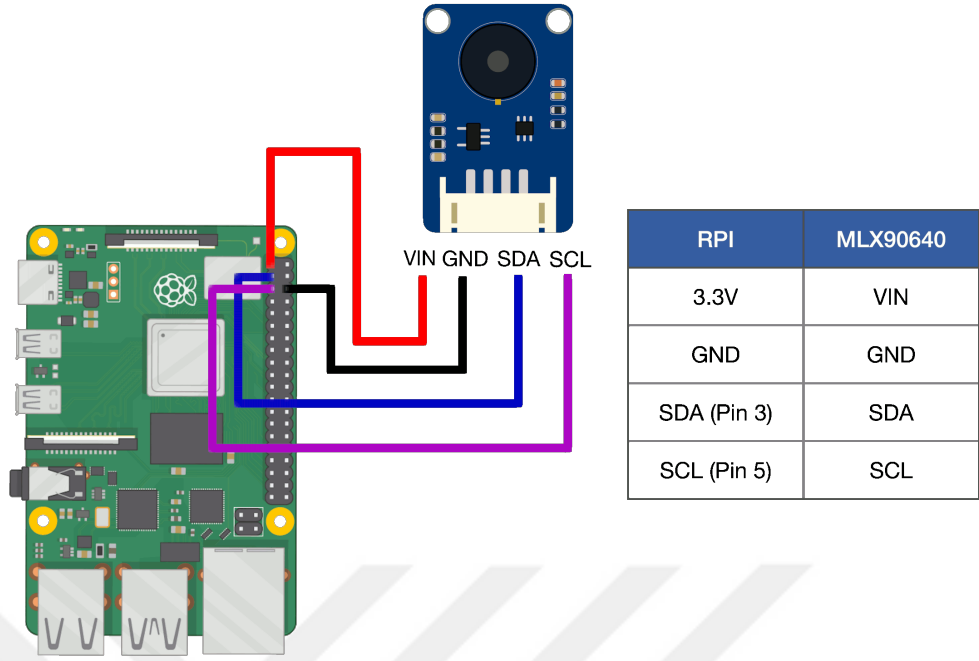
Görüntü tabanlı konum kestirim kodunun geliştirilmesi için bilgisayarlı görü hesaplarının yapılacağı bilgisayar olarak Raspberry Pi 4B seçilmiştir. Bu kontrol kartında, Broadcom BCM2711 mikroçipi üzerine kurulu olmakla birlikte VideoCore IV GPU grafik işlem birimine sahiptir. Booting ve veri depolaması için hafıza kartı kullanıldı. Dahili USB portları, HDMI video çıkışı, ses çıkışı, MIPI kamera girişi, GPIO arayüzü ve 5V USB Type-C güç girişi bulunmaktadır. Raspberry Pi 4 Model B cihazında, Raspbian, Ubuntu gibi Linux işletim sisteminin farklı dağıtımları çalışabilmektedir. Resmi olarak onaylanmış işletim sistemleri dışında Pardus ARM, Arch Linux ARM ve Windows 10 IoT Core işletim sistemlerini de desteklemektedir. Bu çalışmada Linux işletim sisteminin Ubuntu dağıtımı kullanılmıştır.



Şekil 2.5 Görüntü işleme birimi

### 2.2.4 Kamera

Alta bakan şekilde kullanılacak olan seyrüsefer kamerası olarak Raspberry Pi Kamera modülü kullanılmıştır. Kolay bulunabilirlik, düşük maliyet ve yardımcı bilgisayarla uyumlu çalışma ve ayrıca termal kamera modeli bulunması gibi sebeplerden ötürü tercih edilmiştir.



Şekil 2.6 Kamera modülü

### 2.2.5 İtki Sistemi

İtki sistemi seçiminde öncelikle tasarlanan hava aracı tahmini kalkış ağırlığı belirlenir. Motor, pervane, esc, batarya, otopilot, görev bilgisayarı, gövde ağırlığı, kamera gibi hava aracında bulunan ekipmanlar düşünüldüğünde yaklaşık en yüksek 2kg kalkış ağırlığı olacağı öngörülmüştür. Döner kanat quadrotor itki sistemi tasarımında, hava aracı 50% gazda asılı durumda kalması istenir. Bu sebeple motor 100% gazda iken 1 kg itki verecek pervane seçimi yapılır. Pervane modellerinde ilk iki rakam çap ikinci iki rakam hatveyi belirtir. Kullanılan motorlar fırçasız doğru akım motorudur. Bu tip motorlar ESC adı verilen motor sürücüleri ile kullanılırlar. ESC seçiminde ise motor pervane ikilisinin 100% gazda çektiği akımın 20% fazlası kadar fazla akım geçirilebilme özelliği dikkate alınır. Bu minvalde 920KV Readytosky 2212 motor ile 10\*45 boyutlarında pervane ve 30A esc seçimi yapılmıştır.

### 2.2.6 Dronekit Kütüphanesi

Dronekit, Ardupilot açık kaynak otopilot yazılımı ile haberleşebilen otopilot ile beraber çalışacak programlara uygulama oluşturmak için geliştirilmiş python ve android tabanlı bir kütüphanedir. Hava aracına yüksek otonomi kabiliyeti katan yazılımların geliştirmesi için ve hesaplama açısından yoğun veya zamana duyarlı görevleri gerçekleştirerek (örneğin, bilgisayarla görme, rota planlama veya 3D modelleme) otopilota destek yazılımları hazırlanmasında kullanılır. Dronekit, MAVLink üzerinden

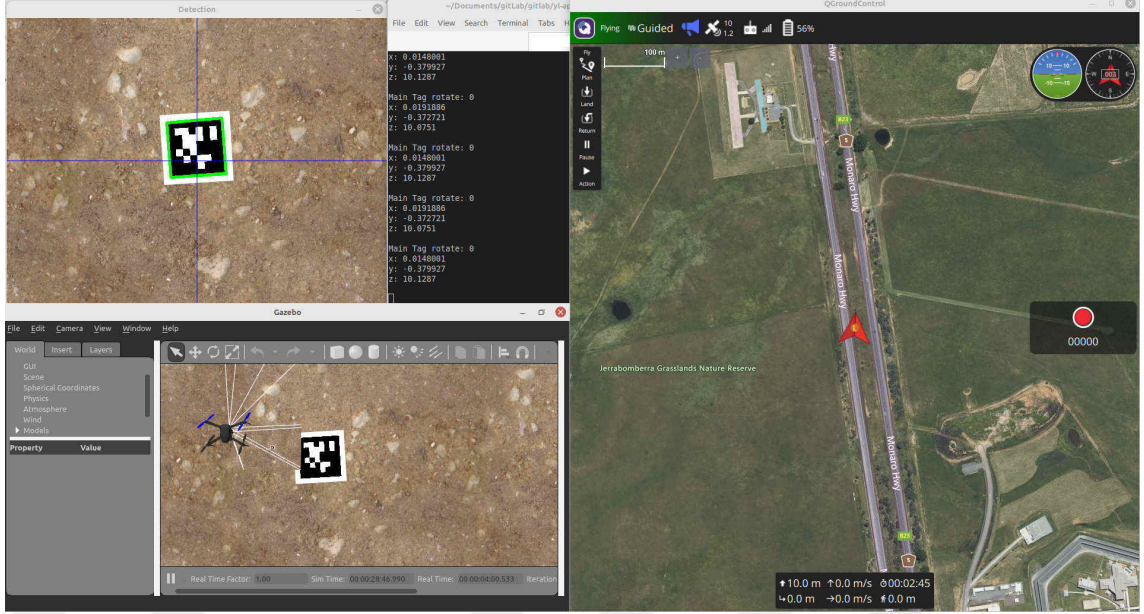


Şekil 2.7 İtki sistemi, Motor-Pervane

hava araçları dolayısıyla otopilotlar ile iletişim kurmaktadır. Veri bağı ile bağlı bir aracın telemetri, durum ve parametre bilgilerine programlı erişim sağlamakta ve hem görev yönetimini hem de araç hareketi ve operasyonları üzerinde doğrudan kontrol sağlamaktadır. DroneKit kütüphanesi kullanılarak; bir veya birden fazla İHA ile bağlantı kurabilme, aracı belirlenen konuma gönderebilmek ve araca entegre bilgisayar üzerinde çalıştırılabilme gibi önemli özelliklere sahiptir. Bu çalışmada, kameradan alınan görüntüler işlenerek elde edilen veriler ile gerekli pozisyon hesaplamaları yapıldıktan sonra, İHA'nın otonom inişi için gerekli olan haberleşmenin sağlanabilmesi ve komutların gönderilmesi için Dronekit kütüphanesi kullanılmıştır.

### 2.2.7 Gazebo Ortamı

Gazebo, açık kaynaklı bir 3D robot simülatörüdür. Donanımlı tasarımı ile gerçekçi senaryolar oluşturabilmektedir. Robotları tasarlamayı, algoritma testleri yapmayı ve yapay zeka sistemini eğitmeyi mümkün kılmaktadır. Bununla birlikte simülasyon ortamına eklenen modeller ile kamera üzerinden gerçek zamanlı olarak görüntü alınabilmektedir. Bu çalışmada, quadrotor hava aracının matematiksel modeli Gazebo ortamında gerçekleştirilmiş, sonrasında gazebo'da bulunan kamera sensöründen alınan veri ile önerilen algoritma geliştirilmiş ayrıca gazebo'daki matematiksel model çıktısı algoritmayı doğrulamak için referans bilgisi olarak kullanılmıştır.



Şekil 2.8 Gazebo simülasyon ortamı ve yer kontrol arayüzü

## 2.3 Quadrotor Sistemi Modellemesi

Dikey iniş kalkış kabiliyetine sahip quadrotor sistemi, yüksek manevra yeteneğine sahip, dört motor ve pervaneli hava aracı sistemidir. Şekil 2.2’de görülen hava aracı sistemi dört pervanesiyle altı serbestlik dereceli değişkenleri kontrol edebilmektedir.

Simülasyon sistemi yazılım geliştirmelerde zaman ve maliyet açısından çok önemlidir. Gerçek hayattaki dinamikler ne kadar gerçekçi olarak modele aktarılırsa model ile geliştirilen algoritmalar o kadar başarılı olur. Hava aracının gerçek hayattaki hareketleri, dinamik ve kinematik denklemler ile hava aracının kütle, atalet, sürüklenme katsayıları kullanılarak modellenmektedir. Bu modellemede aşağıda sıralanan bazı kabuller yapılmaktadır.

- Hava aracı ağırlık merkezi motorların kesişim noktasındadır.
- Hava aracı yapısalı rijid ve simetrik.
- Pervane itki ve sürüklenme kuvvetleri motorun dönüş hızının karesi ile orantılıdır.
- Modellenen quadrotor hava aracı X konfigürasyonundadır.

### 2.3.1 Eksen Takımları

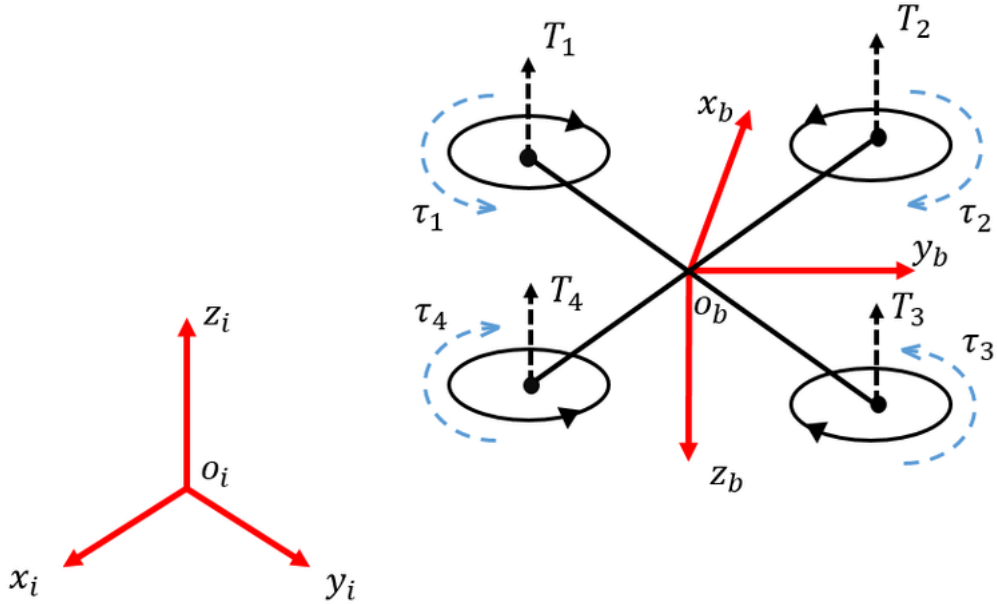
Hava aracının duruş ve konum bilgileri için bazı referans alınan eksen takımları vardır. Aşağıda quadrotor modellemesinde kullanılacak eksen takımları anlatılmıştır.

### 2.3.1.1 Yer Eksen Takımı

Newton hareket denklemleri yere sabit bir eksene göre ölçülen ivmeler ile kullanılabilir olmaktadır. Bu sebeple, dinamik model denklemlerinin çözümleri için sabit dünya eksen takımına bağlı bir referans yer eksen takımı tanımlanır. Tanımlanan referans noktasında;

- X eksenini Kuzeye doğru,
- Y eksenini Doğuya doğru,
- Z eksenini dünyanın merkezine doğru

olacak şekilde oluşturulur.



Şekil 2.9 Yer eksen takımı

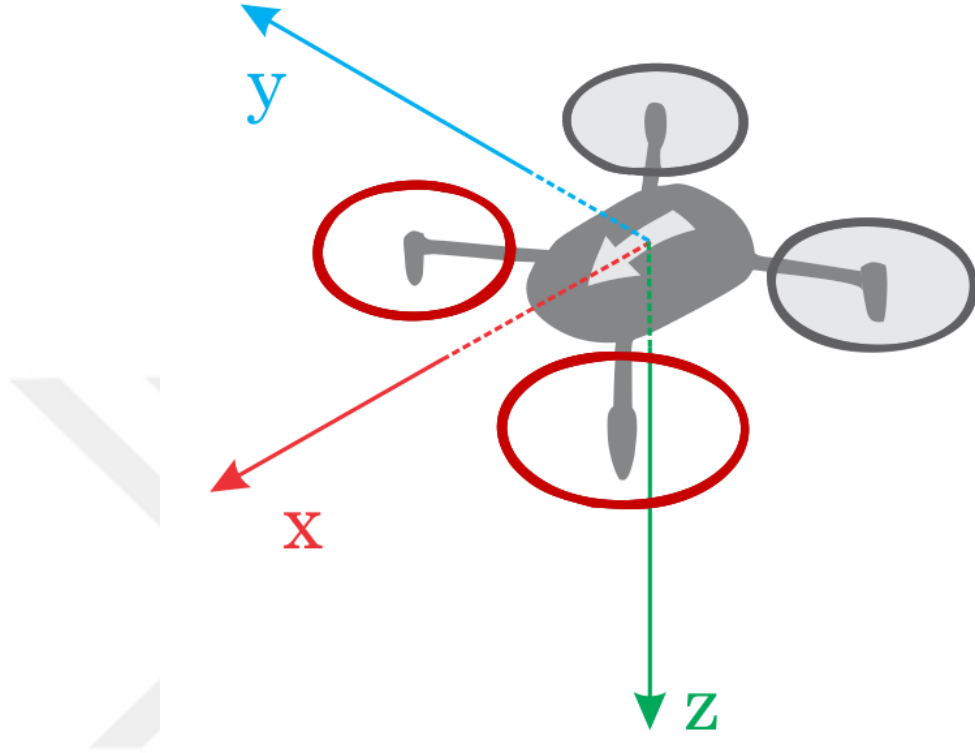
### 2.3.1.2 Gövde Eksen Takımı

Quadrotorun ağırlık merkezi gövde eksen takımının merkezidir. Kuvvet ve momentler, birbirine dik üç eksenlerden oluşan gövde eksen takımının merkezinde oluşur. Gövde eksen takımında;

- X eksenini hava aracının burun yönünde,
- Y eksenini hava aracının burna göre sağa yönünde,

- Z eksenini ise gövdenin tam aşağısı yönünde

tanımlanmaktadır.



Şekil 2.10 Gövde eksen takımı

### 2.3.2 Eksen Takımları Arasındaki Dönüşümler

Quadrotorun modellenmesinde iki eksen takımı tanımlanmıştır. Newton-Euler denklemlerinden türetilen dinamik model yer eksen takımına göre oluşturulur. Quadrotorun oryantasyonu bu eksen takımına göre oluşturulur. Bu sebeple gövde eksen takımı ile yer eksen takımı arasında dönüşümlerin yapılması gerekmektedir. Bu dönüşüm için yaygın üç metot kullanılmaktadır.

- Euler Açıları
- Quaternionlar
- Direction Cosine Matrix (DCM)

Simülasyon modelinde sezgiselliği ve anlaşılma kolaylığı sebebiyle Euler açıları gösterimi kullanılmıştır. Üç eksenindeki dönüşüm matrisleri denklem (2.1), (2.2) ve (2.4)'te verilmiştir.

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.2)$$

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.3)$$

Eksen dönüşümünde sırasıyla Z,Y ve X eksenleri şeklinde yapılır. Dönüşüm matrisi bu üç matrisin çarpımı ile hesaplanır. denklem (2.4) ve (2.5)'te C ifadesi cosinüs, S ifadesi ise sinüs yerine geçmektedir.

$$R_Y^G = \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \quad (2.4)$$

Dönüşüm matrisi denklem (2.5)'teki gibi elde edilmektedir.

$$R_Y^G = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & c(\theta)s(\phi) \\ c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (2.5)$$

Dönüşüm matrisleri ortogonaldır. Ayrıca tersi transpozuna eşittir.

$$(R_Y^G)^{-1} = (R_Y^G)^T \quad (2.6)$$

X, quadrotorun durumlarını ifade etmek üzere, gövde eksen takımı ile yer eksen takımı arasındaki ilişki denklem (2.7)'de verilmiştir.

$$X_Y = R_Y^G X_G \quad (2.7)$$

### 2.3.3 Açısal Hız Hesapları için Kinematik Denklemler

Euler açıları yer eksen takımında ölçülmezler, gövde eksen takımında ölçülürler. Bundan dolayı, öylece kullanılamazlar. Dolayısıyla denklem (2.8)'de verilen denklem ile gövde eksen takımındaki üç eksendeki hızlar  $(p, q, r)$ , euler açısal hızlarına

dönüştürülürler.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_\phi R_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.8)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.9)$$

Denklem (2.9)'dan euler açısal hız denklemleri çıkarmak için denklem (2.10) elde edilir. Denklem (2.10)'un integrali ile yer eksen takımı euler açıları hesaplanır.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sec(\theta) & \sec(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.10)$$

#### 2.3.4 Lineer Olmayan Dinamik Model

Quadrotor dinamik modeli için Newton - Euler formülü kullanılmıştır. Newton denklemleri yer eksen takımına göre dir. Bu durumda motor ve pervanenin oluşturduğu gövde eksen takımındaki kuvvet ve momentler yer eksen takımına taşınmalıdır. Newton hareket denklemlerinde rijid cisim hareketleri, rotasyon ve öteleme hareketlerinin birleşimi şeklinde modellenir. Rotasyonel dinamikler yatış, dikilme ve baş duruş açılarıdır. Öteleme dinamikleri ise X,Y ve Z eksenindeki konumlardır. Bu dinamikler denklem (2.11) ve (2.12)'den elde edilir.

$$J\dot{\omega} + \omega \times J\omega = \sum M_{top} \quad (2.11)$$

$$m\dot{V}^B + \omega \times JV^B = \sum F_{top} \quad (2.12)$$

Denklem (2.11) ve (2.12)'de quadrotora etkiyen kuvvet ve momentler gösterilmiştir. Kontrol edilen her bir motorun ürettiği kuvvet  $U_1$  ile üretilen yatış, dikilme ve baş momentleri ise  $U_2, U_3$  ile gösterilmektedir.

### 2.3.4.1 Kuvvet ve Momentler

Motor ve pervaneden oluşan kuvvet ve momentler motor dönüş hızının karesi ile orantılı olacak şekilde denklem (2.13) ve (2.14)'teki gibi modellenmiştir.

$$F_i = b\Omega_i^2 \quad (2.13)$$

$$M_i = d\Omega_i^2 \quad (2.14)$$

Denklem (2.13) ve (2.14)'deki  $i$  indisi motor numarasını ifade etmektedir.  $b$  ve  $d$  katsayısı ise pervane aerodinamik itki ve tork katsayısıdır.  $b$  ve  $d$  katsayısı pervane çapı, hatvesi, hava yoğunluğu, motor dönüş hızı ve pervane itki ve sürüklenme katsayısı, motor torku gibi değişkenlere bağlıdır.  $l$  ise motorların dönme eksenine olan uzaklığıdır. Gövde eksen takımında oluşan toplam kuvvet denklem (2.15) ve toplam moment denklem (2.16)'daki gibidir.

$$F_{pervaneB} = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -(b\Omega_1^2 + b\Omega_2^2 + b\Omega_3^2 + b\Omega_4^2) \end{bmatrix} \quad (2.15)$$

$$M_{pervaneB} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -(l * b\Omega_1^2 - l * b\Omega_2^2 + l * b\Omega_3^2 - l * b\Omega_4^2) \end{bmatrix} \quad (2.16)$$

Yer eksen takımında olan yer çekimi kuvveti de quadrotora etkiyen kuvvetler arasındadır. Bu kuvvet de eklenmelidir. Burada  $g$  yer çekimi ivmesidir.

$$F_{yerçekimiY} = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (2.17)$$

İHA üzerine etki eden momentlerden biri de pervanelerden kaynaklı oluşan cayroskop etkisidir. Karşılıklı duran pervaneler birbirine zıt yönde dönse de, net moment sıfır olmamaktadır.  $J$  pervane ataleti,  $\Omega$  pervane dönüş hızı olmak üzere cayroskopik moment denklem (2.18)'de gösterilmiştir.

$$F_{caYG} = \omega \times \begin{bmatrix} 0 \\ 0 \\ J\Omega \end{bmatrix} \quad (2.18)$$

Hava aracı üzerinde oluşan toplam kuvvet ve momentler denklem (2.19)'da gösterilmiştir.

$$F_{top} = \begin{bmatrix} 0 \\ 0 \\ -U_1s \end{bmatrix} + mR_G^Y \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (2.19)$$

$$M_{top} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \omega \times \begin{bmatrix} 0 \\ 0 \\ J\Omega \end{bmatrix} \quad (2.20)$$

Gövde eksen takımına göre açısal hızlar denklem (2.21)'de tanımlanmaktadır.

$$\dot{\omega} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \quad (2.21)$$

Açısal hızlar gövde eksen takımında olduğu için denklem (2.22)'de gösterilen atalet matrisinden bağımsızdır.

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.22)$$

$I_{xx}$ ,  $I_{yy}$  ve  $I_{zz}$  kendi eksenlerindeki atalet değerleridir ve birimi  $kg.m^2$ 'dir. Rotasyonel hareketin denklemleri aşağıda verildiği gibidir.

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J\Omega \end{bmatrix} \quad (2.23)$$

Denklem (2.23) düzenlendiğinde gövde açısal hız ifadesi aşağıdaki gibi olmaktadır.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy}-I_{zz}qr}{I_{xx}} \\ \frac{I_{zz}-I_{xx}pr}{I_{yy}} \\ \frac{I_{xx}-I_{yy}pq}{I_{zz}} \end{bmatrix} + \begin{bmatrix} \frac{U_2}{I_{xx}} \\ \frac{U_3}{I_{yy}} \\ \frac{U_4}{I_{zz}} \end{bmatrix} + \begin{bmatrix} \frac{-J_r q \Omega}{I_{xx}} \\ \frac{-J_r p \Omega}{I_{yy}} \\ 0 \end{bmatrix} \quad (2.24)$$

Denklem (2.24)'ün integrali alınarak gövde eksen takımında açısal hızlar bulunmaktadır. Euler açıları ise gövde açısal hızlarının dönüşüm matrisi ile

çarpımı sonrasında elde edilir.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.25)$$

Euler açısal hızlarının da integrali alındığında euler açıları hesaplanmaktadır.

### 2.3.5 Ötelemenin Hareket Denklemleri

Kuvvetler hava aracında ivmeye sebep olur, ivme ile yer değiştirme meydana gelir. İvmenin bulunmasından sonra hız, sonrasında da pozisyon elde edilir. Gövde eksen takımı hızları yer eksen takımına dönüştürülür.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_Y^G \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.26)$$

Gövde eksen takımındaki ivme, denklem (2.27)'deki gibi elde edilmektedir.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R_Y^G \begin{bmatrix} 0 \\ 0 \\ -U_1/m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (2.27)$$

### 2.3.6 Motor Modeli

Quadrotor sisteminde motor olarak fırçasız doğru akım motoru kullanılmaktadır. Quadrotorun dinamik ve kinematik araç dinamiği modelinden sonra motor modeli yapılacaktır. Fırçasız doğru akım motoru modellenirken elektrik ve mekanik iki model kullanılmaktadır.  $\omega$  motor dönüş hızı,  $\tau$  motorun zaman sabiti,  $d$  sürükleme katsayısı,  $J$  pervane ataleti,  $r$  dişli oranı,  $\eta$  verimlilik,  $R$  motor iç direnci,  $k_m$  motor tork sabiti,  $u$  terminal gerilimi olmak üzere motor model denklemleri denklem (2.28) ve (2.29)'da gösterilmiştir.

$$\dot{\omega} = -\frac{1}{\tau}\omega - \frac{d}{\eta r^3 J}\omega^2 + \frac{1}{k_m \tau}u \quad (2.28)$$

$$\tau = \frac{RJ}{k_m^2} \quad (2.29)$$

Motor, ESC ve pervane modeli birlikte kullanılmıştır. Otopilot çıktısı PWM değerine karşılık gelen itki değeri, ölçülen gecikme değeri eklenerek modellenmiştir. Öncelikle, 10%'dan 90%'a kadar ayırık halde gaz komutu değerlerinde elde edilen itki değerleri

grafik haline getirilip ikinci derece polinom oluşturulmuştur. Denklem (2.30)'da giriş değeri uygulanan PWM değeri, çıkış değeri ise elde edilen itki değeri olmuştur.

$$y = ax^2 + bx + c \quad (2.30)$$



# 3

## GÖRÜNTÜ İŞLEME

---

Bu bölümde kameradan alınan görüntü ile konum kestirimi anlatılmıştır. Algoritma geliştirilmesi ve ilk doğrulama testleri simülasyon ortamında yapılmıştır. Gazebo simülasyon ortamında monoküler kamera sensör modeli bulunmaktadır. Simülasyon ortamında modellenen quadrotor hava aracının altına yerleştirilmiş şekilde kamera konulmuştur. Hava aracı hareket halindeyken kamera sensör çıktısı eş zamanlı olarak C++ kodunda alınıp, işlenmektedir. GNSS konum hassasiyet bilgisi, sinyalin dağ, bina gibi fiziksel engellere takılıp alıcıya ulaşamamasından, atmosferik etkilerden ve alıcının konumundan dolayı düşmektedir. Ayrıca GNSS'lerin konum doğruluğu çektiği uydu sayısı ve anten yerleşimine bağlı olarak değişmekle beraber yatayda 1.5 metre, düşeyde ise 5 metre olabilmektedir.[13] Görsel referans işaretleri kullanılarak görüntü işleme ile hesaplanan konum bilgisi ile santimetre seviyesinde doğruluğa ulaşılmaktadır.

### 3.1 Görsel Referans İşaretçi Sistemleri

Görsel referans işaretleri, tanınması ve birbirinden ayırt edilmesi kolay olacak şekilde tasarlanmış, QR kodlara ve barkodlara benzeyen yapay işaretlerdir. Bu işaretler, algılama ve tanımlama için önemli noktalar. Bunlar, belirli bir figürle tanınarak, bilinen boyut ve şekle sahip yapay işaretlerdir. QR kodlarda kameranın etikete hizalanması gerekmektedir ve QR kodlarda yüzlerce bayt depolanabilmektedir. QR kodun okunabilmesi için, etiketin üzerindeki ışık dağılımının eşit bir şekilde olması gerekmektedir. Görsel referans işaretinde daha az bilgi saklanırken, düşük ışık koşullarında ve kamera hizalanmasına gerek olmadan okunabilmektedir. Barkodun görüntüdeki konumunun önemsiz olduğu 2D barkod sistemlerinden farklı olarak, görsel referans işaretleri etiketin kameraya göre konumunu ve oryantasyonunu bulmayı sağlar. Görsel referans işaretleri aynı zamanda tek bir görüntüde birden fazla işaretleyiciyi algulamak için tasarlanmıştır. Genel olarak, siyah beyaz olarak inşa edilirler. Ancak, bazıları renkli yapıdadır.[14] Genellikle kütüphanelerinde veri kodlama ve kod çözme sistemi, bir kurala göre oluşturulmuş etiket aileleri

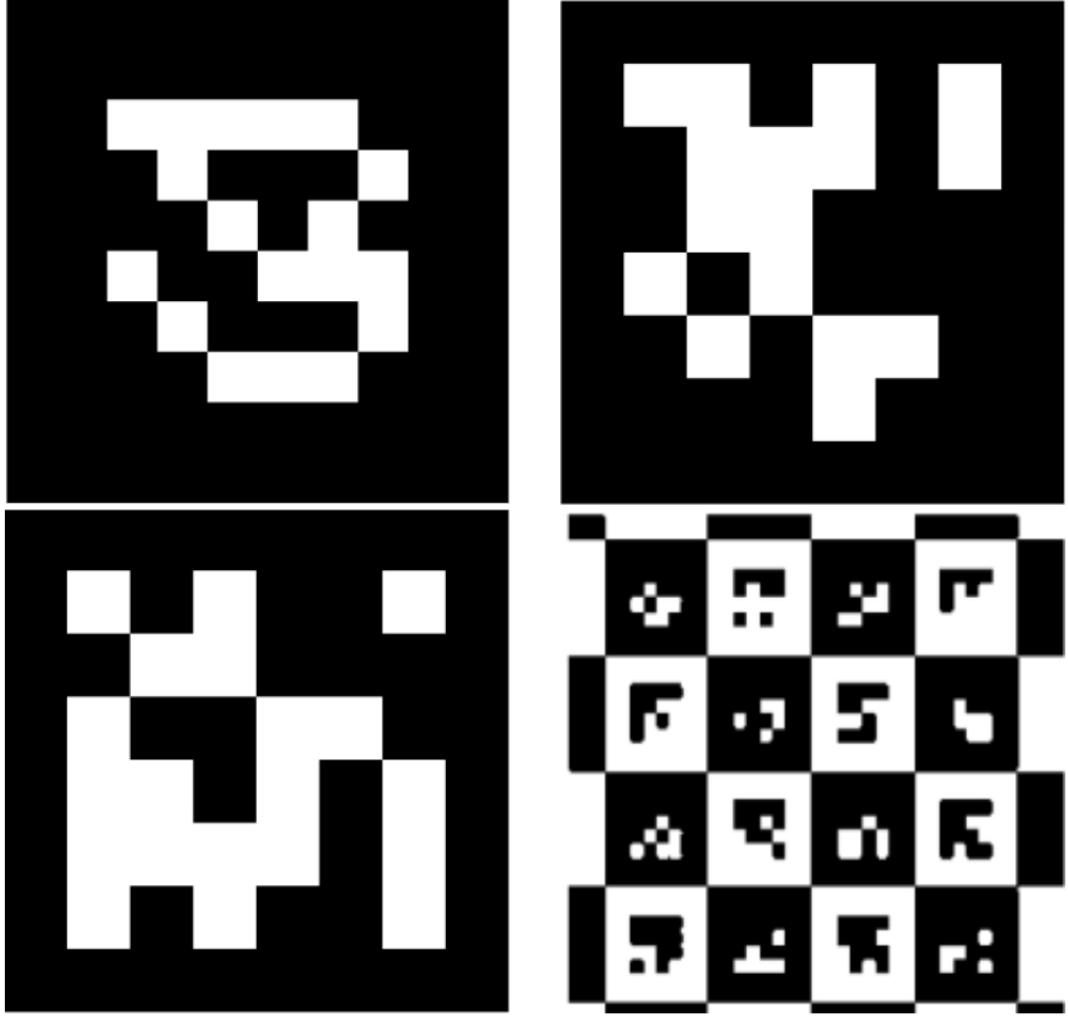
ve belirteçlerini algılama ve doğrulama hesapları mevcuttur.[15] En popüler etiket sistemleri ARTag, AprilTag, ArUco ve CALTag olarak bilinir.[16–19] Bunları kullanmanın avantajlarından biri düşük maliyet ve kurulum kolaylığıdır. Literatürde etiket olarak da isimlendirilen bu işaretler çoğunlukla kolayca basılmak ve sert yüzeylere yapıştırılmak üzere tasarlanmıştır. Etiketlerin sıradan kağıt gibi kolay erişilebilir nesnelere üzerinde kullanılabilir olması, sistemin maliyetini büyük ölçüde düşürür. Bu sistemler, nesne tanıma ve izlemeyi gerçekleştirmek için iki boyutlu görüntüden görel konum sağlayabilir. Bu sistemlerin etiketlerin tespiti ve poz tahmini için kendi algoritmaları vardır. Bu algoritmalar hız ve doğruluk açısından karşılaştırılır. Şekil 3.1’de etiket sistemlerinden bazıları sırasıyla ArTag, AprilTag, ArUco ve CALTag gösterilmektedir. ArTag, artırılmış gerçeklik uygulamaları için geliştirilmiştir. AprilTag, ArTag’e göre daha iyi Hamming Distance sağlayan algoritmaya sahiptir. ArUco, hızlı kamera kalibrasyonu ve poz kestirimi amacıyla tasarlanmıştır. CALTag ise ArTag’deki lisans kısıtlamalarından dolayı geliştirilmiş bir etikettir. Görsel referans işaretleri lisans modeli, hız, doğruluk, farklı ışık şartlarında çalışabilme konularında farklılık gösterirler.

Her farklı etiketin benzersiz kimliği ve karşılık gelen bir bilgisayarlı görme algoritması tarafından tanınması ve kodunun çözülmesi için benzersiz bir grafik gösterimi vardır. Görsel referans etiketlerinin yaygın olarak kullanıldığı alanlar aşağıda listelenmiştir.[19–21]

- Artırılmış Gerçeklik
- Kamera Kalibrasyonu
- Kapalı ortamlarda konumlandırma
- 3D Bilgisayar Grafikleri
- Tanımlama

### 3.2 Geliştirilen Algoritma

Görsel etiket sistemleri kendi kütüphanelerinde etiketin tespiti ve teşhisi, oryantasyon ve konum hesaplaması için metotları bulunmaktadır. Bu metotlar kendi etiket sistemlerinde çalışmaktadırlar. Geliştirilen algoritma herhangi etiket sistemiyle çalışmakta olup, tespit ve poz hesabı yapabilmektedir. Bunun için girdi olarak tespit edilecek etiketin 8x8 taslak ikili (binary) bilgisi ve etiketin bir kenar uzunluğu yeterli olmaktadır. Farklı ışık şartları ve farklı kameralarda çalışması için bazı filtre katsayıları

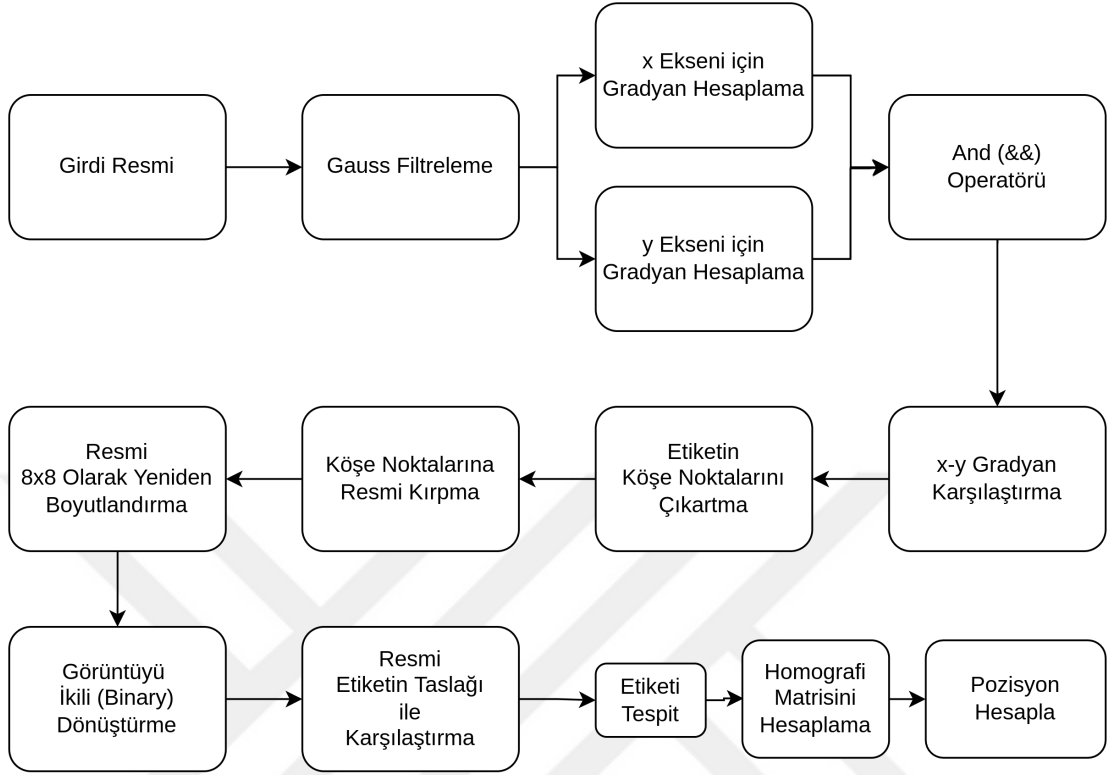


Şekil 3.1 ArTag, AprilTag, ArUco ve CALTag etiket sistemlerinden örnekler



Şekil 3.2 AprilTag'in robotik araçlar için seyrüsefer uygulaması

ve kamera matrisi parametrik olacak şekilde ayarlanmıştır. Algoritma adımları Şekil 3.3'de gösterilmiştir. İlk olarak etiketlin tespiti yapılmaktadır. Sonrasında etiket



Şekil 3.3 Algoritma adımları

ve kamera arasındaki konum hesabı yapılmaktadır. Daha sonra ise Kalman filtresi uygulanmaktadır.

Kameradan alınan görüntü gri görüntüye dönüştürülür. Normalizasyon uygulanır. Daha sonra  $x$  ve  $y$  ekseninde gradyan uygulanır. Gradyan işleminde  $3 \times 3$  kernel matrisi kullanılır. Bu işlemde  $5 \times 5$  kernel matrisi kullanmak daha stabil sonuçlar vermesine katkı sağlayacak olsa da yüksek hesaplama yükü sebebiyle kullanılmamıştır.  $5 \times 5$  kernel matrisi kullanıldığında kodun çalışma frekansı büyük ölçüde düşmüştür. Bu sebeple  $3 \times 3$  kernel kullanımı ile devam edilmiştir. Gradyan işlemi uygulanırken etiketi daha net tespit edebilmek için  $x$  ve  $y$  ekseninde çizgi devamlılığının kontrolü yapılmak amacıyla filtrelemeler uygulanmıştır. Böylelikle köşe noktaları tespit edilirken daha net sonuçlar alınmaktadır. Denklem (3.1)'de ise gradyan büyüklüğü hesaplanmaktadır. Bu aşamadan sonra Şekil 3.5'de görüldüğü üzere çıktı alınmaktadır. Fotoğraftaki şekillerin kontur bilgilerinin bulunmasından sonra köşelerin bulunma işlemi uygulanacaktır. Önceki adımda uygulanan filtreleme işlemi önem arz etmektedir. Filtreleme yapılmadığında fotoğraf üzerinde çok fazla şekil bulunmakta ve dolayısıyla çok fazla köşe noktası bulunmaktadır. Bu da kodun yavaş çalışmasına sebep olmaktadır.

Farklı ışık şartları, gölge durumları, titreşim ve sallantı gibi fiziksel olaylar fotoğraf üzerinde gürültüyü de beraberinde getirir.[22] Bu gürültüler nesnenin tespitine dayanan bu hesaplarda kabul edilemez sonuçlar doğurur. Bu sebeple gürültüleri elimine etmek için gri fotoğrafa gaussian blur işlemi uygulanır.

$$G_{mag} = \sqrt{g_x^2 + g_y^2} \quad (3.1)$$

$$Gauss_{(x,y)} = \frac{1}{2\pi\sigma} \cdot e^{-\frac{(x^2 + y^2)}{2 \cdot \sigma^2}} \quad (3.2)$$

Blur işlemi sonrasında fotoğraftaki potansiyel olarak etiket olabilecek şekiller tespit edilmektedir. Böylelikle, görüntüdeki tüm şekillerin konturları elde edilir. Bu şekillerin kontur yani kenar bilgileri bulunmaktadır. Gradyan işlemi sonucu görüntü üzerinde bulunan şekiller  $x$  ve  $y$  ekseninde devam eden çizgilere dayandığından dolayı gereğinden fazla kontur bilgisi elde edilmektedir. Bu noktada Douglas - Peucker algortiması ile filtreleme yapılarak şekiller üzerindeki kenar bilgileri indirgenmiştir. Sonrasında ise tespit edilen şekillerden dört kenarlı olanlar filtrelenir. Daha sonra dört kenarlı şekillerin köşe koordinatları elde edilir. Kontur bilgileri elde edildikten sonra köşe noktaları bulunur. Bu adımda bulunan dört köşeli şekillerin etiket olup olmadığı kontrol edilir. Etiket farklı oryantasyonlarda ve bakış açılarından tespitinin sağlanması için bulunan şekillere perspektif dönüşüm uygulanır. Perspektif dönüşüm için şeklin dört köşe noktasının  $x$  ve  $y$  koordinatları tek boyutlu dizide küçükten büyüğe olacak şekilde sıralanır. Bu şekilde görüntü üzerindeki şekline hangi rotasyona göre dönüşüm yapılacağı seçilir. Perspektif dönüşüm yapılırken etiketin taslak fotoğrafa göre hangi açıda olduğu bilgisi de tutulur. Bu açı bilgisine göre oluşturulan rotasyon matrisi homografi matrisi ile çarpılacaktır.

Perspektif dönüşüm yapıldıktan sonra kırılan fotoğraf taslak etiket fotoğrafı ile kıyaslanır. Eşleştiği durumda tespit yapılmış olur. Eşleşen şeklin fotoğraf üzerindeki köşe koordinatları konum hesaplamasında kullanılmak üzere döndürülür. Perspektif dönüşüm işlemi sonrasında, köşe noktaları arasında kalan ilgili alan kırılır.  $8 \times 8$  boyuta göre yeniden boyutlandırma yapılır. Yeniden boyutlandırılan gri seviye fotoğraf binary denilen ikili görüntüye çevrilir. İkili görüntü ile, etiketin  $8 \times 8$  ikili taslak görüntüsü karşılaştırılır. Karşılaştırma sonucu aynı çıkarsa etiket tespit edilir. Köşe bilgileri döndürülür.

Kameranın etikete üstten bakıldığı durumda hangi açı ile bulunduğu bilgisi de hesaplanır. 0 ile 45 derece arasında 0 derece, 45 ile 90 derece arasında ise 90 derece

```

Algorithm detectTag ( $I, G_x, G_y, thresh_{min}, thresh_{max}, axis$ )
Input:  $I$ , gray level image.  $t_{min}, t_{max}, G_x$  and  $G_y$  gradient kernel, axis
Output: The detected tag, the corner coordinates

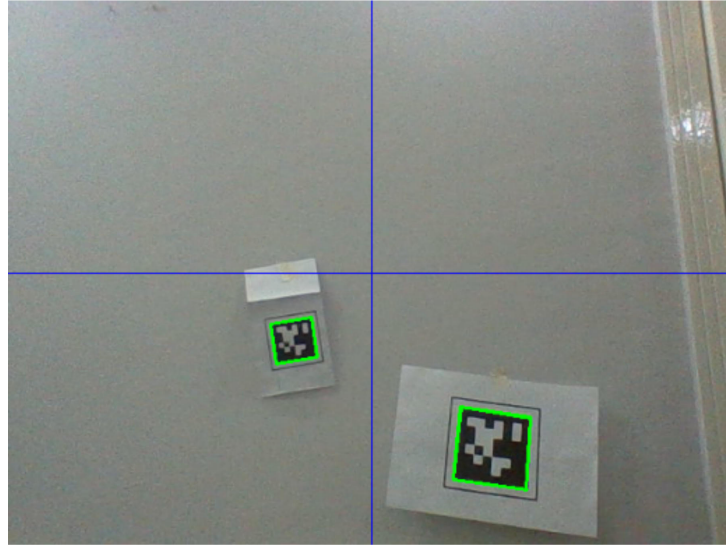
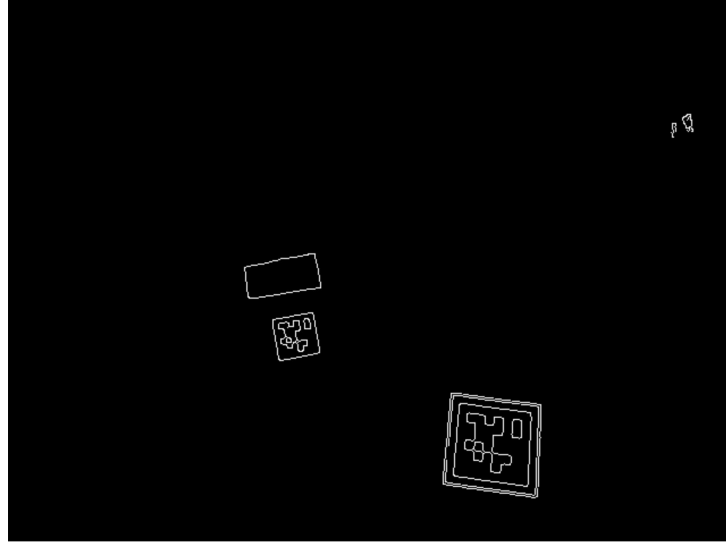
for  $i \leftarrow 0$  to  $I$ . rows do
  for  $j \leftarrow 0$  to  $I$ . cols do
     $gValue_x \leftarrow G_x * I$ 
     $gValue_y \leftarrow G_y * I$ 
    if  $gValue_x > thresh_{min}$  or  $gValue_x < thresh_{max}$  then
       $gValue_x \leftarrow 255$ 
    else
       $gValue_x \leftarrow 0$ 
    if  $gValue_y > thresh_{min}$  or  $gValue_y < thresh_{max}$  then
       $gValue_y \leftarrow 255$ 
    else
       $gValue_y \leftarrow 0$ 
     $G_{mag} \leftarrow \text{sqrt}(gValue_x^2 \times gValue_y^2)$ 
     $imgBlurred \leftarrow \text{Blur}(G_{mag})$ 
     $contours \leftarrow \text{findContour}(imgBlurred)$ 
     $corners \leftarrow \text{findCorners}(contours)$ 
    if  $corners.size == 4$  then
       $\text{perspective}(I, corners)$ 
       $\text{convertBinary}(I)$ 
       $tag_ \leftarrow \text{resize}(I)$ 
      if  $tag_ == \text{template\_tag}$  then
         $detected_ \leftarrow corners$ 
return  $detected_$ 

```

Şekil 3.4 Etiket köşe tespit algoritması

olacak şekilde sırayla, etiket ile kamera arasındaki açı tespit edilir. Bu hesap için etiketin ilk iki köşesini koordinat bilgileri kullanılır.  $x$  ve  $y$  eksenindeki pixel cinsinden konum bilgileri  $\text{atan2}$  fonksiyonu kullanılarak açı bilgisi hesaplanır. Bulunan bu açı homografi matrisi ile çarpılır. Bu sayede kameranın etikete olan görel konum eksen takımına göre doğru elde edilmiş olur.

Denklem (3.3)'te görüldüğü gibi, perspektif dönüşüm matrisi sırasıyla köşe noktalarının sol üst, sağ üst, sol alt ve sağ alt noktalarına göre hesaplanır.  $M$ ,  $3 \times 3$

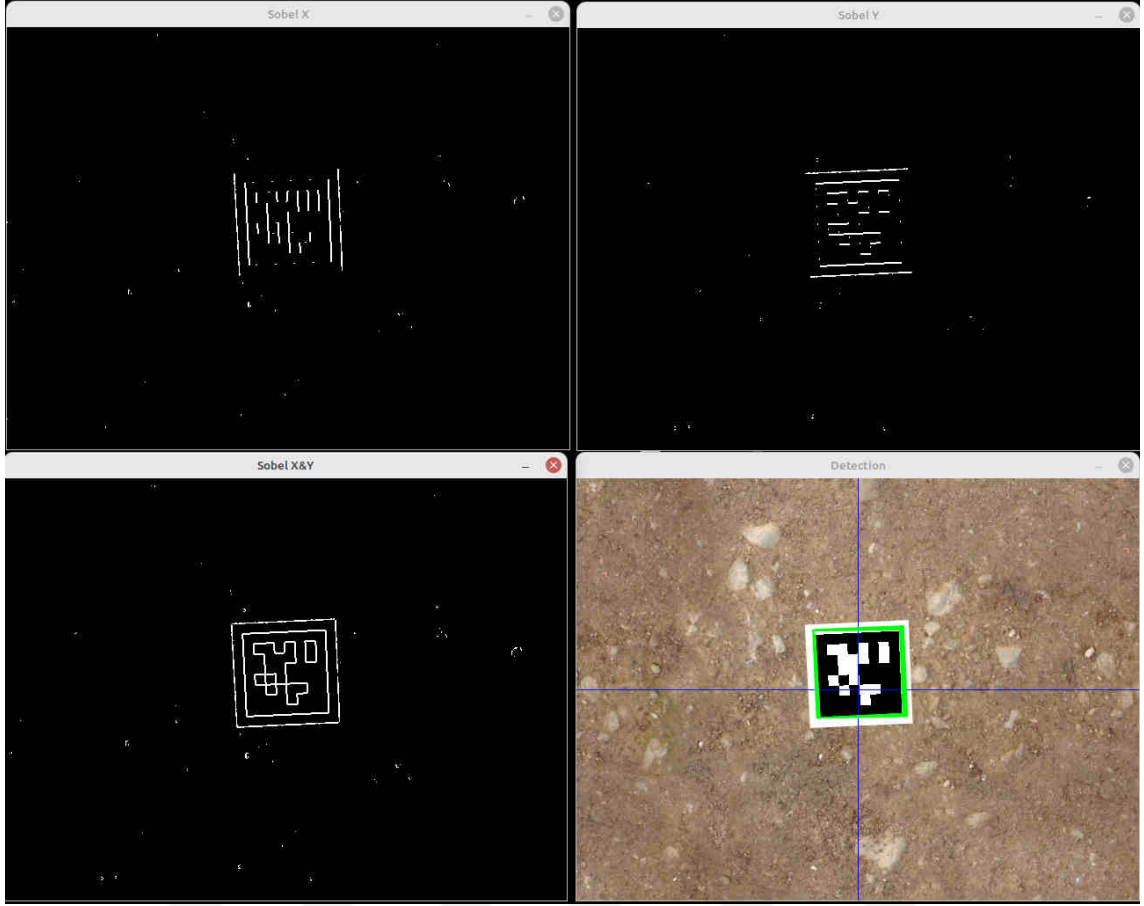


Şekil 3.5 Gradyan, blur ve bazı filtreleme işlemleri sonrası çıktı

perspektif dönüşüm matrisidir.

$$\begin{bmatrix} x' & y' & z' & w' \end{bmatrix} = \begin{bmatrix} x & y & z & w \end{bmatrix} \times \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (3.3)$$

Algoritmanın ikinci aşaması pozisyon hesaplama kısmıdır. Denklem (3.4)'te homografi matrisi, kamera matrisinin dönüşüm matrisi ile çarpımı olarak tanımlanır. Homografi matrisi, belirli bir noktanın görüntülerinin farklı açılardan elde edildiği bir izdüşümdür.[23] Bu ilişki 3x3 boyutlu bir H matrisi ile ifade edilirse, denklem (3.5)



Şekil 3.6 Tespit adımları

kullanılarak homografi matrisi hesaplanabilir.

$$\begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{00} & r_{01} & t_x \\ r_{10} & r_{11} & t_y \\ r_{20} & r_{22} & t_z \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} & h_{03} \\ h_{10} & h_{11} & h_{12} & h_{13} \\ h_{20} & h_{21} & h_{22} & h_{23} \\ h_{30} & h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (3.4)$$

$$x' = H \times x \quad (3.5)$$

Denklem (3.5)'te  $x$  referans görüntünün koordinat bilgisi,  $x'$  ise başka bir açıdan alınan görüntünün koordinat bilgisidir.  $H$  ise  $3 \times 3$  homografi matrisi anlamına gelir.

$$\begin{bmatrix} x'_i \\ y'_i \\ c \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} & h_{03} \\ h_{10} & h_{11} & h_{12} & h_{13} \\ h_{20} & h_{21} & h_{22} & h_{23} \\ h_{30} & h_{31} & h_{32} & h_{33} \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (3.6)$$

İki farklı açıdan alınan görüntüler arasındaki homografi matrisini hesaplamak için aynı düzlemde tek bir doğruya ait olmayan en az dört noktanın koordinat bilgilerinin

bilinmesi gerekir.[24] Bu bağıntı denklem (3.6)'da verilmiştir.

$$x'_i = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}} \quad (3.7)$$

$$y'_i = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}} \quad (3.8)$$

Denklem (3.7) ve (3.8) tek bir koordinat bilgisi için kullanılan denklemlerdir. Bu denklemlerden  $h$  değerlerini elde etmek için denklem (3.9) ve (3.10)'da verilen formüller kullanılmalıdır.

$$a_x^T = h \quad (3.9)$$

$$a_y^T = h \quad (3.10)$$

Denklem (3.11), (3.12) ve (3.13)'te denklem (3.9) ve (3.10)'da kullanılan matrisler verilmiştir.

$$h = [h_{00} \ h_{01} \ h_{02} \ h_{10} \ h_{11} \ h_{12} \ h_{20} \ h_{21} \ h_{22}]^T \quad (3.11)$$

$$a_x = [-x_1 \ -y_1 \ -1 \ 0 \ 0 \ 0 \ x'_2x_1 \ x'_2y_1 \ x'_2]^T \quad (3.12)$$

$$a_y = [0 \ 0 \ 0 \ -x_1 \ -y_1 \ 1 \ y'_2x_1 \ y'_2y_1 \ y'_2]^T \quad (3.13)$$

Denklem (3.12) ve (3.13)'ü çözmek için, verilen bir dizi koordinat bilgisi için, denklem (3.14)'de gösterilen aşağıdaki doğrudan doğrusal dönüşüm (Direct Linear Transformation) formülü kullanılır.

$$A \times h = 0 \quad (3.14)$$

Denklem (3.15),  $n$  koordinat bilgisi için oluşturulan  $A$  matrisi gösterilmektedir.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ & & & & & & \vdots & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n & -y'_n \end{bmatrix} \quad (3.15)$$

Denklem (3.15), denklem (3.16)'da verilen Tekil Değer Ayırıştırma (Singular Value

Decomposition - SVD ) algoritması ile çözülür.

$$A = U\Sigma V^T = \sum_{i=1}^9 \partial_i u_i v_i^T \quad (3.16)$$

SVD'deki  $\partial_i$  değerine karşılık gelen sağ tekil vektör ( $v$ ), homografi matrisini [24] içeren çözümdür. Homografi matrisini hesaplamak için kamera modeli matrisi kamera kalibrasyonu sonrasında hesaplanır. Denklem (3.17)'te verilen kamera matrisi, kameranın odak uzaklığı ( $f_x, f_y$ ) ve optik merkez ( $c_x, c_y$ ) bilgilerini içermektedir. [25] Görüntüdeki bozulmalar görüntü işleme işlemlerinde hatalı sonuçlara sebep olmaktadır. Bu sebeple kamera kaynaklı bozulmaların düzeltilmesi gerekmektedir. Düzeltme işlemi ise genelde yazılımsal olarak yapılmaktadır. Görüntüdeki bozulmalar kameranın sensörü, lens, ışık şiddeti veya farklı açı gibi sebeplerden meydana gelmektedir. Bu bozulmaların en sık olanı radyal bozulmalardır. Radyal bozulmalar fıçı bozulması, içbüyük bozulma ve bıyık biçiminde bozulma olarak üçe ayrılırlar. Kamera kalibrasyon işleminde [26] bu bozulmaları düzelten işlem yapılır. Kalibrasyon işlemi sonucu denklem (3.17)'deki kamera kalibrasyon matrisi elde edilir.

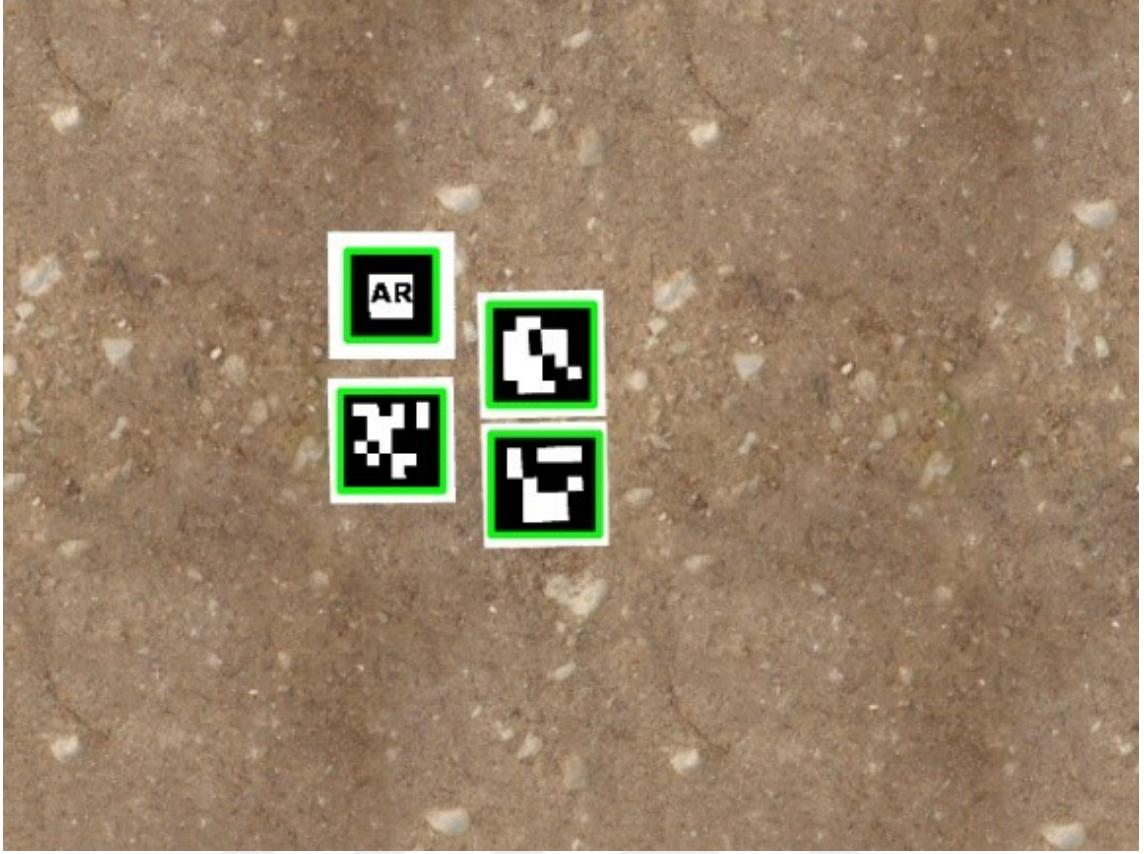
$$\begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

Şekil 3.7'de, önerilen algoritmanın bir sonucu olarak aynı anda dört farklı etiketin tespit edilmesi gösterilmektedir.

### 3.3 Kalman Filtresi Uygulaması

Etikete göre hesaplanan konum bilgisi bazen aracın dolayısıyla kameranın hızlı hareketi ve bozulmaları nedeniyle hatalı olabilmektedir. Bu çalışmada, hesaplanan poz değerlerinin hata ve sapmalarını en aza indirmek için tek boyutlu Kalman filtresi [27] uygulanmıştır. Geliştirilen konum kestirim algoritması, yerel  $x$ ,  $y$  ve  $z$  koordinatlarını hesaplar ve bu koordinatlar Kalman filtresine girdi olarak verilir. Ardından Kalman filtresi, çıktı olarak bu koordinatların daha olası ve daha düzgün sonuçlarını verir. Denklem (3.18) Kalman kazancının denklemini tanımlar.  $K_n$  Kalman kazancı,  $p_{n,n-1}$  kestirimdeki belirsizlik ve  $r_n$  ölçüm belirsizliğidir.

$$K_n = \frac{p_{n,n-1}}{p_{n,n-1} + r_n} \quad (3.18)$$



Şekil 3.7 Farklı aileye ait etiketlerin aynı anda tespiti

Kalman kazancı denklem (3.18)'de hesaplanır ve denklem (3.19)'da gösterildiği gibi 0 ile 1 arasında olması gerekir.

$$0 \leq K_n \leq 1 \quad (3.19)$$

Denklem (3.20)'de çalışmada kullanılan durum kestirim güncelleme denklemi verilmiştir. Burada  $\hat{x}_{n,n-1}$  sistem durumu kestirim vektörüdür. Aşağıdaki denklemde görüldüğü gibi  $1 - K_n$  tahmini değere verilen ağırlık,  $K_n$  ise ölçüm değerine verilen ağırlıktır.

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(z_n - \hat{x}_{n,n-1}) = (1 - K_n)\hat{x}_{n,n-1} + K_n z_n \quad (3.20)$$

Kalman kazancı, verilen bir ölçümle kestirimi ne kadar değiştirmek istediğinizi söyler. Ölçüm belirsizliği büyük ve kestirim belirsizliği küçük olduğunda sıfıra yakındır. Öte yandan, ölçüm belirsizliği küçük ve kestirim belirsizliği büyük olduğunda, bire yakınsar. Kalman kazancı, kestirim ve ölçüm için verilen ağırlıklar değiştirilerek belirlenebilir.

$$P_{n,n} = (1 - K_n)p_{n,n-1} \quad (3.21)$$

Denklem (3.21)'de kestirim belirsizliği güncelleme denklemi verilmiştir. Bu denklem, mevcut durumun kestirim belirsizliğini günceller. Buna “kovaryans güncelleme denklemi” denir.[28]  $(1 - K_n) < 1$  olduğundan, her filtre yinelemesinde kestirim

belirsizliği küçülür. Kalman kazancı düşükse kestirim belirsizliğinin yakınsaması yavaş olacaktır. Ancak Kalman kazancı yüksek olduğunda kestirim belirsizliği hızla sıfıra yakınsar.

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} + \Delta t \hat{\dot{x}}_{n,n} \quad (3.22)$$

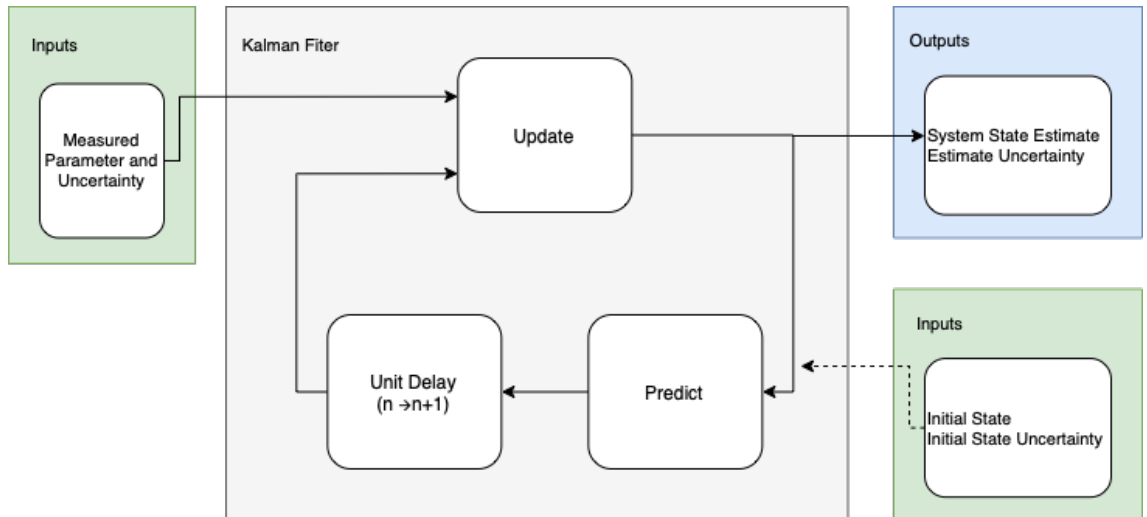
$$\hat{x}_{n,n-1} = \hat{x}_{n,n} \quad (3.23)$$

$$p_{n,n+1} = p_{n,n} + q \quad (3.24)$$

Kestirim belirsizliği ekstrapolasyonu, durum ekstrapolasyonu gibi dinamik bir model denklemi ile yapılır. Denklem (3.22)'de, kestirilen konum bilgisi, mevcut tahmin edilen pozisyon ile delta zamanı ile çarpılan mevcut tahmin edilen hızın toplamına eşittir. Kestirim hız değeri, sabit hız modeli denklem (3.23) varsayıldığında, mevcut hız tahminine eşittir.

Etiket konumu sabit olduğu için sabit hız dinamiği kullanılır. Hesaplanan konumlar, etiket referans noktası alınarak uçağın konumu olarak kabul edilmiştir. Kalman filtresinin uygulanması sırasında sonuçlarda gecikme hatası gözlemlendi. Gecikme hatasını ortadan kaldırmak ve tahminin güvenilirliğini artırmak için sisteme süreç gürültüsü ( $q$ ) eklenmiştir. Süreç gürültüsü artırılarak hata azaltılmıştır. [29]

Bu çalışmada kullanılan Kalman yönteminin basit bir şeması Şekil 3.8'de gösterilmektedir.



**Şekil 3.8** Kalman filtresi blok diagramı

Uygulanan filtrenin girdileri başlangıç durumu ve ölçüm olmak üzere iki kısımda incelenir. İlk sistem durumu ( $\hat{x}_{1,0}$ ) ve ilk durum belirsizliği ( $p_{1,0}$ ) ilk girişler için verilmiştir. Bu girdiler sistemde deneme yanılma yöntemi ile en iyi sonucu verecek

şekilde belirlenmiştir. Ölçüm girdileri, ölçülen sistem durumu ( $z_n$ ) ve belirsizliğidir ( $r_n$ ). Filtrenin çıktıları, sistem durumu tahmini ( $\hat{x}_{n,n}$ ) ve kestirim belirsizliğidir ( $p_{n,n}$ ).  $p_{n,n}$ ,  $(1 - K_n) < 1$ 'den, kademeli olarak azalmalıdır.

**Tablo 3.1** Kalman Filtresi Katsayı Tablosu

Katsayı	Değer
$\hat{x}_{n,n}$	1
$p_{1,0}$	1000
$r_n$	3
q	0.09

# 4

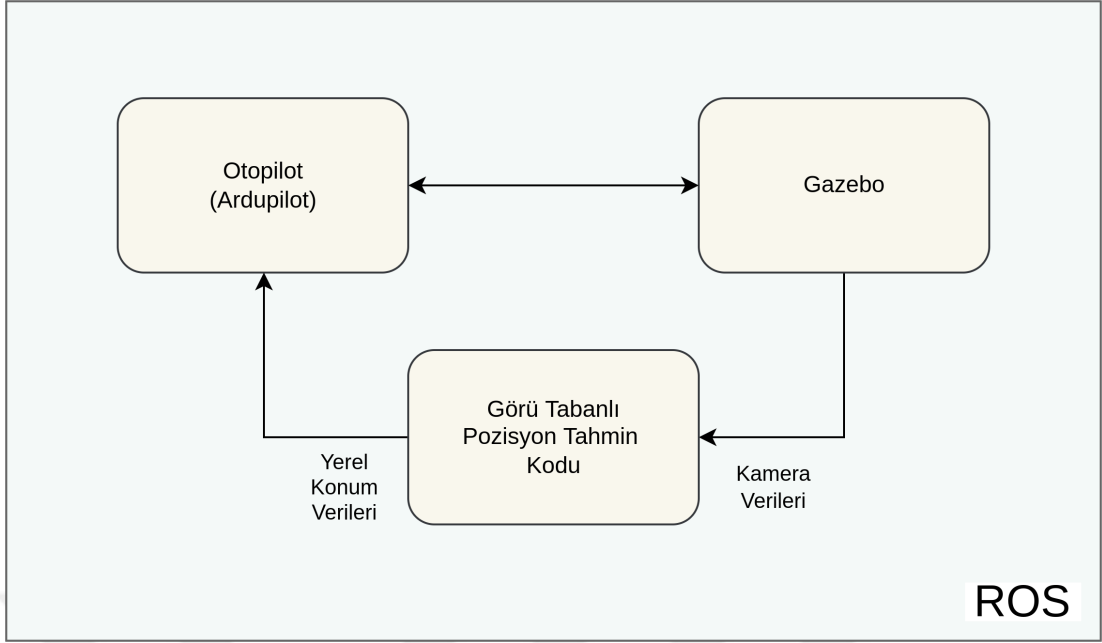
## SONUÇ VE ÖNERİLER

---

Geliştirilen algoritma hem simülasyon ortamında hem de gerçek test düzeneğinde test edilmiştir. Simülasyon ortamı doğruluk oranı yüksek modellerle çalıştığından ötürü algoritma geliştirme aşamasında sıklıkla kullanılmıştır. Geliştirilen algoritma çıktısı konum kestirim değerleri İHA'nın otonom inişinde kullanılmıştır. Simülasyon ortamındaki geliştirmeler sonrasında quadrotora bağlı bir kamera ile metrik ölçümlerin yapıldığı bir test düzeneğinde doğrulanmıştır.

### 4.1 Simülasyon Ortamı

Simülasyon ortamı linux işletim sistemi üzerinde çalışan Robotik İşletim Sistemi [30] (ROS) tabanlı bir yapıda oluşturulmuştur. Quadrotor matematiksel modeli, tezin ikinci bölümdeki dinamik ve kinematik denklemlere göre Gazebo [31] isimli simülasyon ortamında geliştirilmiştir. Gazebo ortamı seçilmesinin sebebi, hem kamera verisinin bir sensör çıktısı olarak veriliyor olması hem de görselleştirme yazılımının olmasıdır. Gazedodan alınan kamera verisi ile görüntü tabanlı algoritma C/C++ dili ile geliştirilmiştir. Konum kestirim kodu, otopilot kodu ve gazebo ROS ortamında birbirleriyle haberleşecek şekilde çalıştırılmaktadır. Otopilot kodundan gelen motor komutları gazebo ortamındaki matematiksel modele girmektedir. Buradaki dinamik denklemlere göre döner kanat hava aracının 6 serbestlik dereceli, üç eksenindeki duruş açıları ve ve açısız hız bilgileri, yine üç ekseninde hız ve konum bilgileri üretilmektedir. Sonrasında bu model çıktıları otopilot koduna gönderilmektedir. Gazebo ortamındaki quadrotor hava aracı üstünde alta bakacak şekilde entegre edilmiş kameradan, görüntü verisi görüntü tabanlı konum kestirim koduna gönderilmektedir. Bu kodda üretilen konum bilgisi de otopilot koduna gönderilmektedir. Oluşturulan simülasyon ortamının blok diagramı Şekil 4.1'de gösterilmiştir.



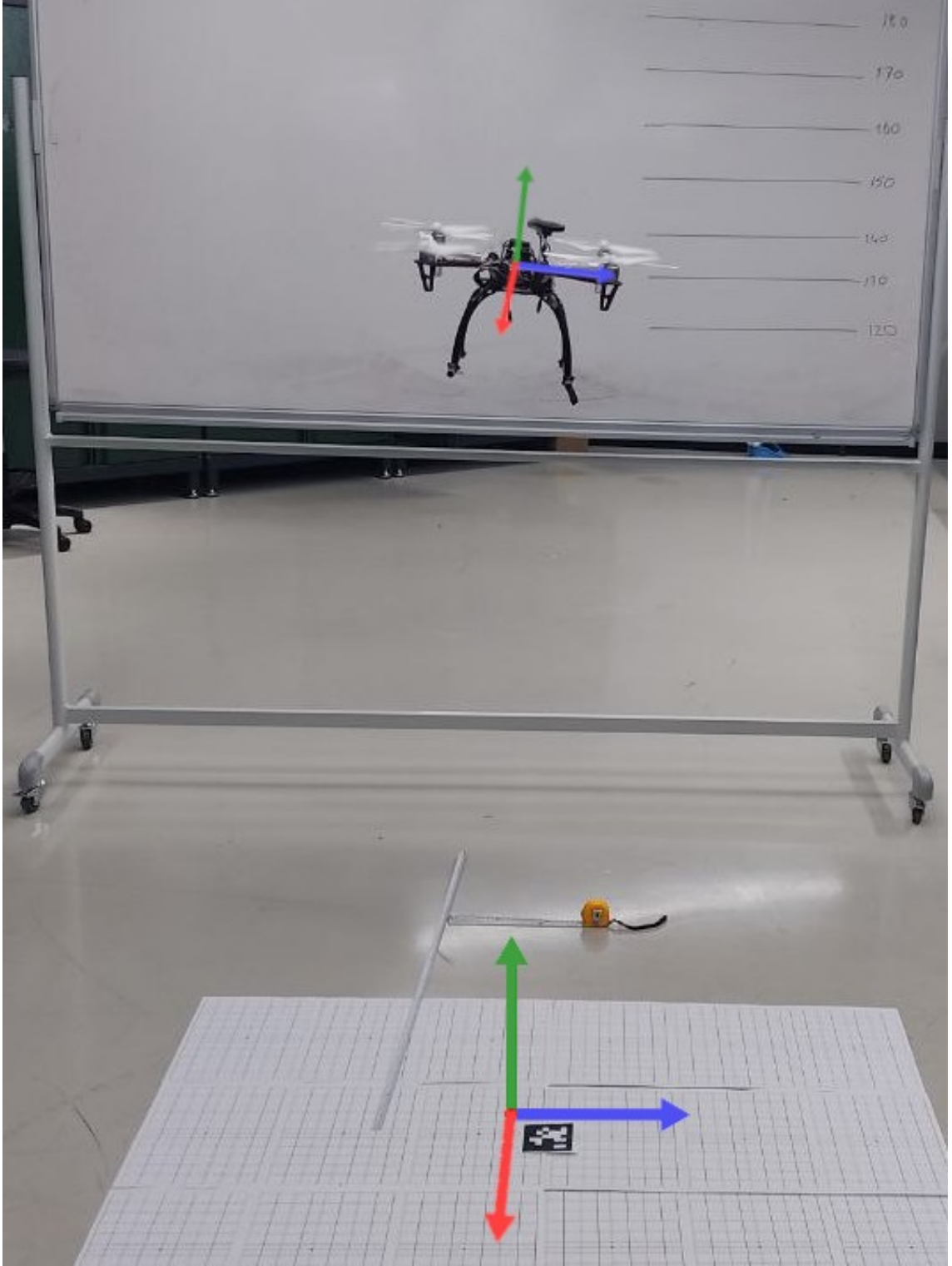
Şekil 4.1 Simülasyon ortamı blok diyagramı

## 4.2 Test Düzeneği Ortamı

Algoritmanın gerçek ortamda doğrulaması için bir test düzeneği oluşturulmuştur. Quadrotorun altında yere bakacak şekilde bir kamera monte edilmiş olup, zeminde ise quadrotor ve etiket arasındaki ölçümün net ve kolay yapılması için metrik çizgilerden oluşan beyaz kağıtlar yerleştirilmiştir. Bahsedilen test ortamı Şekil 4.2'de gösterilmiştir.

## 4.3 Analizler

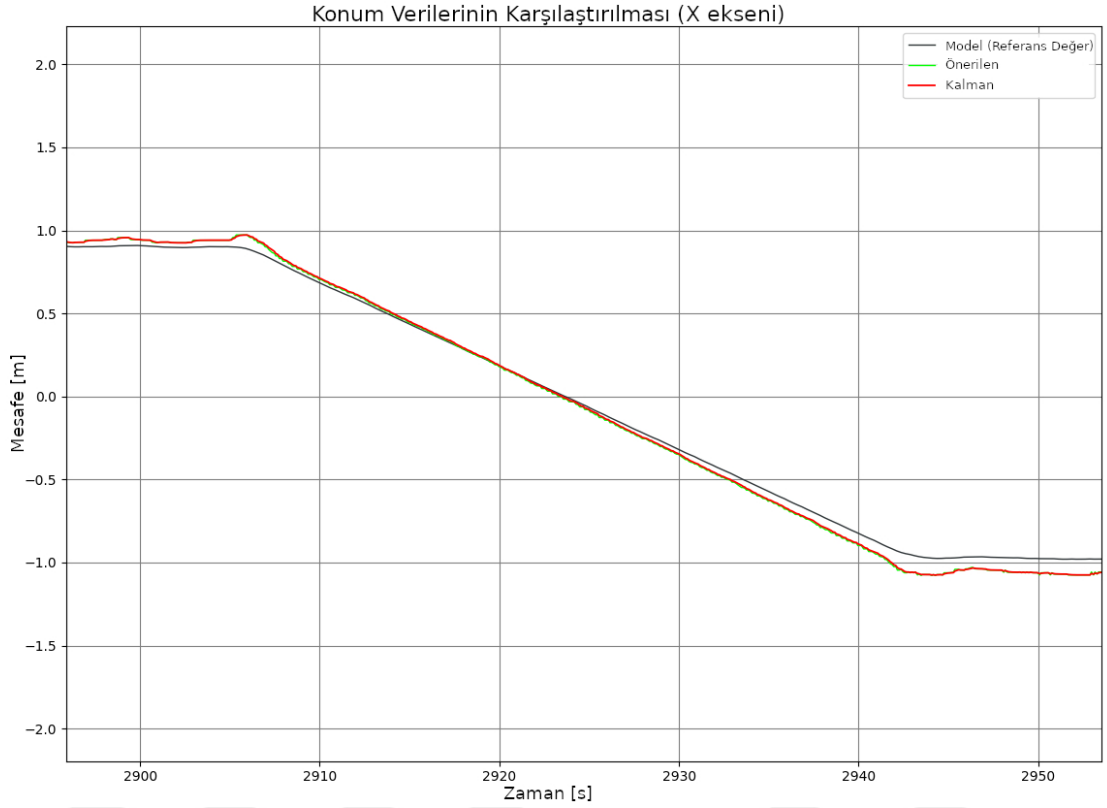
Önerilen metot öncelikle simülasyon ortamında quadrotor araç dinamik modeli ile doğrulanmıştır. Quadrotor dinamik modeli çıktı verileri, algoritma çıktısı olan konum verileri karşılaştırma amacıyla ROS ortamında bulunan matplotlib grafik kütüphanesi ile çizdirildi. Simülasyon çalışma hızı gerçeğe yakın değerler üretmesi için 1000 Hz olarak ayarlandı. Grafiklerde hava aracı matematiksel model çıktısı referans değer olarak kullanılmıştır. Model çıktısı lokal X,Y ve Z konum, geliştirilen algoritma çıktısı ve algoritma çıktısına uygulanmış Kalman filtresi çıktıları aynı grafikte çizdirilerek karşılaştırılmıştır. Grafiklerin  $x$  eksenini zamanı ifade ederken,  $y$  eksenini metre cinsinde konumu ifade etmektedir. Grafiklerde "Model (Referans)" olarak ifade edilen değer matematiksel model çıktısı iken, "Önerilen" olarak gösterilen algoritma çıktısı, "Kalman" olarak ifade edilen değer ise algoritma çıktısına uygulanmış Kalman filtresi sonucu olarak gösterilmektedir. Bu şekilde Kalman filtresinin performansı değerlendirilmiştir. Grafikler, Şekil 4.3, Şekil 4.4 ve Şekil 4.5'de gösterilmiştir. Şekil



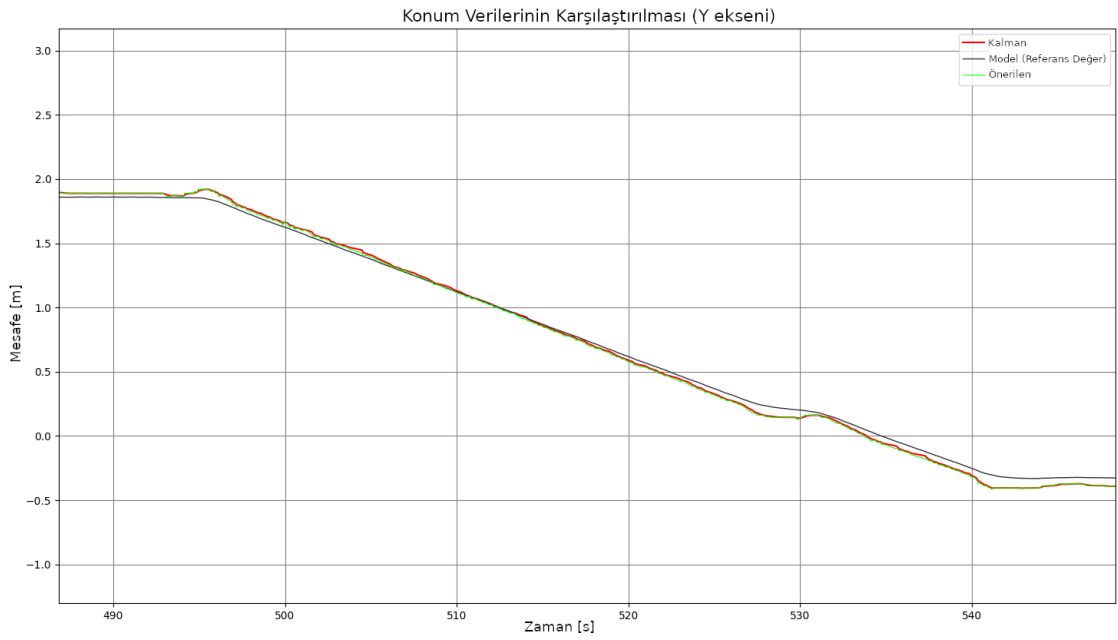
**Şekil 4.2** Test ortamı

4.5'te önerilen metotta konum sapması en kötü 35 cm saparken, Kalman filtresi uygulanmış halinde 15 cm sapmıştır. Şekil 4.3, Şekil 4.4 ve Şekil 4.5 incelendiğinde Kalman filtresi ve önerilen metot sonuçları genelde yakın takip etmekle birlikte Kalman filtresi sonuçları modeli yani referans değeri daha yumuşak takip ettiği gözlenmiştir. Kalman filtresinin etkisi özellikle, kameranın hızlı hareketlerinden ve

görüntüdeki bozulma gibi etkilerden kaynaklı iyi takip edilememe durumlarında etkili olmaktadır. Şekil 4.5'te görüldüğü üzere referans değere daha yakın sonuçlar ürettiği gözlenmiştir.

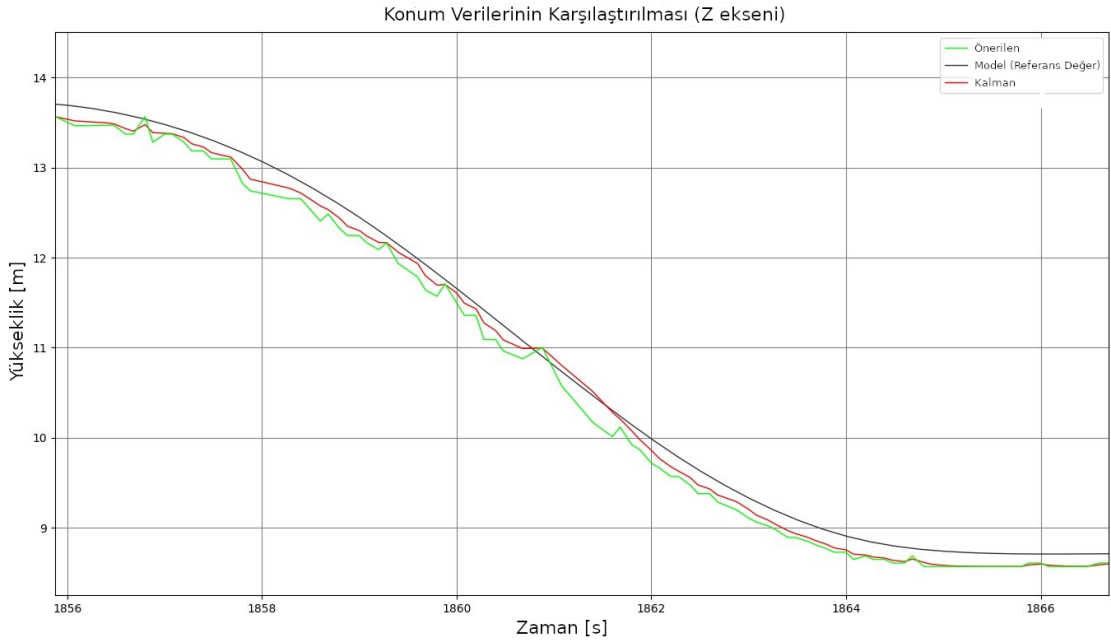


Şekil 4.3 X eksen konum karşılaştırması



Şekil 4.4 Y eksen konum karşılaştırması

Test ortamında ölçülen değerlerin karşılaştırması Şekil 4.6'da verilmiştir. Konum



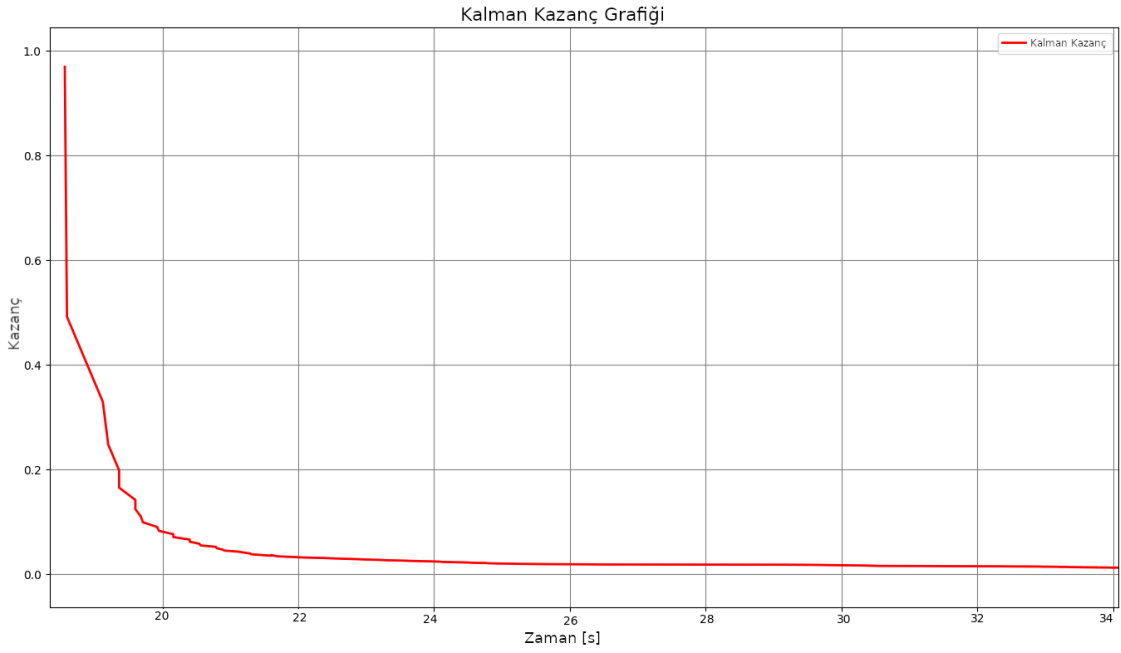
**Şekil 4.5** Z eksen konum karşılaştırması

çıktıları etikete göre kameranın dolayısıyla İHA'nın konum değerleridir. Gerçek ölçülen değer İHA'nın test ortamındaki etikete göre metrik ölçümüdür. Algortima çıktısı quadrotor İHA üstünde bulunan bilgisayar ile hesaplanan çıktıdır.

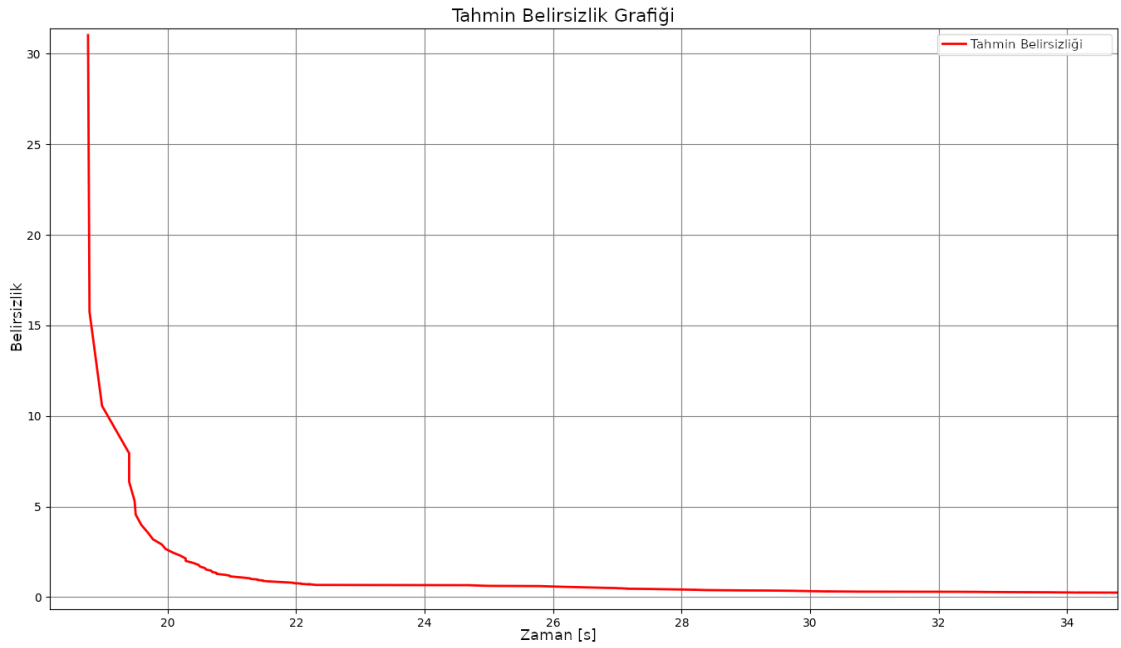
Deney No:	Gerçek Ölçülen Değer			Kalman Filtresi ile Geliştirilen Algortima Çıktısı		
	X	Y	Z	X	Y	Z
1	30 cm	10 cm	100 cm	28 cm	9cm	105cm
2	20 cm	30 cm	150 cm	18cm	32cm	146cm
3	50 cm	40 cm	200 cm	47cm	35cm	205cm
4	70 cm	30 cm	180 cm	76 cm	28 cm	187cm
5	50 cm	90 cm	45 cm	57 cm	88 cm	44 cm

**Şekil 4.6** Test ortamında ölçümlerin kıyaslanması

Kalman kazancı ise Şekil 4.7'de incelenmiştir. Kazanç değeri 0.97'den başlayarak 0'a yakınsamıştır. Kazancın sıfıra yakınsaması ölçüm ağırlığının ve kestirim belirsizliğinin kademeli olarak azaldığının göstergesidir. Belirsizlik değerinin ise 31'den 0'a yakınsaması Şekil 4.8'de görülmektedir. Kalman kazancı yüksek ise kestirim belirsizliği hızlı yakınsamaktadır. Kazanç düştüğü durumda ise, belirsizlik yakınsaması Şekil 4.8'de görüldüğü üzere yavaşlamaktadır. Böylelikle kestirim güvenilirlik oranının arttığı söylenebilir.



Şekil 4.7 Kalman kazancı



Şekil 4.8 Kestirim belirsizliği

#### 4.4 Sonuç

Bu çalışmada, görsel referans işaretçileri kullanılarak geliştirilen görüntü tabanlı bir konum kestirim algoritması geliştirilmiştir. Konum bilgisinin daha kesin çıktılarını almak ve bozulma etkilerinden dolayı kararsız kontrol gibi durumların önüne geçmek için algoritmanın sonucuna tek boyutlu Kalman Filtresi uygulanmıştır. Çalışma, simülasyon ve gerçek deneysel ortamda test edilmiştir. Test sonuçları incelendiğinde,

çıktıların gerçek değere yakın, kabul edilebilir bir hassasiyetle özellikle Z ekseninde GPS sisteminden daha hassas şekilde hesaplandığı yorumlanmıştır. Bu uygulama, GPS olmadığı durumlarda veya sensör hataları gibi koşullar altında konum kestirimi için bir alternatif olarak önerilmektedir. Gelecekteki potansiyel bir çalışma, uçak gemileri gibi iniş platformunun 3 boyutlu hareketine sahip olduğu koşullarda konum tahmini geliştirmesi olacaktır. Ayrıca aracın etikete oryantasyonu hesaplanabilir, etiketin kadrajda olmadığı durumlarda yine kamera sensörü kullanarak optik akış algoritması gibi algoritmalarla farklı sensör füzyon yaklaşımları entegre edilerek etiket olmadığı durumlarda da kararlı konum kestirim çözümü sunulabilir.



- [1] K. B. Ariyur, K. O. Fregene, “Autonomous tracking of a ground vehicle by a uav,” in *2008 American Control Conference*, IEEE, 2008, pp. 669–671.
- [2] H. Başak, M. Gülen, “İnsansız hava aracı kazalarının önlenmesi için risk ölçümü ve yönetimi modeli,” *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 14, no. 1, pp. 55–65, 2010.
- [3] S. Arora, S. Jain, S. Scherer, S. Nuske, L. Chamberlain, S. Singh, “Infrastructure-free shipdeck tracking for autonomous landing,” in *2013 IEEE international conference on robotics and automation*, IEEE, 2013, pp. 323–330.
- [4] D. Iwakura, W. Wang, K. Nonami, M. Haley, “Movable range-finding sensor system and precise automated landing of quad-rotor mav,” *Journal of System Design and Dynamics*, vol. 5, no. 1, pp. 17–29, 2011.
- [5] P. Chmielewski, K. Sibilski, “Ground speed optical estimator for miniature uav,” *Sensors*, vol. 21, no. 8, p. 2754, 2021.
- [6] S. C. Diamantas, P. Dasgupta, “Active vision speed estimation from optical flow,” in *Conference Towards Autonomous Robotic Systems*, Springer, 2013, pp. 173–184.
- [7] A. Gautam, M. Singh, P. B. Sujit, S. Saripalli, “Autonomous quadcopter landing on a moving target,” *Sensors*, vol. 22, no. 3, p. 1116, 2022.
- [8] B. Herissé, T. Hamel, R. Mahony, F.-X. Russotto, “Landing a vtol unmanned aerial vehicle on a moving platform using optical flow,” *IEEE Transactions on robotics*, vol. 28, no. 1, pp. 77–89, 2011.
- [9] A. Vurgun, “İnsansız hava araçlarında görüntü işleme tabanlı iniş sistemi tasarımı ve gerçekleştirilmesi,” M.S. thesis, Fen Bilimleri Enstitüsü.
- [10] G. Chahine, C. Pradalier, “Survey of monocular slam algorithms in natural environments,” in *2018 15th conference on computer and robot vision (CRV)*, IEEE, 2018, pp. 345–352.
- [11] B. Joshi *et al.*, “Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 7227–7233.
- [12] S. Suzuki *et al.*, “Topological structural analysis of digitized binary images by border following,” *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [13] J.-M. Zogg, *GPS: Essentials of Satellite Navigation: Compendium: Theorie and Principles of Satellite Navigation, Overview of GPS/GNSS Systems and Applications*. U-blox, 2009.

- [14] J. DeGol, T. Bretl, D. Hoiem, “Chromatag: A colored marker and fast detection algorithm,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1472–1481.
- [15] Z. Zhang, Y. Hu, G. Yu, J. Dai, “Deeptag: A general framework for fiducial marker design and detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [16] M. Fiala, “Artag, a fiducial marker system using digital techniques,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, IEEE, vol. 2, 2005, pp. 590–596.
- [17] R. Muñoz-Salinas, M. J. Marin-Jimenez, E. Yeguas-Bolivar, R. Medina-Carnicer, “Mapping and localization from planar markers,” *Pattern Recognition*, vol. 73, pp. 158–171, 2018.
- [18] J. Wang, E. Olson, “Apriltag 2: Efficient and robust fiducial detection,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4193–4198.
- [19] B. Atcheson, F. Heide, W. Heidrich, “Caltag: High precision fiducial markers for camera calibration,” in *VMV*, vol. 10, 2010, pp. 41–48.
- [20] D. Wagner, D. Schmalstieg, “Artoolkitplus for pose tracking on mobile devices,” 2007.
- [21] L. Makabe, H. Santo, F. Okura, Y. Matsushita, “Shape-coded aruco: Fiducial marker for bridging 2d and 3d modalities,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2655–2664.
- [22] A. K. Boyat, B. K. Joshi, “A review paper: Noise models in digital image processing,” *arXiv preprint arXiv:1505.03489*, 2015.
- [23] E. Özüağ, S. Ertürk, “A homography matrix decomposition based video synchronization approach,” in *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2014, pp. 2043–2046.
- [24] V. Klema, A. Laub, “The singular value decomposition: Its computation and some applications,” *IEEE Transactions on automatic control*, vol. 25, no. 2, pp. 164–176, 1980.
- [25] Y. Ito, Y. Oda, “Estimation of camera projection matrix using linear matrix inequalities,” in *2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)*, IEEE, 2016, pp. 72–75.
- [26] R. Hartley, A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003, pp. 178–194.
- [27] A. Becker, “Kalman filter overview,” URL: [https://www.kalmanfilter.net/default.aspx\(hämtad 2019-05-14\)](https://www.kalmanfilter.net/default.aspx(hämtad%202019-05-14)), 2018.
- [28] C. K. Chui, G. Chen, *et al.*, *Kalman filtering*. Springer, 2017.
- [29] M. Gonsalves, R. Dinesh, K. Santha, “Two-dimensional adaptive filtering using the kalman algorithm,” in *TENCON 2006-2006 IEEE Region 10 Conference*, IEEE, 2006, pp. 1–4.

- [30] Stanford Artificial Intelligence Laboratory et al., *Robotic operating system*, version ROS Melodic Morenia, May 23, 2018. [Online]. Available: <https://www.ros.org>.
- [31] N. Koenig, A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727.



## TEZDEN ÜRETİLMİŞ YAYINLAR

---

### **Konferans Bildirisi**

1. Mücteba Uzunođlu, Muharrem Mercimek and Recep Behlül Şahin, "Vision-Based Position Estimation with Markers For Quadrotors" 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, 2022

