

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**MİKRO HAVA ARAÇLARININ BİLİNMEYEN ORTAMLARDA
GÖRÜNTÜ TEMELLİ KONTROLÜ**

YÜKSEK LİSANS TEZİ

Cihat Bora YİĞİT

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

HAZİRAN 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**MİKRO HAVA ARAÇLARININ BİLİNMEYEN ORTAMLARDA
GÖRÜNTÜ TEMELLİ KONTROLÜ**

YÜKSEK LİSANS TEZİ

**Cihat Bora YİĞİT
(518091041)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Erdinç Altuğ

HAZİRAN 2012

İTÜ, Fen Bilimleri Enstitüsü'nün 518091041 numaralı Yüksek Lisans Öğrencisi **Cihat Bora YİĞİT**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**MİKRO HAVA ARAÇLARININ BİLİNMEYEN ORTAMLARDA GÖRÜNTÜ TEMELLİ KONTROLÜ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Yrd. Doç. Dr. Erdinç Altuğ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Şeniz Ertuğrul**
İstanbul Teknik Üniversitesi

Yrd. Doç. Dr. Pınar Boyraz
İstanbul Teknik Üniversitesi

Teslim Tarihi : **4 Mayıs 2012**
Savunma Tarihi : **7 Haziran 2012**

ÖNSÖZ

İTÜ BAP birimi tarafından desteklenen bu tez çalışmasında insansız hava araçlarının daha önceden bilinmeyen ortamlarda görüntü ile kontrolü üzerine çalışılmıştır.

İnsansız hava araçları kullanımı son yıllarda artmaktadır ve buna bağlı olarak bu araçlarla ilgili yapılan çalışmalar da artış göstermektedir. İnsansız hava araçları genellikle üzerlerinde kamera taşırlar ve bu veriyi yalnızca yer istasyonuna gönderirler. Bu tez çalışmasında insansız hava araçlarının taşıdıkları kameralar yardımı ile kontrollerinin sağlanması amaçlanmıştır. Ayrıca, ileride bu tip çalışmalar yapılabilmesi için örnek bir platform oluşturulmuştur.

Bu tez çalışmasında değerli görüşleri ile beni yönlendiren ve emeğini esirgemeyen tez danışmanım Sayın Yrd. Doç. Dr. Erdiç Altuğ'a ve bu projenin gerçekleştirilmesinde maddi destek sağlayan İTÜ BAP birimine teşekkürü bir borç bilirim. Ayrıca tez boyunca her konuda bilgisini ve emeğini esirgemeyen değerli arkadaşım Ar. Gör. Gökçe Burak Tağlıoğlu'na, hiçbir zaman çekinmeden soru sorabildiğim hocam Ar. Gör. Mesut Acar'a, gömülü sistemler konusunda yaptığı yardımlardan dolayı A. Çağatay Sarı'ya, laboratuvar arkadaşım Bilal Taş'a ve imalatla yardımcı olan Teknisyen Ahmet Savaş'a çok teşekkür ederim. Proje boyunca hep yanımda olan aileme desteklerinden dolayı minnettarım.

Mayıs 2012

Cihat Bora YİĞİT
(Makina Mühendisi)

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	iii
1. GİRİŞ	1
1.1 Tezin Amacı	3
2. QUADROTOR TASARIMI.....	5
2.1 Mekanik Tasarım.....	6
2.1.1 Gövde	6
2.1.2 Motor ve pervaneler	8
2.2 Elektronik Tasarım	10
2.2.1 Otopilot	11
2.2.2 IMU	11
2.2.3 Gömülü sistem kartı	12
2.2.4 ESC (Elektronik hız kontrol ünitesi).....	12
2.2.5 Piller	13
2.2.6 Kamera	13
2.2.7 UBEC (Voltaj regülatörü).....	14
2.2.8 XBee (İletişim arayüzü)	14
2.2.9 Kumanda	15
3. QUADROTOR KONTROLÜ	17
3.1 Quadrotor Modeli	17
3.2 DCM Algoritması.....	24
3.3 PID Kontrol	28
3.4 Benzetimler	29
3.5 Gerçekleme.....	33
4. EŞ ZAMANLI KONUMLANDIRMA VE HARİTALAMA	35
4.1 Giriş	35
4.2 Kalman Filtresi ve Genişletilmiş Kalman Filtresi.....	35
4.3 SLAM.....	37
4.4 V-SLAM.....	37
4.4.1 Olasılık haritası	38
4.4.2 Hareket modeli	41
4.4.3 Kamera modeli	43
4.4.4 Başlangıç	44
4.4.5 Tahmin etme	48
4.4.6 Ölçüm alma	51
4.4.7 Durum vektörünü güncelleme.....	52
4.4.8 Uzaklık tahmini.....	52

4.4.9 Diğer çalışmalar ile farklar.....	52
5. DENGELEME DENEYİ	55
5.1 Giriş	55
5.2 Deney Düzenegi	56
5.3 Sonuçlar.....	57
6. SONUÇLAR VE ÖNERİLER.....	61
6.1 Öneriler.....	62
EKLER.....	69
ÖZGEÇMİŞ.....	79

KISALTMALAR

İHA	: İnsansız Hava Aracı
GPS	: Küresel Konumlandırma Sistemi
IMU	: Ataletsel Ölçüm Ünitesi
SLAM	: Eşzamanlı Konumlandırma ve Haritalama
V-SLAM	: Görsel Eşzamanlı Konumlandırma ve Haritalama
KF	: Kalman Süzgeci
EKF	: Genişletilmiş Kalman Süzgeci
ESC	: Elektronik Hız Kontrolü
PWM	: Band genişliği modülasyonu
DCM	: Direction Cosine Matrix
PID	: Oransal, Türevsel ve İntegral Kontrol Yöntemi
EKF	: Genişletilmiş Kalman Süzgeci
SFM	: Hareketten Yapı Çıkarsama (Structure From Motion)

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Hava aracında kullanılan parçaların ağırlıkları	5
Çizelge 2.2 : Motor karşılaştırması.....	10
Çizelge 3.1 : Motor detaylı ölçümü.	21
Çizelge 3.2 : Quadrotror modelinin sayısal değerleri	23
Çizelge 3.3 : 0.1 aşma ve 0.75 s. oturma süresi için PID katsayıları.....	30
Çizelge 3.4 : Aşmasız ve 1.5 saniye oturma süresi için PID katsayıları	32
Çizelge A.1 : Beagleboard XM kurulum konfigürasyonu.....	70
Çizelge A.2 : Beagleboard XM teknik özellikler	71
Çizelge A.3 : Ardupilot teknik özellikler	71
Çizelge A.4 : ArduIMU teknik özellikler	72
Çizelge A.5 : Kamera teknik özellikler	73

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Solidworks programında hava aracı tasarımı.	6
Şekil 2.2 : Quadrotor gövdesi.	7
Şekil 2.3 : Motor ve pervane test düzeneği.	8
Şekil 2.4 : Motor hızı ayarlayan devre.....	9
Şekil 2.5 : Hava aracı iletişim diyagramı.	11
Şekil 2.6 : Quadrotor üzerinde parçaların yerleşimi.....	16
Şekil 3.1 : Eksen takımları ve motor dönüş yönleri.	18
Şekil 3.2 : PWM-Hız doğru uydurma sonuçları.	21
Şekil 3.3 : Hız ile kaldırma kuvveti doğru uydurma sonuçları.....	22
Şekil 3.4 : DCM blok diyagramı [23]'ten uyarlanmış.....	27
Şekil 3.5 : Quadrotor kontrol blok diyagramı.....	28
Şekil 3.6 : Simulasyon ana pencere	29
Şekil 3.7 : Açısız pozisyonların PID birim basamak cevapları	30
Şekil 3.8 : Doğrusal olmayan modele göre birim basamak cevapları	31
Şekil 3.9 : Yunuslama açısı simulasyon ve deney sonuçları	32
Şekil 3.10 : Yalpalama açısı simulasyon ve deney sonuçları	32
Şekil 3.11 : Uçuş deneyleri.....	33
Şekil 4.1 : Görüntü işleme algoritması akış şeması.....	38
Şekil 4.2 : V-SLAM referans resimler için arama alanları.....	40
Şekil 5.1 : Tek eksenli deney düzeneği ve hava aracı	56
Şekil 5.2 : Tek eksenli deney düzeneği üzerinde yunuslama hareket.....	56
Şekil 5.3 : Dengeleme deney düzeneği.....	57
Şekil 5.4 : Yunuslama ve yalpalama açılarına ait zaman-açı grafiği.....	58
Şekil 5.5 : Yönelme açısına ait zaman-açı grafiği	59
Şekil 5.6 : Dengeleme deneyine ait deney sonuçları	60
Şekil B.1 : Ardupilot blok diyagramı.	74
Şekil D.1 : Pervane kuvvet ölçüm deneyinde kullanılan devre şeması.	78

MİKRO HAVA ARAÇLARININ BİLİNMEYEN ORTAMLARDA GÖRÜNTÜ TEMELLİ KONTROLÜ

ÖZET

İnsansız hava araçları günümüzde büyük önem kazanmıştır. Bu araçların askeri ve sivil pek çok uygulama alanı vardır ve uygulama alanları gün geçtikçe artmaktadır. Bu araçlar genellikle üzerlerinde bulunan kamera ile yer kontrol istasyonuna görüntü gönderirler. Son yıllarda, bu kameraları kullanarak görüntü işleme metotları ile araçların otonomluğunu geliştirmeyi amaçlayan çalışmalar yapılmaktadır. Bu çalışmada öncelikle deney platformu olarak açık kaynak kodlu bileşenler içeren bir hava aracı geliştirilmesi amaçlanmış, daha sonra bu platformu kullanarak görüntü işleme teknikleri ile bu aracın kontrolünün sağlanması amaçlanmıştır.

İHA dört rotorlu bir helikopter (quadrotor) olarak oluşturulmuştur. Mekanik ve elektronik parçalar istenilenleri sağlayacak şekilde seçilmiş ve uygun biçimde birleştirilmiştir. Parçaların seçimi için bazı deney düzenekleri oluşturulmuş, deneyler yapılmış ve sonuçlara göre parça seçimi yapılmıştır. Tüm parçaların seçiminde hafifliğe özen gösterilirken, mekanik parçaların dayanımı elektronik parçaların ise kullanıcı müdahalesine açıklığı ön planda tutulmuştur.

Hava aracının kontrolünün sağlanabilmesi için matematiksel modeli oluşturulmuş ve görüntü kullanılmaksızın ataletsel ölçüm birimi yardımı ile kontrolünün sağlanabilmesine çalışılmıştır. Oluşturulan model simulasyon ortamına aktarılmış ve kontrolcü katsayıları denenerek katsayılar için ön değerlendirme yapılmıştır. Daha sonra bu katsayılar hava aracı üzerinde denemiş ve uygun katsayılar bulunarak hava aracının ataletsel ölçüm birimi yardımı ile kontrolü sağlanmıştır.

Görüntü işleme için kullanılan yöntem V-SLAM yaklaşımı olmuştur. Hava aracının daha önceden düzenlenmemiş veya bilinmeyen bir ortamda çalışmasının amaçlanması öncelikle haritalama ve bu haritaya göre konumlandırma ihtiyacı doğurmuştur. Hava aracı üzerinde kullanılmaya uygun, gerçek zamanlı çalışabilen bir algoritma incelenmiş ve önce bilgisayar ortamında, daha sonra hava aracı üzerinde taşınabilmesi amacı ile gömülü bir bilgisayarda gerçekleştirilmiştir. Hava aracının hızlı hareketlerine karşılık anlamlı görüntüler alınabilmesi için yüksek hızlı kamera kullanılmıştır.

Görüntü işleme sistemi, hava aracı kontrolörü ve yer bilgisayarı arasında iletişim kurularak deney platformu hazırlanmıştır. Bu platform, güvenli olması amacıyla, hava aracının üç ekseninde hareketine izin veren ve düşmesini engelleyen bir deney düzeneği üzerinde denenmiştir. Görüntü işleme algoritmasının verdiği sonuçların ataletsel ölçüm ünitesine benzer olduğu görülmüştür. Ayrıca yönelme açısı görüntüden alınan geribesleme sayesinde kullanıcı tarafından kumandadan gönderilen referans açısal konuma oturması sağlanmıştır. İleriki çalışmalarda bu referansın da görüntü işleme algoritması tarafından üretilmesine (navigasyon) çalışılacaktır.

VISION BASED CONTROL OF MICRO AIR VEHICLES IN UNKNOWN ENVIRONMENTS

SUMMARY

Unmanned aerial vehicles (UAV) has become increasingly important nowadays. They have been widely used in civil applications such as aerial photography and military applications such as reconnaissance and combat. These vehicles can be classified as fixed wing and vertical take-off and landing (VTOL). VTOL vehicles can be used in small workspaces due to their maneuver capabilities. However successful control of these vehicles requires precise information about the vehicle attitude. Camera is usually carried on boards of the systems, as a requirement of their applications. Usually these cameras are used to observe something from aerial vehicle side of view and connected to ground control system with a wireless communication system. Computer vision systems using the onboard cameras can be an alternative to inertial measurements system which are mostly used on aerial vehicles to measure attitude. First aim of this project is to develop an open source air vehicle platform. Second one is the stabilisation of the vehicle by using a vision algorithm.

Quadrotor is a four-rotor helicopter and can be classified as a VTOL vehicle. A quadrotor consists of some mechanical and electronic parts. Mechanical parts are frame, motors, rotors and other connection parts. Motors and rotors are chosen after some experiments. To find the correct motor for the system a thrust test rig is created. Two different brushless motors and different rotors are run on the rig. Best performed motor-rotor couple is chosen. Electronics consists of usually an autopilot and inertial measurement unit (IMU), electronics speed controllers (ESC), receiver, battery and vision system. Autopilot is responsible for the stabilisation of quadrotor vehicle. IMU is attitude measurement unit and IMU used in the project has a magnetometer and altimeter. Altimeter can measure the altitude of the helicopter and magnetometer gives information of yaw angle. ESC is a circuit that can run brushless motors using PWM input, which comes from directly an RC receiver or an autopilot system. Receiver is a bridge between user and quadrotor, it can sense inputs from user by using radio waves and transfers it to Ardupilot. A lithium-ion battery is used in this project due to its low volume/charge ratio. Vision system includes an embedded computer and a high speed camera. Embedded computer is Beagleboard Xm. High speed camera is Firefly Mv and its maximum frame rate is 60 frame per second. All of the components used in this system are given in more details in the project and the methods are explained in order to choose the correct components.

Working principle of a quadrotor is based on the difference between speed of rotors. Helicopter is designed in '+' shape and every corner has a motor-rotor couple. When decelerating, the front motor and accelerating the rear motor gives a difference, which means a change in pitch angle and movement on the x axis. Pitch and roll axes work based on this principle. Yaw axis is changed differentiating motor couple speeds. To balance the yaw angle, opposed motors turns in the same direction. When

front and rear motor is decelerating, comparing right and left, quadrotor rotates around yaw axis. Thus quadrotor control is an easy control problem when compared to other helicopter models. However, to stabilize the quadrotor synchronization of motors are required.

Firstly, to control the system, model of the quadrotor is analysed. Quadrotor is an under-actuated platform, it has four inputs and six outputs. The model is simplified and linearized to calculate a controller for the system. For attitude control, an enhanced PID algorithm is used in autopilot card and controller coefficients are tuned using the simplified model. Both linear and non-linear models are created in simulation environment (Matlab Simulink) and autopilot is modeled as well. Calculated coefficients are simulated and the results give the best coefficients for fine tuning on real system. For the experiments, inertial measurement system which comes with autopilot is used for attitude information. Gyros and accelerometers are combined to give attitude by using Direction Cosine Matrix (DCM) algorithm.

First simulation results for an underdamped system gave successful result both for linear and non-linear model; however, experiments show that controller design has to make system over-damped to stabilise the quadrotor. Secondly, coefficients are calculated in the light of this information. Both simulation and experiments are successful and stable flight is observed after fine tuning of the controller.

ArduPilot is an open-source platform, therefore an existing algorithm 'ArduCopter NG' is used to implement PID control design to the system. Controller coefficients which are calculated, are added to the algorithm. Open-source software of the autopilot gives the chance of changing some parts of it, while using vision system instead of IMU, related parts are changed. Also the communication between Beagleboard, ArduPilot and ground control station are coded using same software.

Control in an unknown environment requires mapping and localization simultaneously. In literature, SLAM (simultaneous localisation and mapping) is well defined problem and most of the examples use sensors such as radar. Alternative method for the SLAM is structure from motion approach, however these algorithms generally are computationally expensive algorithms and usually used for augmented reality applications. Recently, vision has started to be used as a sensor for SLAM. V-SLAM approach is explained in this project and an EKF based V-SLAM algorithm is implemented using a PC. However, aim of the project is to stabilise the quadrotor without using a ground control system help. Thus, an embedded computer is used to run the vision algorithm.

Vision algorithm uses corners as reference points and takes 21 x 21 pixels area around the corner as the parts of the map. Dynamic model is given as constant velocity model and system treats accelerations as disturbance. Quadrotor states are given in a vector size of 13. Every camera point adds system 6 new states. To cope with the blurring image problem under fast velocity situations, a fast camera (60 FPS) is used on the quadrotor. The camera model is defined as standard pinhole camera model, and calibrated using well-known solutions in literature. EKF algorithm makes 'predict-measure-update' cycle in every step so that algorithm can run on beagleboard (embedded computer, Angstrom operating system, 1 GHz Arm processor-512 MB RAM) 10 FPS. Having one camera on board due to limitations of thrust provided by motors, makes system a 'monocular' vision system. Problem of using monocular vision for estimating location of camera, is estimating depth of a

feature location on the image plane. This problem is solved by using parallax in the algorithm. Parallax can be defined as the different movements on camera appearance of two different objects, one of them is closer, other one is far.

Initialization of the vision system is very important in these kind of incremental vision algorithms. For simplification vision algorithm are always started at zero degree orientation because the purpose of the project is not designing a new initialisation system.

Finally, in order to put together quadrotor and vision system, communications between autopilot, embedded computer, RF receiver and ground logging computer is designed. Embedded computer uses camera and vision algorithm to estimate attitude of the quadrotor and sends the results (orientation) to the autopilot. Autopilot controls motors using this estimates and user reference orientation values which is received by RF receiver. Autopilot also sends some useful information for logging to ground logging computer. These data are processed using Matlab to plot the results of control.

For a safe experiment, micro air vehicle is connected to a test rig. Quadrotor can rotate between +20 and -20 degrees for roll and pitch angles. It can rotate freely for yaw angles. First experiment is about verifying vision algorithm by using IMU as ground truth. Experiments show that there is similar results between IMU and vision system; however, there is a one second delay due to the embedded board-autopilot communication, vision processing and camera-embedded board communication. Second experiment is the control of quadrotor using the vision data for attitude estimation. These experiments shows that, quadrotor can stabilise itself near hover position, and can track reference signal on yaw axis.

At last, the performance of the vision system can be refined using optimisation techniques. Delay between IMU and can be minimized using a better communication system. Response of the vision system can be accelerated using EKF and estimations without measurements making application a multi-thread one. Dynamic model can be introduced into the EKF instead of using constant velocity model. Also usage of this vision system gives opportunity to navigate in an unknown environment. For this purpose, IMU and other sensors on the quadrotor can be fused with vision system to get better results.

1. GİRİŞ

İnsansız hava aracı (İHA); insan taşıma amacı olmayan, çeşitli askeri ve sivil uygulamalarda kullanılabilen ve uçarak hareket etme kabiliyetine sahip araçlara verilen isimdir. Bu araçlar yerden kumanda edilebildikleri gibi önceden belirlenen bir yörünge üzerinde de otonom bir şekilde hareket edebilirler.

İHA'lar günümüzde askeri ve sivil amaçlarla pek çok alanda kullanılmaktadır. Askeri kullanım alanlarından bazıları gözlem yapma, hedef belirleme, kara mayınlarının yerini belirleme ve yok etme, radar sistemi bozma ve düşman aktivitesini belirlemedir [1]. Aynı kaynakta verilen sivil aktivitelerden bazıları ise hava fotoğrafçılığı; tarımsal alan, yangın, enerji hattı, çevre kirliliği, afet alanı ve trafik gözlemleme olarak verilebilir. Buna ek olarak İHA kullanımı gün geçtikçe artmaktadır ve bu kullanım alanları da her geçen gün genişlemektedir. İHA'ların günümüzde en önemli kullanım alanının ve gelişiminde en büyük yatırım sahibi askeriye olmasına karşılık önümüzdeki yüzyıl içerisinde sivil hayatta da pek çok alanda kullanımının artacağı belirtilmiştir [2].

Klasik hava araçları yerine İHA kullanılması için İHA'ların klasik araçlara göre avantajlarının olması gerekir. Askeri açıdan bakıldığında, bu araçlar genellikle insan taşıyan araçlara göre daha küçük boyutlu oldukları için düşman tarafından fark edilmeleri ve etkisiz hale getirilmeleri daha zordur. Aynı zamanda ekonomik açıdan da insanlı araçlarla karşılaştırıldıklarında daha az maliyetlidirler. Sivil açıdan bakıldığında da insanlar tarafından yapılması zor veya tehlikeli olan işlerde bu araçlar kullanılmaktadır.

İHA'lar temel olarak iki sınıfa ayrılabilirler: Sabit kanatlı ve dikey havalanan ve konabilen (VTOL). Sivil uygulamaların genellikle düşük irtifa uçuşları, havada sabit kalabilme, dikey havalanma, dikey konabilme ve küçük boyutlu olma gibi özellikler gerektirdiğini belirtilmektedir [3]. Bu çalışmada öncelikli amaç kapalı bir alanda kararlı bir uçuş yapılması olduğu için üzerinde çalışılacak hava aracı olarak bir VTOL belirlenmiştir. VTOL araçları arasında quadrotorlar üzerinde özellikle yüksek

ivmeli hareket kabiliyetleri açısından son yıllarda pek çok üniversitede araştırma yapılmaktadır [3].

İHA'lar genellikle üzerinde gözlem amacı ile yer merkezine görüntü aktaran veya seyir halindeyken kayıt yapıp sonra görüntülerin izlenmesine imkân veren kameralar taşırlar. Yapılan çalışmalar bu kameraların görüntü izleme algoritmaları kullanılarak bir sensör olarak kullanılmasını sağlamak yönündedir. Görüntü işleme quadrotorlar üzerinde farklı amaçlar için kullanılmaktadır. Bunlar görsel servolama, görsel navigasyon, gökyüzünde izleme ve havada lokalizasyon ve harita çıkarmadır [4-7]. Bu sistemler hava araçlarının otonom iniş kalkış yapması, otonom şarj olması veya yakıt alması, obje takibi ve hava araçlarının birlikte çalışabilmesi veya otonom navigasyon amaçlı kullanılmaktadır.

Quadrotor navigasyonu için literatürde özellikle GPS kullanımı öne çıkmaktadır. Ancak GPS dış etkilere kolayca bozulabilir veya kapalı ortamlarda çalışmaz. Ayrıca askeri uygulamalarda güvenilir de değildir.

Özellikle üzerinde çalıştığımız konu görüntü işleme ile konumlandırmadır. Görsel konumlandırmada temel olarak iki farklı yaklaşım söz konusudur: Hareketten yapı çıkarsama (structure from motion) metodu ve Kalman süzgeci tabanlı SLAM [8],[4], [9]. Hareketten yapı çıkarsama kullanarak navigasyon yapabilen bir insansız hava aracı örneği literatürde mevcuttur [5]. Görüntü işleme kullanarak navigasyon yapabilen başka örnekler de ortaya konulmuştur [10], [11].

Hava aracının oryantasyonunun doğru tahmin edilebilmesi kararlı bir uçuşun sağlanabilmesi açısından çok önemlidir. Bu konuda bazı araştırmacılar katadioptrik görüntü sistemleri üzerinde çalışmıştır. Bu teknolojinin temel avantajı geniş bir görüş alanı sağlamaktır. Yunuslama ve yalpalama açılarının tahminine yönelik çalışmalar mevcuttur [12], [13]. Bu çalışmalarda Rastgele Markov Alanları veya RGB temelli Mahalanobis uzaklığı kullanılarak ufuk çizgisi bulunulmuştur. Ancak bu çalışmalar ufuk çizgisinin her zaman gözüktüğü durumlar için düşünülmüştür ve bu hep mümkün değildir. Ayrıca yönelme açısı da tahmin edilememektedir. Homografi ve ufuk çizgisinin hibrid bir şekilde kullanıldığı bir çalışma yapılmıştır [14]. Ufuk çizgisinin tespitinde kullanılan yöntemlerin sınırlamalarından kurtulmak için kentsel alanlardaki görüntüde genellikle bulunan çizgileri kullanan çalışma yapılmıştır [15]. Bu çalışmada yönelmeyi her zaman tespit etmek mümkün olmamakla beraber, gökyüzünün tespiti gerekmektedir. Bu nedenle yoğun şehir ortamında ve kapalı

alanlarda bu alıřmaları kullanmak mmkn olmamaktadır. Daha yeni bir alıřma kaıř noktaları ve sonsuz homografi kullanmakta ve helikopterin oryantasyonunu kentsel alanlarda tahmin edebilmektedir [16]. Bu alıřmaya ait daha yeni bir gerekleme alıřması mevcuttur [17]. Bu alıřmada tm ynl kamera kullanılarak quadrotor oryantasyonu tespit edilmiř, Kalman filtresi yardımıyla geliřtirilmiř ve bařarılı bir řekilde helikopter dengelemesini saėlamıřtır. Ancak bu alıřma da hem aık alanlar iin tasarlanmıř hem de ortamda bulunması gereken bir takım izgilere ihtiya duymaktadır.

Bu tezde birinci blmde bir giriř yapılmıř ve insansız hava araları, quadrotor, grnt iřleme ve navigasyon konuları hakkında literatrdeki rneklerle yer verilmiřtir.

İkinci blmde yapılan quadrotor tasarımı, quadrotor zerinde mekanik ve elektronik olarak kullanılan malzemeler incelenmiř ve quadrotorun istenilen amaca uygun nasıl oluřturulduėu gsterilmiřtir.

nc blmde quadrotor dinamiėinin matematik modellemesi, ataletsel lm nitesinin alıřmasında kullanılan algoritma ve kullanılan kontrolrn yapısından bahsedilmiřtir. Bu blmdeki iřlemler quadrotor zerinde bulunan dřk seviye iřlemci zerinde gereklenen iřlemlerdir.

Drdnc blmde grnt iřleme aıklanmıř ve kullanılan geniřletilmiř Kalman filtresi tabanlı V-SLAM (grsel konumlandırma ve haritalama) algoritması ve bu algoritmanın nasıl gereklendiėi anlatılmıřtır. Bu blmde anlatılan algoritmalar quadrotor zerinde bulunan yksek seviye iřlemci tarafından gerekleřtirilen iřlemlerdir.

Beřinci blmde quadrotorun  eksenli hareketine izin veren bir deney dzeneėi zerinde sadece grnt iřleme algoritmasını kullanarak dengelemesini saėlamak iin yapılan deney ve sonuları anlatılmaktadır.

Altıncı blm sonuları ve gelecekteki alıřmalar iin nerileri ierir.

1.1 Tezin Amacı

Bu alıřmada ncelikle aık kaynak kodlu yazılım kullanan ve yrttė iřlemlerin hemen her noktasında mdahale edilebilme imknını sunan bir elektronik donanıma sahip ve ekonomik bir insansız hava aracı oluřturulması hedeflenilmiřtir. Bununla

beraber gelişen görüntü işleme algoritmaları ile bir hava aracının kararlı uçuşunun bu algoritmalar ile sağlanabilmesi amaçlanılmıştır.

2. QUADROTOR TASARIMI

İnsansız hava aracının görüntü işleme programı ile birlikte çalışabilmesi için üzerine gömülü sistem bilgisayar ve kamera takılmaktadır. Bu ek ağırlıklar nedeniyle hazır alınan bir quadrotor tasarlanıldığı şekilde uçamamaktadır. Ayrıca görüntü işleme programı ile İHA'nın birlikte çalıştırılması gerekmektedir. Hazır quadrotorlar genellikle kapalı kaynak kodlu otopilot barındırdıkları için kontrol algoritmalarına müdahale etme şansı verilmemektedir. Bu nedenlerden ötürü görsel navigasyonu sağlaması amacı ile bir quadrotor tasarımı yapılmış ve uygun parçalar seçilerek quadrotorun montajı gerçekleştirilmiştir.

Parçaların tamamının seçiminde ağırlık çok önemlidir. Kullanılan tüm parçaların ağırlıkları Çizelge 2.1'de verilmiştir. Bazı parçaların ağırlıkları monte edildikleri bağlantı elemanları ile birliktedir.

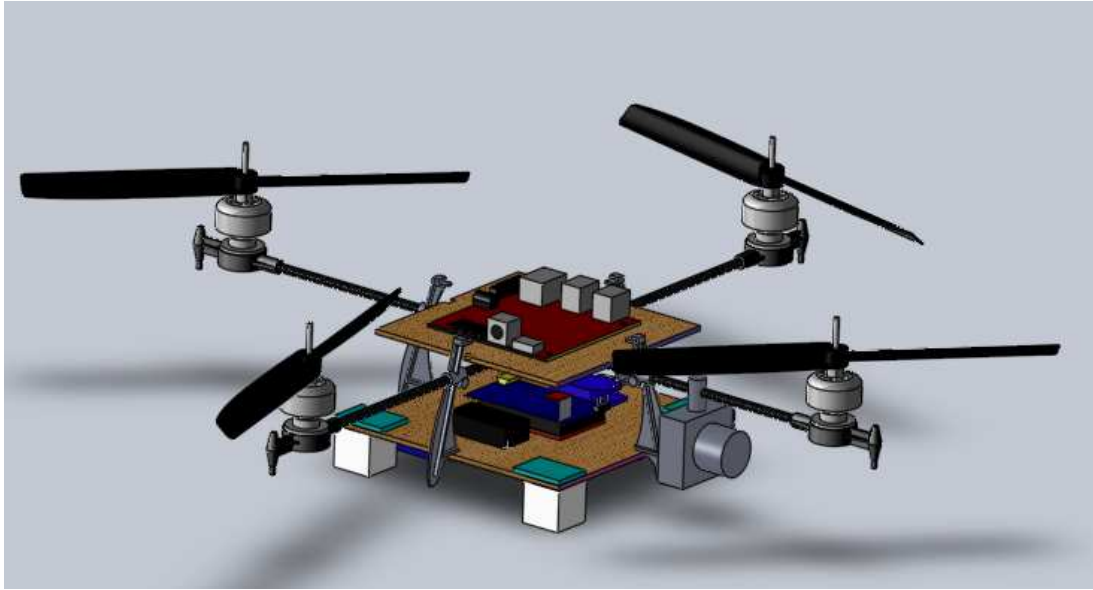
Çizelge 2.1 : Hava aracında kullanılan parçaların ağırlıkları.

Parça adı	Adet	Ağırlık [g]	Toplam [g]
Gövde	1	110	110
Motorlar	4	70	280
Pervaneler	4	10	40
Ardupilot ve IMU	1	60	60
BeagleBoard (Gömülü Sistem Kartı)	1	150	150
Kamera ve Lens	1	50	50
XBee	1	11	11
Alıcı	1	14	14
ESC	4	30	120
Pil	1	125	125
Bağlantı parçaları, kablolar ve bağlayıcı elemanlar	1	85	85
Genel Toplam			1045

2.1 Mekanik Tasarım

Bir quadrotor artı şeklinde birbirine birleştirilmiş bir düzeneğin her bir ucuna birer motor ve bu motorların uçlarına pervane bağlamak suretiyle oluşturulmuş hava aracıdır. Motorların iki tanesi saat yönünde dönerken diğer iki tanesi saatin tersi yönde dönecek şekilde çalıştırılır. Bunun nedeni quadrotor üzerinde oluşan torkun ortadan kaldırılmasıdır. Eğer tüm motorlar aynı yönde çalıştırılırsa, quadrotor üzerinde bir yönelme hareketi oluşur. Pervaneler motorların ters yönde dönmelerine bağlı olarak ters açılı olarak seçilmişlerdir. Yani pervanelerin yapısı gereği ters yönde dönmelerine rağmen tüm pervaneler yukarı yönlü bir kuvvet oluştururlar.

Kullanılacak olan quadrotorun öncelikle çalıştırılmak istenildiği ortam kapalı alanlardır. Bu nedenle quadrotorun oda içerisinde yükselebileceği mesafelerden düşmesi halinde sağlam kalabilmesi amaçlanmıştır. Quadrotor'un Solidworks ortamında çizilmiş modeli Şekil 2.1'de görülebilir.



Şekil 2.1 : Solidworks programında hava aracı tasarımı.

2.1.1 Gövde

Quadrotor üzerinde bulunan elektronik parçaları ve motorları taşıyan ana parçaya gövde denilebilir. Quadrotor gövdeleleri genellikle hafif olması açısından alüminyum, karbon fiber veya ahşap gibi malzemelerden üretilir. Quadrotor hem akademik çalışmalarda hem de hobi uğraşları açısından çok kullanılan bir model olduğu için tüm malzemelerini hazır olarak bulmak mümkündür. Bu çalışmada quadrotorun gövdesi hazır olarak satın alınmıştır. Draganflyer III marka quadrotorun

yedek parçası olarak satılan gövde karbon fiber malzemeden yapılmış olması, hafif ve dayanıklı olması nedeniyle seçilmiştir (Şekil 2.2). Ayrıca gövde üzerindeki karbon fiber borular dış çapı aynı olmak şartıyla değişik boy ve et kalınlıkları ile değiştirilebilmektedir. Motorları merkeze bağlayan bu borular civatalar ile sıkıştırılarak bağlandığından motorların veya pervanelerin bir yere çarparak tehlike görmesi anında yön değiştirerek pervanelerin kırılmasını önlemektedir. Ancak yapılan deneyler göstermiştir ki, bu hareket nedeniyle motorlar birbirlerine paralel bağlanamamaktadır. Bu nedenle motorları tutan parça ile karbon borular üzerine yeni bir civata takılarak motorların karbon boru etrafında dönmeleri engellenmiştir.



Şekil 2.2 : Quadrotor gövdesi.

Hazır alınan gövde üzerinde kullanılacak elektronik kart ve motorlara göre bazı değişiklikler yapılmıştır. Gövdenin ortasında kalan alan IMU, otopilot gömülü bilgisayar ve pilin taşınacağı alandır. Bu alan hazır gelen gövde üzerinde ihtiyaç duyulan alandan küçüktür. Bu nedenle burada kullanılmak üzere iki adet parça hazırlanmıştır. Bu parçalar 130 x 130 ve 160 x 160 mm boyutlarında 3 mm kalınlığında üretilmiştir. Parçalar üzerine otopilot ve gömülü bilgisayar takılacak şekilde uygun noktalardan delinerek kullanılmıştır.

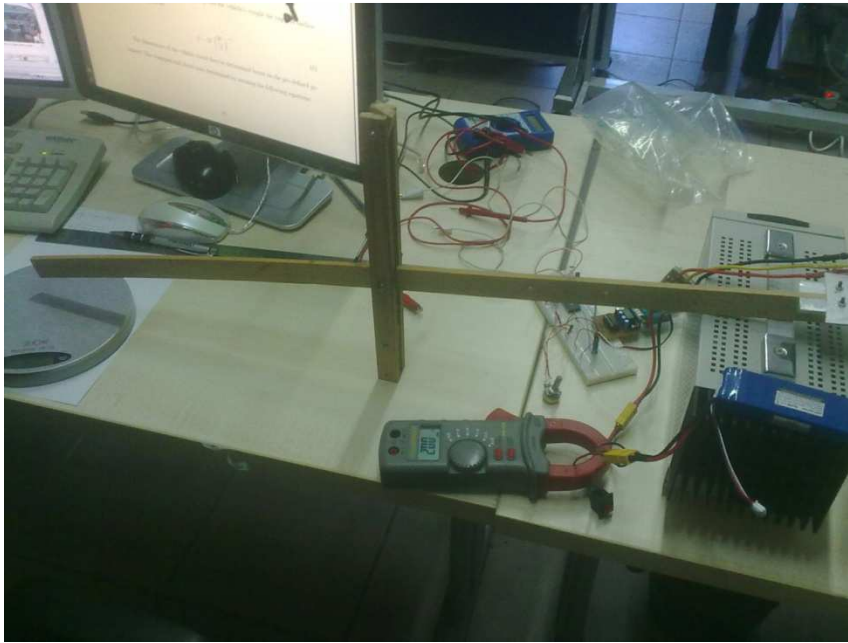
Gövde tasarımında en önemli noktalardan birisi ağırlık merkezinin gövde merkezi ile arasındaki dikey uzaklıktır. Eğer ağırlık merkezi gövde birleşim noktasının üzerinde kalırsa, kontrol açısından problem bir ters sarkaç problemine dönüşür. Buna karşılık ağırlık merkezinin aşağıda kalması durumunda problem bir sarkaç problemi olur ve

kontrolü daha kolay bir sistem elde edilir. Bu nedenle sistem tasarımında ağırlık merkezinin aşağıda kalmasına dikkat edilmiştir.

2.1.2 Motor ve pervaneler

Literatürde kullanılan quadrotorlar üzerinde genellikle iki çeşit motor kullanılmaktadır. Bunlardan birincisi fırçalı motor ile birlikte dişli kullanımıdır. İkinci kullanım şekli ise bir fırçasız motorun doğrudan pervaneye bağlandığı şekildir. Bu çalışmada ikinci şekil uzun ömürlü olması ve montaj kolaylığı gibi nedenlerden dolayı seçilmiştir.

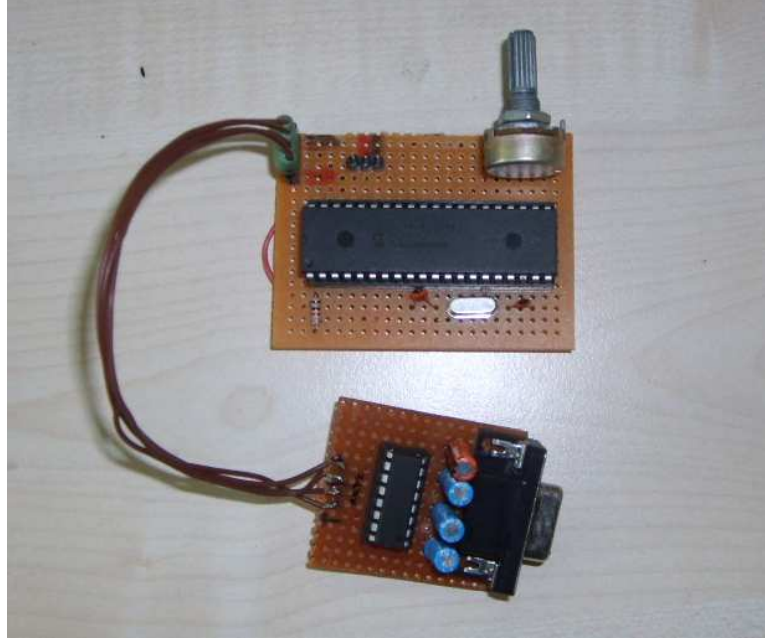
Quadrotor tasarımında motor ve pervaneler kesinlikle birbirinden ayrı düşünülerek seçilmemelidir. Bunun nedeni her motorun belirli boyuttaki pervaneler ile verimli çalıştığı gerçeğidir. Bu nedenle iki farklı motor ve beş farklı pervane alınarak bir matris oluşturulmuştur. Motorları denemek için bir deney düzeneği ve bir elektronik devre tasarlanmış ve imal edilmiştir. Bu deney düzeneği Şekil 2.3'te görülebilir.



Şekil 2.3 : Motor ve pervane test düzeneği.

Resimde görülen düzenek bir sabit kol (dikey olan) ve bir hareketli kolun (yatay) civatalar yardımı ile birleştirilmesi ile elde edilmiştir. Bir terazi gibi çalışmaktadır. Hareketli kolun sağ tarafına farklı motorların bağlanabileceği alüminyum saclardan elde edilen bir düzenek bağlanmıştır. Bu düzenek üzerine açılan delikler kullanılan 2 farklı motorun da bağlanabilmesini sağlamaktadır.

Şekil 2.4'te görülen devre sayesinde motor hızı kullanıcı tarafından kontrol edilebilmektedir. Ayrıca devre üzerinde bulundurduğu seri port bağlantısı sayesinde motora aktardığı PWM değerini bilgisayara da iletilir. Bu şekilde kullanıcı motorun girdisi hakkında bilgi sahibi olur.



Şekil 2.4 : Motor hızı ayarlayan devre.

Motor hız ayarlama devresi bir PIC16F877, potansiyometre, max 234 entegresinden ve bazı devre elemanlarından oluşur. Devreye ait şema ekte mevcuttur. Potansiyometreden alınan analog sinyal mikroişlemci içerisinde dijitalle çevrilir (10 bit, 0-1024 aralığı). Motor sürücü devresi olan ESC'ler ise hobi uçak kumandaları ve servolarına uygun şekilde 1000-2000 PWM aralığında çalışırlar. Bu nedenle potansiyometreden alınan değer doğrudan 1000 değeri ile toplanarak çıkış değeri elde edilir. Çıkış hem ESC'ye sinyal gönderir hem de bu veriyi seri port aracılığı ile bilgisayara gönderir.

Motorun çektiği akım değeri de veriminin bir göstergesi olduğu için kaydedilmektedir. Üç farklı kuvvet değeri etrafında motorların pervane çeşitlerine bağlı performansları Çizelge 2.2'de verilmiştir.

Çizelge 2.2 : Motor karşılaştırması.

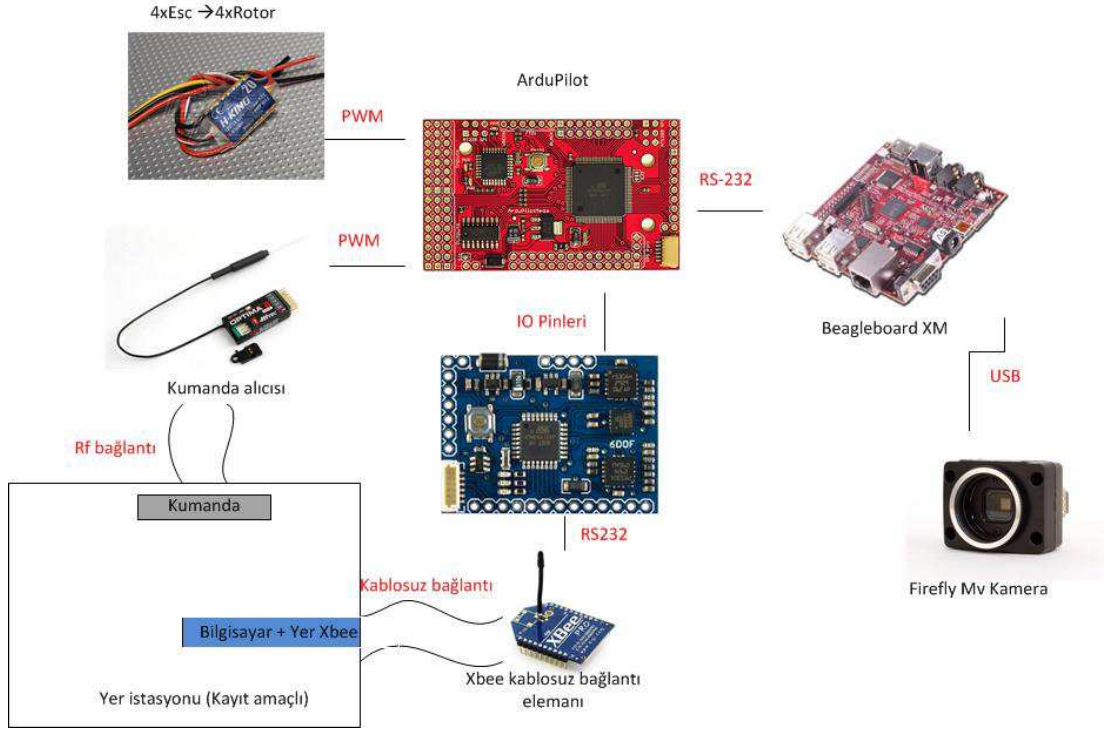
TOWERPRO 2410-08T	10x6 / 2S	10x6 / 3S	8x4 / 2S	8x4 / 3S	9x4.7 / 2S	9x4.7 / 3S	10x5 / 2S	10x5 / 3S
150gr / Akım	1.5	1.8	x	2.1	2.2	1.7	2.8	2.1
150gr / PWM	1675	1303	x	1426	1478	1316	1485	1339
200gr / Akım	3.3	2.4	x	2.7	3	2.3	3.5	2.6
200gr / PWM	1773	1342	x	1503	1532	1370	1535	1380
500gr / Akım	x	x	x	x	x	7	5.3	7.5
500gr / PWM	x	x	x	x	x	1600	285	1588
Maks / Akım	x	x	x	4.5	4.5	x	x	x
Maks / PWM	x	x	x	2000	2000	x	x	x
Maks	x	x	30	335	485	x	x	x

TURNIGY	10x6 / 2S	10x6 / 3S	8x4 / 2S	8x4 / 3S	9x4.7 / 2S	9x4.7 / 3S	10x5 / 2S	10x5 / 3S
150gr / Akım	X	1.6	x	1.8	X	1.7	X	1.9
150gr / PWM	X	1303	x	1360	X	1290	X	1460
200gr / Akım	X	2.6	x	2.8	X	2.6	X	2.8
200gr / PWM	X	1342	x	1586	X	1380	X	1586
500gr / Akım	x	x	x	x	x	x	X	X
500gr / PWM	x	x	x	x	x	x	X	X
Maks / Akım	x	9.5	x	4.6	X	8.2	x	8.9
Maks / PWM	x	2000	x	2000	X	2000	x	2000
Maks	x	520	X	300	X	470	x	485

Yukarıdaki tabloya göre towerpro brushless outrunner 2410-08T 890 kv motor ve 10x6 pervane seçilmiştir.

2.2 Elektronik Tasarım

Quadrotor üzerinde uçuş kontrolünü sağlayabilmek amacı ile otopilot ve IMU kullanılmıştır. Görsel navigasyonun sağlanması için de gömülü sistem bir kart kullanılmıştır. Elektronik parçaların aralarındaki iletişim Şekil 2.5'te açıklanmıştır.



Şekil 2.5 : Hava aracı iletişim diyagramı.

2.2.1 Otopilot

Quadrotorun motor hızlarını değiştirerek açısal ve düzlemsel konumunu istenilen değerlere ulaştırabilmesi amacıyla hazır bir otopilot kartı kullanılmıştır. Kart üzerinde kullanılan algoritmaya müdahale edebilmek amacı ile açık kaynak kodlu Ardupilot kartı quadrotor üzerinde kullanılmıştır. Üzerinde Atmel marka mikroişlemcisi bulunan bu kart programlanmamış şekilde satılmaktadır ve internet üzerinde kart kullanımıyla quadrotor sürülmesi ile ilgili açık kaynak kodları mevcuttur. Bu kodlardan bazıları denenmiş ve en kararlı çalışan ArducopterNG isimli kod otopilot üzerinde kullanılmıştır. Bu kodun çalışması ile ilgili açıklamalar ve yapılan değişiklikler quadrotor kontrolü bölümünde açıklanacaktır.

2.2.2 IMU

İHA'larda eğim ölçmek amacı ile genellikle iki farklı yaklaşım kullanılmaktadır. Birincisi ve sıklıkla kullanılanı, ivmeölçer ve jiroskop kullanarak eğim ölçen ataletsel ölçüm üniteleridir (IMU). IMU'ya alternatif ise termofil (thermophile) sensörleridir. Termofil sensörler sıcaklık değişikliklerinden etkilendikleri için bu çalışmada IMU kullanılması uygun görülmüştür.

Çalışmada kullanılan IMU ardupilot ile uyumlu ArduIMU'dur. 6 serbestlik derecesine (3 eksen jiroskop ve 3 eksen ivmeölçer) sahiptir. Otopilot ve IMU özellikleri ekte mevcuttur.

2.2.3 Gömülü sistem kartı

Yapılan çalışmada görüntü işleme yapabilme ve otopilot ile haberleşerek istenilen pozisyon değerine ulaşılabilmesi için otopilota komutlar gönderen gelişmiş bir işlemciye ihtiyaç duyulmuştur. Bu nedenle hem hafif hem de düşük boyutlu olmaları nedeni ile gömülü sistem kartları incelenmiştir.

Bu çalışmada BeagleBoard XM marka bir gömülü sistem kartı kullanılmıştır. Bu kartlar bir bilgisayar gibi işletim sistemi üzerinde çalışmanın yanı sıra 83 x 83 mm boyutlarındadır.

Gömülü sistem bilgisayarı üzerine OpenCV kütüphanesini destekleyen ve sadece belirli ihtiyaçları karşılayacak yönde düzenlenmiş bir işletim sistemi kurulmuştur. Bu işletim sistemi internet sitesinden Ek A1'de verilen konfigürasyon ayarlarına göre indirilmiştir [18].

2.2.4 ESC (Elektronik hız kontrol ünitesi)

Fırçasız motorlar 3 fazlı motorlardır ve doğrudan bir pil bağlantısı ile sürülemezler. Bu nedenle pilden gelen doğru akımı, 3 fazlı alternatif akıma çevirmek gereklidir. ESC (elektronik hız kontrol ünitesi) adı verilen devreler pilden aldıkları doğru akımla, 3 fazlı bir alternatif akım yaratırlar ve PWM sinyallerini kullanarak, motoru besleyen alternatif akımın frekansını değiştirirler. Fırçasız motorun üç çıkışı içerisinde, iki tanesine besleme yaparken üçüncü çıkışı da zıt elektromotor kuvveti (Back EMF) ile geribesleme almak için kullanır.

Bu projede kullanılan motorların yaklaşık olarak çekecekleri akımın 10A'den daha az olması planlanmıştır. Ayrıca pil seçiminde seçilecek pilin en fazla 3 hücreli bir lityum polimer olacağı düşünülmüştür. Akım konusunda emniyetli bir seçim yapabilmek için bir motora en fazla 20 A besleme yapabilen H-king 20A Fixed Wing Brushless Speed Controller kullanılmıştır. Bu model 3 hücreli lityum polimer piller ile çalışabilmektedir.

ESC kullanımında dikkat edilmesi gereken en önemli nokta ESC kalibrasyonudur. ESC ilk kullanımında bir pil ve bir motora bağlanarak kumanda yardımıyla kalibre

edilmelidir. Bu işlemde motorlardan gelen sesler kullanıcıya yol gösterir. Kalibrasyonda kumanda ayarları, motor yönü, pil çeşidi ESC içerisine tanımlanır. Motorların dönüş yönü ESC kalibrasyon menüsüne girmeden, kablo bağlantısında herhangi iki kablonun yerini değiştirerek de tersine çevirilebilir.

2.2.5 Piller

Pil seçiminde dikkat edilecek kriterler sağladığı akım değeri, bu akımı sağlama süresi, sağladığı gerilim değeri ve pilin ağırlığıdır. Lityum polimer piller bu tip uygulamalarda sıklıkla kullanılmaktadır. Bunun nedeni daha hafif olmalarına karşılık hem anlık verebildikleri yüksek akım değerleri, hem de daha fazla şarj kapasitelerinin olmasıdır.

Lityum polimer pil, birden fazla pilin seri bağlanması ile farklı gerilim değerleri verebilir. Pil üzerinde 2S veya 3S olarak görülen değerler pilin gerilim değerini belirtir. Her bir hücre 3.7 V gerilim sağlar. Bu projede kullanılan motorun ihtiyacına bağlı olarak bir adet 3S (11.1V) lityum polimer batarya kullanılmıştır.

Quadrotorun 5 dakika havada kalabilmesi amaçlanmıştır. Quadrotorun havada kalmak için çektiği akım 12 A ve görüntü işleme sisteminin ihtiyaç duyduğu akım ise 1 A olarak ölçülmüştür. Bu durumda ihtiyaç 5 dakika için 1083 mAh olarak hesaplanmıştır. Bu özelliğe uygun 3 hücreli 11.1 volt çıkışa sahip 1500 mAh'lık bir pil sisteme eklenmiştir.

2.2.6 Kamera

Görüntü işleme sistemi bir kamera ve Beagleboard'dan oluşur. Bu çalışmada tek kamera üzerinden alınan renksiz resimler ile hava aracının kontrolü sağlanmaya çalışılmaktadır.

Hava araçları hızlı manevra kabiliyetine sahip oldukları için, insan gözünün, veya insan gözü ile aynı hızda çalışan kameraların net göremeyecekleri görüntüler üzerinde çalışırlar. Doğada uçan canlıların genellikle görme sistemleri bu duruma uyum sağlamıştır ve insan gözünden daha hızlı çalışırlar. İnsan gözü 50 Hz üzerindeki görüntüleri sadece bulanıklık olarak görürken güvercinlerin gözleri 116-146 Hz aralığında çalışır ve net görüntüyü hızla uçarken elde edebilir [19]. Bu nedenle bu çalışmada hızlı bir kameranın kullanılmasına karar verilmiştir.

Hava aracında kullanılan kameranın ağırlığı ve tuttuğu yer de önemli olduğu için mümkün olduğu kadar küçük boyutlarda ve hafif bir kamera araştırılmıştır. Ayrıca kameradan gelen görüntüler Beagleboard üzerine aktarılacağı için ve Beagleboard Linux işletim sistemi ile çalıştığı için bu işletim sistemi altında çalışabilecek bir kamera aranmıştır.

Yukarıda verilen özellikleri sağlayan Firefly-MV bu çalışmada kullanılmıştır. Firefly-MV 60 FPS’de görüntü almaktadır.

2.2.7 UBEC (Voltaj regülatörü)

ESC’ler içerisinde bulunan ve BEC (battery eliminator circuit) adı verilen devreler gerilim regülatörü gibi çalışırlar. Belirli aralıkta bir gerilim değeri verildiğinde çıkış olarak sabit bir gerilim üretirler. Otopilot motorlar için kullanılan ESC’lerden bir tanesinin BEC’ini kullanarak beslenmektedir.

Beagleboard 5V gerilim ile çalışırken yapılan ölçümler sonucunda, üzerinde kamera takılı iken ihtiyaç duyduğu akım değeri 1A civarında olduğu belirlenmiştir. Quadrotor üzerinde çalışan ESC’lerin BEC devreleri 2 A maksimum akım verebilmektedir. Ancak yapılan deneylerde bu devrelerin aşırı ısınması nedeni ile 5-30V giriş değerine karşılık 5V gerilim ve 5A maksimum akım verebilen bir UBEC devresi ile Beagleboard’un beslenilmesi düşünülmüştür.

2.2.8 XBee (İletişim arayüzü)

Quadrotor uçuşu esnasında üzerindeki sensörler hakkında canlı bilgi alabilmek için bir bilgisayar, yer istasyonu olarak kullanılmaktadır. Burada yer istasyonunun kontrol üzerinde hiçbir işlevi yoktur. XBee kablosuz kartları (bir adet quadrotor üzerinde bir adet yer bilgisayarında) kullanılarak, quadrotorun istenilen verileri yer bilgisayarına ulaştırması sağlanır.

XBee hem hafifliği hem de kolay kullanımı nedeniyle tercih edilmiştir. XBee kablosuz haberleşmeyi seri port protokolü kullanarak yapmayı sağlamaktadır. Üzerinde bulundurduğu FTDI çipi sayesinde bilgisayara USB üzerinden bağlanırken, otopilot üzerinde bu bağlantı için hazırlanmış bir yer bulunur. Deneyler sırasında elde edilen veriler bilgisayar tarafında “Labview” tabanında hazırlanmış olan “Configurator” programı ile gözlenebilirken, kayıtların da tutulabilmesi amacı “C#” programlama dili üzerinde XBee’den gelen bilgileri bir dosya içerisine kaydeden bir

program yazılmıştır. Bu program ile quadrotor üzerindeki XBee'ye bağlanıldığında "Configurator" programından farklı bir kod gönderilerek kayıt tutulacağı otopilota söylenmekte ve otopilot kayıt tutma işlemi için göndermesi istenilen bazı değişkenler bilgisayara gönderilmektedir.

2.2.9 Kumanda

Otopilot kullanıcıdan quadrotorun yönlendirilmesi ile ilgili bilgiyi almak için radyo frekansı kullanan kumanda kullanır. Kullanılan kumandalar hobi araçlara göre düzenlendiği için alıcı tarafına PWM cinsinden veriler gönderir. Bu sisteme göre sinyal periyodu 20 ms'dir. Bu sürenin ilk 2 ms'lik kısmında gönderilmek istenilen sinyal belirtilir. Kumanda üzerinde bir kontrol çubuğu minimum seviyesine getirilirse, kumanda alıcısına 1 ms dijital 1 sinyali, 19 ms dijital 0 sinyali gönderilir. Kontrol çubuğu maksimum duruma getirildiğinde 2 ms dijital 1, 18 ms dijital 0 sinyali alıcıya gönderilir. Aradaki değerler 1000 parçaya bölünür. PWM değerleri μ s cinsinden gönderilen dijital 1 komutu ile adlandırılır. Minimum durum 1000 PWM, maksimum durum 2000 PWM'e karşılık gelir.

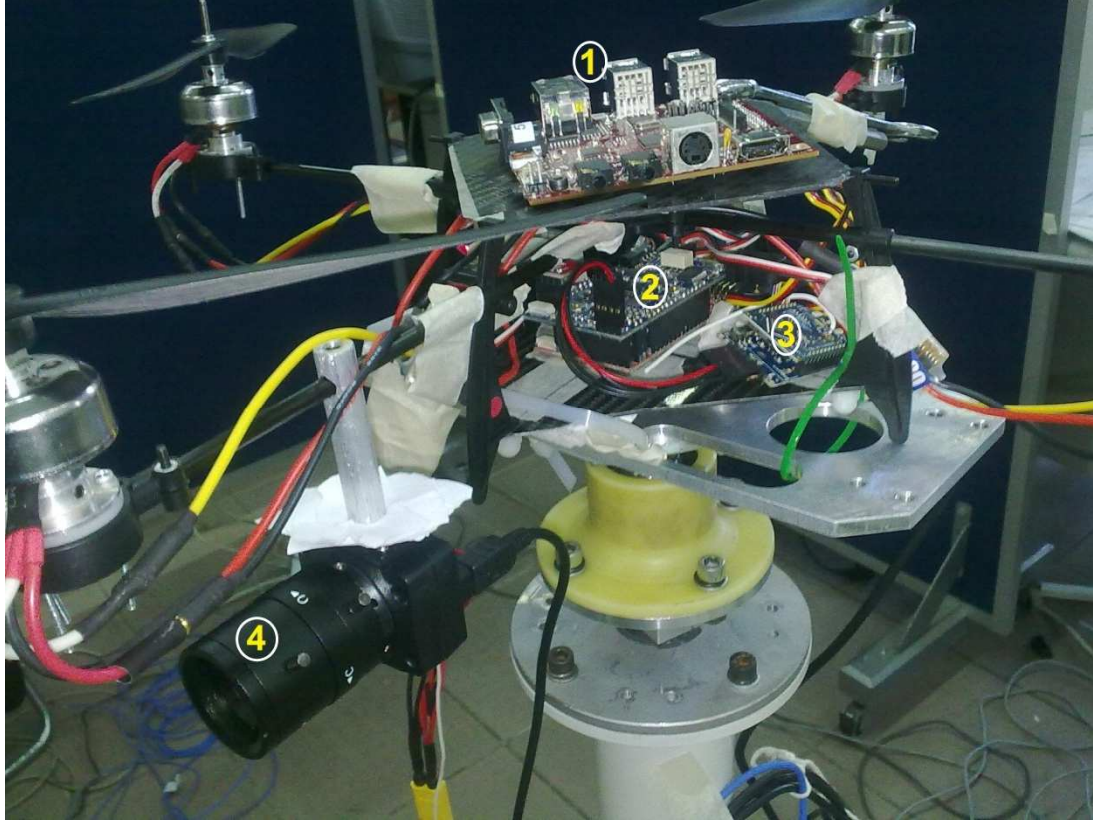
Kumanda üzerindeki dört kanal yükseklik, yunuslama açısı, yalpa açısı ve yönelme açısı referansı vermek için kullanılır. Bu durumda kumanda üzerindeki kontrol çubukları (yükseklik referansı hariç) ortada tutulursa, sisteme 1500 PWM verilir. Bunun anlamı quadrotorun oryantasyonunu tüm açılar sıfır derece olacak şekilde tutmaktır.

Kumanda üzerindeki yükseklik referans kanalı motorların çalıştırılması, durdurulması ve hızlarının artırılıp azaltılmasında rol oynar. Motorların durdurulması 1040 PWM değerine karşılık gelmektedir.

Bu çalışmada kullanılan kumanda 6 kanallı bir kumandadır. 5. ve 6. üzerinden elde edilen bilgi Beagleboard'a gönderilerek uçuş modları değiştirilir. Kumanda kalkış ve iniş gibi kullanıcı kontrolünde gerçekleşmesi gereken işlemlerin yanı sıra güvenlik nedeniyle sistemde bulunmaktadır. 5. ve 6. kanal kullanılarak quadrotorun açısal pozisyon kontrolü kullanıcı ile Beagleboard arasında değiştirilir. Aynı zamanda Beagleboard Linux işletim sistemi ile çalıştığı için ve bu kartın beslemesi motorlar ile aynı kaynaktan yapıldığı için motorlar ve otopilot ile birlikte elektriğinin doğrudan kesilmesi sakıncalıdır. Bu nedenle kumandadaki bu kanallar kullanılarak besleme kesilmeden önce işletim sistemine kapanma komutu gönderilmektedir.

Şekil 2.6’da verilen numaralandırmaya göre hava aracın üzerinde yukarıda anlatılan bileşenlerin yerleşimi şöyledir:

- 1- Beagleboard XM gömülü bilgisayar
- 2- Ardupilot Mega ve ArduIMU (otopilot ve IMU)
- 3- XBee kablosuz haberleşme modülü
- 4- Firefly MV kamera



Şekil 2.6 : Quadrotor üzerinde parçaların yerleşimi.

3. QUADROTOR KONTROLÜ

3.1 Quadrotor Modeli

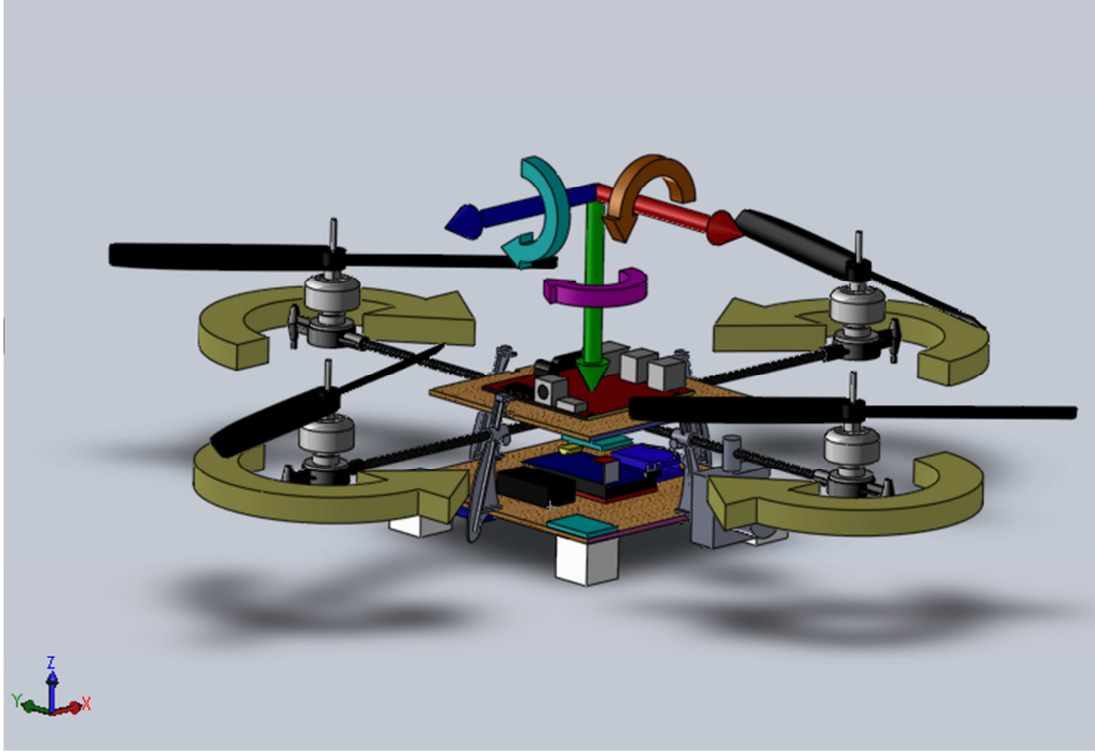
Quadrotor artı şeklinde bağlanmış iki doğrusal parçanın uçlarına dört adet motor ve bu motorların uçlarına birer pervane bağlanarak oluşturulan bir hava aracıdır. Bu motorlar gövde üzerinde sabit bir şekilde dururlar. Aynı şekilde diğer helikopterlerde olduğu gibi pervaneler ile motor arasında da hiçbir servo mekanizma yoktur. Quadrotor tüm hareketlerini üzerinde bulunan dört motorun hızlarını değiştirerek yapar. Bu çalışmada kullanılan quadrotor artı şeklinde kullanılmaktadır. Bu kullanım şeklinde motorlar ön, arka, sağ ve sol olarak adlandırılabilir. Ayrıca çarpı şeklinde kullanılan quadrotorlar da literatürde mevcuttur.

Bu çalışmada kullanılan eksen takımı quadrotorun ileriye doğru hareketi x eksenini olacak şekilde, yukarı hareketi de z ekseninde olacak şekilde seçilmiştir. Buna ek olarak x eksenini etrafında yaptığı açıya yalpalama açısı, z eksenini etrafında yaptığı açıya yönelme açısı ve y eksenini etrafında yaptığı açıya ise yunuslama açısı denilir. Çalışmada yalpalama açısı Φ , yunuslama açısı θ ve yönelme açısı da Ψ ile gösterilecektir.

Quadrotor hareket ettirilmek istendiğinde, motor hızları değiştirilir. Yalpalama hareketi elde etmek için ön ve arka motor arasında hız farkı yaratılır. Aynı şekilde yunuslama hareketi için de sağ ve sol motorlar arasında hız farkı yaratılır. Yönelme hareketinin elde edilebilmesi için ön motor ile arka motor toplam hızları ile sağ ile sol motor toplam hızları arasında fark yaratılır. Quadrotorun yönelme hareketi yapmadan yükselebilmesi için ön ve arka motorun aynı yöne, sağ motor sol motorla aynı yönde fakat ön ve arkaya ters yönde dönmesi gerekir. Yükseklik ise tüm motorların hızlarının artırılması ve azaltılması ile elde edilir.

Şekil 3.1’de quadrotora ait eksen takımı görülmektedir. Motorlar altında gösterilen oklar motorların dönüş yönlerini gösterir. Quadrotor merkezi üzerindeki oklar quadrotor çerçevesi yani quadrotora sabitlenmiş eksen takımını gösterir. Kırmızı ok x eksenini, mavi ok y eksenini ve yeşil ok z eksenidir. Turkuvaz ok yunuslama hareketi

(θ), kahverengi ok yalpa hareketi (Φ) ve pembe ok yönelme hareketini (Ψ) gösterir. Resmin sol alt tarafında kalan eksen takımı ise sabit eksen takımıdır.



Şekil 3.1 : Eksen takımları ve motor dönüş yönleri.

Quadrotor dinamik denklemleri basitleştirilmiş bir şekilde anlatılmıştır [20]. Bu modele göre her bir motorun yukarı yönde uyguladığı kuvvetler motor numarasına göre F_1 , F_2 , F_3 ve F_4 olarak adlandırılmıştır. Ω_i ilgili motorun dönme hızı olduğuna göre motor kuvvetleri:

$$F_i = b\Omega_i^2 \quad (3.1)$$

şeklinde tanımlanır. Buna bağlı olarak motorların quadrotor üzerinde oluşturduğu toplam kuvvet,

$$F_T = b \sum_{i=1}^4 \Omega_i^2 \quad (3.2)$$

m sistemin toplam kütlesi olmak üzere, oluşan toplam ivme,

$$a_F = \frac{b}{m} \sum_{i=1}^4 \Omega_i^2 \quad (3.3)$$

a_F ivmesinin yerde bulunan sabit eksen takımına göre ivmesi Ra_F olmak üzere, kuvvet dengesi yazılırsa,

$$\dot{V} = -ge_z + Re_z a_F \quad (3.4)$$

denklemini elde edilir. Burada R rotasyon matrisidir.

$$e_z = [0 \quad 0 \quad 1]^T \quad (3.5)$$

şeklinde tanımlıdır ve z eksenindeki büyüklükleri göstermek için kullanılır.

Quadrotor çerçevesinin açısal hızı ile rotasyon matrisi arasında aşağıdaki şekilde bir ifade vardır.

$$\dot{R} = RS(\omega) \quad (3.6)$$

Bu denklemde ω quadrotor çerçevesinin açısal hız vektörüdür. $S(\omega)$ ise bu vektörün 3x3 eksi bakışlıdır.

$$\omega = [\omega_1 \quad \omega_2 \quad \omega_3]^T \quad (3.7)$$

$$S(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3.8)$$

Quadrotor açısal hızlarının oluşturduğu açısal momentumlar L_x, L_y, L_z

$$L_{x,y,z} = I\omega \quad (3.9)$$

şeklinde ifade edilir. Burada I quadrotorun ilgili eksenlerdeki ataletidir.

Açısal hızların oluşturduğu tork, açısal momentumun türevi olduğundan

$$\tau_B = \dot{L} \quad (3.10)$$

İfade yerine yazıldığı takdirde x vektörel çarpımı temsil etmek üzere

$$\tau_B = \omega \times I\omega + I\dot{\omega} \quad (3.11)$$

ifadesi elde edilir. Gövde ve pervanelerin eksenleri etrafındaki dönüş hareketlerinden doğan tork ifadesi aşağıda verilmiştir. J herbir pervanenin ataletidir.

$$\tau_G = \sum_{i=1}^4 J(\omega \times e_z) \Omega_i (-1)^i \quad (3.12)$$

Motorların oluşturduğu kuvvetten kaynaklanan tork ifadesi ise

$$\tau_a = \begin{bmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (3.13)$$

l motorlar arası mesafe ve d sürüklenme faktörü olarak verilmiştir.

Sistem üzerindeki tork dengesi aşağıdaki gibi yazılır:

$$\tau_G + \tau_B = \tau_a \quad (3.14)$$

Yukarıdaki denklemde torkların açık ifadeleri yerlerine yazıldığında çıkan sonucun daha rahat görülebilmesi için girişler aşağıdaki şekilde tanımlanır.

$$U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (3.15)$$

$$U_2 = b(\Omega_4^2 - \Omega_2^2) \quad (3.16)$$

$$U_3 = b(\Omega_3^2 - \Omega_1^2) \quad (3.17)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (3.18)$$

Tork denkleğinin çözümünde açısal ivmeler şöyledir:

$$\ddot{\Phi} = \dot{\Psi}\dot{\Theta}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J}{I_x}\dot{\Theta}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{l}{I_x}U_2 \quad (3.19)$$

$$\ddot{\Theta} = \dot{\Psi}\dot{\Phi}\left(\frac{I_x - I_z}{I_y}\right) + \frac{J}{I_y}\dot{\Phi}(-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) + \frac{l}{I_y}U_3 \quad (3.20)$$

$$\ddot{\Psi} = \dot{\Theta}\dot{\Phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \quad (3.21)$$

Ve ivmeler bu şekilde bulunur.

$$\ddot{x} = (C\Psi S\Theta S\Phi + S\Psi C\Phi)\frac{U_1}{m} \quad (3.22)$$

$$\ddot{y} = (S\Psi S\Theta C\Phi - C\Psi S\Phi)\frac{U_1}{m} \quad (3.23)$$

$$\ddot{z} = -g + (C\Theta C\Phi)\frac{U_1}{m} \quad (3.24)$$

Bu denklemlerde S sinüsü, C kosinüsü ifade eder.

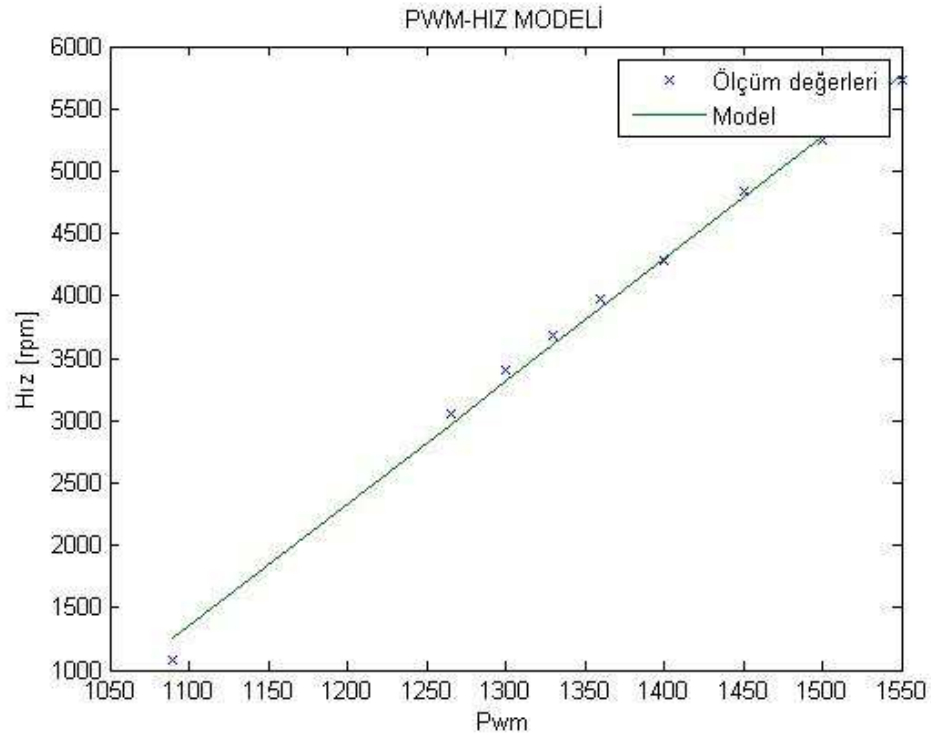
Bu denklemlerde verilen atalet momentlerinin kullanılan quadrotor için gerçek değerleri Solidworks programı ortamında yapılan çizimler sonucunda elde edilmiştir. Motorlar ile ilgili katsayıların belirlenmesi için motorlar bölümünde anlatılan deney, kullanılan motor ve pervane için tekrarlanmış ve bu sefer daha çok nokta üzerinde ölçüm alınırken, aynı zamanda pervane dönüş hızları da ölçülmüştür. Bu şekilde

PWM-pervane hızı-kaldırma kuvveti arasındaki ilişki matematiksel olarak ifade edilebilir hale gelmiştir. Bu deneyin sonucu Çizelge 3.1’de verilmiştir.

Çizelge 3.1 : Motor detaylı ölçümü.

PWM	AKIM (A)	KALDIRMA KUVVETİ (gr)	HIZ (rpm)
1090	0.3	0	1080
1265	1.4	154	3050
1300	1.8	169	3412
1330	2.2	200	3690
1360	2.6	230	3974
1400	3.3	269	4285
1450	4.4	344	4845
1500	5.6	399	5240
1530	6.2	426	5450
1550	7.1	467	5730

Şekil 3.2’de PWM-hız doğru uydurma işlemine ait grafik verilmektedir.

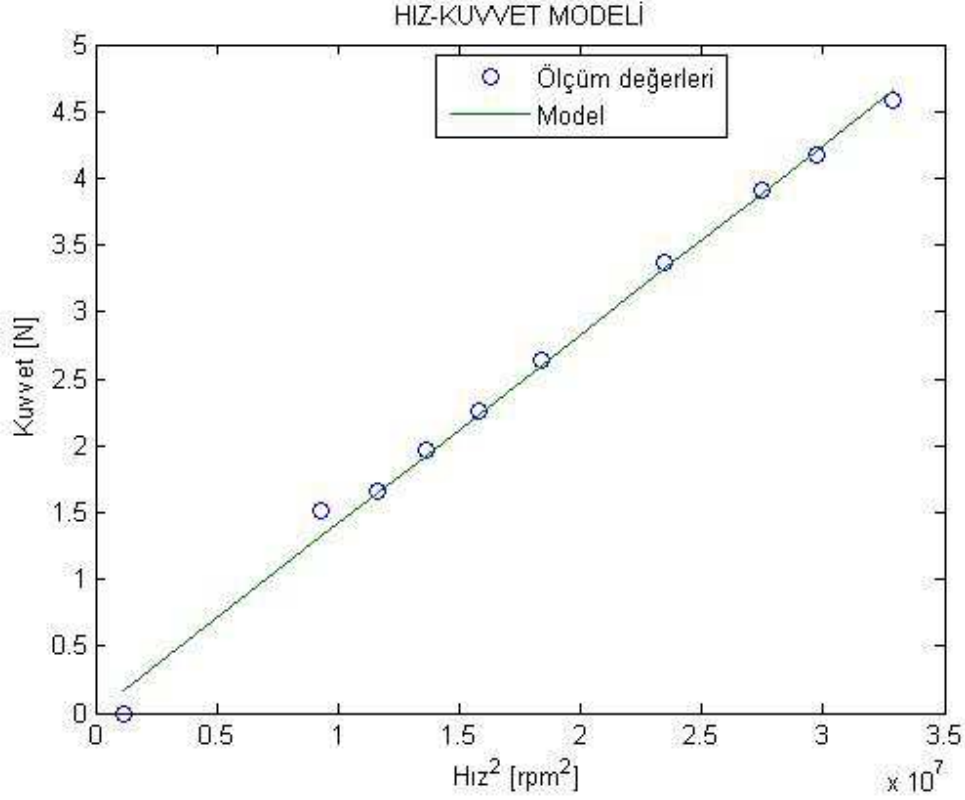


Şekil 3.2 : PWM-Hız doğru uydurma sonuçları.

Elde edilen değerler p PWM değeri olmak üzere (Şekil 3.2’ye göre):

$$\Omega = 9.829xp - 9463 \quad (3.25)$$

denklemini verir. Modelin başında motor modeli olarak çıktısı kuvvet olan girdisi ise hızın karesi olan lineer bir bağıntı verilmiştir. Bu lineer bağıntı yine doğru uydurma yöntemi ile MATLAB aracılığıyla bulunmuştur. Bu işlem sonucunda b katsayısı 1.4163×10^{-7} olarak bulunmuştur. İşlem sonucu oluşan doğru Şekil 3.3'te verilmektedir.



Şekil 3.3 : Hız ile kaldırma kuvveti doğru uydurma sonuçları.

Modelde tanımlanan d katsayısı için tek bir ölçüm alınabilmektedir. 3412 rpm devirde motorun 0.1226 Nm'lik bir tork ile kendi eksenini etrafında dönmeye çalıştığı fark edilmiştir. Bu durumda d katsayısı için 1.0533×10^{-8} denilebilir.

Çizelge 3.2'de quadrotor modelinde kullanılan sayısal değerler verilmektedir.

Çizelge 3.2 : Quadrotor modelinin sayısal değerleri.

Sembol	Anlamı	Sayısal Değeri [Birim]
I_x	Yalpalama ataleti	$10.35 \times 10^{-3} \text{ [kgm}^2\text{]}$
I_y	Yunuslama ataleti	$11.41 \times 10^{-3} \text{ [kgm}^2\text{]}$
I_z	Yönelme ataleti	$17.66 \times 10^{-3} \text{ [kgm}^2\text{]}$
J	Rotor ataleti	$3.94 \times 10^{-5} \text{ [kgm}^2\text{]}$
b	Motor itme katsayısı	$1.42 \times 10^{-7} \text{ [N/rpm}^2\text{]}$
d	Motor sürüklenme katsayısı	$1.05 \times 10^{-8} \text{ [Nm/rpm]}$
l	Quadrotor kol uzunluğu	0.211 [m]
m	Sistemin toplam kütlesi	1045 [gr]

Quadrotorun çalışma noktası, tüm açıların sıfır olduğu nokta olarak tanımlanırsa doğrusallaştırma işlemi sonucunda sistemin transfer fonksiyonu:

Bu denklemleri PID tasarımına uygun kılmak ve basitleştirmek için nispeten önemsiz jiroskopik etkiler atılırsa denklemler şöyledir [21]:

$$\ddot{\Phi}, \ddot{\theta}, \ddot{\Psi} = \frac{l}{I_{x,y,z}} U_{2,3,4} \quad (3.26)$$

Ve çalışma noktasının sıfır açısal konum seçilmesi halinde yükseklik denkleminin lineerleştirilmiş hali:

$$\ddot{z} = -g + \frac{U_1}{m} \quad (3.27)$$

şeklinde elde edilir. Quadrotor eksik eyleyicili (under-actuated) bir sistem olduğu için (giriş sayısı 4 çıkış sayısı 6) tüm çıkışları birden fazla PID ile aynı anda kontrol etmemiz mümkün olmadığı için 4 çıkış kontrol için seçilmiştir [22].

Verilen ifadelere Laplace dönüşümleri uygulandığında sistemin transfer fonksiyonları,

$$\frac{\Phi(s)}{U_2(s)} = \frac{l}{I_x s^2} \quad (3.28)$$

$$\frac{\theta(s)}{U_3(s)} = \frac{l}{I_y s^2} \quad (3.29)$$

$$\frac{\Psi(s)}{U_4(s)} = \frac{l}{I_z s^2} \quad (3.30)$$

$$\frac{Z(s)}{U_1(s)} = \frac{s - mg}{ms^3} \quad (3.31)$$

şeklinde ifade edilir.

3.2 DCM Algoritması

IMU (ataletsel ölçüm ünitesi) içerisinde çeşitli sayılarda ivmeölçer ve jiroskopu bir arada bulunduran tümleşik bir elemandır. Toplam ivmeölçer ve jiroskop eksen sayısına göre isimlendirilir. Bu projede kullanılan IMU 6 eksenlidir, yani 3 eksenli bir jiroskop ve 3 eksenli bir ivmeölçer bulundurur.

İvmeölçerler bulundukları eksenlerdeki statik ve dinamik ivmeleri ölçerler. Bu durumda üzerinde hiçbir titreşim bulunmayan bir ivmeölçer yerçekimi yönünde yerleştirildiğinde 1 g (yerçekimi ivmesi) değerinde bir çıkış verir. 3 adet birbirine dik yerleştirilmiş bir ivmölçer ile de yerçekiminin yönü bulunabilir. Bunun sonucunda yunuslama ve yalpalama açıları ölçülebilir. Ancak bu ivmeölçerler hareketli bir cisim üzerinde bulunuyorsa, bu cismin ivmelenmeleri sonuçların bozulmasına yol açar. Eğer sistem üzerinde titreşim varsa gürültü daha da büyür. Quadrotor hem ivmeli hareketler yapan hem de pervane kaynaklı titreşime maruz bir sistemdir.

Jiroskoplar açısal hızı ölçerler. Jiroskoplara bakarak açı tayin edilmek istenirse, elde edilen değer bir defa integrali alınması ile açı elde edilebilir. Ancak buradaki sorun integrasyon işleminin hatayı da biriktiriyor olmasıdır. Bunun sonucunda sonuçlar gerçek değerlere göre ötelenmiş olarak gözükür ve zaman içerisinde bu hata sürekli büyür.

Yukarıda anlatılan nedenler dolayısıyla sadece jiroskoplar veya sadece ivmeölçerler ile quadrotor üzerinde açı ölçümü yapmak mümkün değildir. Ancak Kalman filtresi veya DCM algoritması kullanımı ile bu altı sensörden titreşimden ve ivmelenmeden etkilenmeyen, zaman içinde hata biriktirmeyen bir oryantasyon ölçüm sistemi

oluşturmak mümkündür. Kullanılan Ardupilot kodu ArducopterNG içerisinde hazır olarak DCM algoritması kullanılmaktadır. Tezin ana amacı IMU'dan daha doğru okuma yapmak olmadığı için kodun bu kısmı üzerinde değişiklik yapılmamıştır. Ancak quadrotor kontrolünün daha iyi anlatılabilmesi için bu bölümde kod içerisindeki hesaplamalar kısaca anlatılmıştır. Bu bölümde yapılacak açıklamalarda [23] kaynak olarak alınmıştır.

Bu algoritma içerisinde jiroskoplar temel ölçüm elemanı olarak kullanılmıştır. Jiroskoplar açısal hızı ölçtükleri için bir kez integre edilmeleri halinde açısal pozisyon elde edilebilir. Ancak burada sensör verileri ayrık zamanlı ve dijitalleştirilmiş veriler olduğu için iki hata ortaya çıkar. Bunlardan birincisi integrasyon hatasıdır. Ayrık zamanlı bir veride iki veri arasındaki zamanda ne olduğunu kullandığımız metot belirler. Örneğin, bu iki zaman arası açısal hız sabit kabul edilebilir. Ancak bu kabul integre edildiğinde biriken hatalar sonuçların yanlış olmasına neden olur. İkinci hata ise kuantalama (quantization) hatasıdır. Burada analog dijital çeviricinin ne kadar yüksek çözünürlükle bu işlemi gerçekleştirirse gerçekleştireceği aslında analog veriden bir miktar kaybedecek olması hatanın kaynağıdır.

Rotasyon matrisleri ortogonaldır. Yani sütun vektörlerinin boylarının 1 olması ve bu vektörlerin birbirlerine dik olmaları gibi 6 adet sınır şartı yazılabilir. Jiroskop verisinde ortaya çıkan kayma bu şartları bozar.

Yukarıda anlatılan nedenler dolayısıyla sadece jiroskoplar veya sadece ivmeölçerler ile quadrotor üzerinde açı ölçümü yapmak mümkün değildir. Ancak Kalman filtresi veya DCM algoritması kullanımı ile bu altı sensörden titreşimden ve ivmelenmeden etkilenmeyen, zaman içinde hata biriktirmeyen bir oryantasyon ölçüm sistemi oluşturmak mümkündür. Kullanılan Ardupilot kodu ArducopterNG içerisinde hazır olarak DCM algoritması kullanılmaktadır. Tezin ana amacı IMU'dan daha doğru okuma yapmak olmadığı için kodun bu kısmı üzerinde değişiklik yapılmamıştır. Ancak quadrotor kontrolünün daha iyi anlatılabilmesi için bu bölümde kod içerisindeki hesaplamalar kısaca anlatılmıştır.

$$r(t) = r(0) + \int_0^t d\theta(\tau) \times r(\tau) \quad (3.32)$$

$$d\theta(\tau) = \omega(\tau)d\tau \quad (3.33)$$

$r(0)$ başlangıç vektörü ve $\int_0^t d\theta(\tau) \times r(\tau)$ ise jiroskoptaki değişimi gösterir. Rotasyon matrisinin her bir satırı veya sütunu dönme hareketi yapan vektörler gibi görülmektedir. Yukarıdaki denklemler quadrotor üzerindeki eksen takımına göre yazılmıştır. Quadrotor eksen takımına göre yeryüzü eksen takımının hareketleri ters yönde hareket eder.

$$r_W(t) = r_{W(0)} + \int_0^t r(\tau) \times d\theta(\tau) \quad (3.34)$$

$$d\theta(\tau) = \omega(\tau)d\tau \quad (3.35)$$

$r_W(t)$ yeryüzü eksen takımının eksenlerinden bir tanesi, helikopterden görüldüğü şekliyle tanımlanırsa,

$$r_{W(t+dt)} = r_{W(t)} + r_{W(t)} \times d\theta(t) \quad (3.36)$$

$$d\theta(t) = \omega(t)dt \quad (3.37)$$

ifadesine ulaşılır. Bu durumda herhangi bir andaki oryantasyon aşağıdaki gibidir:

$$R(t + dt) = \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix} \quad (3.38)$$

$$d\theta_x = \omega_x dt \quad (3.39)$$

$$d\theta_y = \omega_y dt \quad (3.40)$$

$$d\theta_z = \omega_z dt \quad (3.41)$$

Rotasyon matrisinin her bir sütunu bir vektör olarak görülürse, bu üç vektörün birbirine dik olması gerekir. Jiroskop üzerinde oluşan hatalar rastgele olduğu için hatalı değerlerde bu diklik bozulur. Bu özellikten yararlanılarak oluşan hatanın bir kısmı telafi edilebilir ve bu vektörler tekrar ortagonal hale getirilir. Ortogonal iki vektör skaler çarpıma sokulduğunda sonuç 0 olur. Ancak oluşan hata sonucunda bu değer değişmektedir. Ortaya çıkan değer hata olarak ele alınır ve iki eşit parça olarak x ve y eksenlerine dağıtılır:

$$err = x.y \quad (3.42)$$

$$x_{ort} = x - \frac{err}{2}y \quad (3.43)$$

$$y_{ort} = y - \frac{err}{2}x \quad (3.44)$$

x ve y birbirlerine dik şekilde elde edildiklerinde bunlara dik olan z'yi bulmak için vektörel çarpım kullanılır.

$$z = x \otimes y \quad (3.45)$$

Bu vektörlerin herbiri bir birim vektör olduğu için boylarının da bir olması gerekmektedir. Çıkan sonuçlar farklı olduğu için bir normalizasyon işlemine ihtiyaç duyulur. Algoritmada mikroişlemciye kolaylık olması amacı ile normalizasyon işlemi şu şekilde tanımlanmıştır:

$$x_{normal} = 0.5(3 - x_{ort}x_{ort})x_{ort} \quad (3.46)$$

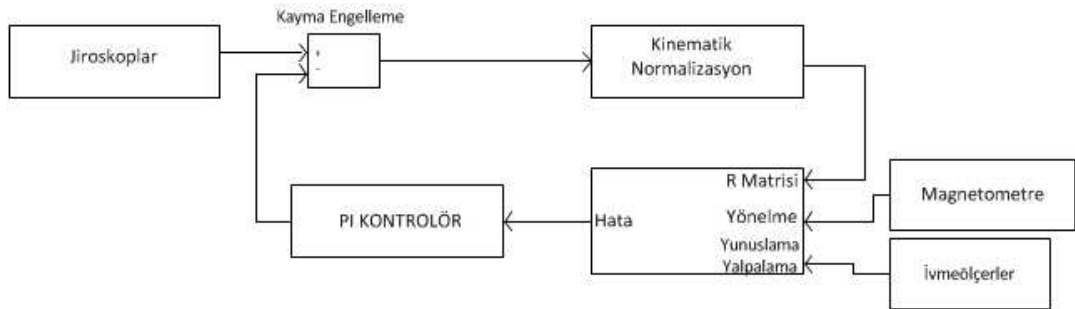
$$y_{normal} = 0.5(3 - y_{ort}y_{ort})y_{ort} \quad (3.47)$$

$$z_{normal} = 0.5(3 - z_{ort}z_{ort})z_{ort} \quad (3.48)$$

Yukarıdaki işlemler sonucunda jiroskopta oluşan hataların bir kısmı yok edilmektedir. Ancak işlemler sonucu hata birikmektedir. Sonuç olarak pozisyon değeri kaymaktadır. Bu kaymayı engellemek için ivmeölçer kullanılır. Quadrotorun belirli bir süre için ivmeleneceği kabulü yapılırsa ivmeölçer $g_{referans}$ olarak kullanılabilir. İvmeölçerin titreşim hareketlerinden etkilenmemesi için alınan değerler bir alçak geçiren filtreden geçirilmektedir. Kaymadan dolayı oluşan hata değeri rotasyon matrisinin 3. sütunu ile referans yerçekimi ivmesinin vektörel çarpımı ile elde edilir:

$$P = \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} x g_{referans} \quad (3.49)$$

Sistem çalışması Şekil 3.4'teki blok diyagramı ile de gösterilebilir.



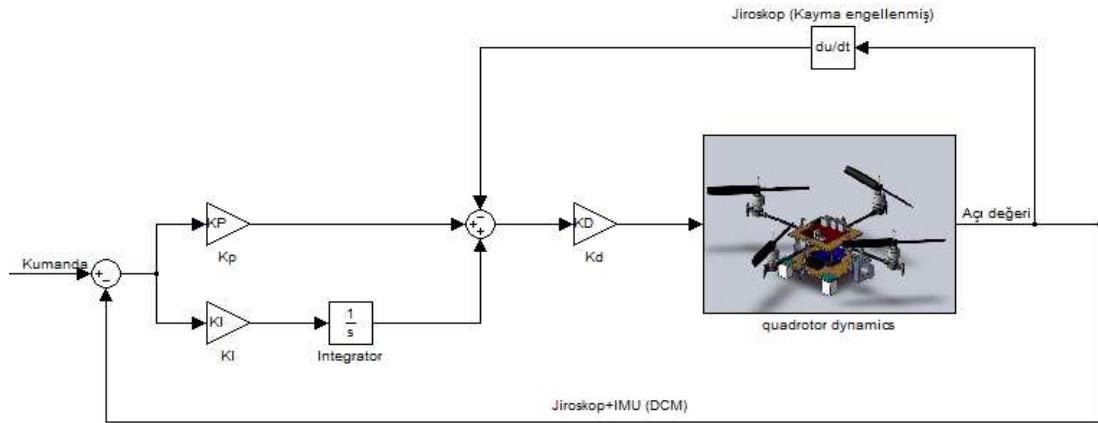
Şekil 3.4 : DCM blok diyagramı [23]'ten uyarlanmış.

3.3 PID Kontrol

Quadrotorun kontrolünde ArducopterNg kodu içerisindeki “stable mode” kısmında gerçekleştirilen PID kontrol algoritmasından yararlanılmıştır.

PID kontrol hata değişkeninin oranı, türevi ve integrali ile yapılan bir kontrol şeklidir. Temel olarak ele alınan hata değişkeninin belirli bir oranı (K_p), değişim oranının belirli bir oranı (K_d) ve değişkenin sürekli olarak toplanması sonucu oluşan integralinin belirli bir oranı (K_i) toplamı ile çalışır. Amacı istenilen kumanda değeri ile ölçülen değer arasındaki hatayı minimize etmektir. Hem teorik olarak hem de gerçekleştirilmesi kolay olduğu için endüstride yaygın bir şekilde kullanılmaktadır. Performans kriterleri doğrultusunda PID kontrolörün katsayılarının elle ayarlanması pratikte sıklıkla karşılaşılan bir durumdur. Buna karşın pek çok halde bu katsayıların analitik olarak hesaplanması mümkündür.

Klasik PID tek döngü ile oluşturulur ve bu PID'nin en basit halidir. Ancak farklı durumlar için PID algoritmaları temel mantık aynı olacak şekilde modifikasyonlara uğramıştır. Genişletilmiş PID iki döngü ile çalışan bir PID'dir ve quadrotor kontrolünde sıklıkla kullanılır. Kullanılan PID blok diyagramı Şekil 3.5'te görülebilir.



Şekil 3.5 : Quadrotor kontrol blok diyagramı.

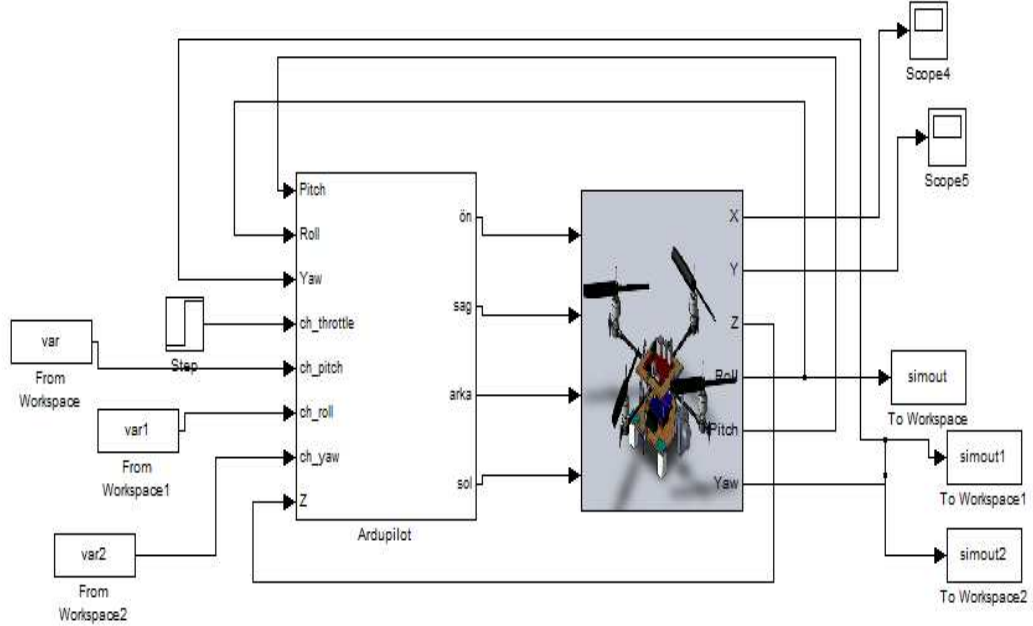
3.1 bölümünün sonunda quadrotor çıkış açıları için verilen transfer fonksiyonlarına yukarıda anlatılan PID kontrol uygulanırsa kapalı sistem transfer fonksiyonu:

$$G_{cl}(s) = \frac{K_p K_d A s + K_i K_d A}{s^3 + K_d A s^2 + K_p K_d A s + K_i K_d A} \quad (3.50)$$

olur. A burada $l/I_{x,y,z}$ 'dir. Görüleceği üzere sistemin karakteristik denkleminde 3 adet kök ve 3 adet bilinmeyen parametre vardır. Sistem kök yerleştirme yaklaşımına müsaittir. Bu nedenle, bu yöntem kullanılarak, kalıcı hal hatası ve osilasyonu olmayan bir sistem oluşturulmaya çalışılacak ve kontrol eforunun çok büyümemesi göz önünde bulundurulacaktır.

3.4 Benzetimler

Simulasyon için oluşturulan lineer ve basitleştirilmiş modele ek olarak, çalışmada verilen quadrotor modeli üzerinde bir miktar değişiklik yapılmıştır [24]. Ayrıca kontrolör olarak Ardupilot üzerinde çalışan kodun, dengeleme kısmı Simulink ortamında modellenmiştir (Şekil 3.6). Bu model girişleri kumanda girişi şeklinde (PWM olarak) almaktadır ve kod içerisinde belirtilen saturasyon değerleri (motorları korumak için) aynı şekliyle simulasyon ortamına aktarılmıştır. Ardupilot kısmının çıkışları dört motor için PWM değerleri olacak şekilde ayarlanmıştır. Quadrotor dinamiği kısmında ise PWM-motor hızı-kaldırma kuvveti arasındaki ilişki dahil edilmiştir.



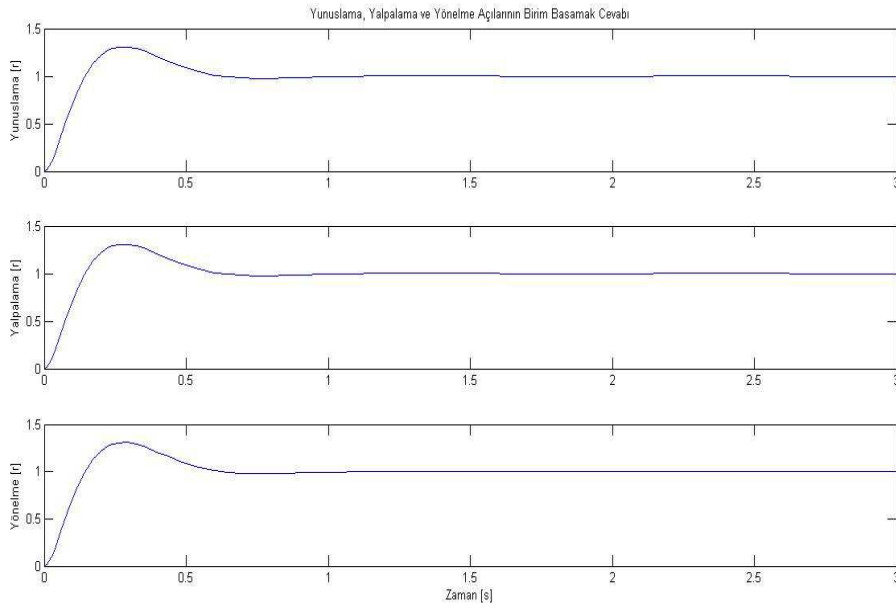
Şekil 3.6 : Simulasyon ana pencere.

Önceki bölümde anlatıldığı üzere hava aracından beklenen performans kriterleri s-düzlemi üzerinde üç kök oluşturmakta ve bu üç kökün yerine bakılarak K_p , K_i ve K_d

değerleri elde edilebilmektedir. Öncelikle yunuslama ekseninde hava aracının %10 kadar aşma yapmasına izin verilmesi ve oturma zamanının 0.75 saniye olarak belirlenmesi düşünülmüştür. Kalıcı hal hatası sistemin kapalı çevrim transfer fonksiyonunun statik kazanç katsayısından anlaşılacağı üzere sıfırdır. İstenilen performans kriterlerine göre eşlenik kökler $-5.33 \pm 7.28i$ noktalarında bulunmuştur. Üçüncü kök ise daha hızlı seçilerek sisteme etki etmesi önlenmiştir ve bu köklerin reel kısmından 5 kat uzakta reel eksene yerleştirilmiştir. Üçüncü kök -26.67 'dir. Karakteristik denklem bu köklere göre yazılırsa sistemin kapalı çevrim transfer fonksiyonu içerisindeki üç bilinmeyen bulunabilir. Çizelge 3.3'te yunuslama, yalpalama ve yönelme açılarına ait kontrolör katsayıları verilmiştir.

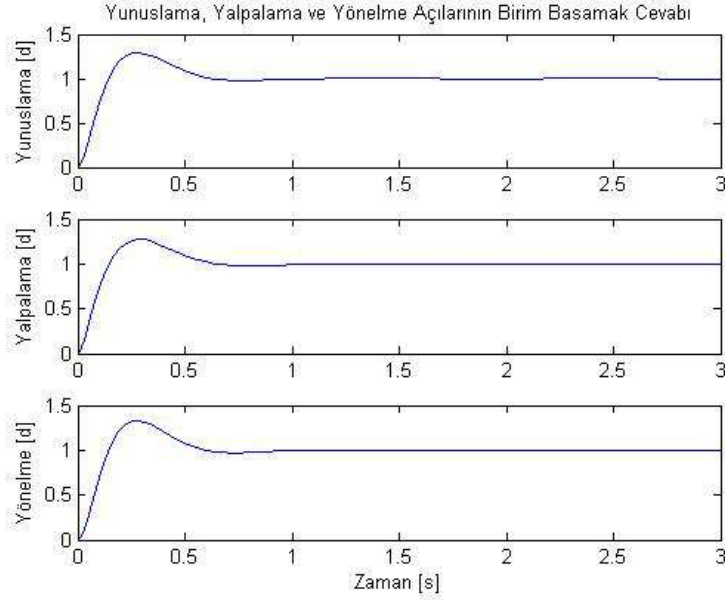
Çizelge 3.3 : 0.1 aşma ve 0.75 s. oturma süresi için PID katsayıları.

	K _p	K _i	K _d
Yunuslama	9.8	58	1.92
Yalpalama	9.8	58	1.75
Yönelme	9.8	58	2.98



Şekil 3.7 : Açısal pozisyonların PID birim basamak cevapları.

Kullanılan model bazı etkilerin ihmal edildiği ve lineerleştirilmiş bir model olduğu için katsayıların belirlenmesinde kullanılabilir. Ancak bu model ile elde edilen simulasyon sonuçları (Şekil 3.7) hava aracının dinamiğini tam yansıtamayacağı için bulunan katsayılar ile doğrusal olmayan ve Ardupilot modelinin dahil edildiği simulasyon ortamında simulasyon tekrarlanmıştır (Şekil 3.8).



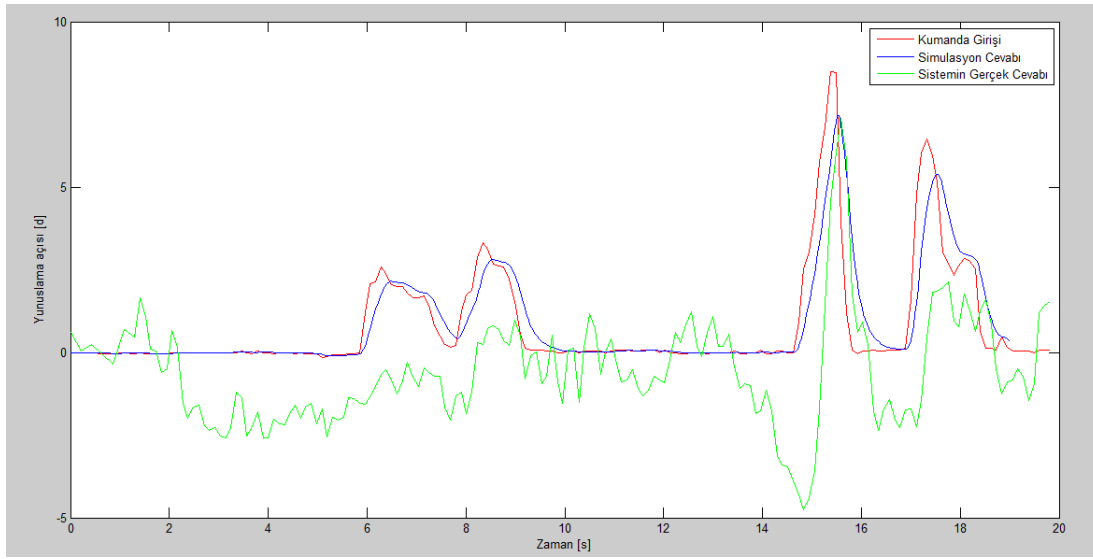
Şekil 3.8 : Doğrusal olmayan modele göre birim basamak cevapları.

Yukarıdaki şekilde %10 aşmalı durumda sistemin birim basamak cevabı, ardupilotun birebir modellendiği ve sisteme gelen non-lineer etkilerin dahil edildiği simulasyon ortamında üretilmiştir. Görüldüğü üzere sistem kararlı kalmakta ve istenilen zaman diliminde cevaba oturmaktadır. Ancak gerçek sistem üzerinde verilen katsayılar ile yapılan denemelerde sistemin salınıma girdiği ve kararsızlığa gittiği görülmüştür. Bu davranışın modelde görülememesinin nedeni modelde yapılan basitleştirmelerdir. Örneğin motorların zaman sabitleri modellenmemiş ve motorların verilen girdiye anında cevap verdikleri kabul edilmiştir. Salınımı azaltmak için istenilen durumlarda değişikliğe gidilmiş ve aşma olmaması ve sönüm katsayısının 1'den büyük alınması düşünülmüştür. Ayrıca sistemin istenilen oturma zamanına yetiemediği de görüldüğü için bu kriter de iki katına çıkarılmıştır. Bu durumda ortaya çıkan katsayılar Çizelge 3.4'te verilmiştir.

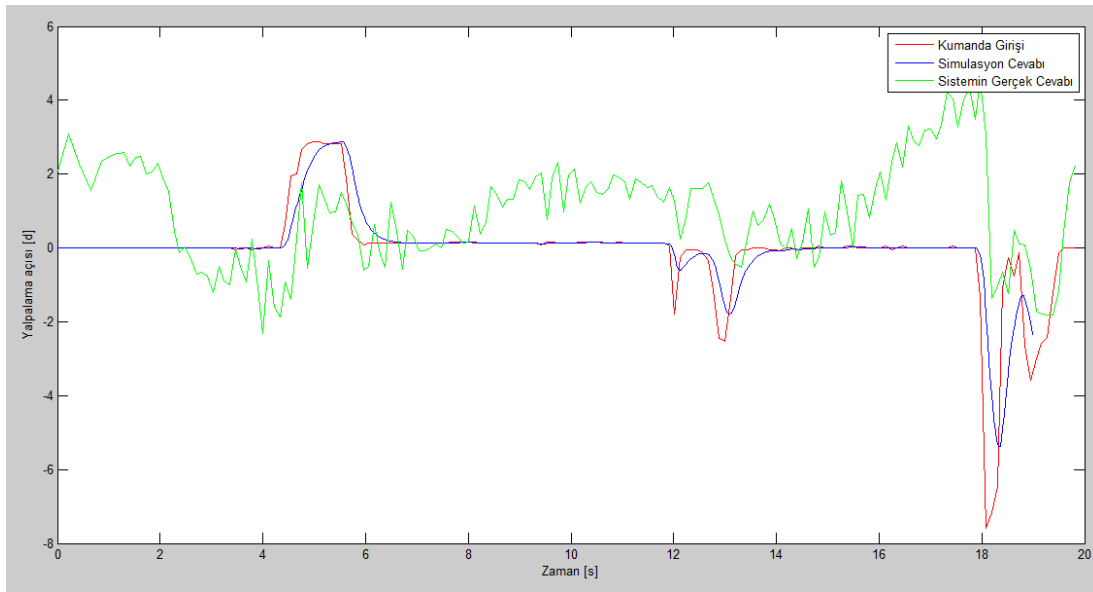
Çizelge 3.4 : Aşmasız ve 1.5 saniye oturma süresi için PID katsayıları.

	Kp	Ki	Kd
Yunuslama	4.45	0.65	1.61
Yalpalama	4.45	0.65	1.46
Yönelme	4.45	0.65	2.50

Bu değerler kullanılarak oluşturulan simülasyon sonuçları ve deneysel veriler ise Şekil 3.9’da ve Şekil 3.10’da verilmiştir.



Şekil 3.9 : Yunuslama açısı simülasyon ve deney sonuçları.



Şekil 3.10 : Yalpalama açısı simülasyon ve deney sonuçları.

Deney sonuçlarından görüldüğü üzere hava aracı ilk 15 saniyelik durumuda dengede kalmayı başarmakta ancak yer etkisinin (özellikle kendi oluşturduğu rüzgarın yerden yansıyıp kendisine tekrar dönmesi nedeniyle) etkisi altında kalmaktadır. Bu süreden sonra kumanda ile verilen referans değerlere verdiği tepkiler, simülasyonda elde edilenle yakınlık göstermekte ancak başlangıç değerleri aynı olmadığı için aynı noktalara oturmamaktadır. Ancak elde edilen bu katsayılar ile quadrotor kararlı bir uçuş göstermektedir.

3.5 Gerçekleme

Uçuş denemeleri için tek eksenli deney düzeneği üzerinde denemelerle başlanmıştır. İlk denemelerde beagleboard ve kamera yokken denemeler yapılmış daha sonra bunlar hava aracı üzerine yüklenmiştir.

Kullanılan kod içerisinde IMU okumaları ve motora gönderilen veriler hızlı döngü içerisinde 200 Hz'de yapılmaktadır. Barometreden yapılan ölçüm ve yer bilgisayarı ile iletişim ise 10 Hz gibi yavaş bir döngü içerisinde yapılmaktadır.

Başarılı bir uçuş denemesine ait resim Şekil 3.11'de görülebilir. Denemeler esnasında pilin sıklıkla bitmesi nedeniyle 40A çıkış verebilen bir güç kaynağı ile uçuş testleri yapılmıştır. Quadrotorun düşüşler esnasında zarar görmesini önlemek amacı ile uçuşun yapıldığı bölgede ve quadrotorun alt yüzeyinde ve laboratuvar zemininde sönümleyici malzemeler kullanılmıştır.



Şekil 3.11 : Uçuş deneyleri.

4. EŞ ZAMANLI KONUMLANDIRMA VE HARİTALAMA

4.1 Giriş

Quadrorotorun önceden bilinmeyen ve önceden şekillendirilmemiş bir alanda kontrolünün sağlanabilmesi için sensörlerinden gelen bilgiyi kullanarak kendisini konumlandırması gerekir. Konumlandırma işlemini yapabilmek için de bir haritanın var olması gereklidir. Konumlandırma ve haritalama yapabilmek için geleneksel olarak kullanılan yöntem SLAM yaklaşımıdır. Genellikle bir lazer yardımı ile robot etrafındaki objelerin haritasının elde edilmesi ve bu haritaya göre robotun pozisyonunun belirlenmesinde kullanılır. Ancak bu çalışmada görüntü işleme kullanılacağı için konumlandırma ve haritalama yaklaşımlarına alternatif olarak SFM'i de göstermek mümkündür. SFM yaklaşımı, kamera hareketinden hava aracının bulunduğu ortamın üç boyutlu haritasını çıkarmayı amaçlar ve bunu yaparken de referans noktaları sürekli takip ettiği için aynı zamanda konumlandırma da yapmış olur. Bu yaklaşım genellikle artırılmış gerçeklik uygulamaları için kullanılmaktadır. Ancak algoritmanın ilk çıkış amacından da anlaşılacağı gibi burada öncelikli problem konumlandırma değil, haritanın çıkarılması ya da başka bir deyişle bir objenin üç boyutlu modelinin çıkarılmasıdır. Bu algoritmanın hem yüksek işlemci ve bellek maliyeti hem de haritalamayı ön planda tutması nedeni ile bu çalışmada SLAM yaklaşımı üzerinden gidilmiş ve genişletilmiş Kalman filtresi tabanlı bir SLAM algoritması ile haritalama ve konumlandırma yapılmıştır.

4.2 Kalman Filtresi ve Genişletilmiş Kalman Filtresi

Teorik olarak Kalman filtresi lineer kuadratik problem için bir tahmincidir ve beyaz gürültülü bozucu etkisine maruz, beyaz gürültülü ölçümleri olan lineer sistemlere uygulanır [25]. Kalman filtresini pratik olarak ise yirminci yüzyılın en büyük buluşları arasındadır, hatta önemi elektronikteki silikona eşdeğerdir [25]. Kalman filtresi sistemin durumlarını tek ölçüme bakarak kestirmek yerine, zaman içerisinde

pek çok ölçüme bakarak tahmin eder ve rastgele gürültülü ve bozucu sistemlerde daha kesin gözlemler yapılmasını sağlar.

Non-lineer sistemler üzerinde genişletilmiş Kalman filtresi (EKF) kullanılmaktadır. EKF aşağıdaki şekilde çalışır [26].

$$\hat{x}_{(k|k-1)} = f(\hat{x}_{(k-1|k-1)}, u_{k-1}) \quad (4.1)$$

$$P_{(k|k-1)} = F_{k-1}P_{(k-1|k-1)}F_{k-1}^T + Q_{k-1}$$

$\hat{x}_{(k|k-1)}$ Durum tahminlerini içeren vektördür. $P_{(k|k-1)}$ ise kovaryans tahmin matrisidir. Q sistemdeki bozucu etkileri (kovaryansını) göstermektedir. Alınan ölçümler ile tahmin edilen değerler arasındaki fark şu şekildedir:

$$y_k = z_k - h(\hat{x}_{(k|k-1)}) \quad (4.2)$$

Kovaryans matrisinin ölçümlere göre düzenlenmesi inovasyon kovaryans matrisini verir. Burada R ölçümlerin gürültü kovaryans değeridir.

$$S_k = H_k P_{(k|k-1)} H_k^T + R_k \quad (4.3)$$

Bu değerler ile Kalman kazancı şöyle bulunur:

$$K_k = P_{(k|k-1)} H_k^T S_k^{-1} \quad (4.4)$$

Tahmin edilen durum ve kovaryans değerleri Kalman kazancı aracılığı ile güncellenir ve Kalman filtresinin çıktısını verir.

$$\hat{x}_{(k|k)} = \hat{x}_{(k|k-1)} + K_k y_k \quad (4.5)$$

$$P_{(k|k)} = (I - K_k H_k) P_{(k|k-1)}$$

Yukarıda verilen dinamik matris ve ölçüm matrisi, EKF için şu şekilde tanımlanır:

$$F_{k-1} = \frac{\partial f}{\partial x} |_{\hat{x}_{(k-1|k-1)}, u_{k-1}} \quad (4.6)$$

$$H_k = \frac{\partial h}{\partial x} |_{\hat{x}_{(k|k-1)}}$$

Bu ifadeler lineerleştirme işleminden gelmektedir.

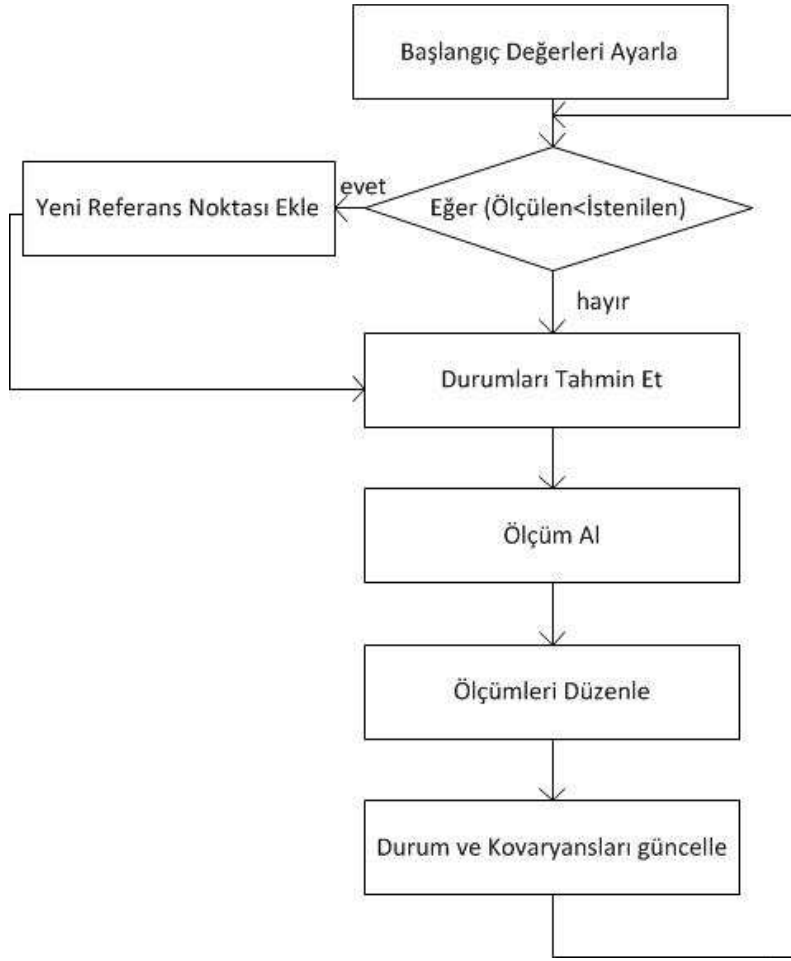
4.3 SLAM

SLAM (eş zamanlı konumlandırma ve haritalama) bir robotun harita çıkarırken aynı anda kendisini bu haritaya göre konumlandırması problemidir. Robot sabit bir eksen takımına göre kendi koordinatlarını belirlemesi, navigasyon çevriminde geribesleme şeklinde kullanılacaktır.

SLAM problemi sadece konumlandırmaya göre daha zor bir problemidir çünkü her bir adımda sadece robotun durumu değil, harita üzerindeki noktaların da durumlarının güncellenmesi gerekmektedir. Bu nedenle sistemin durum uzay modeli, sadece konumlandırma uygulamalarına göre çok daha büyük boyutlu olur.

4.4 V-SLAM

V-SLAM ya da görüntü temelli konumlandırma ve haritalama, Slam algılayıcısı olarak sadece kameranın kullanılması ile sağlanır. Bu çalışmada EKF temelli bir V-SLAM algoritması kullanılmıştır [4]. Algoritmanın gerçekleşmesinde görüntü işleme ve matris işlemlerinin kolay yapılabilmesi için OpenCV 2.1 kütüphanesi kullanılmıştır. Program Visual Studio ortamında C++ dili kullanılarak yapılmıştır. Burada C++ kullanılmasının temel iki nedeni, OpenCV'nin bu dile destek veriyor olması ve yazılacak algoritmanın gömülü linux işletim sistemi altında kullanılacak olmasıdır. Bu kısımda algoritmanın nasıl çalıştığı ve matematiği anlatılacak, bu kısmın sonunda diğer çalışmaların gerçeklemesi ile arasındaki farklar anlatılacaktır. Şekil 4.1'de algoritmanın çalışmasını anlatan bir akış diyagramı bulunmaktadır. Akış diyagramından anlaşıldığı üzere başlangıç değerlerinin ayarlanmasından sonra algoritma bir sonsuz döngüye girmektedir. Bundan sonra yapılan işlemler standart "tahmin et – ölç – güncelle" mantığı üzerinden ilerlemektedir. Bu algoritmaya özel her döngü başlangıcında ölçülen referans nokta sayısına bakılmakta ve istenilen değerden düşük olması durumdan yeni referans noktalar sisteme eklenmektedir. Yeni referans eklenmesinin ardından algoritma tahmin etme aşamasına devam eder ve çevrimi tamamlar. Ölçümleri düzenleme kısmı alınan ölçümlerin matris içerisinde uygun yerlere yerleştirilmesi ve ölçülmesi tahmin edilmiş ancak görüntüde bulunamamış referans noktaların silinmesi işlemlerini içerir.



Şekil 4.1 : Görüntü işleme algoritması akış şeması.

4.4.1 Olasılık haritası

SLAM algoritmalarında oluşturulan haritalar, kullanılan sensör ile takip edilebilecek ve referans niteliği taşıyacak olan ortamdaki bazı özelliklerdir. Burada sensör olarak kamera kullanıldığı için genellikle görüntü içerisinde keskin köşe ve kenarlar kullanılmaktadır. Algoritmada kullanılan köşe tanımlaması ekte açıklanmaktadır. Bu çalışmada görüntü içerisindeki köşeler referans noktaları olarak ele alınmış ve etraflarında kalan 21 x 21 piksellik alan da referans resmi olarak alınmıştır.

Görüntüden alınan referans resimleri program çalıştığı süre boyunca takip edilmektedir. Resimler görüntü alanına girdiklerinde ise bir sensör verisi gibi çalışmakta ve Kalman filtresini beslemektedir. Kalman filtresi bu noktalardan aldığı veriler ışığında quadrotorun konumunu günceller. Bu yüzden bir referans resmi kaliteli kılan şeyler, yeryüzü eksen takımına göre sabit olması ve farklı açılardan görülebilir olmasıdır. Aynı zamanda tekrarlı desenlerde karışıklık olabildiği için iyi referans resimleri olmazlar. Kalman filtresi sayesinde bir referans noktasının görüntü

içerisinde hangi alanda olabilmek olasılığı olduğu S matrisi yardımı ile anlaşılmaktadır. Her bir referans noktasının görüntü üzerinde tahmin edilen pozisyonu etrafında bir dikdörtgen oluşturulmakta ve bu referans noktası bu alan içerisinde aranmaktadır. Görüntü işleme açısından arama işlemi maliyetli bir işlem olduğu için bu işlem algoritmanın hızlı çalışmasına büyük katkılar yapmaktadır. Şekil 4.2’de V-SLAM algoritmasının referans noktalar etrafında arama alanını nasıl çizdiği görülebilir. Bu kısım ile ilgili detaylar tahmin etme başlığı altında anlatılacaktır.

Bu çalışmada görüntüde en az 5 görülebilir referans resmi olması sağlanmaya çalışılmıştır. Eğer bu sayı azalırsa program otomatik olarak “Goodfeaturestotrack” algoritmasını kullanarak görüntüdeki keskin köşeleri yakalar ve etrafındaki resmi yeni harita noktası olarak ekler. Bu işlem esnasında başlangıç kısmında anlatılan işlemler programda yürütülür. Quadrotorun hareketine bağlı olarak harita sürekli genişlemelidir. Ancak görülmüştür ki, eğer harita dinamik olarak genişletilirse, programın çalışması yavaşlar ayrıca hafıza sorunları doğar. Bu nedenle program içerisinde haritanın alabileceği maksimum referans resmi sayısı 15 olarak tanımlanmıştır. Bu durumda quadrotor geride bıraktığı kısımdaki harita verilerini siler. Bu, gömülü bilgisayar üzerindeki işlemci gücü ve hafıza azlığı açısından iyidir. Ancak haritanın unutulması ortamın haritasının çıkarılması açısından doğru değildir. Fakat bu çalışmada temel amaç lokalizasyon olduğu için harita önemsenmemektedir.

Harita üzerindeki referans resimler 21 x 21’lik bir matris içerisinde grilik seviyesi cinsinden tutulurken, aynı zamanda sistemin durum vektörü içerisinde de yerlerini alırlar. Kullanılan algorithmada robot durum vektörü:

$$x_v = \begin{bmatrix} r^W \\ q^{WR} \\ v^W \\ \omega^R \end{bmatrix} \quad (4.7)$$

şeklinde tanımlanmaktadır. Burada W harfi yeryüzüne bağlı sabit eksen takımını, R quadrotora bağlı eksen takımını ifade eder. r^W quadrotorun x, y ve z eksenlerindeki doğrusal pozisyonlarını gösterir. q^{WR} quadrotorun açısal pozisyonunu quaternion kullanarak ifade eder. v^W üç eksen için doğrusal hız durumu, ω^R ’de üç eksen için açısal hız durumunu verir. Bu şekilde quadrotorun durumlarını veren vektörü 13 durumlu bir vektör olur.



Şekil 4.2 : V-SLAM referans resimler için arama alanları.

Kalman filtresi içerisinde kullanılan durum vektörü ise her bir referans nokta tahmini de içerecek şekilde oluşturulmuştur. Bu vektörün ilk 13 değeri quadrotoru temsil eder. Daha sonraki değerler referans noktaların durum vektörlerini gösterir. SLAM'ın tanımında söylendiği gibi bu algoritma çalışırken sadece quadrotorun

pozisyon bilgilerini güncellemekle kalmaz aynı zamanda haritayı da güncel tutabilmek için harita noktalarını da günceller.

$$\hat{x} = \begin{bmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{bmatrix} \quad (4.8)$$

Bu vektörde her bir referans noktası içerisinde 6 durumu barındırır.

$$y_{pi} = \begin{bmatrix} r_i^w \\ \widehat{h_i^w} \end{bmatrix} \quad (4.9)$$

r_i^w referans noktasının ilk görüldüğü andaki quadrotor pozisyon bilgisidir. h_i^w ise quadrotorun derinliği ile ilgili bilgiler veren 3 durumlu bir vektördür. Bu vektörle ilgili detaylar uzaklık tahmini bölümünde anlatılacaktır.

4.4.2 Hareket modeli

Kalman filtresi kullanımında en önemli öğelerden bir tanesi hareket modelinin tanımlanmasıdır. Hareket modeli sayesinde sistem bir sonraki adım için durumları tahmin edebime yeteneğine sahip olmaktadır. Aslında bu sistemde quadrotora ait hareket modeli belli olmasına karşın basitleştirme olması amacı ile sabit hız modeli kullanılmıştır. Sabit hız modeli, sistemin hareketini girişlerden bağımsız bir şekilde sabit hızlıymış gibi ele alırken, ivmelenmeleri gürültü olarak sisteme dahil eder. Sabit hız modeli şu şekilde ifade edilir:

$$f_v = \begin{bmatrix} r_{yeni}^w \\ q_{yeni}^{WR} \\ v_{yeni}^w \\ \omega_{yeni}^R \end{bmatrix} = \begin{bmatrix} r^w + (v^w + V^w)\Delta t \\ q^{WR} \otimes q((\omega^R + \Omega^R)\Delta t) \\ v^w + V^w \\ \omega^R + \Omega^R \end{bmatrix} \quad (4.10)$$

Bu model her iterasyon adımında uygulanır ve sistemin bir sonraki adımda hangi pozisyonda olacağı tahmin edilir. Ancak ivmelenmelerin de hesabın içine katılması sistemin sabit hızla çalışacağı kabulünden çıkartır. Bu nedenle ivme yani sistemin bozucusu için kovaryans matrisi aşağıdaki şekilde hesaplanır.

$$Q_v = \frac{\partial f_v}{\partial n} P_n \frac{\partial f_v^T}{\partial n} \quad (4.11)$$

n gürültü vektörüdür ve şöyle tanımlanır:

$$n = \begin{bmatrix} V^W \\ \Omega^R \end{bmatrix} \quad (4.12)$$

Dinamik matrisin hızlara göre türevi aşağıdaki şekilde elde edilir

$$\frac{\partial f_v}{\partial n} = \begin{bmatrix} [I]_{3 \times 3} \Delta t & [0]_{3 \times 3} \\ [0]_{4 \times 3} & \frac{\partial q_3}{\partial q_1} \frac{\partial q(\omega \Delta t)}{\partial \omega} \\ [I]_{3 \times 3} & [0]_{3 \times 3} \\ [0]_{3 \times 3} & [I]_{3 \times 3} \end{bmatrix} \quad (4.13)$$

Burada $\partial q_3 / \partial q_1$ quaternion çarpımının türevidir ve aşağıdaki şekilde çıkar:

$$\frac{\partial q_3}{\partial q_1} = \begin{bmatrix} q(1) & -q(2) & -q(3) & -q(4) \\ q(2) & q(1) & -q(4) & q(3) \\ q(3) & q(4) & q(1) & -q(2) \\ q(4) & -q(3) & q(2) & q(1) \end{bmatrix} \quad (4.14)$$

Aşağıdaki işlem ise quaternion euler dönüşüm işleminden elde edilen sonucun açısal hıza göre türevini vermektedir. Matris içerisindeki ifadelerin açık halleri de verilmiştir.

$$\frac{\partial q(\omega \Delta t)}{\partial \omega} = \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \\ D_1 & D_2 & D_3 \end{bmatrix} \quad (4.15)$$

$$A_1 = \left(\frac{\Delta t}{2}\right) \left(\frac{\omega(1)}{|\omega|}\right) \sin\left(|\omega| * \frac{\Delta t}{2}\right)$$

$$A_2 = \left(\frac{\Delta t}{2}\right) \left(\frac{\omega(2)}{|\omega|}\right) \sin\left(|\omega| * \frac{\Delta t}{2}\right)$$

$$A_3 = \left(\frac{\Delta t}{2}\right) \left(\frac{\omega(3)}{|\omega|}\right) \sin\left(|\omega| * \frac{\Delta t}{2}\right)$$

$$B_1 = \left(\frac{\Delta t}{2}\right) \frac{\omega(1)^2}{|\omega|^2} \left(\left(\frac{\Delta t}{2}\right) \cos\left(|\omega| * \frac{\Delta t}{2}\right) + \frac{1}{|\omega|} \left(1 - \frac{\omega(1)^2}{|\omega|^2}\right) \sin\left(|\omega| * \frac{\Delta t}{2}\right)\right)$$

$$C_2 = \left(\frac{\Delta t}{2}\right) \frac{\omega(2)^2}{|\omega|^2} \left(\left(\frac{\Delta t}{2}\right) \cos\left(|\omega| * \frac{\Delta t}{2}\right) + \frac{1}{|\omega|} \left(1 - \frac{\omega(2)^2}{|\omega|^2}\right) \sin\left(|\omega| * \frac{\Delta t}{2}\right)\right)$$

$$D_3 = \left(\frac{\Delta t}{2}\right) \frac{\omega(3)^2}{|\omega|^2} \left(\left(\frac{\Delta t}{2}\right) \cos\left(|\omega| * \frac{\Delta t}{2}\right) + \frac{1}{|\omega|} \left(1 - \frac{\omega(3)^2}{|\omega|^2}\right) \sin\left(|\omega| * \frac{\Delta t}{2}\right)\right)$$

$$B_2 = \left(\frac{\omega(1)\omega(2)}{|\omega|^2} \left(\frac{\Delta t}{2}\right)\right) \left(\left(\frac{\Delta t}{2}\right) \cos\left(|\omega| \left(\frac{\Delta t}{2}\right)\right) - \frac{1}{|\omega|} \sin\left(|\omega| \left(\frac{\Delta t}{2}\right)\right)\right)$$

$$\begin{aligned}
B_3 &= \left(\frac{\omega(1)\omega(3)}{|\omega|^2} \left(\frac{\Delta t}{2} \right) \right) \left(\left(\frac{\Delta t}{2} \right) \cos \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) - \frac{1}{|\omega|} \sin \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) \right) \\
C_1 &= \left(\frac{\omega(2)\omega(1)}{|\omega|^2} \left(\frac{\Delta t}{2} \right) \right) \left(\left(\frac{\Delta t}{2} \right) \cos \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) - \frac{1}{|\omega|} \sin \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) \right) \\
C_3 &= \left(\frac{\omega(2)\omega(3)}{|\omega|^2} \left(\frac{\Delta t}{2} \right) \right) \left(\left(\frac{\Delta t}{2} \right) \cos \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) - \frac{1}{|\omega|} \sin \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) \right) \\
D_1 &= \left(\frac{\omega(3)\omega(1)}{|\omega|^2} \left(\frac{\Delta t}{2} \right) \right) \left(\left(\frac{\Delta t}{2} \right) \cos \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) - \frac{1}{|\omega|} \sin \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) \right) \\
D_2 &= \left(\frac{\omega(3)\omega(2)}{|\omega|^2} \left(\frac{\Delta t}{2} \right) \right) \left(\left(\frac{\Delta t}{2} \right) \cos \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) - \frac{1}{|\omega|} \sin \left(|\omega| \left(\frac{\Delta t}{2} \right) \right) \right)
\end{aligned}$$

Dinamik matrisin jacobian matrisi EKF içerisinde kullanılacaktır. Bu matris aşağıdaki şekilde elde edilir:

$$\frac{\partial f_v}{\partial x_v} = \begin{bmatrix} [I]_{3 \times 3} & [0]_{3 \times 3} & [I]_{3 \times 3} \Delta t & [0]_{3 \times 3} \\ [0]_{3 \times 3} & \frac{\partial q_3}{\partial q_2} & [0]_{3 \times 3} & \frac{\partial q_3}{\partial q_1} \frac{\partial q(\omega \Delta t)}{\partial \omega} \\ [0]_{3 \times 3} & [0]_{3 \times 3} & [I]_{3 \times 3} & [0]_{3 \times 3} \\ [0]_{3 \times 3} & [0]_{3 \times 3} & [0]_{3 \times 3} & [I]_{3 \times 3} \end{bmatrix} \quad (4.15)$$

Bu denklemde $\frac{\partial q_3}{\partial q_2}$ ifadesi ise quaternion çarpımında birinci ifadeye göre türev alınması demektir ve aşağıdaki şekilde ifade edilir:

$$\frac{\partial q_3}{\partial q_2} = \begin{bmatrix} q(1) & -q(2) & -q(3) & -q(4) \\ q(2) & q(1) & q(4) & -q(3) \\ q(3) & -q(4) & q(1) & q(2) \\ q(4) & q(3) & -q(2) & q(1) \end{bmatrix} \quad (4.16)$$

4.4.3 Kamera modeli

Bu çalışmada standart iğne deliği (pinhole) kamera modeli kullanılmıştır. Kalibrasyon Caltech kalibrasyon aracı ile yapılmıştır [27]. Bu kameranın modeli

$$h_i = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 - f k_u \frac{h_{Lx}^R}{h_{Lz}^R} \\ v_0 - f k_v \frac{h_{Ly}^R}{h_{Lz}^R} \end{bmatrix} \quad (4.17)$$

şeklinde olup, u_0 ve v_0 kamera merkezlerini piksel cinsinden ifadesidir. F_{ku} ve f_{kv} fokal uzunluklar olup standart kamera kalibrasyon parametreleridir. Bu çalışmada distorsiyon modeli, parametrelerin çok küçük çıkması ve basitleştirme nedeniyle modele dahil edilmemiştir. h_L^R kamera görüntüsüne düşmesi beklenen noktanın ilgili eksenindeki uzaklığıdır.

4.4.4 Başlangıç

Başlangıç durumu sistemde iki durum için tanımlanmaktadır. Bunlar global ve lokal başlangıç durumları olarak adlandırılabilirler. Global başlangıç durumu sistem ilk çalışmaya başladığı anda tüm değişkenleri ilgilendiren bir durumken, lokal başlangıç durumu sisteme yeni bir referans noktası eklenildiğinde değişkenlerin belirli bir kesiminde oluşan değişiklikleri tanımlamak için kullanılabilir.

Global başlangıç aşamasında hafızada tutulacak değişkenler için yer açılırken aynı zamanda durum değişkenleri ve kovaryans matrisi için de ilk değerler belirlenir. Durum matrisinin başlangıç hali aşağıdaki gibidir:

$$x_{vi} = \begin{bmatrix} [0]_{3 \times 1} \\ 1 \\ 0 \\ 0 \\ 0 \\ [2.2 \cdot 10^{-16}]_{3 \times 1} \\ [2.2 \cdot 10^{-16}]_{3 \times 1} \end{bmatrix} \quad (4.18)$$

Pozisyon değerleri ve oryantasyon başlangıç anında 0 kabul edilirken doğrusal ve açısal hız değerleri için sıfıra çok yakın fakat sıfırdan farklı bir değer verilmektedir. Diğer yandan P matrisi ise:

$$P_i = \begin{bmatrix} 2.2 \cdot 10^{-16} [I]_{3 \times 3} & [0]_{3 \times 4} & [0]_{3 \times 3} & [0]_{3 \times 3} \\ [0]_{4 \times 3} & 2.2 \cdot 10^{-16} [I]_{4 \times 4} & [0]_{4 \times 3} & [0]_{4 \times 3} \\ [0]_{3 \times 3} & [0]_{3 \times 4} & [1]_{3 \times 3} & [0]_{3 \times 3} \\ [0]_{3 \times 3} & [0]_{3 \times 4} & [0]_{3 \times 3} & [0.5]_{3 \times 3} \end{bmatrix} \quad (4.19)$$

Şeklinde tanımlanır. Çalışmada kullanılacak referans nokta sayısı 15 olduğu için durum vektörü boyu

$$l = 13 + 15 \times 6 = 103 \quad (4.20)$$

şeklinde elde edilir. Aynı şekilde kovaryans matrisinin de boyutu 103x103'tür. Elde edilen matrisler sıfırlar ile oluşturulan bu matrislerin içine yerleştirilir:

$$\hat{x} = \begin{bmatrix} [x_{vi}]_{13 \times 1} \\ [0]_{90 \times 1} \end{bmatrix} \quad (4.21)$$

$$P = \begin{bmatrix} [P_i]_{13 \times 13} & [0]_{13 \times 90} \\ [0]_{90 \times 13} & [0]_{90 \times 90} \end{bmatrix} \quad (4.22)$$

Bu işlemler yapıldıktan sonra sisteme ilk beş referans nokta eklenir.

Lokal başlangıç olarak tanımlayabileceğimiz bir referans nokta eklenmesi esnasında yapılan değişiklikler ise yine bu iki matris üzerindedir. Eğer sistem yeni başlatılmışsa 15 referans noktası henüz dolmamıştır ve yeni bulunan referans noktası buraya yazılabilir. Ancak eğer tüm referans noktaları kullanılıyor ve yeni bir referans noktası eklenmesi gerekiyor ise önceki referans noktalarından o ana kadar en güvenilir olanı çıkartılır ve yeni referans noktası oraya yazılır. En güvenilir referans noktasına karar vermek için her sistemin çalıştığı her karede referans noktaları için puanlar tutulur. Eğer referans noktası görülemeyecek bir yerdeyse ve de kameradan görünmüyorsa 1, eğer görüntü içerisinde kalacağı tahmin edilmiş ve görülmüyorsa 10 puan verilir. Yeni referans noktası eklenirken hangi nokta daha çok puan toplamışsa o çıkarılır.

Referans noktası eklendiğinde durum matrisinde yapılacak değişiklikler şu şekilde bulunur:

$$h_L^R = \begin{bmatrix} \frac{u_0 - u}{f_{ku}} \\ \frac{v_0 - v}{f_{kv}} \\ 1 \end{bmatrix} \quad (4.22)$$

Quaternionlar ile rotasyon matrisi R^{RW} elde edilir. Referans noktasının yüzey normali:

$$n = R^{RW} h_L^R \quad (4.23)$$

şeklinde ifade edilir. Eklenecek 6 elemanlı vektör şu şekildedir:

$$y_{pi} = \begin{bmatrix} r^W(1,1) \\ r^W(2,1) \\ r^W(3,1) \\ \tan^{-1}\left(\frac{nx}{nz}\right) \\ \tan^{-1}\left(-\frac{ny}{nx^2 + nz^2}\right) \\ \rho \end{bmatrix} \quad (4.24)$$

Kovaryans matrisinde yapılacak deęişiklikler için ise aşığıdaki işlemler yapılır. n vektörünün her bir elemanı gerçek dünya koordinatlarında referans nokta ile bu noktanın ilk görüldüğü andaki kamera pozisyonu arasındaki mesafeyi belirttiğı için bu elemanlara x_w, y_w, z_w diyebiliriz. Öncelikle θ ve φ (azimuth ve elevation açıları) açısal pozisyona göre türetilirse, zincir kuralı yardımıyla:

$$\frac{\partial \theta}{\partial q_{wr}} = \frac{\partial \theta}{\partial g_w} \frac{\partial g_w}{\partial q_{wr}} \quad (4.25)$$

$$\frac{\partial \varphi}{\partial q_{wr}} = \frac{\partial \varphi}{\partial g_w} \frac{\partial g_w}{\partial q_{wr}}$$

Bu denklemlerde $g_w [x_w \ y_w \ z_w]$ vektörüne karşılık gelmektedir. Yukarıdaki türev ifadeleri açık biçimde:

$$\frac{\partial \theta}{\partial g_w} = \begin{bmatrix} \frac{z_w}{x_w^2 + z_w^2} \\ 0 \\ -\frac{x_w}{x_w^2 + z_w^2} \end{bmatrix} \quad (4.26)$$

$$\frac{\partial \varphi}{\partial g_w} = \begin{bmatrix} \frac{x_w y_w}{x_w^2 + y_w^2 + z_w^2 + \sqrt{x_w^2 + z_w^2}} \\ -\frac{\sqrt{x_w^2 + z_w^2}}{x_w^2 + y_w^2 + z_w^2} \\ \frac{z_w y_w}{x_w^2 + y_w^2 + z_w^2 \sqrt{x_w^2 + z_w^2}} \end{bmatrix}$$

elde edilir. Ayrıca:

$$\frac{\partial g_w}{\partial q_{wr}} = \begin{bmatrix} \frac{dR}{dq_0} XYZ_c & \frac{dR}{dq_x} XYZ_c & \frac{dR}{dq_y} XYZ_c & \frac{dR}{dq_z} XYZ_c \end{bmatrix} \quad (4.27)$$

şeklinde ifade edilir. Rotasyon matrisi R 'nin her bir quaternion elemanına göre türevi quaternion rotasyon matrisi dönüşüm işleminin türetilmesi ile elde edilir.

$$\frac{\partial R}{\partial q_o} = \begin{bmatrix} 2q(1) & -2q(4) & 2q(3) \\ 2q(4) & 2q(1) & -2q(2) \\ -2q(3) & 2q(2) & 2q(1) \end{bmatrix} \quad (4.28)$$

$$\frac{\partial R}{\partial q_x} = \begin{bmatrix} 2q(2) & 2q(3) & 2q(4) \\ 2q(3) & -2q(2) & -2q(1) \\ 2q(4) & 2q(1) & -2q(2) \end{bmatrix}$$

$$\frac{\partial R}{\partial q_y} = \begin{bmatrix} -2q(3) & 2q(2) & 2q(1) \\ 2q(2) & 2q(3) & -2q(4) \\ -2q(1) & 2q(4) & -2q(3) \end{bmatrix}$$

$$\frac{\partial R}{\partial q_z} = \begin{bmatrix} -2q(4) & -2q(1) & 2q(2) \\ 2q(1) & -2q(4) & 2q(3) \\ 2q(2) & 2q(3) & 2q(4) \end{bmatrix}$$

Yukarıdaki denklemler sonucunda artık referans noktası vektörünün hava aracına göre türevlerini yazmak mümkündür:

$$\frac{\partial y}{\partial q_{wr}} = \begin{bmatrix} [0]_{3 \times 4} \\ \frac{\partial \theta}{\partial q_{wr}} \\ \frac{\partial \varphi}{\partial q_{wr}} \\ [0]_{1 \times 4} \end{bmatrix} \quad (4.29)$$

$$\frac{\partial y}{\partial r_w} = \begin{bmatrix} [I]_{3 \times 3} \\ [0]_{3 \times 3} \end{bmatrix}$$

$$\frac{\partial y}{\partial x_v} = \begin{bmatrix} \frac{\partial y}{\partial r_w} \\ \frac{\partial y}{\partial q_{wr}} \\ [0]_{6 \times 6} \end{bmatrix}$$

Ölçüm modeline göre türev:

$$\frac{\partial y}{\partial gw} = \begin{bmatrix} [0]_{3 \times 3} \\ \frac{\partial \theta}{\partial gw} \\ \frac{\partial \varphi}{\partial gw} \\ \frac{\partial \psi}{\partial gw} \end{bmatrix} \quad (4.30)$$

$$\frac{\partial gw}{\partial gc} = R_{wc}$$

$$\frac{\partial gc}{\partial h} = \begin{bmatrix} -\frac{1}{f_{ku}} & 0 \\ 0 & -\frac{1}{f_{kv}} \\ 0 & 0 \end{bmatrix}$$

$$\frac{\partial y}{\partial h} = \begin{bmatrix} \frac{\partial y}{\partial gw} \frac{\partial gw}{\partial gc} \frac{\partial gc}{\partial h} & 0 \\ 0 & 1 \end{bmatrix}$$

şeklinde elde edilir. Bu noktada kovaryans matrisinin hangi bölgesinde değişiklik yapacağımıza bakmamız gerekiyor. P_i kovaryans matrisinin ilk 13 x 13'lük bölümü olmak üzere değiştirmek istediğimiz referans noktası 5. nokta ise:

$$P = \begin{bmatrix} P_i & P_{1i} & P_{i2} & \dots & P_{i5} & \dots & P_{i15} \\ P_{i1} & P_{11} & P_{12} & \dots & P_{15} & \dots & P_{115} \\ P_{2i} & P_{21} & P_{22} & \dots & P_{25} & \dots & P_{215} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{5i} & P_{51} & P_{52} & \dots & P_{55} & \dots & P_{515} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{15i} & P_{151} & P_{152} & \dots & P_{155} & \dots & P_{1515} \end{bmatrix} \quad (4.31)$$

matris üzerinde boya ile gösterilmiş kısımlar üzerinde değişiklik yapılmak istenir. Satırdaki değişiklikler için

$$P_{str} = \frac{\partial y}{\partial x_v} P_i \quad (4.32)$$

uygulanır. Sütun için ise bunun transpozu alınır. Kesişim noktası olan P_{55} için ise

$$P_{ksm} = \frac{\partial y}{\partial x_v} P_i \frac{\partial y^T}{\partial x_v} + \frac{\partial y}{\partial h} P_{ek} \frac{\partial y^T}{\partial h} \quad (4.33)$$

bulunur, T transpozu ifade eder ve P_{ek} ise

$$P_{ek} = \begin{bmatrix} R_i & 0 \\ 0 & \rho^2 \end{bmatrix} \quad (4.34)$$

Şeklinde verilir ve R_i tüm referans noktaları için 1, ρ^2 ise 0.25 alınmıştır.

4.4.5 Tahmin etme

Kalman filtresinin döngü içerisindeki ilk adımı durum vektörünü ve kovaryans matrisini tahmin etmektir. Bu tahminler sayesinde referans elemanının hava aracına göre konumu tayin edilir. Bu iki noktada avantaj sağlar: eğer referans noktası kamerada görünmeyecekse bu referans nokta hiç aranmaz ve bu döngü süresince başka hiçbir işleme katılmaz, eğer referans noktası kamera tarafından görüntülenebiliyorsa bu nokta o referans noktası için kamera üzerinde oluşacak gauss dağılımının ortalama noktası kabul edilir. Bu kısım içinde anlatılacak olan S matrisinin yardımı ile de gauss sınırları belirlenir ve bir referans noktasını arayacağımız sınırlar ortaya çıkar. Bu yaklaşım görüntü işleme açısından büyük hız kazanımlarını sağlar. Eğer bir referans noktasını arayacağımız alan genişliyoorsa, hava aracının konumu hakkında bilgimiz kesinliğini yitiriyor demektir.

Durum vektörü için tahminler aşağıdaki şekilde elde edilir:

$$x_{(k|k-1)} = \begin{bmatrix} r_w + v_w \Delta t \\ q_{wr} \otimes q(v_w \Delta t) \\ v_w \\ \omega_w \\ x_{(k|k)}(14:103) \end{bmatrix} \quad (4.35)$$

Bu vektörde Δt döngü süresini gösterir. Yeni açısıl konum quaternion çarpımı ile gerçekleştirilir ve $\otimes$ işareti ile gösterilir. Sistem için kabulümüz, tüm referans noktaların sabit kalacakları yönünde olduğu için bu noktalar önceki pozisyonlar ile eşit tutulur.

Kovaryans matrisi tahmini ise dinamik matrisin jacobian'ı yardımı ile bulunur.

$$\frac{\partial f_v}{\partial y} = \begin{bmatrix} \frac{\partial f_v}{\partial x_v} & [0]_{13 \times 45} \\ [0]_{45 \times 13} & [I]_{45 \times 45} \end{bmatrix} \quad (4.36)$$

$$P_{(k|k-1)} = \frac{\partial f_v}{\partial y} P_{(k-1|k-1)} \frac{\partial f_v^T}{\partial y}$$

Ölçümlere ait tahminlerde bulunabilmek için ölçüm matrisi H'nin oluşturulması gereklidir. EKF kullanımı nedeni ile H matrisi de daha önce verildiği gibi jacobian ile oluşturulacaktır.

$$H = \begin{bmatrix} \frac{\partial h}{\partial x_v} & \frac{\partial h}{\partial y} & [0]_{2 \times 6} & [0]_{2 \times 6} & \dots \\ \frac{\partial h}{\partial x_v} & [0]_{2 \times 6} & [0]_{2 \times 6} & \frac{\partial h}{\partial y} & \dots \end{bmatrix} \quad (4.37)$$

Matrisin elemanları açık formda aşağıdaki gibidir:

$$\frac{\partial h}{\partial x_v} = \begin{bmatrix} \frac{\partial h}{\partial r_w} & \frac{\partial h}{\partial q_{wr}} & [0]_{2 \times 6} \end{bmatrix} \quad (4.38)$$

$$\frac{\partial h}{\partial r_w} = \frac{\partial h}{\partial h_{rl}} \frac{\partial h_{rl}}{\partial r_w}$$

$$\frac{\partial h}{\partial h_{rl}} = \begin{bmatrix} -\frac{f k_u}{h c_z} & 0 & h c_x f \frac{k_u}{h c_z^2} \\ 0 & -f \frac{k_v}{h c_z} & \frac{h c_y f k_v}{h c_z^2} \end{bmatrix}$$

Referans nokta için paralaks parametreleri

$$\theta = y(4), \varphi = y(5), \rho = y(6) \quad (4.39)$$

$$m_i = \begin{bmatrix} \cos(\varphi) \sin(\theta) \\ -\sin(\varphi) \\ \cos(\varphi) \cos(\theta) \end{bmatrix}$$

Referans noktanın konumu:

$$hc = R_{rw}((y_i - rw)\rho + m_i) \quad (4.40)$$

referans noktanın bileşenleri:

$$hc_x = hc(1), hc_y = hc(2), hc_z = hc(3) \quad (4.41)$$

kamera ile referans nokta arasındaki mesafenin kamera hareketine göre türevi:

$$\frac{\partial h_{rl}}{\partial r_w} = -T_{qr}^{-1} \rho \quad (4.42)$$

$$T_{qr1} = \begin{bmatrix} q(1)^2 + q(2)^2 - q(3)^2 - q(4)^2 \\ 2(q(2)q(3) + q(1)q(4)) \\ 2(q(4)q(2) - q(1)q(3)) \end{bmatrix}$$

$$T_{qr2} = \begin{bmatrix} 2(q(2)q(3) - q(1)q(4)) \\ q(1)^2 - q(2)^2 + q(3)^2 - q(4)^2 \\ 2(q(3)q(4) + q(1)q(2)) \end{bmatrix}$$

$$T_{qr3} = \begin{bmatrix} 2(q(4)q(2) + q(1)q(3)) \\ 2(q(3)q(4) - q(1)q(2)) \\ q(1)^2 - q(2)^2 - q(3)^2 + q(4)^2 \end{bmatrix}$$

$$T_{qr} = [T_{qr1} \quad T_{qr2} \quad T_{qr3}]$$

şeklindedir. Ölçüm matrisinin açılal pozisyona göre türevi zincir kuralına göre:

$$\frac{\partial h}{\partial q_{wr}} = \frac{\partial h}{\partial h_{rl}} \frac{\partial h_{rl}}{\partial q_{wr}} \quad (4.43)$$

$$\frac{dh_{rl}}{dq_{wr}} = \begin{bmatrix} \frac{dR}{dq_0} h_c & \frac{dR}{dq_x} h_c & \frac{dR}{dq_y} h_c & \frac{dR}{dq_z} h_c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Ölçüm matrisinin referans noktaya göre türevi:

$$\frac{\partial h}{\partial y} = \frac{\partial h}{\partial h_{rl}} \frac{\partial h_{rl}}{\partial y} \quad (4.44)$$

$$\frac{\partial m_i}{\partial \theta_i} = R_{rw} \begin{bmatrix} \cos(\varphi) \cos(\theta) \\ 0 \\ -\cos(\varphi) \sin(\theta) \end{bmatrix}$$

$$\frac{\partial m_i}{\partial \varphi_i} = R_{rw} \begin{bmatrix} -\sin(\varphi) \sin(\theta) \\ -\cos(\varphi) \\ -\sin(\varphi) \cos(\theta) \end{bmatrix}$$

$$\frac{\partial h_{rl}}{\partial y} = \left[\rho * R_{rw} \quad \frac{\partial m_i}{\partial \theta_i} \quad \frac{\partial m_i}{\partial \varphi_i} \quad R_{rw}(y_i - r_w) \right]$$

şeklinde ifade edilir. Yukarıda anlatılan denklemler kullanılarak ölçüm matrisinin jacobian matrisi bulunabilir.

Ölçüm tahminleri aşağıdaki şekilde elde edilir:

$$h_{rl} = R_{cw}([y_i - r_{wc}]\rho + m_i) \quad (4.45)$$

R_{cw} açışal pozisyonun rotasyon matrisine çevrilmiş halidir. Gerçek koordinatları bilinen bir noktanın kamera üzerinde nereye düşeceğini bulmakta kullanılır. Bu matrisin tersi de kamera üzerindeki bir noktanın gerçek koortinalardaki yerini bulmakta kullanılır.

$$R_{cw} = R_{wc}^{-1} \quad (4.46)$$

$$r_{wc} = x_{v(k|k-1)}(1:3)$$

Noktanın kameranın önünde olup olmadığına bakılır.

$$h_{rl}(3) < 0 \text{ ve } \rho > 0 \quad (4.47)$$

şartı sağlanmışsa, bu referans nokta için yapılan işlemler devam eder, şart sağlanmıyorsa, bu nokta artık ölçüm olarak ifade edilmez.

4.4.6 Ölçüm alma

Kameradan alınan ölçümler, referans noktalarının tanınip x ve y eksenindeki uzaklıklarının bulunması ile gerçekleşir. Bu noktaları tanıyabilmek için eşleştirme algoritmaları kullanılmaktadır. OpenCV kütüphanesi içerisinde farklı yöntemlerdeki eşleştirme algoritmalarını biraraya toplayan “cvmatchtemplate” komutu bulunmaktadır. Bu komutla ilgili detaylı bilgi ekte mevcuttur.

4.4.7 Durum vektörünü güncelleme

Alınan ölçümlere göre hangi referans noktaları bulunabilmişse, sadece bunlarla alakalı parametreler elde tutulmakta diğerleri elenmektedir. Eşleşen referans noktalarının durumları ölçüm bilgileri ve kovaryansları kullanılarak (4.6) denklemlerinde verilen işlemler ile yeni durum ve kovaryans matrisi belirlenir.

4.4.8 Uzaklık tahmini

Tek kamera ile lokalizasyon yaparken en önemli problemlerden birisi, etrafımızdaki cisimlerin kameradan ne kadar uzakta olduklarını bilememektir. Çift kameralı sistemlerde bu bilgi insan gözünde olduğu gibi doğrudan elde edilebilir. Ancak hava araçlarında her ağırlığın, enerji anlamında bir maliyet ve taşıma kapasitesinde düşüş anlamlarına gelmesinden dolayı, ikinci kamera kullanımından kaçınılmıştır.

Tek kameralı sistemlerde, görüntüdeki bir cismin kameraya olan uzaklığını hesaplamak için kamera hareketinden yararlanılır. V-SLAM algoritmasında derinlik bilgisi paralax kullanımı ile gerçekleştirilmiştir. Paralaks, kamera hareket halindeyken, kameraya uzak cisimlerin yakın cisimlere göre görüntü üzerinde daha yavaş hareket etmeleri şeklinde açıklanabilir.

Sisteme yeni eklenen bir referans nokta 6 adet durum vektörü getirir. Bu durum vektörünün ilk üç elemanı bu referans noktanın ilk eklendiği andaki hava aracı pozisyonudur. Diğer üç parametre ise ufuk (azimuth), bakış (elevation) ve açısı derinliğin çarpma işlemine göre tersidir.

Başlangıçta yeni eklenen bir referans noktasının derinliğinin tersi 0.1 olarak eklenmektedir. Bu yöntemle göre kameranın ortasından çıkan bir yarı sonsuz doğru parçası referans noktanın olabileceği tüm derinlikleri gösterir. Verilen derinlik varyans değerine göre (0.5) ilk durumda referans noktasının 1 metre ile sonsuz arasında olduğu kabul edilir. Sonraki gelen karelerde varyans değeri küçülmemekte ve derinlik değeri net olarak ortaya çıkmaktadır. Burada kullanılan yöntem, standart XYZ kullanımına göre fazladan üç parametre getirmesine karşılık derinlik tahmininde daha başarılı olmaktadır [27].

4.4.9 Diğer çalışmalar ile farklar

Bu bölümde anlatılan tüm formüller [4]'te verilen makalede açıklanan sistem üzerinden anlatılmıştır. Ancak açık formda verilen bazı ifadeler, literatürdeki

alıřmalarda aık řekilde verilmemektedir. Bu formller, [4]'n yazarları tarafından paylařılan kodlardan elde edilmiřtir.

Gerekleme kısmında [4] yazarları MonoslamGlow isimli bir ktphane ıkartmıřtır. Ancak bu ktphane Oxford niversitesi grnt iřleme laboratuvarı tarafından geliřtirilen  ktphanenin imkanlarını kullanmaktadır. Bu ktphanelerin uygun versiyonlarını bulmak ve derlemek sıkıntılar doęurmaktadır. Bu nedenle, dkmantasyonu geniř olması nedeniyle sadece opencv ktphanesi yardımıyla kod tekrar yazılmıřtır. Kod OpenCV ile yazılmasının bir dięer nedeni bu řekilde geliřtirilmeye daha aık olacaęının dřnlmesidir.

Kod gerekleme iřleminde [4]'te yapılan ancak bu alıřmanın amaları dıřında kaldıęı iin koddan ıkarılan bazı blmler vardır. Bunlardan en nemlisi, referans noktaların yzey normallerinin tam olarak saptanmaya alıřılmasıdır. Orijinal kod daha ok haritalamaya nem verdięinden dolayı bu yzey doęrultularını hesaplamaya alıřmaktadır. Ancak burada kullanılan algoritmada yzey normalleri nemli grlmemiřtir.

Kamera modeli ve sistem bařlangıcı kısmı daha basitleřtirilmiřtir. [4] kamerada distorsiyon modeli kullanmasına karřın burada distorsiyon modeli yoktur. Sistem bařlangıcı iin makalede aılıřta bilinen bir resmin kamera karřısında konulması saęlanmış ve leklendirme buna gre yapılmıřtır. Ancak hava aracı zerinde sistemin aısal olarak sıfır pozisyonunda bařlatıldıęı kabul vardır.

5. DENGELEME DENEYİ

5.1 Giriş

Hava aracının kontrolü ataletsel ölçüm ünitesinden aldığı bilgilere dayanarak yapılmaktadır. Ancak kontrol değişkenleri olan açısal pozisyonlar ve yükseklik değeri yukarıda anlatılan algoritma yardımıyla elde edilebilir. IMU'dan bağımsız bir şekilde uçuş denemesi başarıyla gerçekleştirilmiştir [29]. Çalışmada bir tüm yönlü kamera ve bu kameranın görebileceği düz çizgiler çizilmiştir. Hava aracının uçuş senaryosu yoğun nüfusu olan yerleşim yerlerinde navigasyon yapılırken, binaların oluşturacağı çizgilerin hava aracı tarafından algılanıp, dengesini sağlayabilmesidir.

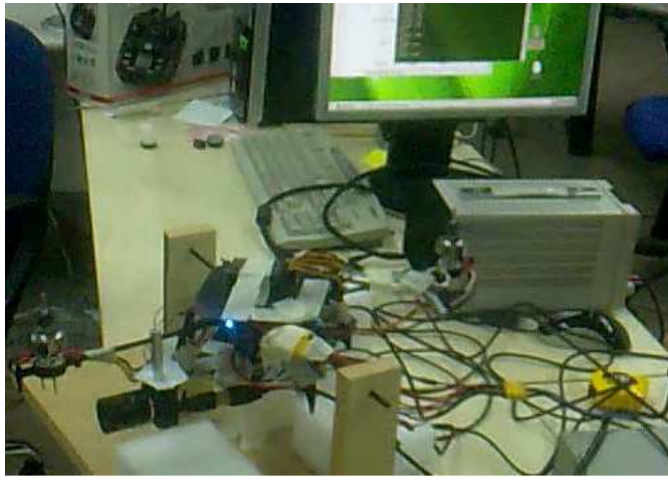
Bu çalışmada hava aracı daha önceden bilinmeyen ortam koşulları içinde sadece görüntü işlemeyi sensör olarak kullanarak açısal pozisyon ve yükseklik kontrolünü yapması amaçlanmıştır. Kamera görüntüsünü etkileyecek, karanlık, sis ve duman gibi şartların olmadığı kabul edilmiştir. Hava aracının kapalı alanlar içinde çalışacağı düşünülmüştür.

Görüntü işleme sistemi dördüncü bölümde anlatılan denklemler ile gerçekleşmiştir. Visual Studio ortamı bu amaçla kullanılmış ve programlama dili olarak C++ kullanılmıştır. Program yazılırken bazı görüntü işleme (görüntü eşleştirme ve hızlı köşe bulma) fonksiyonlarının hazır kullanılabilmesi ve matris oluşturulabilmesi için OpenCV 2.1 kütüphanesi kullanılmıştır. Elde edilen algoritma bilgisayar ortamında (Core2Duo işlemcili) 25 FPS hızında çalışmaktadır.

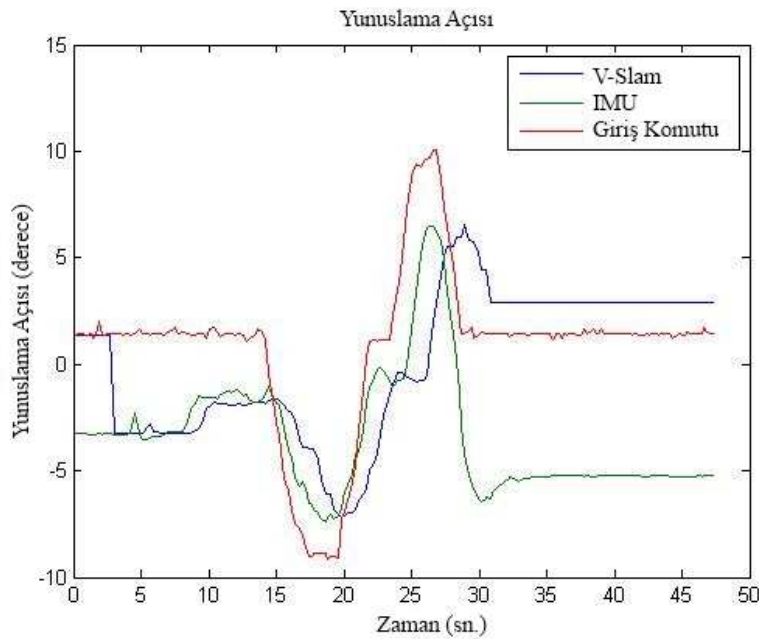
Deneylerde görüntü işleme algoritması oryantasyonun algılanmasını sağlayacak bir sensör gibi çalışacağı için, hava aracına referans vermeden önce, el ile hava aracının açı değerler değiştirilmiş ve IMU ile değerler karşılaştırılmıştır. Daha sonra IMU devreden çıkarılarak, otopilotun görüntü işlemeden gelen açı değerlerini kullanması sağlanmıştır.

5.2 Deney Düzeneği

Hava aracının sağlıklı bir şekilde bu deneye sokulabilmesi ilk deneyler Şekil 5.1’de gösterilen tek eksenli deney düzeneği üzerinde yapılmıştır. Bu deney düzeneği hava aracı iki motoru çıkarılarak karşılıklı yerleştirilmiş iki blok arasına tek eksenle dönebilecek şekilde takılmaktadır. Bu deney düzeneğinin amacı, hava aracının tek bir eksenle (yunuslama veya yalpalama) kontrolünün sınanmasıdır. Bu deneye ait sonuçlar Şekil 5.2’de görülmektedir. Algoritma ilk 30 saniye içerisinde kumandadan gönderilen referans takip edilmeye çalışılmıştır. V-SLAM algoritması 30. saniyede durdurulduğu için buradan sonrası düz çizgi şeklindedir.



Şekil 5.1 : Tek eksenli deney düzeneği ve hava aracı.



Şekil 5.2 : Tek eksenli deney düzeneği üzerinde yunuslama hareketi.

Üç boyutlu deney için Şekil 5.3’de görülen deney düzeneği kullanılmıştır. Bu deney düzeneği üç eksenli dönme hareketine izin verebilecek bir ekleme sahiptir. Yunuslama ve yalpalama ekseninde hareket -20 ile +20 derece arasında sınırlandırılırken, yönelme ekseninde sonsuz bir dönüş mevcuttur.



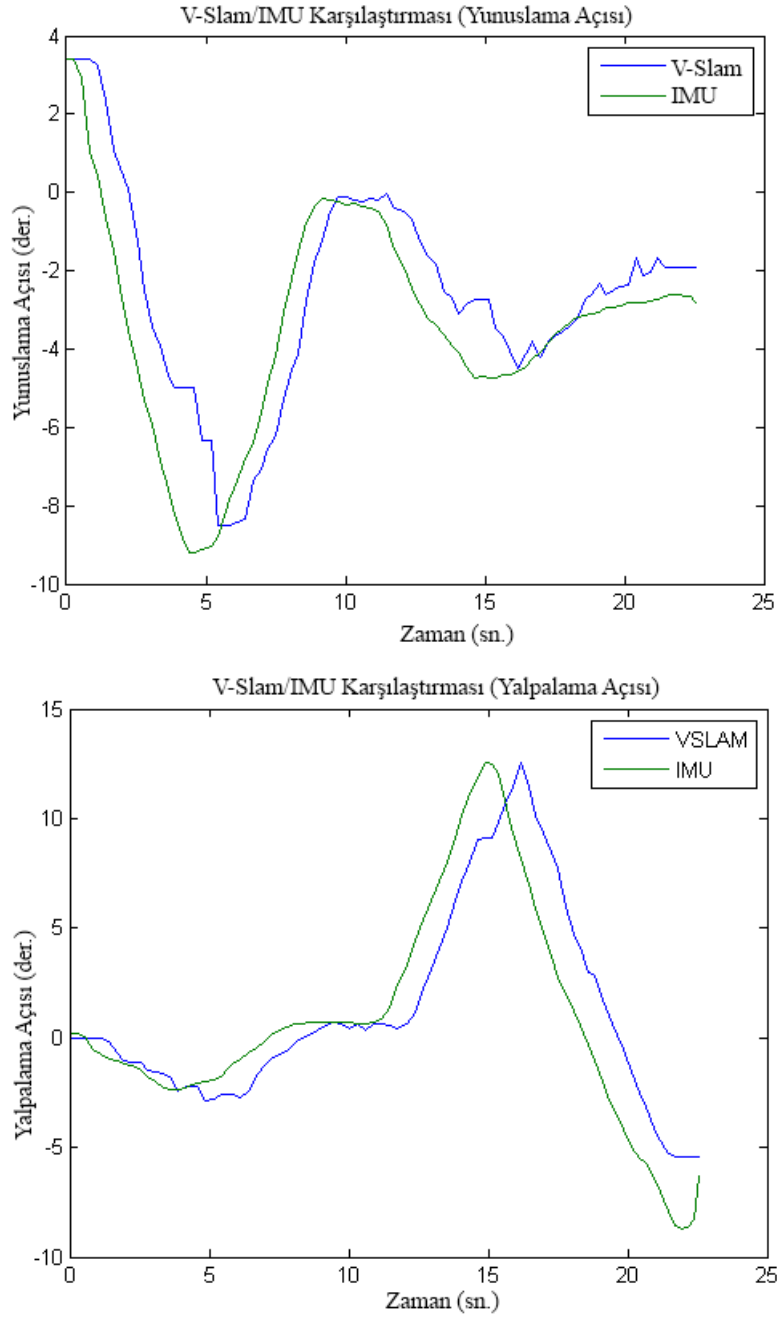
Şekil 5.3 : Dengeleme deney düzeneği.

Deney düzeneği üzerinde sistem modeli değişmesine karşın, PID ayarları tekrar yapılarak, görüntü işleme sisteminin çalışması kontrol edilebilmektedir.

5.3 Sonuçlar

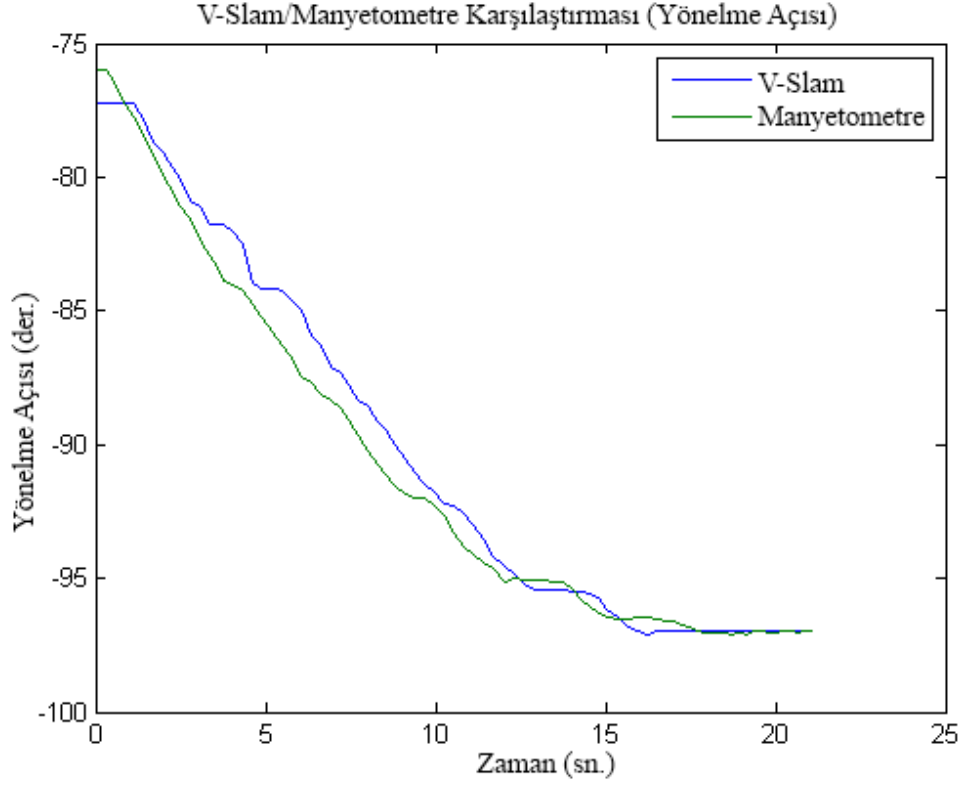
İlk olarak hava aracı deney düzeneği üzerine bağlanmış ve el ile hareketler verilerek IMU ile V-SLAM algoritması karşılaştırılmıştır. Beagleboard üzerinde çalışan görüntü işleme kodu, her bir durum güncellemesi yaptığı anda ardupilot’a açılış değerlerini gönderir. Bu işlem yaklaşık olarak 100 ms’de bir tekrarlanır. Aşağıda

(Şekil 5.4) el ile hareket ettirilmiş hava aracına ait IMU ve V-SLAM değerleri görülmektedir.



Şekil 5.4 : Yunuslama ve yalpalama açılarına ait zaman-açı grafikleri.

Yönelme açısı için durum yunuslama ve yalpalama açılarından farklıdır. Bu açının doğrulama ölçümü için magnetometre kullanılmıştır. Bu deneye ait sonuçlar Şekil 5.5'te verilmiştir.



Şekil 5.5 : Yönelme açısına ait zaman-açı grafiği.

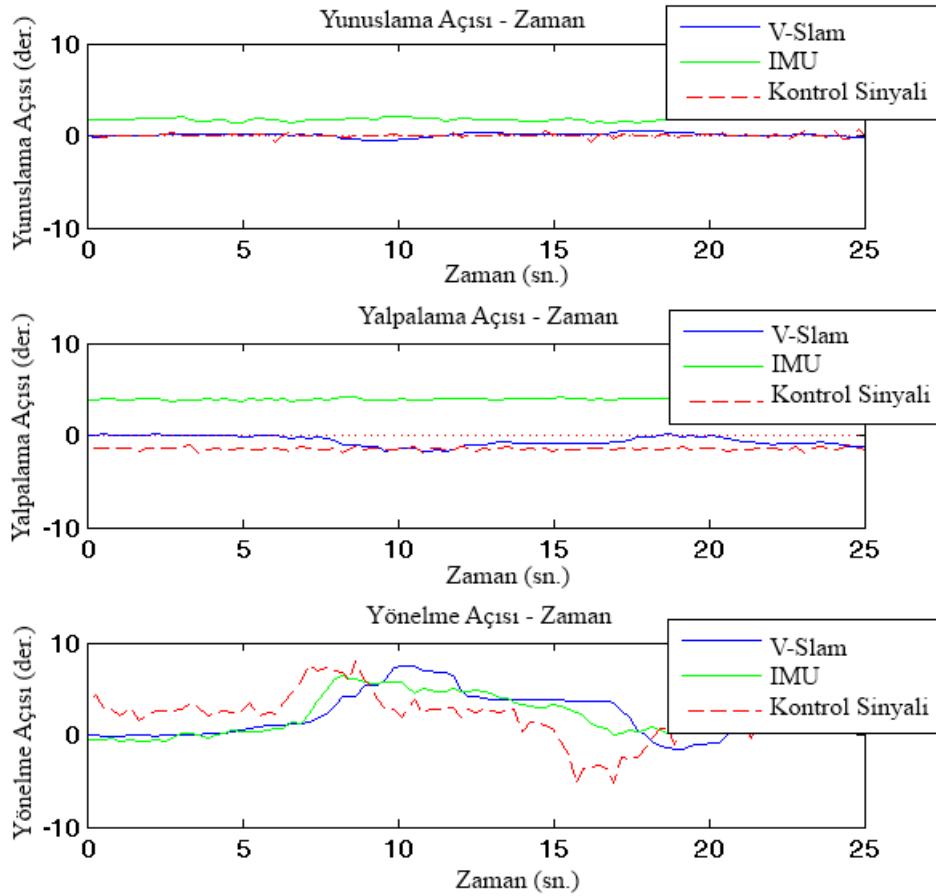
Bu deneylerden sonra yunuslama açısı hatası ortalama değer 0.80 ve standart sapma 1.53 ve yalpa açısı hatası için ortalama değer 0.44 ve standart sapma 2.36 derece olarak bulunmuştur. Bu hataların en büyük nedeni yaklaşık 1 saniyeye varan gecikme süresidir. Bu gecikmenin kamera beagleboard arasında ve beagleboard ardupilot arasındaki iletişim esnasında olduğu düşünülmektedir.

Ardupilot kodunda yapılan değişiklik ile quadrotorun beagleboard tarafından gelen açı değerleri ile açısal konum kontrolü yapması sağlanmıştır. Bu durumda IMU üzerinden ölçülen değerler sadece doğruluk tablosu olarak kullanılacaktır. Bu deneyde kumanda ile referans değerler ardupilot'a ulaştırılmıştır. İstenilen referans değerleri beagleboard'dan gelen değerler ile karşılaştırılarak kontrol sinyali üretilmiştir. Kullanılan pid değerleri bu deney düzeneği için daha önceden elde edilen PID değerleri ile aynıdır.

V-SLAM algoritmasında en önemli noktalardan birisi, algoritma mutlak (absolute) çalışmadığı için başlangıç noktası çok önemlidir. Sistemin mutlak çalışabilmesi için sürekli göreceği bir nesnenin olması referans sağlayabilirdi ancak, mevcut durumda hava aracı etrafındaki objeler önceden düzenlenmemiş ve bilinmemektedir. Ayrıca

sürekli görebileceği bir nesne de yoktur. Bu nedenle başlangıç noktası için basit bir şekilde IMU değerlerine bakılmış ve sistem açılışına kullanıcının klavye yardımı ile o anki IMU değerlerini Beagleboard'a aktarmasını sağlayan bir kod eklenmiştir. Bu sayede IMU ve V-SLAM kodlar açılışta eşitlenmiş olur.

Deneyler sonucunda dengeli uçuşa yakın durumlarda sistemin verilen yönelme açısını takip edebildiği ve bu esnada yalpa ve yunuslama açılarını da koruyabildiği gözlemlenmiştir. Aşağıdaki resimde (Şekil 5.6) bu deneye ait sonuçlar görülebilir:



Şekil 5.6 : Dengeleme deneyine ait deney sonuçları.

6. SONUÇLAR VE ÖNERİLER

İnsansız hava araçları günümüzde askeri ve sivil pek çok alanda kullanılmaktadır. Araştırmacılar için ise son yıllarda ilgi çekici bir konu haline gelmiştir.

Görüntü işleme de son yıllarda araştırmacıları ilgisini çekmekle beraber, farklı alanlarda kullanılmaya başlanmıştır. Kullanım alanları arasında mobil robotlar da bulunmaktadır. Kamera küçük boyutlu ve çok bilgi verebilen bir sistem olduğu için küçük mobil robotlar üzerinde kullanımı büyük avantajlar sağlamaktadır. Hava araçları üzerinde de yer ve ağırlık sorunu bulunduğu için üzerlerinde bulundukları kamera ile aracın otonomlaştırılmaya çalışılması bu yönden avantajlar sağlamaktadır.

Bu tez çalışmasında öncelikli olarak açık kaynak kodlu elektronik bileşenleri olan bir hava aracı oluşturulmaya çalışılmıştır. Bu çalışmada hava aracının tasarımına ve imalatına ait bilgiler verilmiştir. Elektronik ve mekanik olarak parça seçimi açıklanmış ve yapılması gerekli deneyler ve sonuçları hakkında bilgi verilmiştir. Seçilen parçalar birleştirilerek görüntü işlemeye ve uçuşa uygun bir deney platformu oluşturulmuştur.

Hava aracının kararlılığından emin olabilmek için ilk olarak görüntü işlemeden bağımsız bir şekilde ataletsel ölçüm biriminin ölçümleri ile kontrolü sağlanmıştır. Çalışmada ilk önce quadrotor modeli karmaşık yapısından dolayı basitleştirilmiş bir şekilde ortaya konulmuştur. Daha sonra bu model kontrolcü tasarımına uygun hale getirilmiş ve uygun kontrolcü katsayıları belirlenmiştir. İlk belirlenen kontrolcü katsayıları ile gerçek sistemin kararsız çalıştığı görüldüğü için sistemden beklenenler değiştirilmiş ve ikinci adımda bulunan katsayılar ile yeni deneyler yapılmıştır. Sistemin gerçekleşmesine ait bilgiler verilmiş, kullanılan kodda yapılan değişiklikler ve ölçüm algoritması anlatılmıştır.

Görüntü işleme kullanarak konumlandırma ve haritalama yapabilen literatürdeki iki temel yöntemden EKF temelli SLAM algoritması seçilmiştir. Burada EKF'nin böyle bir sistemde yapılacak yeniliklere daha açık olması, kapalı ortamlarda (düşük referans noktası) daha hızlı çalışması ve haritalamadan daha çok konumlandırmayla ilgilenmesi nedeniyle seçilmiştir. Çalışmada SLAM ve Kalman filtresi hakkında

temel bilgiler verildikten sonra, kullanılan algoritma içerisindeki ifadeler tek tek açıklanmış ve kullanılan türev sonuçları ifade edilmiştir. Son olarak algoritmanın orijinal gerçeklemesi ile çalışmada elde edilen gerçeklemesi arasındaki farklar ortaya konulmuş ve nedenleri verilmiştir.

Deney sonuçları göstermiştir ki, kullanılan görüntü işleme algoritması kullanılarak hava aracı sıfır noktası civarında iken yunuslama ve yalpalama eksenlerinde dengede kalabilmekte, yönelme ekseninde ise verilen referans değerlerine oturabilmektedir. Sistem, güvenlik nedeniyle üç eksen de hareketine izin veren bir deney düzeneği üzerinde çalıştırılmıştır.

6.1 Öneriler

Yapılan deneylerde görüntü işleme sisteminin gömülü bilgisayar üzerinde maksimum 10 Hz hızda çalıştığı görülmüştür. Bu, sistemin yunuslama ve yalpalama açısından kontrolü için yeterli bir hız olmadığı için Kalman filtresi ve çoklu iş parçacığı yaklaşımı ile 50 Hz mertebelerine çıkarılması düşünülmektedir. Bu durumda ölçümler yine 10 Hz’de alınırken, ölçümlerin arasında kalan zaman dilimlerinde Kalman filtresinin tahmin etme özelliğinden yararlanılabilir.

Üç eksen de dengede bir sistemin elde edilmesi durumunda hava aracı deney düzeneğinden çıkartılıp, gerçek uçuş şartlarında değerlendirilebilir. Bu sayede daha gerçekçi bir deney ortamı oluşturulmuş olur.

Görüntü işleme sistemi konumlandırma yapabildiği için, sistem navigasyon amaçlı kullanılabilir. Sadece görüntü işleme kullanılarak hem sabit hız modeli için hem de hava aracının kendi lineerleştirilmiş modeli için Kalman filtresi kullanılarak navigasyon yapılabilir. Bu çalışmada ilk aşama olduğu için ve basitleştirme amaçlı sabit hız modeli kullanılmıştır ancak sabit hız modeli yumuşak hareketler için daha uygundur. Buna karşılık quadrotor ani manevralar yapabilen bir hava aracıdır. Hava aracının Kalman filtresine dahil edilmesi, kontrol girişlerinin de modele dahil edilmesi anlamına gelir, ve bu sayede Kalman filtresi daha gerçekçi tahminlerde bulunacaktır.

Navigasyon için sadece görüntü işleme algoritmasını kullanmanın yanı sıra, sistem üzerinde bulunan jiroskop ve ivmeölçerler de Kalman filtresi içinde sensör füzyonu içerisinde kullanılabilir. Bu şekilde sistemin ivmelenmeleri ve açısal hızları tahmin

etmesi yerine doğrudan üzerinden ölçüm alabileceği sensörlere bağlanmış olur. Ek olarak navigasyonda Kalman filtresinin içerisine oryantasyon verisi doğrudan DCM algoritmasının çıktısı olarak verilebilir.

Navigasyon sisteminin kurulması halinde doğruluğunun sınanabilmesi amacı ile yeni bir görüntü işleme sistemi ile dışarıdan hava aracının durumu izlenebilir. Bu şekilde bu deneyden elde edilen sonuçlar doğrulanabilir.

Son olarak kamera sayesinde otonom iniş kalkış, objelerden kaçınma ve obje takibi gibi özellikler eklenerek tam otonom bir uçuşa yaklaşılabılır.

KAYNAKLAR

- [1] **Austin, R.** (2010). Unmanned aircraft systems uavs design, development and deployment, United Kingdom, 1-2.
- [2] **Nonami, K.** (2010). Autonomous flying robots unmanned aerial vehicles and micro aerial,13
- [3] **Kendoul, F.** (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems
- [4] **Davison, A. J., Reid, I.D., Molton, N. ve Strasse, O.** (2007). MonoSLAM: Real-time camera Slam, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol 29, no 6
- [5] **Blösch, M., Stephan, W., Scaramuzza, D. ve Siegwart, R.** (2010). Vision Based MAV Navigation in Unknown and Unstructured Environment, *IEEE International Conference on Robotics and Automation*
- [6] **Nützi, G., Weiss, S., Scaramuzza, D. ve Siegwart, R.** (). Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM
- [7] **Liu, Y., Dai, Q.** (2010). A Survey of Computer Vision Applied in Aerial Robotic Vehicles, *International Conference on Optics, Photonics and Energy Engineering*
- [8] **Klein, G., Murray, D.** (2007) Parallel Tracking and Mapping for Small AR Workspaces, *Mixed and Augmented Reality, ISMAR, 6th IEEE and ACM International Symposium*
- [9] **Holmes, S., Klein, G. ve Murray, D.** (2008) A Square Root Unscented Kalman Filter for Visual monoSlam, *IEEE International Conference on Robotics and Automation*
- [10] **Kendoul, F. Fantoni, I ve Nonami, K.** (2009) Optic Flow-Based Vision System for Autonomous 3D Localization and Control of Small Aerial Vehicles, *Robotics and Autonomous Systems*, 57, 6-7
- [11] **Beard, R., Kingston ,D. Quigley, M. Snyder, D., Christiansen R., Johnson, W., McLain, T. ve Goodrich, M. A.** (2005) Autonomous Vehicle Technologies for Small Fixed-Wing UAVs, *Journal of Aerospace Computing, Information, and Communication*, Vol.2
- [12] **Demonceaux, C., Vasseur, P., Pegard, C.** (2006) Robust Attitude Estimation with Catadioptric Vision, *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 3448-3453, Beijing, China
- [13] **Bazin J. C., Kweon I. S., Demonceaux C., Vasseur P.** (2006) Omnidirectional Vision on UAV for Attitude Computation, *IEEE International Conference on Robotics and Automation*, pp. 2842-2847

- [14] **Bazin J. C., Kweon I. S., Démonceaux, C. Vasseur, P.** (2007) UAV Attitude Estimation by Combining Horizon-Based and Homography-based Approaches for Catadioptric Image, *IFAC/EURON Intelligent Autonomous Vehicles, Toulouse, France*
- [15] **Démonceaux, C., Vasseur, P., Pegard C.** (2007) UAV Attitude Computation by Omnidirectional Vision in Urban Environment, *IEEE International Conference on Robotics and Automation*
- [16] **Bazin, J. C., Kweon I. S., Démonceaux C, Vasseur, P.** (2008) UAV attitude estimation by Vanishing Points in Catadioptric Images, *IEEE International Conference on Robotics and Automation*, pp. 2743-3749
- [17] **Tarhan, M., Altuğ, E.** (2010) EKF Based Attitude Estimation and Stabilization of a Quadrotor UAV Using Vanishing Points in Catadioptric Images, *Journal of Intelligent and Robotic Systems Volume 62, Numbers 3-4*, 587-607
- [18] **Url-1** <<http://narcissus.angstrom-distribution.org/>>, alındığı tarih 25.04.2012
- [19] **Jones, M. P., Pierce, K. E. ve Ward, D.** (2007) Avian Vision: A Review of Form and Function with Special Consideration to Birds of Prey, *Journal of Exotic Pet Medicine, Vol 16, No 2, pp 69-87*
- [20] **Altuğ, E.** (2003) Vision Based Control of Unmanned Aerial Vehicles with Applications to an Autonomouse Four-Rotor helicopter, Quadrotor, University of Pennsylvania Phd. Thesis
- [21] **Bouabdallah, S., Noth, A ve Siegwart, R.** (2004) PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor, *Intelligent Robots and Systems (IROS)*
- [22] **Salih, A. L., Moghavvemi, M., Mohamed, H. A. F. ve Gaeid, K. S.** (2010) Flight PID Controller Design for a UAV Quadrotor, *Scientific Research and Essays, Vol. 5(23), pp. 3660-3667*
- [23] **Premierlani, W. ve Bizard, P.** (2009) Direction Cosine Matrix IMU: Theory. Adres: <http://code.google.com>
- [24] **Erginer B.** (2007) Quadrotor VTOL Aracının Modellenmesi ve Kontrolü, İTÜ Yüksek Lisans Tezi, İstanbul
- [25] **Grewal, S. M. ve Andrews, A. P.** (2008) Kalman Filtering Theory and Practice Using Matlab, A John Wiley and Sons. Inc. Publication, pp. 1-2
- [26] **Extended Kalman Filter.** (t.y). Wikipedia. Alındığı tarih : 20.04.2012, adres: http://en.wikipedia.org/wiki/Extended_Kalman_filter
- [27] **Civera, J., Davison A. J. ve Montiel, M. M.** (2008) Inverse Depth Parametrization for Monocular SLAM, *IEEE Transaction on robotics, Vol 24, No 5*.
- [28] **Url-2** <http://www.vision.caltech.edu/bouguetj/calib_doc/>, alındığı tarih: 26.04.2012
- [29] **Tarhan M.** (2009) Quadrotor Hava Aracının Katadioptrik İmgelerdeki Kaçış Noktaları Kullanılarak Kontrol Edilmesi, İTÜ Yüksek Lisans Tezi, İstanbul

- [30] **Beagleboard.** (t.y.). Wikipedia. Alındığı tarih: 01.05.2012, adres:
<http://en.wikipedia.org/wiki/BeagleBoard>
- [31] **Url-3** <<http://code.google.com/p/ardupilot/>>, alındığı tarih: 01.05.2012
- [32] **Url-4** <http://www.ptgrey.com/products/fireflymv/fireflymv_usb_firewire_cm_os_camera.asp>, alındığı tarih : 01.05.2012
- [33] **Quaternion** (t.y.). Wikipedia. Alındığı tarih: 25.04.2012 , adres:
<http://en.wikipedia.org/wiki/Quaternion>
- [34] **Conversion between quaternions and Euler angles** (t.y.). Alındığı tarih:
20.04.2012, adres:
<http://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles>
- [35] **Baldski, G. ve Kaehler, A.** (2008) Learning OpenCV, O'Reilly, s 215 ve s 317

EKLER

EK A1 : Beagleboard Konfigürasyonu

EK A2 : Elektronik Bileşenlere Ait Teknik Özellikler

EK B : Hava Aracı Kontrolcü Simulink Modeli

EK C1 : Quaternion İşlemleri

EK C2 : Görüntü İşleme Algoritmaları

EK D : Pervane Kuvvet Ölçüm Deneyinde Kullanılan Devre Şeması

EK A1

Çizelge A.1 : Beagleboard Xm kurulum konfigürasyonu.

Additional Packages:

angstrom-gdm-autologin-hack
angstrom-task-gnome
bash-sh
boost-dev
cpufrequtils
dosfstools
e2fsprogs
e2fsprogs-mke2fs
epiphany
ffmpeg
gedit
gnome-mplayer
gst-plugin-gles
gst-plugins-bad-meta
gst-plugins-base-meta
gst-plugins-good-meta
gst-plugins-ugly-meta
gstreamer-ti
initscripts
kernel-modules
mplayer
nano
network-manager-applet
networkmanager
networkmanager-openvpn
omapfbplay
omapfbplay-dce
opencv-dev
opencv-samples
opencv-samples-dev
openjdk-6-jdk
shadow
sysvinit
sysvinit-pidof
task-gnome-apps
task-native-sdk
xdotool

EK A2

Çizelge A.2 : Beagleboard Xm teknik özellikler [30].

	Özellik	
İşlemci	Texas Instruments Cortex A8 1GHz	
POP Hafıza	Micron 4 Gb MDDR SDRAM (512 MB) 200 MHz	
PMIC TPS65950	Güç Regulatorleri	
	Audio Codec	
	Reset	
	USB OTG PHY	
Hata Ayıklama Desteği	14-pin JTAG	GPIO Pinleri
	UART	3 Led
PCB	3.1" X 3.0"	6 Katman
Göstergeler	Güç, Güç Hatası	2-Kullanıcı Kontrolünde
	PMU	USB Güç
Ethernet	10/100	USB cihaz üzerinden
SD/MMC bağlantısı	MicroSD	
Video	DVI-D	S-Video
Kamera	Konektör	Leopard Imaging Desteği
Güç Konektör	USB	DC Güç

Çizelge A.3 : Ardupilot teknik özellikler [31].

	Özellik
İşlemci	16 MHz Atmega 1280
Flash program hafızası	128k
SRAM	8k
EEPROM	4k
Analog giriş ve ADC	16 (12 bit)
Dijital IO	40
Seri port	2 bağımsız
Besleme	RC alıcı veya ayrıca pil ile

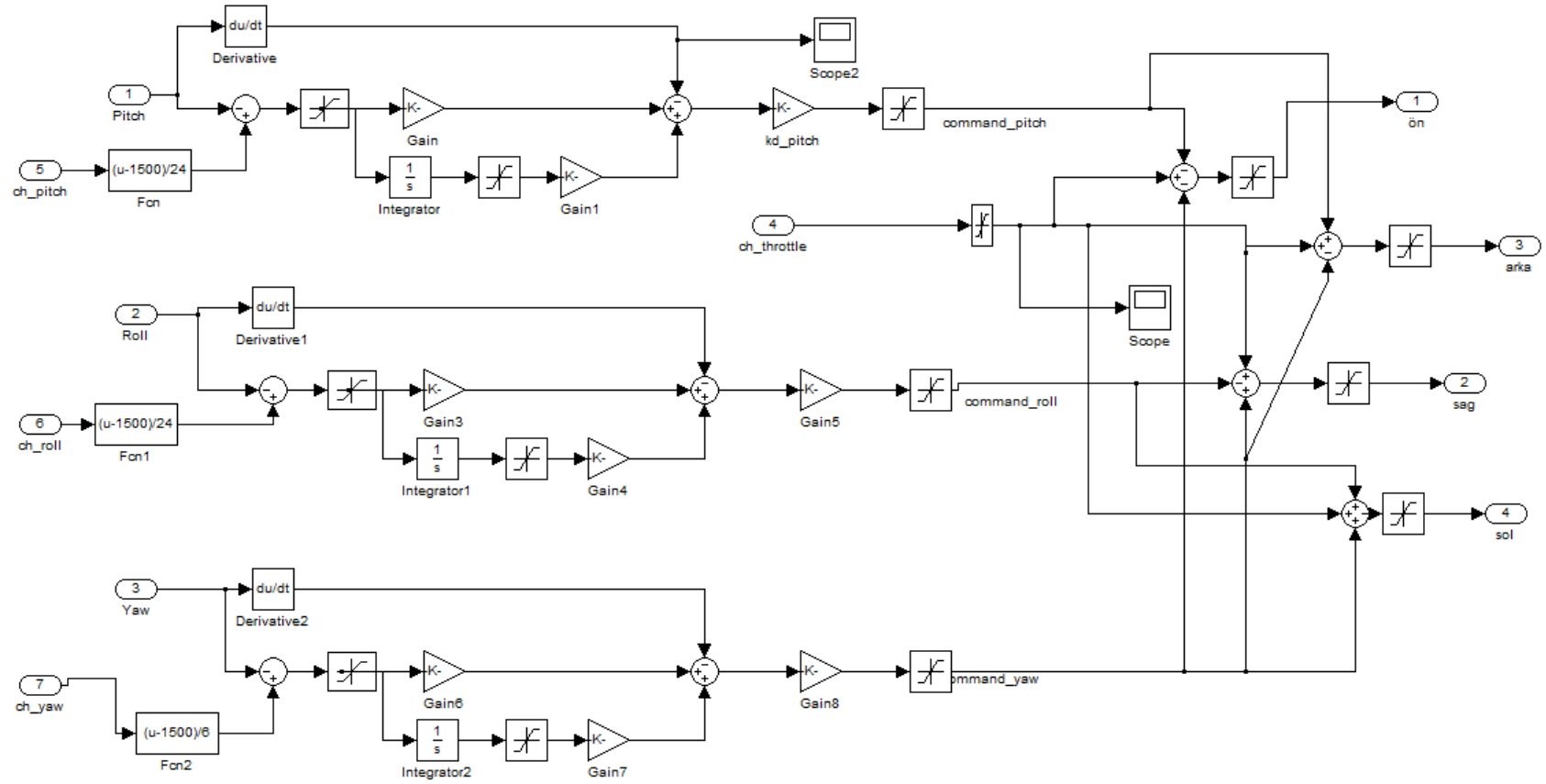
Çizelge A.4 : ArduIMU teknik özellikler [31]

	Özellik
İvmeölçer	ADX330
Jiroskoplar	LISY300AL
Flash program hafızası	128k
Kara Kutu	16 MB data logger
Ekstra Sensörler	Sıcaklık Sensörü
	Barometre
İletişim Portları	Seri
	I2C
	FTDI

Çizelge A.5 : Kamera teknik özellikler [32].

	Özellik
Sensör Tipi	1/3" progressive scan CMOS
Shutter Tipi	Global shutter Micron TrueSNAP teknolojisi kullanan
Sensör Modeli	Micron MT9V022
Maximum Çözünürlük	752(H)/480(V)
Piksel Boyutu	6.0 µm x 6.0 µm
ADC	10-bit ADC
Video Veri Çıkışı	8 ve 16 bit dijital veri
Kare Hızı	640x480 – 60FPS
Görüntü Veri Formatı	Y8-Y16 (monochrome) 8 bit ve 16 bit ham Bayer verisi
Bağlantı	5-pin Mini-B USB 2.0
Gain	Otomatik/manuel, 0dB-12dB
Shutter	Otomatik/manuel, 0.12 ms-512 ms
Lens	Cs Mount

EK B



Şekil B.1 : Ardupilot blok diyagramı.

EK C1

İki quaternion ifadesi çarpımı [33] $(a_1, b_1, c_1, d_1) \otimes (a_2, b_2, c_2, d_2)$

$$\begin{aligned} &= (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2, \\ &\quad a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2, \\ &\quad a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2, \\ &\quad a_1 + b_1 c_2 - c_1 b_2 + d_1 a_2) \end{aligned} \tag{C 1.1}$$

Euler quaternion dönüşümü [34]

$$q = [q_0 \ q_1 \ q_2 \ q_3] \tag{C 1.2}$$

$$q_0 = \cos\left(\frac{\alpha}{2}\right)$$

$$q_1 = \sin\left(\frac{\alpha}{2}\right)\cos(\beta_x)$$

$$q_2 = \sin\left(\frac{\alpha}{2}\right)\cos(\beta_y)$$

$$q_3 = \sin\left(\frac{\alpha}{2}\right)\cos(\beta_z)$$

Quaternion Euler Dönüşümü

$$\begin{bmatrix} \Phi \\ \theta \\ \Psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \\ \arcsin(2(q_0 q_2 - q_3 q_1)) \\ \arctan\left(\frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \end{bmatrix} \tag{C 1.3}$$

EK C2

Goodfeaturestotrack [35]

Görüntü işleme teknikleri kullanılarak bir obje takip edilecekse veya kamera hareket halinde iken sahne takip edilecekse, boş beyaz bir duvara bakarak bunu yapmak pek mümkün olmaz. Ya da birbirini tekrar eden yüzeyler üzerinden alınan bir parçanın, bir sonraki resimde tekrar bulunabilmesi çok zordur. Bu nedenle görüntü işlemede sahne içerisinde tekil bir nokta bulunur ve bu noktalara köşe adı verilir.

Resmin ikinci türevi ile elde edilen Hessian matrisi:

$$H(p) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (\text{C 2. 1})$$

Hessian resmi üzerinden alınan küçük pencerelerden elde edilen otokorelasyon matrisi:

$$M(x, y) = \begin{bmatrix} \sum_{-K \leq i, j \leq K} w_{i,j} I_x^2(x+i, y+j) & A \\ \sum_{-K \leq i, j \leq K} w_{i,j} I_x(x+i, y+j) I_y(x+i, y+j) & B \end{bmatrix} \quad (\text{C 2. 2})$$
$$A = \sum_{-K \leq i, j \leq K} w_{i,j} I_x(x+i, y+j) I_y(x+i, y+j)$$
$$B = \sum_{-K \leq i, j \leq K} w_{i,j} I_y^2(x+i, y+j)$$

W ifadesi ağırlık terimidir. Harris'in ifadesine göre ikinci türev resmi üzerinde bulunan otokorelasyon matrisi üzerinde iki özdeğer büyük çıkarsa (belirli bir eşikten) bu bize tekil noktayı verir. Harris'ten sonra Shi ve Tomasi köşe noktasının bu özdeğerlerin minimum değerinin belirli bir eşikten daha yüksek olması durumunda kabul edilebileceğini söylemiştir. Bu algoritma da bunu kullanmaktadır.

Cvmatchtemplate [35]

Görüntü işlemede şablon eşleştirme işlemi için farklı metotlar söz konusudur. Ancak bu metotlar genel olarak şablon resmin karşılaştırıldığı resim üzerinde dolaştırılması ve farklılıklarının karşılaştırılması ilkesine dayanır. En basit şekliyle iki resim

arasında kalan farkların hesaplandığı bir resimde 0 noktası veya minimum nokta şablon resmin ana resim üzerinde görüldüğü nokta kabul edilebilir. Bu tez çalışmasında biraz daha karmaşık bir yöntem (daha sağlıklı sonuçlar veren) olan `ccoeff_normed` kullanılmıştır. Buradaki işlemler [kitap]'ta şu şekilde açıklanır.

$$R_{ccoeff}(x, y) = \sum_{x', y'} [T'(x', y') \cdot I'(x + x', y + y')]^2 \quad (\text{C 2.3})$$

T şablon resim (template) ve I şablonun arandığı resimdir.

$$T'(x', y') = T(x', y') - \frac{1}{(w \cdot h) \sum_{x'', y''} T(x'', y'')} \quad (\text{C 2.4})$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{(w \cdot h) \sum_{x'', y''} I(x + x'', y + y'')}$$

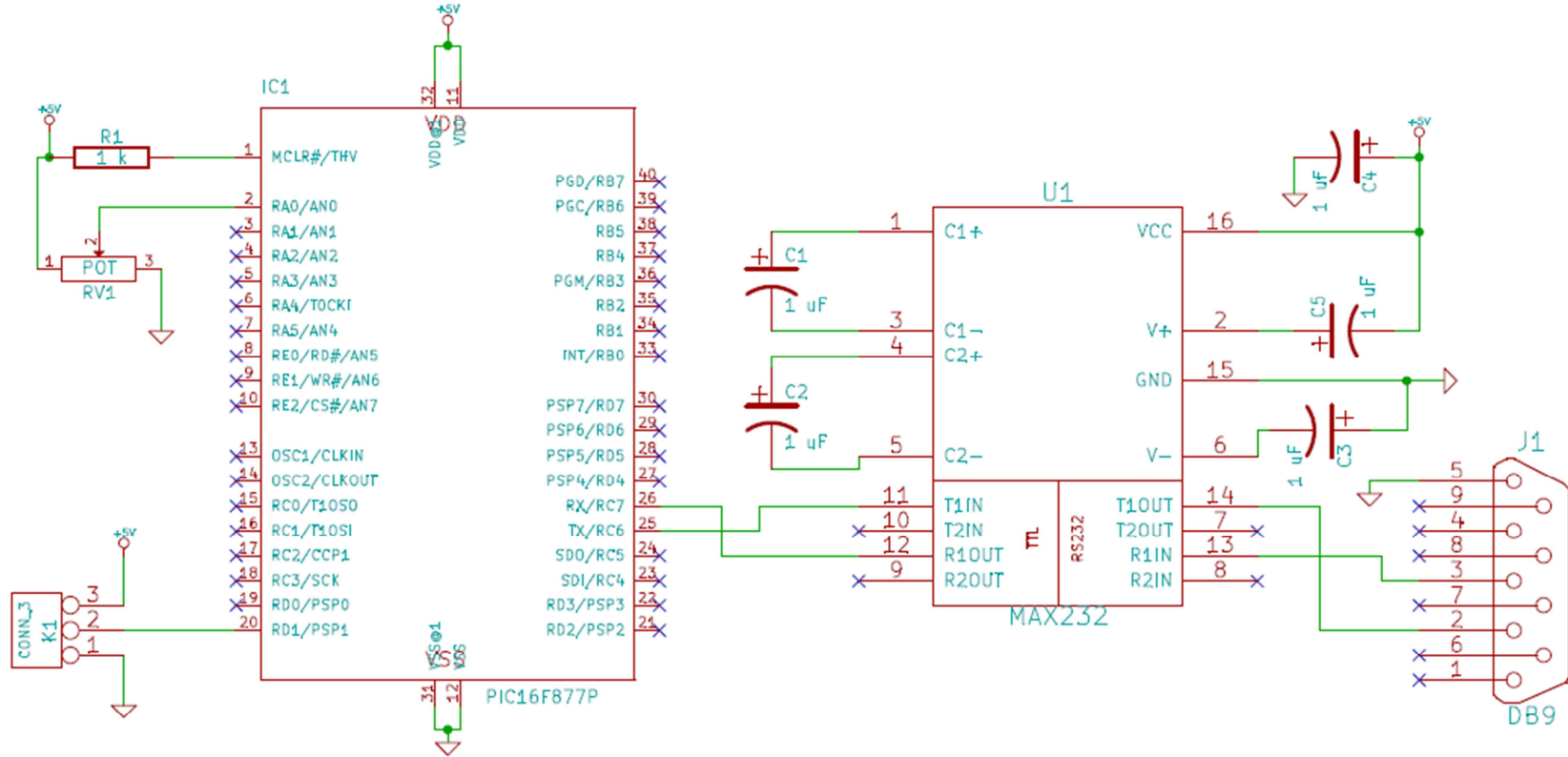
Normalleştirme işlemi için gerekli işlem yapılırsa

$$Z(x, y) = \sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2} \quad (\text{C 2.5})$$

Sonuç aşağıdaki şekilde bulunur. Bu işlem sonucunda 1 mükemmel eşleşmeye denk gelir. Algoritma içinde kullanımında eşik değeri %80 eşleşmeye ayarlanmıştır.

$$R_{ccoeff_{normed}}(x, y) = \frac{R_{ccoeff}(x, y)}{Z(x, y)} \quad (\text{C 2.6})$$

EK D:



Şekil D.1 :Pervane kuvvet ölçüm deneyinde kullanılan devre şeması.

ÖZGEÇMİŞ

Ad Soyad: Cihat Bora Yiğit

Doğum Yeri ve Tarihi: İstanbul ve 17.06.1987

E-Posta: yigitci@itu.edu.tr

Lisans: İTÜ Makine Mühendisliği