

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**KURTARMA ROBOTLARINDA HAZIR ELEKTRONİK KARTLAR
İLE İLERİ KONTROL UYGULAMALARI**

YÜKSEK LİSANS TEZİ

Mehmet Volkan BÜKEY

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

TEMMUZ 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**KURTARMA ROBOTLARINDA HAZIR ELEKTRONİK KARTLAR
İLE İLERİ KONTROL UYGULAMALARI**

YÜKSEK LİSANS TEZİ

**Mehmet Volkan BÜKEY
518091050**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç Dr. Emin Faruk KEÇECİ

TEMMUZ 2012

İTÜ, Fen Bilimleri Enstitüsü'nün 518091050 numaralı Yüksek Lisans Öğrencisi **Mehmet Volkan BÜKEY** ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**Kurtarma Robotlarında Hazır Elektronik Kartlar ile İleri Kontrol Uygulamaları**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç.Dr. Emin Faruk KEÇECİ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Dr. Erdinç ALTUĞ**
İstanbul Teknik Üniversitesi

Yrd. Doç. Dr. Aydemir ARISOY
Hava Harp Okulu

Teslim Tarihi : **6 Temmuz 2012**
Savunma Tarihi : **17 Temmuz 2012**

ÖNSÖZ

Bu projenin her aşamasındaki desteklerinden dolayı Doç. Dr. Emin Faruk KEÇECİ'ye, Cihat Bora YİĞİT'e, Gökçe Burak TAĞLIOĞLU'na ve aileme en içten duygularıyla teşekkürü bir borç bilirim.

Bu yüksek lisans tezinin deneysel aşamalarının gerçekleştirilmesi için gereken mali destek ITU- BAP birimi tarafından 36482 numaralı proje kapsamında sağlanmıştır.

Temmuz 2012

Mehmet Volkan BÜKEY
(Elektrik Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1. GİRİŞ	1
1.1 Arama ve Kurtarma Robotlarında Kullanılan Sensörler	2
1.2 Kamera Görüntüsünün Düzeltilmesi	4
1.3 Kendini Dengeleyen Platform	5
1.4 Amaç	5
1.5 Bölümlerin Dağılımı	6
2. ATALETSEL ÖLÇÜM CİHAZI.....	7
2.1 İvmeölçer.....	10
2.2 Jiroskop	16
2.3 Kalman Filtresi Metodu	18
2.4 Kalman Filtresi ile Veri Birleştirme	22
3. ROBOTUN TASARIMI VE KULLANILAN EKİPMANLAR.....	27
3.1 Mekanik Donanım	27
3.1.1 Mobil robot şasesi	28
3.1.2 Kamera dengeleme sistemi	28
3.1.3 Servo motorlar.....	33
3.2 Elektronik Donanım	34
3.2.1 Mikroişlemci kart	34
3.2.2 IMU	37
3.2.2.1 İvmeölçerin özellikleri	37
3.2.2.2 Jiroskobun özellikleri	40
3.2.3 Motor sürücü kartı.....	42
3.2.4 Kablosuz haberleşme ünitesi.....	45
3.2.5 Kumanda devresi.....	47
3.2.6 Kablosuz kamera.....	48
3.3 Yazılım ve Haberleşme	49
3.3.1 Mikroişlemci programlama dili	49
3.3.2 Haberleşme protokolü	52
3.3.2.1 ADXL345 I ² C yazmaçları.....	57
3.3.2.2 ITG3200 I ² C yazmaçları	58
4. ROBOTUN GERÇEKLEMESİ VE KONTROLÜ.....	61
4.1 Kamera Dengeleme Sisteminin Çalışma Prensipleri.....	62
4.2 Bağlantı Şeması.....	63
4.2.1 Enerji bağlantıları.....	65

4.2.2 Sinyal bağlantıları	66
4.3 Mikroişlemci Kodları	67
4.3.1 Mobil robot kodu.....	67
4.3.2 Kumanda devresi kodu.....	70
5. TESTLER VE SONUÇLAR	73
5.1 İvmeölçer Sonuçları.....	74
5.2 Kalman Filtresi Sonuçları.....	75
5.2.1 Ölçüm gürültü kovaryansının Euler açıları üzerindeki etkisi.....	77
5.2.2 İşlem gürültü kovaryansının Euler açıları üzerindeki etkisi.....	82
5.2.3 Kalman filtresi sonuçlarının analizi	85
5.2.3.1 Tepki hızının incelenmesi	85
5.2.3.2 Titreşimin etkilerinin incelenmesi.....	87
5.3 Kamera Görüntüleri.....	88
5.4 Gelecek Çalışmalar.....	90
KAYNAKLAR.....	93
EKLER.....	97
ÖZGEÇMİŞ	111

KISALTMALAR

ASCII	: Bilgi alışverişi için standart Amerikan kodu
LSB	: En önemsiz bit
MSB	: En önemli bit
I²C	: Inter-Integrated Circuit
IMU	: Ataletsel ölçüm cihazı
PID	: Proportional-Integral-Derivative
PWM	: Darbe genişlik modülasyonu
SCL	: Saat vurumu hattı
SDA	: Veri sinyal hattı
SPI	: Serial Peripheral Interface
USB	: Evrensel seri veri yolu

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Palet motorlarının özellikleri.....	28
Çizelge 3.2 : Parallax standart servo motorun teknik özellikleri.....	33
Çizelge 3.3 : Arduino Duemilanove mikroişlemci kartına ait teknik özellikler.....	35
Çizelge 3.4 : ADXL345 için farklı ölçüm aralıklarındaki ölçek değerleri.....	39
Çizelge 3.5 : ADXL345 için farklı ölçüm aralıklarındaki hassasiyet değerleri.....	39
Çizelge 3.6 : MC33926 çift motor sürücü karta ait teknik özellikler.....	42
Çizelge 3.7 : MC33926 çift motor sürücü kartta bulunan bağlantılar ve görevleri..	44
Çizelge 5.1 : Euler açılarına ait standart sapma değerleri.....	78

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Üç ekseninde rotasyon hareketleri.	8
Şekil 2.2 : Z-Y-X sıralamasına göre yapılan rotasyon hareketleri.	9
Şekil 2.3 : İvmeölçer kutu ve küre modelinin yerçekimsiz alandaki görünümü.	11
Şekil 2.4 : İvmeölçerin X+ yönünde hareketi sonucunda uygulanan kuvvetler.	12
Şekil 2.5 : İvmeölçerin düz yerleştirilmesi sonucunda uygulanan kuvvetler.	13
Şekil 2.6 : Y+ yönünde 45 derecelik dönme sonucunda uygulanan kuvvetler.	13
Şekil 2.7 : İvmeölçerden elde edilen ivme vektörünün görünümü.	14
Şekil 2.8 : X ve Y eksenindeki rotasyonların koordinat sisteminde gösterimi.	15
Şekil 2.9 : Kalman filtresi denklem döngüsü.	22
Şekil 3.1 : Mobil robotun paletli gövdesinin görünümü.	28
Şekil 3.2 : Kamera dengeleme sisteminin görünümü.	29
Şekil 3.3 : Arduino Duemilanove mikroişlemci kartının görünümü.	36
Şekil 3.4 : Mobil robot üzerinde kullanılan ataletsel ölçüm cihazı.	37
Şekil 3.5 : Farklı görev döngüsü için uygulanan gerilim sinyalleri.	43
Şekil 3.6 : MC33926 çift motor sürücü karta ait bağlantılar.	44
Şekil 3.7 : XBee kablosuz haberleşme modülünün görünümü.	46
Şekil 3.8 : XBee Shield ve Arduino Duemilanove birleşik görünümü.	46
Şekil 3.9 : Joystick Shield ve Arduino Duemilanove birleşik görünümü.	47
Şekil 3.10 : Kablosuz kamera ve alıcı ünitenin görünümü.	49
Şekil 3.11 : I ² C veri yolunun fiziksel yapısı.	53
Şekil 3.12 : Veri transferinin başlaması için gereken lojik değerler.	54
Şekil 3.13 : Veri transferinin durması için gereken lojik değerler.	54
Şekil 3.14 : Adres bilgisi ve okuma/yazma bitini barındıran baytın lojik değerleri.	55
Şekil 3.15 : Bir baytlık veri transferi sırasında hatların lojik değerleri.	56
Şekil 4.1 : Robotun üstten görünümü.	61
Şekil 4.2 : Robotun karşıdan görünümü.	61
Şekil 4.3 : Kamera dengeleme sisteminin çalışma prensibi.	63
Şekil 4.4 : Devre bağlantı şeması.	64
Şekil 4.5 : Hareketli koldan farklı yönler için elde edilen veriler.	71
Şekil 5.1 : X ve Y eksenleri için ivmeölçerden elde edilen Euler açıları.	74
Şekil 5.2 : X ve Y eksenleri için ivmeölçerden hareketsiz durumda ölçülen veriler.	78
Şekil 5.3 : X eksenini Kalman filtresi ve ivmeölçer sonuçları (R=0,4461).	79
Şekil 5.4 : Y eksenini Kalman filtresi ve ivmeölçer sonuçları (R=0,4369).	79
Şekil 5.5 : X eksenini Kalman filtresi ve ivmeölçer sonuçları (R=10).	80
Şekil 5.6 : Y eksenini Kalman filtresi ve ivmeölçer sonuçları (R=10).	80
Şekil 5.7 : X eksenini Kalman filtresi ve ivmeölçer sonuçları (R=0,01).	81
Şekil 5.8 : Y eksenini Kalman filtresi ve ivmeölçer sonuçları (R=0,01).	81
Şekil 5.9 : X eksenini Kalman filtresi ve ivmeölçer sonuçları (Q _a ve Q _b =0,001).	83
Şekil 5.10 : Y eksenini Kalman filtresi ve ivmeölçer sonuçları (Q _a ve Q _b =0,001).	83
Şekil 5.11 : X eksenini Kalman filtresi ve ivmeölçer sonuçları (Q _a ve Q _b =0,1).	84
Şekil 5.12 : Y eksenini Kalman filtresi ve ivmeölçer sonuçları (Q _a ve Q _b =0,1).	84

Şekil 5.13 : Kamera ve gövde yöneliminin karşılaştırılması (X eksen).	86
Şekil 5.14 : Kamera ve gövde yöneliminin karşılaştırılması (Y eksen).	86
Şekil 5.15 : Titreşim sırasında X eksenine ait Euler açılarının değişimi.	87
Şekil 5.16 : Titreşim sırasında Y eksenine ait Euler açılarının değişimi.	88
Şekil 5.17 : Robot üzerinden elde edilen kamera görüntüleri.....	89

KURTARMA ROBOTLARINDA HAZIR ELEKTRONİK KARTLAR İLE İLERİ KONTROL UYGULAMALARI

ÖZET

Robotlar insan yardımıyla ya da otomatik olarak çeşitli görevleri yerine getiren elektromekanik araçlardır. Gelişen eyleyici ve sensör teknolojileri sayesinde robotik sistemlere birçok farklı alanda rastlamak mümkündür. Genellikle robotik sistemler yapılan işte insan gücünün azaltılması ve kalitenin artırılması için kullanılırlar. Buna ek olarak robotlar insanların çalışması için elverişli olmayan ortamlarda da kullanılabilirler. Bu tip robotlar için verilebilecek en tipik örnek arama ve kurtarma robotlarıdır. Arama ve kurtarma robotları yangın, enkaz, yıkılmış maden, sualtı gibi insanların çalışması için tehlikeli olan alanlarda çalışma yapabilirler. Genel olarak bu robotlar çalışılan alandaki canlılar ya da çalışma alanının durumu hakkında bilgi edinmek için kullanılır. Çalışma koşullarından dolayı bütün arama ve kurtarma robotları çalışma yapılacak bölge için özel olarak tasarlanırlar.

Robotlar çalışma sırasında kendilerini çevreleyen alanı ve kendi durumlarını algılamak için sensörlere başvururlar ve çalışma sırasında bu sensörlerden elde edilen bilgilere göre görevleri yerine getirirler. Bu yüzden robotik sistemlerde sensörlerden elde edilen verilerin optimizasyonu robotun çalışma veriminin artırılması için önemlidir.

Arama kurtarma robotları da çalıştıkları alanlara göre farklı sensörler barındırırlar. Bu sensörler genellikle üç farklı kategoride toplanır. Buna göre arama ve kurtarma robotlarının üzerinde enkaz altında mahsur kalan canlıların bulunması için, robotu çevreleyen alan hakkında bilgi edinmek için ve kullanıcıyı robotun durumu hakkında bilgilendirmek için çeşitli sensörler bulunur. Burada bulunan her sensör robotun çalışmasında farklı bir öneme sahiptir ve genellikle kullanılan sensörlerin hepsi burada belirtilen görevlerden en azından birinin gerçekleştirilmesi için özel olarak seçilmiştir.

Kamera sistemleri arama ve kurtarma robotlarında bulunan sensörlerin yerine getirdiği üç görev için de kullanılabilir. Bu yüzden genellikle arama ve kurtarma robotlarının çoğunda kamera sistemi mevcuttur. Mobil robotlarda kullanılan kameralar ile ilgili en önemli sorunlardan biri kameradan elde edilen verinin gövdenin duruşuna göre değişmesidir. Genellikle kamera sistemleri robotun gövdesine sabit şekilde monte edildiği için robotun gövdesi farklı yönlerde hareket ettikçe kameradan elde edilen görüntü de robotun yönelimine göre değişir. Bu durum özellikle arama ve kurtarma robotları için önemli bir sorundur; zira robotun çalışma yaptığı alanlar genellikle moloz ile kaplı olduğundan robotun gövdesi hareket sırasında birçok kez yönelimini değiştirebilir. Bunun sonucunda robottan elde edilen görsel veriye göre robotun kontrol edilmesi zorlaşır. Bu sorunun giderilmesi için ya elde edilen görüntünün görüntü işleme yöntemleri ile düzeltilmesi ya da kameranın robotun yönelim değişikliklerine karşı dengelenmesi gerekir. Görüntü işleme yönteminde işlemin robot üzerinde gerçekleşmesi için gereken işlemci gücü çok

fazladır. Ayrıca bu tür işlemleri standart kameralar ile yapmak mümkün değildir. Bunun için kameranın dengelenmesi yöntemi daha elverişlidir.

Yapılan çalışmada da mobil robot üzerine yerleştirilen bir kamera robota özel olarak üretilmiş bir hareket sistemi ile dengelenmiştir. Burada kameranın dengelenme işlemi servo motorlar aracılığıyla gerçekleştirilmiştir. Robotun gövdesinin yöneliminin belirlenebilmesi için ise bir ataletsel ölçüm cihazı kullanılmıştır.

Ataletsel ölçüm cihazları genellikle bir adet ivmeölçer ve bir adet jiroskoptan meydana gelir. Burada bulunan ivmeölçerde İvmeölçerler bağlı oldukları gövdenin statik ve dinamik ivmesini ölçmek için kullanılır. Jiroskoplar ise gövdenin farklı eksenler üzerinde yaptığı rotasyon hareketlerinin açısal hızını ölçmek için kullanılır. Elde edilen açısal hız verilerinin belli zaman aralıklarında integrali alınarak gövdeye ait rotasyon açıları kolayca bulunabilir. Gövdenin yöneliminin belirlenebilmesi için bu iki sensörden elde edilen veriler belli bir algoritma kullanılarak birleştirilir. Bunun için yapılan uygulamada Kalman filtresi kullanılmıştır.

Kalman filtresi stokastik modeli bilinen bir sistemin ölçülemeyen parametrelerinin hesaplaması için kullanılır. Burada belirlenecek parametrelerin değerleri farklı sensör ölçümleri kullanılarak tahmin edilir. Tahmin değerleri hesaplanırken sistemde bulunan sensörlere ve modelde belirlenecek parametrelere ait standart sapma değerleri kullanılır. Buradaki standart sapma değerlerine göre her ölçüm adımında Kalman filtresinin yaptığı tahmin değerleri belli bir değere yakınsar. Kalman filtreleri esnek ve kolay uygulanabilir yapılarından dolayı birçok farklı uygulamada kullanılırlar.

Yapılan uygulamada söz konusu mobil robot tasarlanıp imal edildikten sonra robotun üzerinde bulunan kamera dengeleme sisteminin çalışma performansı test edilmiştir. Bunun için ilk olarak kullanılan modele göre Kalman filtresi uygun şekilde düzenlenmiş ve filtreden elde edilen sonuçlar incelenmiştir. Daha sonra da dengeleme sistemi aktifken robotun gövdesinin yönelimi değiştirilmiş ve bu durumlarda kameradan elde edilen görsel veri gösterilmiştir. Sonuç olarak tasarlanan dengeleme sistemi gövdenin iki farklı dönme ekseninde yapacağı ± 30 derecelik rotasyon hareketlerine karşı robotun üzerindeki kameranın görüş açısını sabit tutabilmiştir.

ADVANCED CONTROL APPLICATIONS FOR RESCUE ROBOTICS WITH OFF THE SHELF ELECTRONICS

SUMMARY

Robots are electro-mechanical devices that can accomplish tasks automatically or with guidance. They have a wide area of usage because of their effectiveness. Moreover, their capabilities continue to increase with the advancing technology of actuators and sensors. The main purpose of using robots is to save manpower and increase quality and speed. Additionally robots can be used in such way that they can accomplish tasks that are impossible to complete with human effort. For example, with proper modifications robots can function in environments where it can be dangerous for humans to work. Search and rescue robots are typical examples for this kind of robots. They are special robots that can work in hazardous environment such as partially collapsed buildings, mines, wildfire, flooded areas...etc. Generally, their main purpose is to investigate the area and collect information about the environment or victims. Considering their working environment, almost all search and rescue robots are specially designed for working under difficult conditions.

Robots use sensors to perceive their environment and their situation. They perform various tasks with respect to the data received from the sensors. Therefore, optimization of the data taken from the sensor is important for controlling the operation of the robot.

Similarly, search and rescue robots use various kinds of sensors according to their mission or according to where they are used. The sensors contained in search and rescue robots can be used for three different goals such as locating the victims, perceiving the environment of the robot and informing the operator about the robot's situation. Every sensor placed on the robot is specially chosen to accomplish at least one of the tasks mentioned above.

Camera systems can be used for all three goals defined for the sensors. Therefore, almost all mobile robots carry a camera on their body. Nevertheless, it might be hard to work with the camera on mobile robots since the view of the camera changes continuously during the movement of the robot considering that the camera is fixed on the body of the robot.

The changing orientation of the camera is an important problem for search and rescue robots as well. Generally search and rescue robots need to move through lots of rubbles and broken floor considering its working environment and because of this, the body of the robot change its orientation consistently during its movement, which makes it hard to interpret the data taken from the camera.

Visual data can be optimized either through image processing or through rotating the camera platform with respect to the orientation of the robot. Using image-processing techniques on real time projects requires lots of processor power and generally, these operations cannot be done through standard cameras. It requires advanced cameras, which can record visual data from different angles, since it requires more dimensions

to understand the orientation of the visual data. Therefore moving the camera with the changing orientation of the robots is a simpler solution for such problem.

In this work, a camera stabilization system for a mobile robot was built. Hence, a premade mobile robot body was equipped with necessary electronic circuit and a camera was mounted on the robot. Afterwards two servomotors rotated the camera platform in order to stabilize it with respect of the orientation of the robot. Two additional servomotors have been used to change the view of the camera on another axis. The orientation of the robot's body was determined through an inertial measurement unit.

There are several ways to define the orientation of a body. In this project Euler angle representation has been used to calculate the orientation. The orientation of a rigid body can be described with rotation of a moving frame with respect to another fixed frame with one point fixed. Additionally it is assumed that the origin of the moving frame and fixed frame are fixed to the same point. In an Euler angle representation, the orientation of a body is described through three successive rotations about the coordinate axes of either the fixed coordinate system or the moving coordinate system. The angles of these rotations around the coordinate axes are named as Euler angles. The orientation of the body can be determined through calculating these angles. In this project, the required data for calculating Euler angles were collected from an inertial measurement unit.

An inertial measurement unit generally consists of a gyroscope and an accelerometer. Both sensors work together in order to sense the orientation of the body, to which they are attached. Accelerometers are used for sensing the static and dynamic acceleration of the body and gyroscopes are used for sensing the angular velocity of the body, when it is rotating around a particular axis. Both sensors cannot determine the orientation of the body on its own. The data collected from the sensors should be joined to each other through an algorithm. Therefore, a Kalman filter has been used in this project.

Kalman filters are used for estimating unknown parameters of a system, which has a predefined stochastic model. The filter uses several sensor readings in order to make the estimations. Additionally the behaviour of the filter is controlled with changing the standard deviation rates of the sensor and system parameters defined in the Kalman filter equations. The estimations converge to a value with respect to these standard deviation rates on every iteration. Consequently, Kalman filters are used on many different applications due to their feasibility.

The realization of the camera stabilization system was done in three steps. Firstly, the required electronic circuits and mechanical parts were chosen. Afterwards the robot was built with the chosen equipment according to the design. Lastly, several tests were performed on the completed system to determine the working performance of the system.

The camera stabilization system and the mobile robot are designed separately through the realization process. Firstly, a prepared mobile robot chassis was bought and required electronic circuits to control the motion of the motors were added to the system. Afterwards a platform has placed on the top of the chassis and the stabilization system was built on this platform. Here the servomotors were placed on the proper areas, so that they can control the rotation of the camera on three dimensions. Lastly, a microcontroller unit was mounted on the platform and all the electronic equipment except the wireless camera were connected to the

microcontroller unit, so that the operation of the motors and the camera stabilization system can be controlled.

The mobile robot was designed in such way, so that it can be controlled wirelessly. Therefore, necessary electronic equipment was placed on the mobile robot in order to activate the wireless communication. Here the movement of the robot and the yaw movement of the camera were controlled with a remote control unit. Therefore, another microcontroller unit was used and modified in such way, that it can send relevant data about the motors and camera movement to the microcontroller placed on the mobile robot.

After the mobile robot was realised, the camera stabilization system was tested with respect to some predefined performance criterion. Firstly the Kalman filter was arranged properly for the model of the inertial measurement unit and the results taken from the filter were analysed. Afterwards the orientation of the robot was changed when the stabilization system is active and then the visual data was taken from the camera mounted on the mobile robot. Consequently, the stabilization system was able to control the orientation of the camera when the body of the robot was rotating on two different axes by ± 30 degrees.

1. GİRİŞ

Robotlar otomatik olarak veya bir operatör yardımıyla çeşitli görevleri yerine getirebilen elektro-mekanik yardımcılardır. Günümüzde birçok alanda farklı amaçlar için kullanılan robotlar mevcuttur. Gelişen teknoloji ile de robotların çalışma alanları genişlemektedir. Robotlar genellikle kendini tekrarlayan işlerde hem insan gücünü azaltmak hem de hatayı ve zaman kaybını en aza indirgeyerek kaliteyi arttırmak için kullanılırlar. Ayrıca robotlar insan gücüyle gerçekleştirilemeyecek görevleri yerine getirerek çoğu alanda insanlara kolaylık sağlarlar. Günümüzde insanların çalışması için tehlikeli olan alanlarda çalışabilen robotlara en tipik örneklerden biri arama kurtarma robotlarıdır. Bu robotlar enkaz, yangın, yıkılmış maden ya da sualtı gibi zor koşullarda arama ve kurtarma görevlerinde yer alabilirler. Genel amaçları arama ve kurtarma görevini yapacakları bölge ve bu bölgenin içinde mahsur kalan canlılar hakkında bilgi toplamak ve operatöre iletmektir. Fakat günümüzde enkaz içinde lojistik yardım sağlamak [1], moloz kaldırmak [2] ya da içeride kalan insanları dışarı çıkarmak [3] gibi farklı görevleri yerine getirebilen robotlar da mevcuttur. Arama ve kurtarma robotları zor çalışma koşullarından dolayı çalıştıkları alana göre özel olarak imal edilirler.

Arama ve kurtarma robotları hakkında ilk araştırmalar 1995 yılındaki Kobe depreminden sonra yapılmıştır [4]. 90'lı yılların başında robotik sistemler ve otonom mobil araçlar konularında yaşanan gelişmeler arama ve kurtarma görevi yapabilecek robotlar hakkında çalışmalar yapılmasında motive edici olmuştur. Ayrıca 2000 yılından sonra da birçok uluslararası mobil robot yarışmasına arama ve kurtarma robotları kategorisi eklenmiş ve bu sayede bu konuda yapılan bilimsel araştırmalarının sayısı arttırılmıştır [5]. Resmi olarak arama ve kurtarma robotları gerçek bir enkaz alanında ilk kez 11 Eylül 2001 tarihinde Dünya Ticaret Merkezi'ne yapılan saldırılardan sonra kullanılmıştır [4].

Bu bölümde ilk olarak arama ve kurtarma robotlarında sıklıkla kullanılan sensörler ve bu sensörlerin kullanılma nedenleri anlatılacaktır. Burada ayrıca mobil robot uygulamalarında sıklıkla kullanılan kamera ekipmanlarının öneminden

bahsedilecektir. Daha sonra da robotun çalışması sırasında meydana gelen yönelim değişikliklerinin kamera görüntüsü üzerindeki olumsuz etkileri açıklanacak ve bu soruna ait literatürde mevcut olan çözüm yöntemleri incelenecektir.

1.1 Arama ve Kurtarma Robotlarında Kullanılan Sensörler

Robotlar kendilerini çevreleyen bölge hakkında bilgi edinmek için sensörlerden yardım alırlar ve bu sensörlerden aldıkları bilgilere göre görevlerini yerine getirirler. Bu yüzden robotun üzerinde bulunan sensörlerden elde edilen verilerin optimizasyonu robotun istenen şekilde çalışmasında önemli rol oynar. Arama ve kurtarma robotları da kullanım alanlarına ya da kullanım şekillerine göre farklı sensörler barındırırlar. Genellikle bu tip robotlarda bulunan sensörler üç tane farklı görevi yerine getirmek için kullanılır. Bunlar önem sırasına göre aşağıda sıralanmıştır.

- Enkaz içinde mahsur kalan canlıların durumlarının ve konumlarının belirlenmesi
- Robotun etrafında bulunan objelere karşı operatörün uyarılması
- Robotun durumunun belirlenmesi

Yukarıda belirtilen her görev için robotlar üzerinde farklı sensörler bulunur. Burada bu görevler için en yaygın kullanılan sensörlerden bahsedilmiştir.

Arama ve kurtarma sırasında kullanılan robotlar için tanımlanmış ana görev enkaz içindeki canlıların konumunun belirlenmesidir. Bunun için farklı özellikle sensörler kullanılabilir. Yapılan uygulamalarda enkaz içindeki insanların yerleri belirlenirken genellikle ses algılama [6] ya da ısı algılama [7] yöntemine başvurulur. Ayrıca robotun ilerleyeceği rotanın tayin edilmesi için de robotun üzerine hava kalitesini ölçen sensörler monte edilir [8]. Hava kalitesini ölçen sensörler robotun bulunduğu yerdeki hava koşullarının insanların yaşaması için uygun olup olmadığını belirler. Buradan elde edilen bilgilere göre robot bulunduğu bölgede arama yapmaya devam eder ya da başka bir bölgeye doğru hareket eder. Bu sayede robotun insanların yaşayamayacağı bölgelerde arama yaparak zaman kaybetmesi önlenir.

Arama ve kurtarma robotları için uygulama sırasında ikinci önceliğe sahip görev robotun korunmasıdır. Bunun için robotu kontrol eden kişinin robotun çevresindeki

tehlike arz edecek durumlara karşı uyarılması gerekir. Aksi takdirde robot öngörülme­yen bir durum yüzünden kullanılamaz hale gelebilir. Robotun çevresindeki tehlikeleri algılaması için robotun üzerine çeşitli sensörler monte edilebilir. Robotun etrafındaki objeleri ya da boşlukları algılamak için genellikle ultrasonik mesafe ölçerler [9], kızılötesi sensörler [10], lazer mesafe ölçerler [11] kullanılır. Ayrıca özellikle yangın gibi alanlarda kurtarma görevi yapan robotlarda robotun gövdesinin yangından etkilenmemesi için sıcaklık sensörleri [12] kullanılır.

Enkaz alanında arama ve kurtarma görevi yapan robotların görev sıralamasında üçüncü önemli görev robotun durumunun kullanıcıya bildirilmesidir. Buradaki sensörler robotun batarya durumu, gövdenin duruşu, robotun hareket yönü gibi bilgileri kullanıcıya ileterek robotun kontrolünde kolaylık sağlarlar. Genellikle robotlarda batarya durumu bataryanın gerilimi ölçülerek bulunur bu yüzden mobil robot uygulamalarında özel durumlar dışında batarya durumuyla ilgili herhangi bir sensör kullanılmaz. Robotun gövdesinin ya da hareket eden mekanik parçalarının yöneliminin belirlenmesi için genellikle bir ataletsel ölçüm cihazı kullanılır [13]. Böylece robotun bulunduğu zemin üzerinde nasıl durduğu anlaşılabilir. Son olarak robotun hareket yönünün belirlenmesinde ise elektronik pusulalar kullanılır [7]. Pusulalar aynı zamanda robotun üzerinden geçtiği yolların haritasını çıkarmak için de kullanılır.

Bu sensörlere ek olarak yaklaşık her arama ve kurtarma robotu üzerinde robot ile kullanıcı arasında görüntü aktarımını yapacak kamera ekipmanları bulunur [7–13]. Kameralar robotun üzerindeki sensörler için tanımlanmış görevlerin hepsini gerçekleştirmede kullanılabilir. Bu yüzden arama ve kurtarma robotu uygulamalarında robottan elde edilen kamera verisi önemli bir yere sahiptir. Buna rağmen kameradan elde edilen veri kameranın görüş açısı ile sınırlıdır ve standart kameralarda insan gözünün elde ettiği görüş açısını elde etmek mümkün değildir. Ayrıca genellikle kameralar robotun üzerine sabitlenmiş olduğundan robotun gövdesinin yönelimindeki değişikliklere göre kameradan elde edilen görüntünün açısı da değişir. Bunun sonucunda da kameradan elde edilen görüntüye göre robotu kontrol etmek zorlaşır. Bu sorunun giderilmesi için kameradan elde edilen görüntüde meydana gelen değişimlerin giderilmesi gerekir. Bunun için çeşitli yöntemler mevcuttur. Gelecek bölümde görüntünün dengelenmesi için kullanılan genel çözüm yöntemlerinden bahsedilecektir.

1.2 Kamera Görüntüsünün Düzeltilmesi

Bu kısımda hareketli cihazlardaki kamera uygulamalarında yönelim değişikliklerinin kamera görüntüsünde meydana getirdiği olumsuz etkilerin giderilmesi için kullanılan yöntemler anlatılacaktır. Bunun için üç farklı yöntem mevcuttur. Buna göre kamera görüntüsünün dengelenmesinde görüntü işleme yöntemi, optik yöntemler ya da mekanik yöntemler kullanılabilir.

Görüntü işleme yönteminde kamera sabit tutulurken kameradan elde edilen görüntüler dengelenir. Kameranın yönelimi bir sensör vasıtasıyla ve ya elde edilen görüntülerin referans bir görüntü ile karşılaştırılmasıyla belirlenir [14–17]. Yönelim belirlendikten sonra ise kameranın görüntüsü buna göre düzenlenir. İşlemin gerçek zamanlı olarak yürütebilmesi için gereken işlemci yükü çok fazladır. Bunun için basit bir mikroişlemci uygulaması ile bu işlem gerçekleştirilemez. Ayrıca görüntü işleme yöntemini standart kameralarla gerçekleştirmek zordur, zira sadece tek kamera görüntüsü ile yapılan düzeltme işlemlerinde ilk duruma göre veri kayıpları meydana gelir. Fakat bu sistemde görüntünün düzeltilmesi için herhangi bir mekanik düzenek gerekmez. Bu yüzden işlem robotun gövdesinden bağımsızdır.

Optik yöntemlerde kameranın lensi hareket ettirilerek kameradan elde edilen görüntü dengelenmeye çalışılır. Bu yöntemde genellikle kamera üzerinde bulunan bir sensör vasıtasıyla kameranın yönelimi belirlenir. Daha sonra da kameranın içindeki lensler bu yönelim bilgisine göre hareket ettirilir [18,19]. Lensin hareket ettirilmesi işlemi kameranın içinde meydana geldiğinden bu tip sistemler de robotun gövdesinden bağımsız olarak çalışır. Fakat lenslerin hareket ettirilmesi için ya özel kameralar kullanılmalı ya da lenslere uygun küçük eyleyiciler tasarlanmalıdır. Bu yüzden bu yöntemin uygulanması diğer yöntemlere göre daha zordur.

Mekanik yöntemlerde ise kameranın kendisi hareket ettirilerek elde edilen görüntü dengelenir. Optik yöntemlerde olduğu gibi burada da kameranın yönelimi bir sensör vasıtasıyla belirlenir ve buradan elde edilen bilgiye göre kameranın bağlı olduğu platform hareket ettirilir [20–22]. Bu işlem basit bir mekanik hareket düzeneği ile gerçekleştirilebilir. Bu yüzden bu yöntem diğer yöntemlerde göre daha uygulanabilir yapıdadır. Fakat mekanik düzende meydana gelecek sıkışmalar ya da bu düzeneğin hasar görmesi sistemin çalışmasını direkt olarak engelleyecektir. Bu yüzden sistem robotun gövdesine ya da çalışma yöntemine direkt olarak bağlıdır.

Yapılan uygulamada robotun üzerindeki kamera kendini dengeleyen platform sistemi üzerine yerleştirilmiş ve bu sayede kameranın görüntüsü düzeltilmiştir. Literatürdeki robotik uygulamalarda benzer kamera dengeleme sistemlerine rastlanabilir. Özellikle dört rotorlu hava aracı (quadrotor) [21] ve mobil robot uygulamalarında [22] robotun üzerinde bulunduğu kameranın dengelenmesi önemlidir. Dört rotorlu hava araçlarında genellikle bu işlem kameranın bağlı olduğu platformun servo motorlardan oluşan bir dengeleme sistemi ile düzeltilmesi sonucunda gerçekleşir. Yapılacak olan mobil robot uygulamasında da aynı yöntem takip edilecektir.

1.3 Kendini Dengeleyen Platform

Kendini dengeleyen platformlar üzerlerinde bulunan cisimlerin yeryüzüne göre yönelimini sabit tutan elektromekanik cihazlara denir. Bu platformlar genellikle hareket eden araçlarda meydana gelen salınım hareketlerinin aracın içindeki cisimler üzerindeki etkisini gidermek için kullanılır. Literatürde bu tip platformlar ile yapılmış birçok farklı uygulama bulunmaktadır. Özellikle deniz araçlarında [23], görüntüleme cihazlarında [20–22], uydu alıcılarında [24] ve hareketli araçlara monte edilen silahlarda [25] kullanılır.

Yapılan uygulamalarda çoğu zaman hazırlanan bir gövde üzerine farklı eksenlerde dönmeyi sağlayacak dönüş elemanları ve bu dönüş elemanlarını çevirecek eyleyiciler yerleştirilir. Ayrıca sistemin üzerinde platformun yönelimini belirlemek için özel sensörler bulunur. Bu sayede platformun yeryüzüne göre yönelimi belirlenebilir ve eyleyiciler farklı eksenlerde rotasyon hareketleri yapılarak dengeleme işlemi gerçekleştirilebilir.

1.4 Amaç

Bu çalışmada mobil robotlarda kullanılmak üzere bir kamera dengeleme sistemi üretilenektir. Bu sayede robotun hareketi sırasında robot gövdesinin yöneliminde meydana değişikliklerin kameradan elde edilen görsel veri üzerindeki etkisi giderilecektir. Bunun için ilk olarak önceden hazırlanmış bir mobil robot gövdesinin üzerine servo motorlar ile hareket ettirilen bir dengeleme sistemi kurulacaktır. Sonra da bu sistemin üzerine monte edilen bir kamera servo motorlar ile dengelenecektir. Bu işlem sırasında gövdenin yöneliminin belirlenebilmesi için bir adet ataletsel

ölçüm cihazı (IMU) kullanılacaktır. Dengeleme işi başarılı olarak yapıldıktan sonra robotun çalışma performansı çeşitli yöntemlerle test edilecektir.

1.5 Bölümlerin Dağılımı

Bu çalışmada genel olarak mobil robotlarda kullanılacak bir kamera dengeleme sisteminin tasarım, üretim ve test süreçleri anlatılmıştır.

Hazırlanan kontrol sisteminin temeli kameraya ait yönelim hesaplamalarına dayanır. Bunun için 2. Bölümde ilk olarak yönelimin belirlenmesinde kullanılan sensörün çalışma prensibi ve yapılan hesaplamalar anlatılmıştır. Ayrıca bu kısımda sensöre ait verilerin birleştirilmesinde kullanılan Kalman filtresi gösterilmiştir.

Bir sonraki bölümde robotun tasarım süreci ve kullanılan ekipmanlar anlatılmıştır. Bu bölüm mekanik donanım, elektronik donanım ve yazılım olarak üç parçaya ayrılmıştır. Mekanik kısımda robotun gövdesi, kamera dengeleme sisteminin mekanik tasarımı ve yapısı açıklanmıştır. Elektronik kısımda ise robot üzerinde kullanılan elektronik ekipmanların özellikleri ve çalışma şekilleri gösterilmiştir. Son olarak yazılım kısmında robotun üzerinde bulunan mikroişlemcinin kodlama yapısı ve kullanılan haberleşme protokolleri anlatılmıştır.

Robotun tasarım süreci anlatıldıktan sonra 4. bölümde robotun çalışma şekline ve gerçeklemesine değinilmiştir. Burada kamera dengeleme sisteminin çalışma prensibi ayrıntılı şekilde anlatılmıştır. Daha sonra ise sistemdeki elektrik bağlantıları gösterilmiştir. Son olarak da kumanda devresinde ve robot üzerinde bulunan mikroişlemcilerin kodları ayrıntılı şekilde açıklanmıştır.

Yapılan çalışmanın son kısmında ise robotun test süreci ve sonuçlar anlatılmıştır. Burada ilk olarak Kalman filtresinin ataletsel ölçüm cihazından elde edilen veriler üzerindeki etkisi incelenmiştir. Daha sonra ise dengeleme sisteminin kamera görüntüsü üzerinde yaptığı iyileştirmeler gösterilmiştir. Test süreçleri anlatıldıktan sonra elde edilen sonuçlara göre gelecekte yapılabilecek iyileştirme çalışmaları anlatılmıştır.

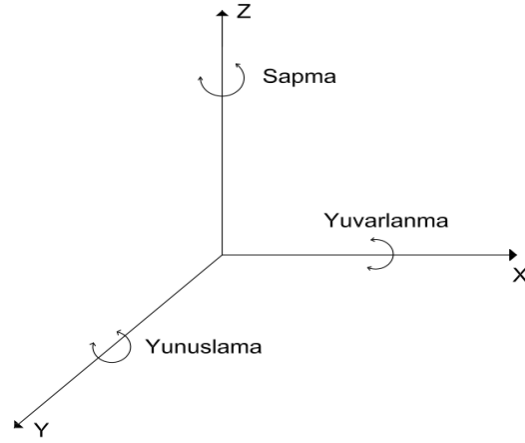
Ek A kısmında ise robot üzerinde ve kumanda devresinde kullanılan mikroişlemci kodları verilmiştir.

2. ATALETSEL ÖLÇÜM CİHAZI

Mobil robot üzerindeki kameranın yöneliminin hesaplanabilmesi için yön kosinüs gösterimi, vida eksen gösterimi, Euler açıları gösterimi gibi yöntemler mevcuttur. Burada yapılan hesaplamalarda Euler açıları gösterimi kullanılacaktır [26,27].

Euler açıları gösteriminde yönelim aynı merkeze sahip iki tane üç boyutlu eksen takımı kullanılarak hesaplanır. Buna göre bu eksen takımlarından birisi referans kabul edilir ve diğer eksen takımının bu referansa göre yönelimi belirlenir. Yapılan uygulamalarda genellikle dünya eksen takımı referans olarak alınır ve yönelimi belirlenecek cismin dönme merkezine sabitlenen ikinci bir eksen takımı kullanılarak cismin yeryüzüne göre yönelimi bulunur. Her iki eksen takımı arasında eksenler etrafındaki rotasyon hareketleri kullanılarak geçiş yapılabilir. Bu rotasyon hareketlerinin meydana getirdiği rotasyon açılara Euler açısı denir. Buna göre kameranın yöneliminin belirlenmesi için üç tane Euler açısının bulunması yeterli olacaktır.

Kamera dengeleme sisteminde Euler açıları rotasyon hareketinin yapıldığı eksene göre isimlendirilmiştir. Şekil 2.1’de kameranın dönme merkezine sabitlenmiş eksen takımına ait rotasyon hareketleri gösterilmiştir. Burada görüldüğü gibi Z eksenine etrafına yapılan rotasyon hareketine sapma hareketi, X eksenine etrafında yapılan rotasyon hareketine yuvarlanma hareketi ve Y eksenine etrafında yapılan rotasyon hareketine de yunuslama hareketi denir. Kamera dengeleme sisteminde sadece yuvarlanma ve yunuslama açısı kontrol edilmiştir. Sapma açısı ise operatörün isteğine göre uzaktan kumanda edilerek değiştirilmiştir. Kameranın sapma hareketi aynı zamanda kameranın bakış yönünü değiştirir. Bu eksende meydana gelen hareketlerin dengelenmeye çalışması sonucunda kamera robotun hareketi sırasında sürekli olarak tek bir yöne doğru bakacaktır. Bunun sonucunda da kameranın bakış yönü çalışma boyunca sabit kalacaktır ve robotun sağa ya da sola doğru hareketlerinde robotun ilerleme yönü görülemeyecektir. Bu yüzden dengeleme sisteminde sapma açısı kontrol edilmemiştir ve robotun hareketi boyunca bu açı sıfır olacak şekilde sabit tutulmuştur.



Şekil 2.1 : Üç eksen de rotasyon hareketleri.

Bir eksen takımının kendi eksenlerinden biri etrafındaki dönüş hareketini belirtmek için rotasyon matrisleri kullanılır. Bu matrislerin yapısı rotasyon hareketinin yapılacağı eksene bağlıdır. Buna göre üç boyutlu bir eksen takımı için rotasyon matrisleri aşağıdaki gibi yazılabilir.

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

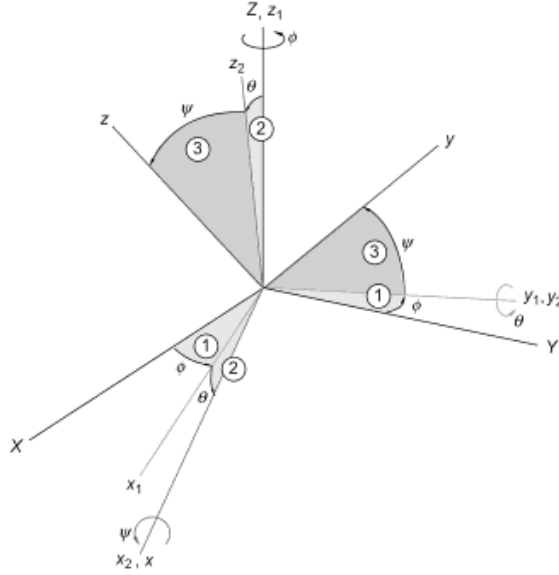
$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

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

Yukarıdaki matrislerde gösterilen ψ , ϕ ve θ açıları eksenlerin etrafındaki rotasyon hareketlerini tanımlayan Euler açılarıdır. Kameranın referans eksen takımına göre yönelimi hesaplanırken yukarıda verilen rotasyon matrislerinin birleşimi kullanılır. Buna göre bu rotasyon matrisleri belirli bir sıralamayla çarpılarak yeni bir matris oluşturulur. Daha sonra bu matris kullanılarak iki eksen takımı arasındaki geçiş yapılır. Rotasyon matrislerinin sıralamasına göre farklı matrisler elde edilebilir.

Yunuslama, yuvarlanma ve sapma açıları ile çalışılan uygulamalarda referans eksen takımından kameraya ait eksen takımına geçebilmek için rotasyon matrisleri Z-Y-X sıralamasına göre çarpılır. Şekil 2.2’de bu sıralamaya göre yapılan rotasyonlar gösterilmiştir [27]. Burada gösterilen sıralamaya göre referans eksen takımı (XYZ)

ilk olarak Z eksenini etrafında ϕ açısı kadar dönme hareketi yapar. Bunun sonucunda X eksenini x_1 konumuna Y eksenini ise y_1 konumuna gelir. Rotasyon hareketi Z eksenini etrafında yapıldığı için bu eksenin konumu değişmez. İkinci rotasyon hareketi Y eksenini etrafında θ açısı kadar yapılır. Bu hareketin sonucunda da x_1 eksenini x_2 konumuna ve Z eksenini de z_1 konumuna gelir. Son olarak X eksenini etrafında ψ açısı kadar rotasyon hareketi yapılır ve cismin yönelimini belirten eksen takımına ait x, y ve z eksenleri elde edilir.



Şekil 2.2 : Z-Y-X sıralamasına göre yapılan rotasyon hareketleri.

Burada belirtilen rotasyon matrisi sıralamasına göre iki eksen takımı arasında geçiş yapmak için kullanılacak matris Denklem (2.4)'teki gibi yazılabilir. Burada yazımın basitleştirilmesi için kosinüsler “c” olarak ve sinüsler de “s” olarak gösterilmiştir.

$$R(\phi, \psi, \theta) = \begin{bmatrix} c\phi c\theta & s\phi c\theta & -s\theta \\ c\phi s\theta s\psi - s\phi c\psi & s\phi s\theta s\psi + c\phi c\psi & c\theta s\psi \\ c\phi s\theta c\psi + s\phi s\psi & s\phi s\theta c\psi - c\phi s\psi & c\theta c\psi \end{bmatrix} \quad (2.4)$$

Yukarıdaki matris kullanılarak XYZ referans eksen takımı xyz eksen takımı cinsinden yazılabilir.

$$E_{xyz} = R(\phi, \psi, \theta)E_{XYZ} \quad (2.5)$$

Denklem (2.5)'te iki eksen takımındaki koordinatları göstermek için E vektörleri tanımlanmıştır. Kameranin dünya koordinat sistemine göre yöneliminin bulunabilmesi için Denklem (2.5)'te verilen dönüşümün diğer yönde olması gerekir.

Bu yüzden yukarıda verilen matrisin tersi alınır. Rotasyon matrislerinin tersi aynı matrisin devriğine eşittir. Bu özellik kullanılarak Denklem (2.6b)'de verilen matris bulunur. Buradaki R_{xx} matrisi kameranın yönelimini gösteren eksen takımından dünya eksen takımına geçiş yapmak için kullanılır.

$$R_{xx} = R(\phi, \psi, \theta)^T \quad (2.6a)$$

$$R_{xx} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix} \quad (2.6b)$$

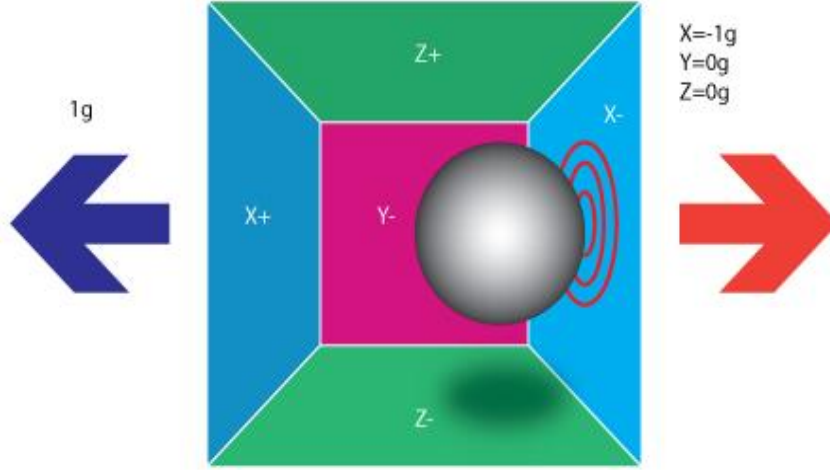
Denklem (2.6b)'deki matriste verilmiş Euler açıları hesaplanarak kameranın yeryüzüne göre yönelimi bulunabilir. Yapılan çalışmada bu işlem için ataletsel ölçüm cihazı kullanılmıştır.

Ataletsel ölçüm cihazları bağlı oldukları gövdenin hızını, belli bir referansa göre yönelimini ve gövdeye uygulanan yerçekimsel kuvvetleri ölçmek için kullanılır. Genellikle bu cihazlarda birer adet jiroskop ve ivmeölçer birlikte kullanılır. Bazı ataletsel ölçüm cihazlarında ise sapma ekseninde meydana gelen hataları gidermek için manyetometre de bulunur. Ataletsel ölçüm cihazları isimlendirilirken üzerinde bulunan sensörlerin ölçüm yapabildikleri serbestlik derecesi sayısı temel alınır. Örneğin üzerinde 3 serbestlik dereceli bir ivmeölçer ve 2 serbestlik dereceli bir jiroskop bulunduran bir ataletsel ölçüm cihazı 5 serbestlik dereceli kabul edilir. Cihaz üzerinde bulunan sensörler arasında gövdenin yönelimini direkt olarak ivmeölçer üzerinden ölçmek mümkündür fakat sensörde meydana gelen hataları düzeltmek için jiroskop kullanılması gerekir.

2.1 İvmeölçer

İvmeölçer bağlı olduğu cismin farklı eksenler üzerindeki ivmelenmesini hesaplamak için kullanılır. Burada ivmelenme statik (yerçekimsel) ve dinamik (harekete bağlı) olarak ikiye ayrılır. Cismin yeryüzüne göre yönelimini belirlemek için gereken veri yerçekimsel ivmedir. Bunun için hareketten doğan dinamik ivmelenmenin etkilerinin göz ardı edilip sadece yerçekimsel ivmeden elde edilen verilerin kullanılması gerekmektedir. İvmeölçerler bu iki veriyi de ölçtükleri için hareketten doğan ivmelenmenin filtrelmesi gerekir [28].

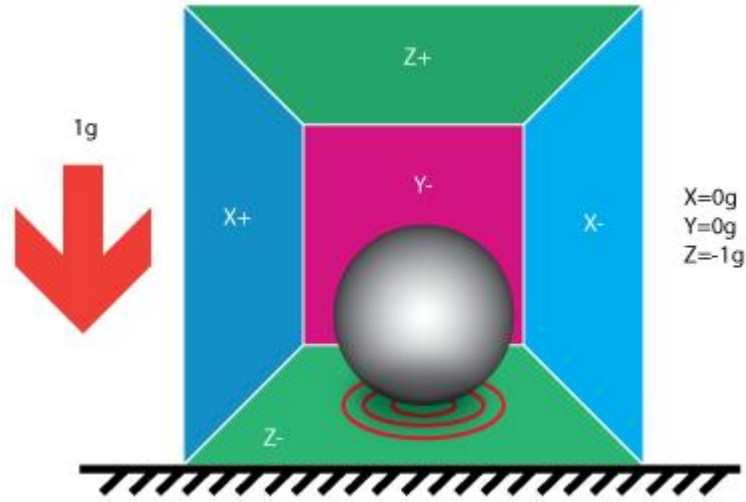
ivmenin çarpımına eşittir. Buna göre m kg ağırlığına sahip bir küre X - yönünde $1 m \times g$ büyüklüğünde bir kuvvet uygulayacaktır. Ayrıca burada görüldüğü gibi ivmeölçerden elde edilen verinin yönü hareket yönünün tersine doğru olacaktır. Bunun için ivme hesaplaması yapılan uygulamalarda ölçülen verilerin buna göre düzenlenmesi gerekir.



Şekil 2.4 : İvmeölçerin $X+$ yönünde hareketi sonucunda uygulanan kuvvetler.

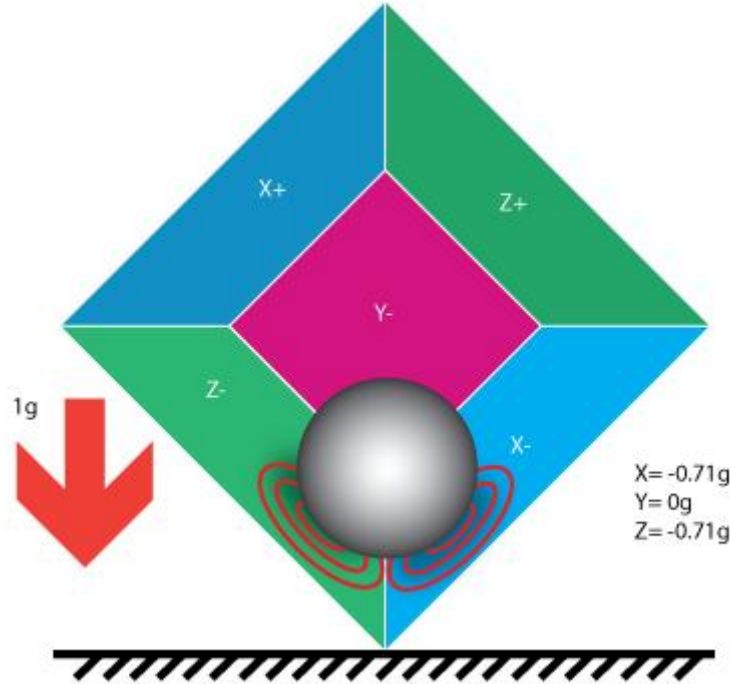
Yukarıda verilen örnekte cismin hareketi sonucunda meydana gelen dinamik ivmelenme ve bunun sonucunda eksenlere uygulanan kuvvetler gösterilmiştir. Yönelimin bulunabilmesi için ise yerçekimsel yani statik ivmenin belirlenmesi gerekir. Eğer ivmeölçerin bağlı olduğu cisim, düz bir alana konulursa dünyanın yerçekiminden dolayı küre tarafından Z - eksenine $1 m \times g$ büyüklüğünde bir kuvvet uygulanır. Burada önemli olan ivmeölçere ait eksen takımının ve yeryüzü eksen takımının eksenlerinin üst üste çakışık halde olmasıdır; zira ivmeölçerin ters şekilde ya da başka bir eksen üzerinde yerleştirmesi sonucunda kuvvetin uygulandığı eksen değişecektir. Bunun sonucunda uygulanan kuvvet Z - yönü yerine başka bir yönde olabilir. Normalde cismin statik ivmesi cisim aynı pozisyonda tutulsa bile cismin bulunduğu konuma göre değişkenlik gösterir. Bunun nedeni yerçekimi kuvvetinin farklı yüksekliklerde değişmesidir. Fakat referans olarak kullanabilmek için ölçüm yapılırken uygulanan kuvvet $1 m \times g$ olarak alınır.

Şekil 2.5'te düz bir alana yerleştirilen cisme uygulanan kuvvetlerin kutu ve küre modelindeki karşılığı verilmiştir. Burada görüldüğü gibi ivmeölçer eksen takımı ve yeryüzü eksen takımı birleşik olduğunda küre Z - yönünde bir kuvvet uygulayacaktır. Bu kuvvetin büyüklüğü önceden belirtilen büyüklükte seçilebilir.



Şekil 2.5 : İvmeölçerin düz yerleştirilmesi sonucunda uygulanan kuvvetler.

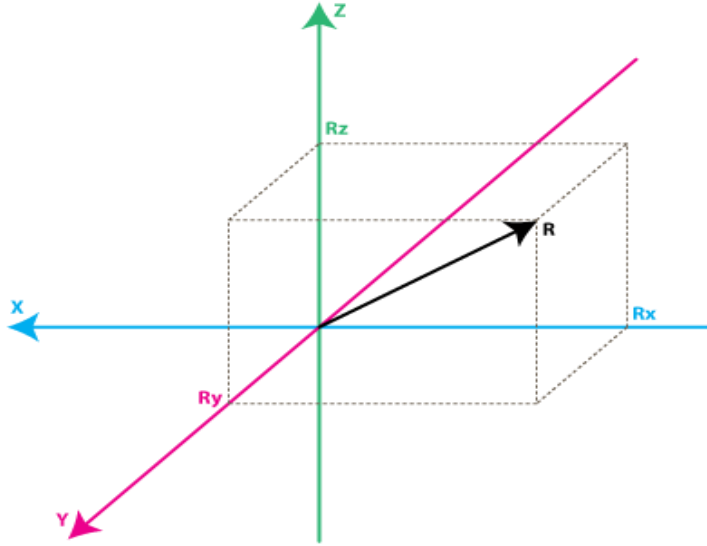
İvmeölçerin bağlı olduğu cisim Y eksenini üzerinde 45 derecelik bir açıyla pozitif yönde çevrildiğinde küre tarafından X ve Z eksenlerine $0,71 \text{ m} \times g$ büyüklüğünde bir kuvvet uygulanacaktır. Şekil 2.6’da söz konusu durumda uygulanan kuvvetler model üzerinde gösterilmiştir.



Şekil 2.6 : Y+ yönünde 45 derecelik dönme sonucunda uygulanan kuvvetler.

İvmeölçerden elde edilen verilerden yönelimin matematiksel olarak hesaplanabilmesi için Şekil 2.7’de verilen koordinat sistemi modelinin kullanılması gerekir. Bu modelde görülen koordinat eksenleri yeryüzü eksen takımını temsil eder. Şekilde verilmiş olan R vektörü ivmeölçerden elde edilen ivme vektörüdür. R_x , R_y ve R_z

değerleri ise R vektörünün bileşenlerini gösterir. Kutu ve küre modelinde sağ kısımda gösterilen ivmeölçer verileri R_x , R_y ve R_z bileşen vektörlerini hesaplamak için kullanılır.



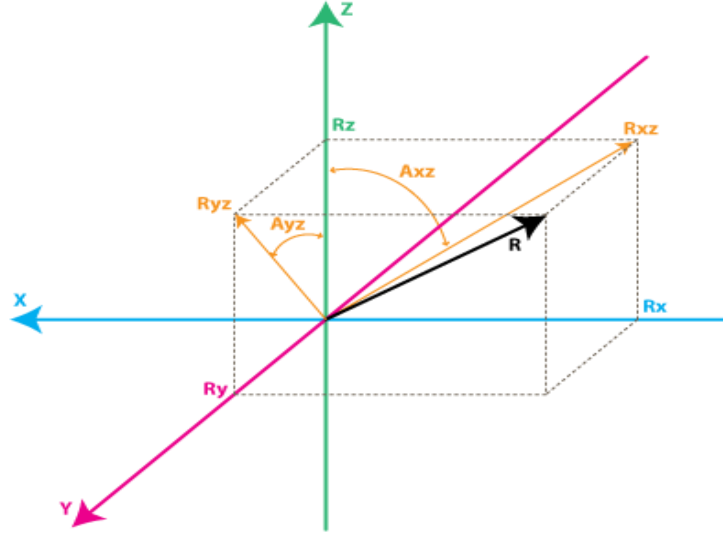
Şekil 2.7 : İvmeölçerden elde edilen ivme vektörünün görünümü.

Şekildeki R vektörü cismin hareketi sonucunda meydana gelen ivmenin ve yerçekimsel ivmenin bir bileşkesidir. Cismin yeryüzüne göre yöneliminin bulunabilmesi için cismin hareketinin R vektörü üzerindeki etkisinin giderilmesi gerekir. R vektörünün büyüklüğü Pisagor teoreminin yardımıyla Denklem (2.7)'deki gibi yazılabilir.

$$R^2 = R_x^2 + R_y^2 + R_z^2 \quad (2.7)$$

İvmeölçerin bağlı olduğu cisim hareketsiz bir şekilde düz bir alana yerleştirildiğinde elde edilen R vektörü sensöre ait koordinat sistemindeki Z ekseninin dünya eksen takımına göre yönelimini gösterir. Cismin yeryüzüne göre yöneliminin değişmesi sonucunda bu vektörün yönü ve büyüklüğü değişir. Denklem (2.6b)'de verilmiş Euler açıları R vektörünün farklı düzlemler üzerine izdüşümü bulunarak hesaplanabilir. Kamera dengeleme sisteminde sadece yunuslama ve yuvarlanma açıları kontrol edildiği için burada bu açılara ait hesaplamalar verilmiştir. Buna göre yunuslama açısı R vektörünün XZ düzlemi üzerindeki izdüşümünün Z eksenine yaptığı açıdır. Benzer şekilde R vektörünün YZ düzlemi üzerindeki izdüşümünün Z eksenine yaptığı açıya da yuvarlanma açısı denir. Şekil 2.8'de cismin X ve Y ekseninde yaptığı rotasyon hareketleri açısal olarak gösterilmektedir. Burada A_{xz}

yunuslama açısını ve A_{yz} yuvarlanma açısını temsil eder. Bu açılar aynı zamanda Denklem (2.6b)'deki X ve Y eksenleri için tanımlanmış Euler açılarıdır. R_{xz} ve R_{yz} vektörleri ise R vektörünün sırasıyla XZ ve YZ düzlemlerindeki izdüşümünü gösterir.



Şekil 2.8 : X ve Y eksenindeki rotasyonların koordinat sisteminde gösterimi.

Ölçüm yapılan cisim X eksenini etrafında çevirilince A_{yz} açısı, Y eksenini etrafında çevirilince de A_{xz} açısı değişecektir. Buna göre Denklem (2.8) ve Denklem (2.9) kullanılarak A_{xz} ve A_{yz} Euler açıları hesaplanabilir.

$$A_{xz} = \arctan\left(\frac{R_x}{R_z}\right) \quad (2.8)$$

$$A_{yz} = \arctan\left(\frac{R_y}{R_z}\right) \quad (2.9)$$

İvmeölçer kullanılan uygulamalarda Denklem (2.8)'de ve Denklem (2.9)'da gösterilen vektör bileşenlerinin birbirine oranlanması sonucunda meydana gelebilecek hatalara dikkat edilmesi gerekir; zira sınır değerlerde çalışırken sensör gürültüsünün hesaplanan açı üzerindeki etkisi çok artmaktadır. Söz konusu durumu anlatmak için Şekil 2.5'te gösterilen durum kullanılacaktır. Buna göre Şekil 2.5'te gösterilen örnekte ivmeölçerden elde edilen R vektörü $[0 \ 0 \ -1]^T$ olarak yazılabilir. Bu durumda sensörün bağlı olduğu cisim Y eksenini etrafında pozitif yönde 90 derece döndürülürse R vektörü $[1 \ 0 \ 0]^T$ olarak değişir. Fakat uygulama sırasında ivmeölçerde meydana gelen ölçüm hatalarından dolayı R_y ve R_z değerleri tam olarak 0 değerini almaz. Sensör gürültüsünden dolayı R_y ve R_z değerleri -0,05 g ile +0,05 g

arasında sürekli deęiřir. Bu durumda Denklem (2.8)'e gre A_{xz} aısı istenen aralıktadır. Fakat Denklem (2.9)'da bulunan R_y/R_z oranı sıfır olması gerekirken sensr grltsnden dolayı byk deęerler alabilir. Bunun sonucunda da A_{yz} aısı iki byklęn oranına gre rastgele bir deęer alır. Yapılan incelemelerde bu hatanın sadece 90 derecelik rotasyonlarda meydana geldięi grlmřtr. Bunun iin kamera dengeleme sisteminin alıřma aralıęı bu deęerleri ařmayacak řekilde seilmiřtir. Ayrıca robotun normal hareketi sırasında bu byklkte bir rotasyon meydana gelemeyeceęi iin burada oluřabilecek hatalar gz ardı edilmiřtir.

 eksenli ivmelerden yapılan lm sonucunda R vektrnn bileřenlerini oluřturan R_x , R_y ve R_z vektrlerinin byklkleri elde edilir. Bu deęerler mikrořlemciye analog ya da dięital olarak iletilir. Analog veri kullanılması durumunda mikrořlemci, vektrlere ait byklkleri belirli bir skalada deęiřen gerilim olarak grr. Dięital veri iletiminde ise vektrlerin byklkleri sayısal veri paketleri olarak mikrořlemciye gnderilir. Normal uygulamalarda llen ivme deęerleri m/s^2 cinsinden elde edilmez. llen ivme deęerlerini m/s^2 ya da g cinsinden ifade edebilmek iin ivmelerden iletilen veriler mikrořlemci tarafında belli bir hassasiyet deęerine blnr. Bu hassasiyet deęeri genellikle sensrn veri fynde belirtilmiřtir [21]. Ařaęıdaki denklemlerde her eksen iin ivmelerden llen verilerin ivme deęerlerine dnřtrlme yntemi verilmiřtir.

$$R_x = DV_{R_x} / S \quad (2.10)$$

$$R_y = DV_{R_y} / S \quad (2.11)$$

$$R_z = DV_{R_z} / S \quad (2.12)$$

Yukarıdaki denklemlerde sensrden elde edilen veri “DV” olarak tanımlanmiřtır. Hassasiyet deęeri ise “S” řeklinde gsterilmiřtir. Bu denklemler kullanılarak her eksen iin ivme deęeri g cinsinden hesaplanabilir.

2.2 Jiroskop

Jiroskoplar baęlı olduęu gvdenin farklı eksenlerde yaptıęı rotasyon hareketlerinin aısal hızlarını lmek iin kullanılırlar [28]. Bu sayede cismin sz konusu eksenler etrafında ne kadar hızlı dndę belirlenebilir. Sensrden elde edilen deęerler

genellikle $^{\circ}/s$ ya da rad/s olarak ifade edilir. Jiroskoplarda da ivmeölçerlerde olduğu gibi bir, iki ve üç eksenle ölçüm yapabilen modeller mevcuttur.

Jiroskop yardımıyla cismin yeryüzüne göre yönelimi hakkında dolaylı şekilde bilgi edinmek mümkündür. Fakat kullanılan yöntem ivmeölçerdekinden farklıdır. İvmeölçerler bağlı oldukları gövdenin yerçekimsel ivmesini ölçebilirler. Buradan elde edilen ölçümler kullanılarak gövdenin yeryüzüne göre yönelimi elde edilebilir. Jiroskoplarda ise bağlı oldukları cismin belli bir eksen etrafında yaptığı dönme hareketi açısı hesaplanabilir. Jiroskoptan elde edilen açısal hız değerlerinin belli bir zaman aralığında integrali alınarak bu aralıkta cismin kat ettiği rotasyon açısı bulunur. Sonrasında bu değer her zaman aralığı için toplanır ve bu şekilde cismin söz konusu eksenle ne kadar dönme hareketi yaptığı belirlenebilir. Bunun sonucunda Denklem (2.8)'de ve Denklem (2.9)'da verilmiş olan rotasyon açılarının değişimi kolayca hesaplanabilir; fakat jiroskop ile ölçüm yapılırken bu açıların başlangıç değerleri belirlenemez. Bu yüzden de cismin yeryüzüne göre yönelimini sadece jiroskop kullanarak ölçmek mümkün değildir. Sonuç olarak ivmeölçer kullanarak elde edilen yönelim bilgisi sabit iken, jiroskoptan elde edilen rotasyon bilgisi kümülatif bir yapıdadır.

Jiroskoptan elde edilen açısal hız verisinin açı değerine dönüştürülmesi için elde edilen değerlerin integralinin alınması gerekmektedir. Bunun için ilk olarak sabit bir zaman aralığı belirlenir. Bu süre içinde yapılan ölçümlerin ortalaması alınarak bir açısal hız değeri hesaplanır. Sonrasında ise ortalama ile elde edilen açısal hız değeri belirlenen zaman aralığıyla çarpılarak integralin sonucu bulunur. İntegralden elde edilen değer ölçüm yapılan zaman aralığı için eksen etrafındaki rotasyon açısının değişimini verir. Bu yüzden buradan edilen açı değerleri her zaman aralığı için toplanır ve başlangıç anından itibaren cismin ölçüm yapılan eksen üzerinde yaptığı dönme hareketi açısal olarak hesaplanır.

Yukarıda belirtilen yöntemler kullanılarak cismin yunuslama ve yuvarlanma açılarındaki değişimler Denklem (2.13)'teki ve Denklem (2.14)'teki gibi hesaplanabilir. Burada cismin sabit zaman aralığı içinde Y eksen etrafında yaptığı rotasyonun ortalama açısal hızı W_{xz} olarak ve cismin aynı zaman aralığı içinde X eksen etrafında yaptığı rotasyonun ortalama açısal hızı da W_{yz} olarak gösterilmiştir. Benzer şekilde ΔA_{xz} yunuslama açısındaki değişimi, ΔA_{yz} ise yuvarlanma açısındaki değişimi temsil eder. Denklemlerde T sabiti sabit zaman aralığını, n değişkeni ise

ölçüm adımını belirtmek için kullanılmıştır. Burada değişimi hesaplanan A_{xz} ve A_{yz} açıları Şekil 2.8’de verilmiş olan Euler açılarıdır.

$$\Delta A_{xz}(n) = W_{xz}(n)T \quad (2.13)$$

$$\Delta A_{yz}(n) = W_{yz}(n)T \quad (2.14)$$

Jiroskop ile yapılan rotasyon açısı hesaplamalarında sensörün tek başına kullanılması zamanla hatalı sonuç elde edilmesine yol açacaktır. Bunun en büyük nedeni rotasyon açısı hesaplanırken integrale dayalı hesaplama yapılmasıdır [30]. İntegral işlemi kullanılan hesaplamalarda sensörden elde edilen veriler her zaman aralığı için toplanır. Bu sırada sensörde meydana gelen ölçüm hataları da integral işlemine girer ve elde edilen sonuçlar üzerindeki hata değeri zamanla artar. Jiroskoplarda meydana gelen bu hataya kayma hatası denir. Kayma hatası giderilmediği sürece ölçüm süresi arttıkça elde edilen sonuç üzerindeki hata da artacaktır. Yapılan uygulamalarda hatasız sonuç elde edebilmek için sensörün kayma hatasının düzeltilmesi gerekir.

Jiroskoplarda da ivmeölçerdekine benzer şekilde ölçülen veri açısal hız birimi cinsinden elde edilmez. Elde edilen verilerle hesaplama yapabilmek için mikroişlemci tarafına iletilen ölçümlerin sensöre ait hassasiyet değerine bölünmesi gerekir.

2.3 Kalman Filtresi Metodu

Bir gövdenin yerçekimsel ivmesini belirlemek için tek başına bir jiroskop ya da ivmeölçer kullanmak yeterli olmayacaktır. Bunun için iki sensörden elde edilen verilerin birlikte kullanılması gerekir.

İvmeölçerler yerçekimsel ivmeyi ölçerek sabit bir cismin yeryüzüne göre yönelimi hakkında bilgi verebilir, fakat sensörün bağlı olduğu cismin farklı hızlarda hareket etmesi ya da sürekli olarak sallanması durumunda elde edilen bilgi hatalı olacaktır. Bunun nedeni ivmeölçerlerin hem yerçekimsel ivmeyi hem de hareket sonucunda meydana gelen dinamik ivmeyi ölçmesidir. Buna göre hareketli bir cisim üzerinde yapılan ölçümlerde sadece yerçekimsel ivme değerlerini elde etmek mümkün olmayacaktır. Jiroskoplara ise ivmeölçerlere göre lineer hareketlerden daha az etkilenirler. Bu sayede jiroskoplara sarsılma ya da lineer hareket sırasında ivme

değişikliği gibi durumlarda ivmeölçere göre daha düzgün ölçüm yapabilirler. Fakat jiroskop ile yapılan hesaplamalarda integral işlemleri kullanıldığından ortam koşullarındaki değişiklikler, sensör gürültüsü gibi nedenlerden meydana gelen ölçüm hatalarının rotasyon açısı üzerindeki etkisi fazladır. Buna göre özellikle zamana bağlı uygulamalarda sadece jiroskoptan elde edilen verilere bakılarak hesap yapılması hatalı sonuçlar elde edilmesine neden olur. Ayrıca önceden belirtildiği gibi cismin başlangıç anındaki yönelimi bilinmeden sadece jiroskoplar kullanarak yönelimin belirlenmesi mümkün değildir.

İki sensörden elde edilen verilerin birleştirilmesi için çeşitli algoritmalar mevcuttur. Bunlardan en yaygın olarak kullanılanları tamamlayıcı filtre [31] ve Kalman filtresidir. Yapılan uygulamada sensör verilerinin birleştirilmesi için Kalman filtresi kullanılmıştır [32].

Kalman filtresi, durumu rastgele olarak değişen dinamik bir sistemin son durumunu tahmin etmek için kullanılan bir matematiksel denklemler kümesidir. Filtre, uygulandığı sistemi tam olarak belirten bir model mevcut olmasa da sistemin farklı zamanlardaki durumları hakkında kestirim yapabilir. Kalman filtresi, ayrık ve sürekli olmak üzere ikiye ayrılmaktadır. Bu kısımda sadece çalışmada kullanılan ayrık Kalman filtresi anlatılacaktır.

Durum ve ölçüm denklemleri ile ifade edilmiş bir ayrık zaman sistemi Denklem (2.15)'te ve Denklem (2.16)'da gösterildiği gibi tanımlanabilir. Burada Denklem (2.15) sisteme ait durum denklemini, Denklem (2.16) ise sensöre ait ölçüm denklemini gösterir.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2.15)$$

$$z_k = Hx_k + v_k \quad (2.16)$$

Burada A durum geçiş matrisini, B kontrol matrisini, u kontrol vektörünü ve H ölçüm hassasiyet matrisini belirtir. Ayrıca denklemlerde ölçüm adımını belirtmek için k indisi kullanılmıştır. Buna göre sistemin son durumu bir önceki durumdaki değeri ile uygulanan kontrol sinyalinin lineer bir kombinasyonudur. Ayrıca bu lineer denkleme w_{k-1} olarak tanımlanmış işlem gürültüsü de eklenir. Benzer şekilde ölçüm değeri de sistemin durumu ve v_k olarak tanımlanmış ölçüm gürültüsünün lineer bir kombinasyonu olarak tanımlanır. İki denklemde bulunan gürültü değerleri

birbirinden bağımsız kabul edilir. İki gürültü değerine ait olasılık denklemi aşağıdaki Gauss dağılımı cinsinden aşağıdaki gibi yazılır.

$$p(w) \approx N(0, Q) \quad (2.17)$$

$$p(v) \approx N(0, R) \quad (2.18)$$

Verilen Gauss dağılımlarında Q değeri işlem gürültü kovaryansını, R değeri ise ölçüm gürültü kovaryansını temsil eder.

Kalman filtresi ile çalışma yapılırken sistemin durum değişkeni x_k tahmini olarak hesaplanır. Her ölçüm adımı için iki tane tahmin değeri mevcuttur. Kalman filtresi denklemlerinde her ölçüm için bu iki değer arasında geçiş yapılır. İlk ölçüm için ise bir başlangıç değeri belirlenir ve buna göre hesaplama yapılır. Genellikle başlangıç değeri sensörlerden bir ölçüm adımı için elde edilen verilerin işlenmemiş halidir.

Kalman filtresiyle ilgili denklemler zaman güncelleme denklemleri ve ölçme güncelleme denklemleri olarak iki grup altında toplanabilir. Kalman filtresi ilk olarak zaman güncellemesinden başlar daha sonra ölçme güncellemesine geçer. İlk adımdan sonra bu işlemler bir döngü halinde devam eder.

Zaman güncelleme denklemleri, bir sonraki adım için öncelikli tahminleri belirler. Bunun için ilk olarak bir önceki ölçümden elde edilen tahmin değeri yeni ölçüm adımı için güncellenir.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (2.19)$$

İkinci adımda ise yeni hata kovaryansı hesaplanır.

$$P_k^- = AP_{k-1}A^T + Q \quad (2.20)$$

Zaman güncelleme denklemleri hesaplandıktan sonra ölçme güncelleme denklemlerine geçilir. Bu denklemlerde tahmin değeri ile ilgili geri besleme yapılır. Burada sistem durum değişkenlerine ait ölçümden önceki tahmin değeri ile ölçümden sonraki tahmin değeri arasında geçiş yapılır. Tahmin değerinin hangi değere bakılarak güncelleneceğini Kalman kazancı belirler. Buna göre Kalman filtresi her ölçüm adımı ya eski durum tahminine göre ya da sensörden yapılan ölçümlere göre güncelleme işlemini yerine getirir.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (2.21)$$

Bir ölçüm adımı için Kalman kazancı hesaplandıktan sonra durum tahmin değerleri arasında geçiş yapılabilir.

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad (2.22)$$

Burada verilen z_k değeri ve H matrisi Denklem (2.16)'da tanımlanmış olan ölçüm değeri ve ölçüm hassasiyet matrisidir.

Denklem (2.21)'de gösterilen Kalman kazancı filtrenin bir sonraki ölçüm adımı için davranışını belirler. Ölçüm adımı için belirlenen ölçme hata kovaryansının (R) sıfıra yakın olması sensöre ait ölçüm hatasının az olduğunu gösterir. Bu yüzden bu değer büyüklüğü limitte sıfıra gittikçe Kalman kazancı H^{-1} değerini alır ve sonuç olarak Denklem (2.22) aşağıdaki gibi yazılabilir.

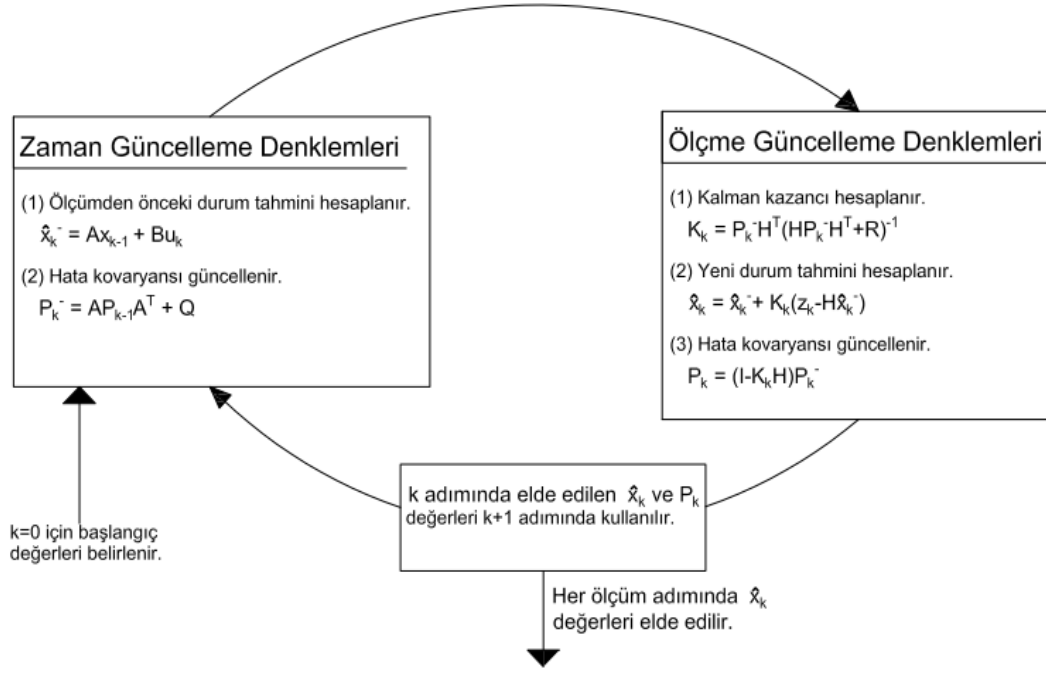
$$\hat{x}_k = H^{-1} z_k \quad (2.23)$$

Denklem (2.23) gelecek bir ölçüm adımı için durum tahmin değerinin sadece sensör ölçümüne bakılarak güncelleneceğini gösterir. Benzer şekilde zaman güncelleme denklemlerinde hesaplanan hata kovaryansı (P_k^-) değerinin limitte sıfıra yaklaşması bir önceki sistem durum tahminine ait değerlerin güvenilir olduğunu gösterir. Bu durumda ise Kalman kazancı sıfır değerini alır ve tahmin değerinin güncellenmesi için sensörden elde edilen değerler yerine bir önceki ölçüm adımından elde edilen tahmin değerleri kullanılır. Sonuç olarak P_k^- ve R değerleri bir ölçüm adımı için tahmin değerinin nasıl güncelleneceğini belirler.

Ölçme güncelleme denklemlerinde en son olarak hata kovaryansı güncellenir. Burada I matrisi birim matristir.

$$P_k = (I - K_k H) P_k^- \quad (2.24)$$

Ölçme güncelleme denklemleri tamamlandıktan sonra tekrar zaman güncelleme denklemlerine geçilir. Her ölçüm adımı için bu döngü tekrarlanır ve her döngü sonunda bir önceki ölçüm adımından gelen hata kovaryansları ve sistem durum tahminleri bir sonraki ölçüme iletilir. Kalman filtresinin denklem döngüsü Şekil 2.9'da verilmiştir.



Şekil 2.9 : Kalman filtresi denklem döngüsü.

Ölçme gürültüsü kovaryansı R , filtre denklemlerinden önce hesaplanır. Bu değer filtreyle bazı çevrimdışı ölçümler yapılarak belirlenebilir. Sistem gürültü kovaryansı yani Q matrisinin hesaplanması ise sistem doğrudan gözlemlenemediği için daha zordur. Pratikte bu değerler her ölçüm için değişir, fakat yapılan uygulamada bu değerlerin sabit olduğu kabul edilmiştir. Bu durumda hem hata kovaryansı P_k hem de Kalman kazancı K_k hızlı bir şekilde kararlı hale gelecek ve sabit kalacaklardır. Yapılan uygulama için ölçüm ve sistem gürültü kovaryanslarının belirlenmesi ve Kalman filtresinden elde edile sonuçlar Bölüm 5.2’de gösterilmiştir.

2.4 Kalman Filtresi ile Veri Birleştirme

Yapılan çalışmada kontrol edilen her rotasyon eksenini için ayrı bir Kalman filtresi tasarlanmıştır. Burada Denklem (2.6b)’de verilen Euler açıları jiroskoptan elde edilen rotasyon açı verilerine bakılarak güncellenmiştir. Sistemdeki ivmeölçer ise jiroskoptaki kayma hatasının giderilmesinde ve Euler açılarının başlangıç değerinin belirlenmesinde kullanılmıştır. Kalman filtresi içinde için iki sensörden de elde edilen Euler açıları ile ilgili veriler kullanılmıştır. Fakat düzeltme işlemi için sadece ivmeölçerden elde edilen veriler temel alınmıştır.

Filtreye ait denklemlerin yazılabilmesi için ilk olarak sistemin durum denklemlerinin belirlenmesi gerekir. Bunun için jiroskobun kayma değeri ile

rotasyon açısı arasındaki ilişkiden faydalanılmıştır. Denklem (2.25)'te iki değer arasındaki matematiksel ilişki gösterilmiştir.

$$a_k = a_{k-1} + (u_k - b_{k-1})T \quad (2.25)$$

Yukarıdaki denklemde k indisi ölçüm adımını, a değişkeni Euler açısını, u değişkeni jiroskoptan elde edilen açı değerindeki değişimi, b değişkeni ise jiroskobun kayma değerini temsil eder. Ayrıca denklemde verilen T değeri mikroişlemci kodunun sabit döngü süresi belirtmek için kullanılmıştır. Modelleme sırasında kullanılan denklemler olabildiğince basit şekilde yazılmıştır. Bu sayede mikroişlemci üzerindeki hesaplama yükü azaltılmış ve filtrenin çalışma hızı arttırılmıştır.

Denklem (2.25)'te verilen ilişki kullanılarak sisteme ait durum ve ölçüm denklemleri yazılabilir. Buna göre Denklem (2.15)'in ve Denklem (2.16)'nın sisteme uygun şekilde yeniden düzenlenmesi sonucunda Denklem (2.26) ve Denklem (2.27) elde edilir.

$$x_k = \begin{bmatrix} a \\ b \end{bmatrix}_k = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} T \\ 0 \end{bmatrix} u_k + w_{k-1} \quad (2.26)$$

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \quad (2.27)$$

Denklem (2.26)'da görülebileceği gibi sistemin durum değişkenleri rotasyon açısından ve jiroskoba ait kayma değerinden meydana gelmektedir. Ayrıca ataletsel ölçüm cihazı tarafından yapılan ölçümlerde sadece Euler açıları elde edildiği için ölçüm denklemlerindeki H matrisi $[1 \ 0]$ olarak alınmıştır.

Sistem ölçüm ve durum denklemleri ile modellendikten sonra Şekil 2.9'da gösterilmiş Kalman filtresi denklemleri yazılabilir. İlk olarak sistemin yeni ölçüm adımında kullanılacak durum tahmini Denklem (2.19)'da verilen ilişkiye göre hesaplanır.

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}_k = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}_{k-1} + \begin{bmatrix} T \\ 0 \end{bmatrix} u_k \quad (2.28)$$

Denklem (2.28)'de jiroskoba ait kayma değerinin her ölçüm adımı için sabit kaldığı görülmektedir. Buna göre Kalman filtresi her adımda ivmeölçerden elde edilen verilere göre kayma değerini düzenleyecektir. Böylelikle her ölçüm adımında jiroskoptan elde edilen değerlerdeki kayma hatası giderilecektir.

Yeni ölçüm adımındaki durum tahmini hesaplandıktan sonra Kalman filtresine ait ikinci denklem yazılabilir. Burada Denklem (2.20)'de verilen zaman güncelleme denkleminde göre yeni hata kovaryansı hesaplanmıştır.

$$P_k^- = \begin{bmatrix} P_{00}^- & P_{01}^- \\ P_{10}^- & P_{11}^- \end{bmatrix}_k \quad (2.29a)$$

$$P_k^- = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k-1} \begin{bmatrix} 1 & 0 \\ -T & 1 \end{bmatrix} + \begin{bmatrix} Q_a & Q_{ab} \\ Q_{ba} & Q_b \end{bmatrix} \quad (2.29b)$$

$$P_{00}^- = P_{00} - T(P_{01} + P_{10}) + T^2 P_{11} + Q_a \quad (2.29c)$$

$$P_{01}^- = P_{01} - TP_{11} + Q_{ab} \quad (2.29d)$$

$$P_{10}^- = P_{10} - TP_{11} + Q_{ba} \quad (2.29e)$$

$$P_{11}^- = P_{11} + Q_b \quad (2.29f)$$

Burada işlem gürültü kovaryansı iki sistem durumu için de tanımlanmıştır. Önceden de belirtildiği gibi bu değerlerin her ölçüm adımı için sabit kaldığı kabul edilmiştir. Sistemin hata kovaryansı güncellendikten sonra ölçme güncelleme denklemlerine geçilir. Buna göre ilk olarak Denklem (2.21)'de verilmiş olan Kalman kazancı güncellenir.

$$K_k = \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k = \frac{\begin{bmatrix} P_{00}^- & P_{01}^- \\ P_{10}^- & P_{11}^- \end{bmatrix}_k \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{\left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00}^- & P_{01}^- \\ P_{10}^- & P_{11}^- \end{bmatrix}_k \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R \right)} \quad (2.30a)$$

$$K_0 = \frac{P_{00}^-}{P_{00}^- + R} \quad (2.30b)$$

$$K_1 = \frac{P_{10}^-}{P_{00}^- + R} \quad (2.30c)$$

Kalman kazancı güncellendikten sonra Denklem (2.22)'de verilmiş yeni tahmin değeri hesaplanabilir. Önceden belirtildiği gibi Kalman filtresi denklemlerindeki düzeltme işlemleri ivmeölçer ölçümlerine göre yapılır. Bu yüzden Denklem (2.31a)'da z_k olarak tanımlanmış değişken ivmeölçerden elde edilen Euler açısını

temsil eder. Buna göre her ölçüm adımında ivmeölçerden ölçülen veriler kullanılarak hesaplanan Euler açıları Kalman filtresinden elde edilen Euler açılarından çıkarılır ve buradan elde edilen hata değerlerine göre her eksen için güncelleme yapılır. Filtrenin hangi değere göre güncellenmesi gerektiği R ve P kovaryanslarına bağlı olarak değişir.

$$\hat{x}_k = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}_k = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}_k^- + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \left(z_k - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}_k^- \right) \quad (2.31a)$$

$$\hat{a}_k = \hat{a}_k^- + K_0(z_k - \hat{a}_k^-) \quad (2.31b)$$

$$\hat{b}_k = \hat{b}_k^- + K_1(z_k - \hat{a}_k^-) \quad (2.31c)$$

Tahmin değeri güncellendikten sonra en son olarak Denklem (2.23)'de verilmiş hata kovaryansı hesaplanır. Bu matris daha sonra yeni ölçüm adımı için kullanılmak üzere tekrar döngüye girecektir.

$$P_k = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_k = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{00}^- & P_{01}^- \\ P_{10}^- & P_{11}^- \end{bmatrix}_k \quad (2.32a)$$

$$P_{00} = P_{00}^-(1 - K_0) \quad (2.32b)$$

$$P_{01} = P_{01}^-(1 - K_0) \quad (2.32c)$$

$$P_{10} = P_{10}^- - K_1 P_{00}^- \quad (2.32d)$$

$$P_{11} = P_{11}^- - K_1 P_{01}^- \quad (2.32e)$$

Tek ölçüm adımı için Kalman filtresi denklemlerinin sonuçları belirlendikten sonra gelecek ölçüm adımı için denklemler tekrar baştan hesaplanır.

3. ROBOTUN TASARIMI VE KULLANILAN EKİPMANLAR

Kamera dengeleme sistemi genel olarak bir sensör veri optimizasyonu uygulamasıdır. Fakat tasarım sürecine sistem buna ek olarak aynı zamanda bir kablosuz mobil robot uygulaması olarak da ele alınmıştır. Yapılan çalışmada hazır bir paletli mobil robot şasesi üzerine servo motorlardan oluşan bir dengeleme sistemi kurulmuş ve robotun üstüne yerleştirilen bir kablosuz kamera bu sistem vasıtasıyla robotun gövdesinin yönelimindeki değişikliklere karşı dengelenmiştir. Burada kameranın hareketini kontrol etmek için robotun üzerine 4 adet 180 derece dönebilen servo motor yerleştirilmiştir. Bu motorlardan ikisi kamerayı dengede tutmak için kullanılmıştır. Diğer servo motorlar ise kameranın bakış yönünün 360 derecelik bir alanda değiştirilebilmesi için birleşik halde yerleştirilmiştir. Ayrıca robotun üzerindeki kameranın yeryüzüne göre yöneliminin bulunması için de bir adet ataletsel ölçüm cihazı kullanılmıştır. Bunlara ek olarak robot uzaktan kablosuz haberleşme vasıtasıyla kontrol edilebilecek şekilde tasarlanmıştır. Bunun için operatör tarafında kullanılacak bir kumanda devresi hazırlanmış ve robotun gövdesinin hareketleri buradan kontrol edilmiştir.

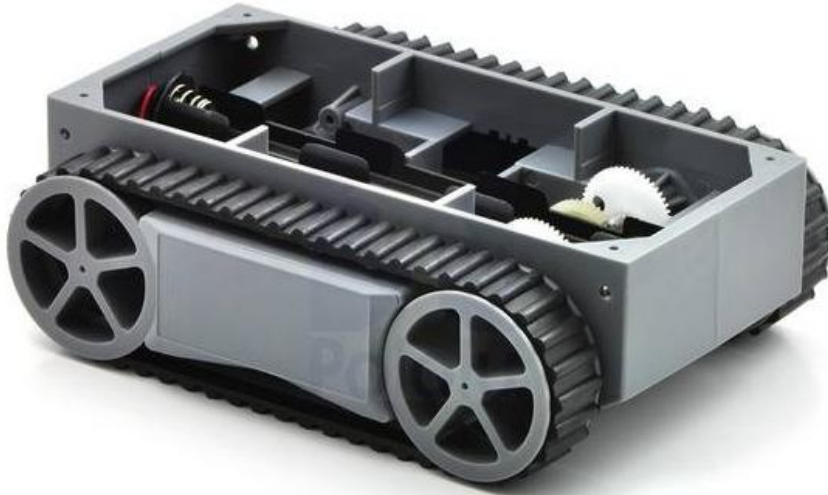
Kamera dengeleme sisteminin tasarım süreci mekanik donanım, elektronik donanım ve yazılım olarak üç parça halinde ele alınacaktır. Mekanik kısımda mobil robotun alt gövde kısmı ve dengeleme sisteminin bulunduğu platform detaylı olarak anlatılacaktır. Elektronik kısımda ise hem mobil robotun uzaktan kontrolünde hem de kamera dengeleme sisteminde kullanılan elektronik ekipmanlara değinilecektir. Son olarak yazılım kısmında kullanılan mikroişlemcinin kod yapısı ve sensör ile haberleşme protokolü anlatılacaktır.

3.1 Mekanik Donanım

Mekanik kısım mobil robota ait hazır şaseden, kamera dengeleme sistemi için kurulmuş platformdan ve bu platformun üzerindeki servo motorlardan meydana gelmektedir. Bu bölümde mekanik kısma ait bütün parçalar ayrıntılı şekilde anlatılmıştır.

3.1.1 Mobil robot şasesi

Yapılan çalışmada esas alınan konu kamera dengeleme sistemi olduğundan mobil robotun hareketini sağlayacak mekanizma hazır olarak temin edilmiştir. Şekil 3.1’de robotun şasesi görülmektedir [33].



Şekil 3.1 : Mobil robotun paletli gövdesinin görünümü.

Şasenin boyutları 18 x 14 x 6 cm olarak verilmiştir. Şasenin iki tarafında bulunan paletleri hareket ettirmek için paletlerin ortasındaki kutularda birer adet doğru akım motoru bulunmaktadır. Kullanılan motorlar hakkında teknik bilgi Çizelge 3.1’de gösterilmiştir [33].

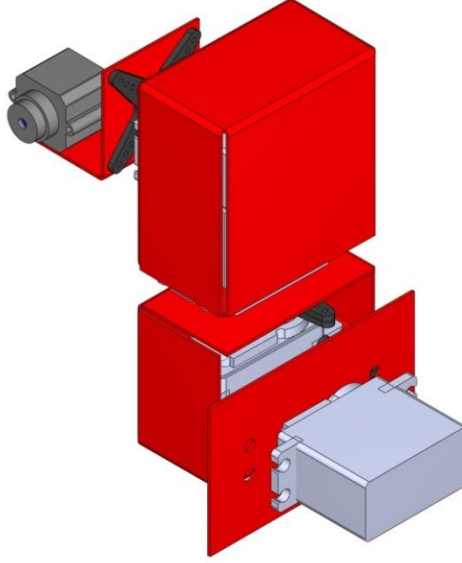
Çizelge 3.1 : Palet motorlarının özellikleri.

Nominal gerilim	7,2 V
Maksimum yükte çektiği akım	2,4 A
Boşta çektiği akım	210 mA
Hız	15 cm/san
Pil gerilimi	7,2 V

Şasenin üzerine köşelerdeki vida boşlukları kullanılarak metal bir platform monte edilmiştir. Bu platformun üzerine servo motorlardan oluşan kamera dengeleme sistemi ve elektronik kartlar yerleştirilmiştir.

3.1.2 Kamera dengeleme sistemi

Yapılan uygulamada birbirine bağlı servo motorlardan oluşan bir dengeleme sistemi kullanılmıştır. Burada motorlar sırayla birbirine bağlanmış ve en uç kısma ise kamera muhafaza kutusu monte edilmiştir. Sistemin görünümü Şekil 3.2’de verilmiştir.



Şekil 3.2 : Kamera dengeleme sisteminin görünümü.

Kamera dengeleme sistemi bir adet kablosuz kameradan, 4 adet 180 derece dönebilen servo motordan ve buradaki ekipmanları muhafaza eden 3 tane metal kutudan meydana gelir. Bütün sistem ilk servo motordan başlayarak birbirine bağlanmıştır. Burada sistem en alttan en üst kısma kadar anlatılacaktır.

Servo motorlardan ilki mobil robotun gövdesinin üzerinde bulunan metal platforma monte edilmiştir. Yatay şekilde platformun üzerine yerleştirilmiş olan servo motor kameranın yuvarlanma açısını kontrol eder. Servo motorun mili bir sonraki servo motoru taşıyan metal kutuya bağlıdır.

İkinci servo motor birinci motorun milinin bağlı olduğu metal kutuya monte edilmiştir. Servo motor mili yukarıya bakacak şekilde kutunun üst kısmına bağlanmıştır. Burada bulunan servo motor üstündeki kutuda bulunan diğer bir servo motor ile birlikte kameranın sapma açısını yani bakış yönünü değiştirir.

İkinci servo motorun milinin bağlı olduğu kutuya diğer servo motorlar yerleştirilmiştir. Burada alt kısımda bulunan servo motor bir önceki kutuda bulunan motor gibi kameranın sapma açısını değiştirir. Burada iki motor birbirini tamamlayıcı olarak çalışır. Buna göre servo motorlardan bir tanesi 180 derecelik hareketini tamamladıktan sonra diğeri dönmeye başlar. Bu sayede sapma açısı üzerinde 360 derecelik bir alanda kontrol yapılabilir. Kutunun üst kısmında bulunan servo motor ise kameranın yunuslama açısını kontrol etmek için kullanılır. Motorun mili kameraya ait pilin ve kameranın bulunduğu kutuya bağlıdır.

Sonuç olarak kullanılan servo motorlar aracılığıyla kameranın üç farklı ekseninde yaptığı dönme hareketi kontrol edilmiştir. Burada kameranın sadece sapma açısı 360 derece değişebilecek şekilde tasarlanmıştır. Diğer rotasyon açıları için ise 180 derecelik bir hareket yapılmasına izin verilmiştir. Tasarım sürecinde robotun normal hareketi sırasında gövdesinin X ya da Y ekseninde 90 dereceden fazla rotasyon hareketi yapmayacağı düşünülmüştür. Bu yüzden bu açıları kontrol eden servo motorlar ± 90 derecede hareket edecek şekilde seçilmiştir. Kameranın bakış yönünü belirleyen sapma açısı ise 360 derece çevrilerek kameranın çevresini tam olarak görebilmesi sağlamıştır.

Kamera dengeleme sisteminde yunuslama ve yuvarlanma açılarını kontrol eden servo motorların kamera üzerindeki etkisi sistemin kinematik denklemleri incelenerek anlaşılabilir. Bunun için ilk olarak sistem üzerinde mevcut eksen takımlarının belirtilmesi gerekir. Daha sonra bu eksen takımları arasında geçiş yapılarak sistemin kinematik denklemi elde edilir. Mevcut sistem üzerinde üç boyutlu 6 adet eksen takımı bulunmaktadır. Bu eksen takımları aşağıda sıralanmıştır.

- 1- Yeryüzü eksen takımı (sabit)
- 2- Robot gövdesi dönme eksenine sabitlenmiş eksen takımı (hareketli)
- 3- Yuvarlanma açısını kontrol eden servo motorun miline sabitlenmiş eksen takımı (hareketli)
- 4- Yunuslama açısını kontrol eden servo motorun miline sabitlenmiş eksen takımı (hareketli)
- 5- Sapma açısını kontrol eden servo motorun miline sabitlenmiş eksen takımı (hareketli)
- 6- Kamera dönme eksenine sabitlenmiş eksen takımı (hareketli)

Yapılan uygulamada kameranın sadece yönelimi kontrol edilmiştir. Bu yüzden eksen takımları arasındaki geçişlerde sadece rotasyon matrisleri kullanılacaktır. Kameranın konumunun da kontrol edilmesi durumunda sisteme ait kinematik denklemlere her eksen takımı için öteleme matrisleri eklenmelidir. Bu durumda yukarıda belirtilen eksen takımlarının konumu da önem kazanacaktır.

Burada birinci ve ikinci eksen takımı robotun gövdesinin yeryüzüne göre yönelimini belirlemek için kullanılır. Üçüncü, dördüncü ve beşinci eksen takımları servo

motorların mil hareketlerini gösterir. Son eksen takımı ise robotun üzerindeki kameranın yeryüzüne göre yönelimini belirtir.

Kamera dengeleme sistemine monte edilmiş kablosuz kamera robotun gövdesinin yönelimindeki değişikliklerden direkt olarak etkilenir. Bu yüzden ikinci ve altıncı eksen takımlarının rotasyon hareketleri birbirine bağlıdır. Yapılan uygulamada sadece yönelim kontrol edildiğinden iki eksen takımı aynı kabul edilmiştir. Ayrıca dengeleme sisteminde sapma hareketi kontrol edilmediği için dördüncü eksen takımı da sistemin kinematik denklemlerine eklenmemiştir.

Birinci ve ikinci eksen takımı arasındaki ilişki Denklem (2.6b)'de verilmiştir. Başlangıç anında bu iki eksen takımına ait eksenler üst üste çakışık haldedir. Buna göre Denklem (2.6b)'de gösterilen bütün Euler açıları başlangıç durumu için sıfıra eşit kabul edilmiştir. Kamera dengeleme sisteminde Z eksen üzerindeki rotasyon hareketleri kontrol edilmediği için buradaki Euler açılarının değerleri sıfıra eşit alınmıştır.

$$R_{12}(\theta, \psi, 0) = \begin{bmatrix} \cos \theta & \sin \psi \sin \theta & \cos \psi \sin \theta \\ 0 & \cos \psi & -\sin \psi \\ -\sin \theta & \sin \psi \cos \theta & \cos \psi \cos \theta \end{bmatrix} \quad (3.1)$$

Yunuslama hareketini kontrol eden servo motora ait eksen takımı sadece Y eksen etrafında hareket eder. Bu yüzden bu eksen takımı üzerindeki hareketler sadece Y eksen etrafında yapılan dönme hareketini gösteren bir rotasyon matrisi olarak tanımlanabilir. Benzer şekilde yuvarlanma hareketini kontrol eden servo motora ait eksen takımı da X eksen etrafında yapılan dönme hareketini gösteren bir rotasyon matrisi olarak tanımlanır. Sonuç olarak dengeleme işlemini gerçekleştiren iki servo motorun hareketleri matematiksel olarak tanımlanabilir.

$$R_{Yservo}(\theta_1) = \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 \\ 0 & 1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 \end{bmatrix} \quad (3.2)$$

$$R_{Xservo}(\psi_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi_1 & \sin \psi_1 \\ 0 & -\sin \psi_1 & \cos \psi_1 \end{bmatrix} \quad (3.3)$$

Başlangıç anında kameranın yönelimini gösteren eksen takımı ile yeryüzü eksen takımının üst üste çakışık halde olduğu kabul edilmiştir. Bu yüzden başlangıç

durumu için Denklem (2.6b)'de tanımlanmış Euler açılarının hepsi sıfıra eşittir. Sonuç olarak başlangıç anında kameranın yöneliminin Euler açısı cinsinden gösterimi Denklem (3.4)'teki gibi yazılabilir.

$$R_0(0,0,0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Robot hareket ettikçe yönelimi değişir ve burada gösterilen açı değerleri de gövdenin yönelimine göre farklı değerler alır. Yuvarlanma açısını ve yunuslama açısını kontrol eden servo motorların görevi burada gösterilen θ ve ψ değerleri değiştikçe kameranın bulunduğu metal kutuyu hareket ettirerek tekrar Denklem (3.4)'te gösterilen duruma getirmektir. Bunun için sistemin X ve Y eksenleri etrafında ters yönde çevrilmesi yeterli olacaktır. Sistemin mekanik tasarımı göz önünde bulundurularak servo motorların çalışma sırası ilk olarak yuvarlanma hareketi daha sonra yunuslama hareketi olacak şekilde belirlenmiştir. Bu yüzden rotasyon matrisleri de bu sıralama ile yazılmıştır.

$$R(0,0,0) = R_{12}(\theta, \psi, 0) R_{Xservo}(-\psi) R_{Yservo}(-\theta) \quad (3.5)$$

Denklem (3.5)'te kameranın yeryüzüne göre yönelimini belirten Euler açıları ile servo motorların hareketleri arasındaki bağlantı gösterilmiştir. Burada görüldüğü gibi yönelimi değiştirilen bir cismin yönelimin başlangıç konumuna getirilmesi için bu eksenler etrafında çalışan servo motorların ters yönde hareket etmesi gerekir. Ayrıca motorların hareket edecekleri açı değerinin büyüklüğü de sistemin yeryüzüne göre yönelimi belirten Euler açılarına eşittir.

Yapılan uygulamada kameranın yunuslama ve yuvarlanma açısı kontrol edilirken yine kameraya ait sapma açısının sıfıra eşit tutulması gerekir. Aksi takdirde kameranın yönelimi istenen şekilde kontrol edilemez. Şekil 3.2'de gösterilen durumda kameranın sapma açısı başlangıç konumuna göre 90 derece saptırılmıştır. Burada görülebileceği gibi bu hareketin sonucunda başlangıç konumunda yuvarlanma hareketini kontrol eden servo motor artık yunuslama hareketini kontrol etmeye başlamıştır. Bu durumda meydana gelen hataları gidermek için robot hareket ettirilirken kameranın sapma açısı otomatik olarak sıfır dereceye getirilmiştir. Buna göre kameranın bakış yönünün değiştirilebilmesi için robotun durması gerekir.

3.1.3 Servo motorlar

Kamera dengeleme sisteminde 4 adet servo motor kullanılmıştır. Servo motorlar içlerinde potansiyometre ve kontrol devresi barındıran özel doğru akım motorlarıdır. Millerine bağlı yüksek çevirme oranlı dişliler ile küçük motorlardan yüksek moment değerleri elde etmek mümkündür. Genellikle piyasada sadece belli bir açı aralığında hareket eden servo motorlar bulunur; fakat sürekli hareket eden modelleri de mevcuttur.

Servo motorlarda üç adet elektriksel bağlantı girişi bulunur. Bunlardan ikisi enerji bağlantısıdır. Buraya servo motorlara enerji verilmesinde kullanılacak pilin enerji uçları bağlanır. Üçüncü bağlantı ucu ise dönme açısını belirleyecek olan sinyal bağlantısıdır. Bu bağlantı servo motorun içerisinde bulunan bir kontrol devresine bağlanmıştır. Bu sayede servo motorlar ek bir kontrol algoritması gerektirmeden sadece uygun sinyallerin uygulanması ile belli açı değerlerine hareket edebilir. Sonuç olarak mikroişlemci ile bu bağlantı ucu birbirine bağlanarak servo motorun hareket kontrolü gerçekleştirilir.

Servo motorların hareketi sinyal girişine uygulanan kare dalga sinyali ile kontrol edilir. Buna göre motorun sinyal ucuna periyodik olarak kare dalga sinyali uygulanır. Uygulanan dalga sinyalinin bir periyot içindeki uzunluğu motorun döneceği açıyı belirler. Buna göre kare dalga sinyalinin uzunluğu arttıkça motor daha fazla açı yol alır. Benzer şekilde sinyalinin uzunluğu azaldıkça da motor eski konumuna geri döner. Sonuç olarak her sinyal uzunluğu için motorun ilerleyeceği açı konumu bellidir. Servo motorların kontrol edilmesi için kullanılan bu yöntemle darbe genişlik modülasyonu (PWM) denir.

Yapılan uygulamada kameranın hareket ettirilmesi için 4 tane Parallax firmasına ait standart model servo motorları kullanılmıştır. Bu motora ait teknik özellikler Çizelge 3.2’de verilmiştir [34].

Çizelge 3.2 : Parallax standart servo motorun teknik özellikleri.

Nominal gerilim	6 V
Maksimum moment (6 V)	3,1 kg-cm
Boşta çektiği akım (6 V)	90-190 mA
Dönme açısı	180°
Ölçüler	5,58 x 1,9 x 40,6 cm
Hız	1,5 sn/180°

3.2 Elektronik Donanım

Robotta kullanılan elektronik ekipman mikroişlemci kartı, ataletsel ölçüm cihazı, palet motorlarını sürmek için kullanılan sürücü kart, kablosuz haberleşme ünitesi, uzaktan kumanda devresi ve kablosuz kamera olarak sıralanabilir. Bu kısımda bütün parçaların kullanım şekilleri ve özellikleri detaylı bir şekilde anlatılmıştır.

3.2.1 Mikroişlemci kart

Robotun üzerindeki bütün kontrol işlemleri mikroişlemci tarafından yapılır. Bu yüzden sistemde bulunan her elektronik parçanın kullanılan mikroişlemci kart ile bağlantısı vardır. Sadece kablosuz kamera kendi başına bir sistem olarak çalışmaktadır. Bu yüzden mikroişlemci kartı ile bir bağlantısı yoktur.

Yapılan çalışmada Arduino Duemilanove isimli bir mikroişlemci geliştirme kartı kullanılmıştır. Arduino tabanlı mikroişlemci kartlar kolay ve pratik yapılarından dolayı son yıllarda birçok prototip projesinde yer almaya başlamıştır. Bu yüzden birçok kitapta ya da internet sitesinde Arduino'ya ait mikroişlemci kartları ile yapılmış hazır projeler bulunabilmektedir [35]. Ayrıca hem Arduino'nun kendi internet sayfasında hem de farklı geliştiricilerin internet sayfalarında kart ile uyumlu kütüphane dosyaları mevcuttur. Bu kütüphane dosyaları kullanılarak çeşitli sensörler üzerinden veri elde etmek ya da farklı haberleşme protokollerini ile iletişim kurmak gibi işlemler kolayca gerçekleştirilebilir. Örnek olarak, yapılan çalışmada ataletsel ölçüm cihazı ile haberleşmek için ve servo motorları kontrol etmek için iki ayrı kütüphane kullanılmıştır.

Arduino'nun sağladığı başka bir kolaylık ise modüler yapısıdır. Arduino mikroişlemci kartları için özel olarak üretilmiş “Shield” isimli eklenti kartları ile devreye birçok yeni özellik katılabilmektedir. Bu eklenti kartları mikroişlemci kartlarının konektörleri ile uyumludur. Buna göre iki kartın da konektörleri üst üste yerleşebilecek şekilde tasarlanmıştır. Bu sayede eklenti kartları ek bir kablo gerektirmeden mikroişlemci ile bağlantı kurabilirler.

Prototip projelerinde Arduino mikroişlemci kartlarını ideal kılan en önemli özellik ise kartların güvenilir yapılarıdır. Üzerlerinde bulunan koruma devreleri ile hatalı bağlantı durumlarında kartın enerjisi kesilerek devreye zarar gelmesi önlenir. Ayrıca kartın besleme geriliminin de 7 – 12 V arasında değişmesi karta uygun pil seçiminde

seenekleri olduka zenginleřtirmektedir. Arduino Duemilanove’a ait teknik zellikler izelge 3.3’te verilmiřtir [35].

izelge 3.3 : Arduino Duemilanove mikroięlemci kartına ait teknik zellikler.

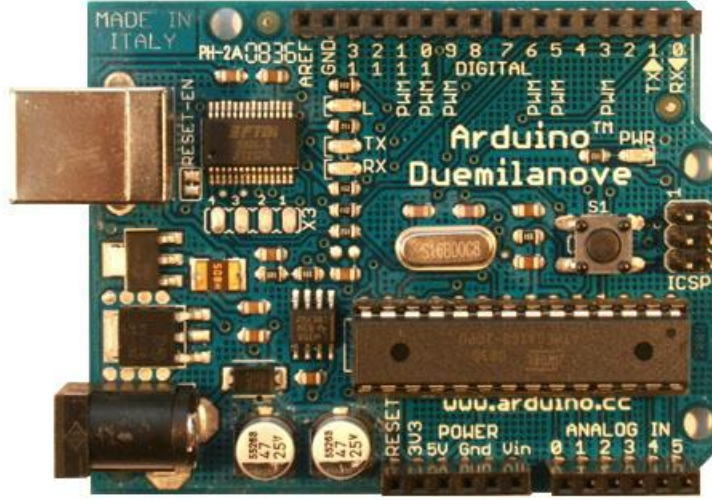
Kullanılan mikroięlemci	ATmega168
alıřma gerilimi	5 V
Besleme gerilim aralıęı (onerilen)	7 – 12 V
Besleme gerilim aralıęı (izin verilen)	6 – 20 V
Dijital giriř/ıkıř sayısı	14 Adet
Analog giriř sayısı	6 Adet
Giriř/ıkıř baęlantısı bařına akım	40 mA
3,3 V baęlantısı iin akım	50 mA
Program hafıza byklę	16 KB
RAM hafıza byklę	1 KB
alıřma hızı	16 MHz

Arduino Duemilanove zerinde 20 adet giriř/ıkıř bulunmaktadır. Bunlardan 14’ dijital giriř/ıkıř olmakla beraber 6’sı da analog giriř olarak kullanılır. Mikroięlemcinin bu giriřlerden alabileceęi ya da verebileceęi en byk doęru akım deęeri 40 mA’dır.

Mikroięlemciye enerji vermek iin  farklı alternatif mevcuttur. Bunlardan ilki bilgisayar ile baęlantıyı mmkn kılan evrensel seri veri yolu (USB) giriřidir. Bilgisayar ile mikroięlemci kart A-to-B USB kablosu kullanılarak USB giriři zerinden birbirine baęlanır. Bu sayede hem mikroięlemciye enerji verilir hem de bilgisayarda hazırlanmıř mikroięlemci kodu karta iletilebilir. Enerji beslemesi iin ikinci alternatif adaptr giriři kısımdır. Mikroięlemciye buradan enerji verebilmek iin kullanılacak enerji kaynaęının elektrik ularına pozitif merkezli 2,1 mm uzunluęunda bir adaptr ucu yerleřtirilir. Daha sonra bu adaptr ucu mikroięlemci kartının adaptr giriřine baęlanarak mikroięlemciye enerji verilebilir. Ayrıca aynı enerji kaynaęından adaptr kullanılmadan kaynaęın pozitif ucu mikroięlemcinin “VIN” giriřine, negatif ucu da “GND” giriřine baęlanacak řekilde karta enerji verilebilir. Fakat “VIN” giriřine uygulanabilecek gerilim deęeri izelge 3.3’te belirtilen gerilim deęerlerini ařmamalıdır.

izelge 3.2’de de belirtildięi gibi kartın besleme gerilimi 6 ile 20 V arasında herhangi bir gerilim deęerinde olabilir. Fakat 12 V’u ařan gerilim deęerlerinde kart ısınıp mikroięlemciye zarar verebilir. Benzer řekilde de kartın 7 V’un altında bir gerilim ile beslenmesi sonucunda ise kartın ıkıřlarından istenen lojik deęerler

alınmayabilir. Şekil 3.3'te Arduino Duemilanove mikroişlemci kartının görünümü verilmiştir. Burada kart üzerinde bulunan bütün giriş/çıkışlar gösterilmiştir [35].



Şekil 3.3 : Arduino Duemilanove mikroişlemci kartının görünümü.

Şekil 3.3'ten de görülebileceği gibi Arduino Duemilanove üzerinde 14 adet dijital giriş/çıkış mevcuttur. Bu mikroişlemci pinlerinden kullanılacak olanların durumu programlama kısmında önceden tanımlanır. Böylece mikroişlemci hangi pinlerin giriş ya da çıkış olarak çalışacağını algılar. Programlamada belirtilmediği sürece kullanılan pinler çıkış olarak görev yapar. Arduino Duemilanove mikroişlemci kartında 6 adet analog giriş bulunur. Bu girişlerin hepsi 10 bit çözünürlüğe sahip bir analog-dijital dönüştürücüye bağlıdır. Bu sayede analog girişe uygulanan 0 V ile 5 V arasındaki gerilim uygulanan gerilime orantılı olarak değişen 0 ile 2^{10} arasında bir değere çevrilir. Bu sayede kullanıcı uygulama sırasında elde edilen gerilim değerini 0 ile 1023 arasında değişen bir değer olarak görür. Özel uygulamalarda ise dönüştürücünün referans olarak kullanacağı gerilim aralığı kod kısmında ya da “AREF” girişine 5 V’u aşmayacak bir gerilim uygulanarak değiştirilir.

Arduino Duemilanove üzerinde bulunan bazı pinlerin kendilerine has özellikleri bulunur. Bu pinler ve özellikleri aşağıda verilmiştir.

- 0 ile 1 numaralı dijital pinler seri iletişim için kullanılır.
- 2 ve 3 numaralı dijital pinler ise hızlı uygulamalarda kullanılan kesme pinleridir. Bu pinler kullanılarak kod içindeki bazı işlemlerin normal kod döngüsünden bağımsız halde çalışması sağlanabilir. Böylece bu pinlere bağlanan sensörler ya da cihazlar ile daha hızlı çalışma yapılır.

- 3, 5, 6, 9, 10 ve 11 numaralı pinler PWM özelliği olan çıkışlardır. PWM yöntemi ile pinlerden sabit bir dijital sinyal yerine belli bir görev döngüsü (duty cycle) için kesintili sinyaller elde edilebilir. Böylelikle PWM özelliği bulunan pinler analog çıkış gibi çalışabilirler ve 5 V'tan daha küçük sinyalleri çıkış olarak üretebilirler.
- Bunlara ek olarak çeşitli haberleşme protokolleri üzerinden iletişim yapabilmek için kullanılan kütüphane dosyalarında bazı pinler özel olarak tanımlanır. Bu kütüphane dosyalarının sağladığı özelliklerin aktif olarak kullanılabilmesi için bu pinlerin de kullanılması gerekir.

Arduino tabanlı mikroişlemci kartlar programlama için ek bir devreye ihtiyaç duymazlar. Buna göre kartlar sadece A-to-B USB kablosu kullanılarak bilgisayara bağlanabilirler. Programlama için ise Arduino'nun internet sayfasında bulunan özel bir ara yüz programı kullanılır. Kullanılan programlama dili kullanılan mikroişlemci için özel olarak düzenlenmiş C++ tabanlı bir dildir. Kodlama hakkında ayrıntılı bilgi Bölüm 4.3'te verilmiştir.

3.2.2 IMU

Yapılan uygulamada gövdenin yönelimini belirlemek için bir adet üç eksenli jiroskop ile bir adet üç eksenli ivmeölçer beraber kullanılmıştır. Bu sensörler Şekil 3.4'te gösterilen ataletsel ölçüm cihazında birleştirilmiştir [36]. Cihazın içinde bulunan ivmeölçer ve jiroskop ile ilgili ayrıntılı bilgi Bölüm 3.2.2.1'de ve Bölüm 3.2.2.2'de verilmiştir.



Şekil 3.4 : Mobil robot üzerinde kullanılan ataletsel ölçüm cihazı.

3.2.2.1 İvmeölçerin özellikleri

Yapılan çalışmada ADXL345 isimli ivmeölçer kullanılmıştır. ADXL345; düşük enerji tüketimi olan, ince ve küçük bir yapıya sahip üç eksenli bir ivmeölçerdir. 16 g gibi yüksek ivme değerlerinde ölçüm yapabilen sensör hem statik ivme hem de

dinamik ivme ölçümünde kullanılabilir. Ayrıca hareket, darbe veya serbest düşüş algılama gibi ek özellikleri de mevcuttur. Bu bölümde sensör ile ilgili gösterilen verilerin hepsi sensöre ait veri füyünden alınmıştır [29].

İvmeölçerler yaptıkları ölçümleri mikroişlemciye analog ve dijital olmak üzere iki farklı şekilde iletebilirler. ADXL345 de mikroişlemci ile haberleşmek için dijital veri paketleri kullanır. Burada haberleşme iki yönlüdür. Sensör mikroişlemci tarafından belirlenen parametrelere göre ölçüm yapar ve elde ettiği verileri mikroişlemciye iletir. Kullanılan veri paketleri farklı çözünürlüklere sahip olabilir. Yapılan uygulamada ise sadece 10 bitlik çözünürlüğe sahip veri paketleri kullanılmıştır. Sensöre ait ölçüm parametreleri ve ölçüm verileri sensörün hafızasında bulunan yazmaçlarda (register) saklanır. Bu yazmaçların hepsinin belirli bir adresi vardır ve mikroişlemci bu adresleri kullanarak veri alışverişini yapar.

Sensör üzerinde dijital veri iletimi için iki farklı iletişim protokolü bulunur. Bu protokoller elektronik haberleşmede Serial Peripherhal Interface (SPI) ve Inter-Integrated Circuit (I²C) olarak adlandırılır. Kamera dengeleme sisteminde veri iletimi için I²C isimli protokol kullanılmıştır. Bu konuda detaylı bilgi Bölüm 3.3.2’de verilmiştir.

ADXL345 üzerinden ölçüm yapabilmek için ilk önce mikroişlemci tarafında ölçüm parametrelerinin belirlenmesi ve sonrasında bu parametrelerin sensöre iletilmesi gerekir. Aynı yöntem ek özelliklerin aktive edilmesi için de kullanılabilir. Sensörden veri elde etme yöntemleri anlatılmadan önce ilk olarak üç tane önemli ölçüm parametresine değinilecektir.

ADXL345 üzerindeki kullanılacak ölçüm parametrelerinden ilki çözünürlük yani ölçüm aralığıdır. Sensör çeşitli ivme aralıklarında (± 2 g, ± 4 g, ± 8 g, ± 16 g) ölçüm yapabilir. Ölçüm aralığının arttırılması daha yüksek ivme değerlerinde çalışmayı mümkün kılarken ölçüm hassasiyetinin yani ölçek değerinin de azalmasına neden olur. Kamera dengeleme sisteminde sadece statik ivme elde edilmek istendiğinden yüksek ivme değerlerinde ölçüm yapılmamıştır. Bunun için ivme ölçüm aralığı ± 2 g olarak alınmıştır. Çizelge 3.4’te farklı ölçüm aralıkları için ölçek değerleri verilmiştir. Burada ölçek değerleri, veri paketindeki en önemsiz bitin (LSB) bir kere artmasına neden olan ivme değişimi olarak tanımlanmıştır. Önceden belirtildiği gibi ivme değerleri g cinsinden gösterilmiştir.

Çizelge 3.4 : ADXL345 için farklı ölçüm aralıklarındaki ölçek değerleri.

Ölçüm Aralığı	Ölçek değerleri (mg/LSB)		
	Minimum	Tipik	Maksimum
±2 g	3,5	3,9	4,3
±4 g	7,0	7,8	8,6
±8 g	14,0	15,6	17,2
±16 g	28,1	31,2	34,3

İkinci önemli parametre ise hassasiyettir. Bu parametre sensör tarafından iletilen verilerin g ivmesi cinsinden hesaplanabilmesini sağlar. Hassasiyet değeri kullanılan ölçüm aralığına göre değişir. Bu ilişki Çizelge 3.5'te verilmiştir. Bu çizelgeye göre ±2 g ölçüm aralığı için kullanılması gereken hassasiyet değeri 256 LSB/g'dir. Bu değer sensörden elde edilen verilere bölünerek her eksen takımı için ivme değeri elde edilir.

Çizelge 3.5 : ADXL345 için farklı ölçüm aralıklarındaki hassasiyet değerleri.

Ölçüm Aralığı	Hassasiyet değerleri (LSB/g)		
	Minimum	Tipik	Maksimum
±2 g	232	256	286
±4 g	116	128	143
±8 g	58	64	71
±16 g	29	32	36

Üçüncü önemli parametre ise veri iletim hızıdır. Veri iletimi burada frekans cinsinden tanımlanmıştır ve bu parametre sensörün mikroişlemciye ölçtüğü verileri iletme hızını belirtir. Sensör 6,25 Hz'ten 3200 Hz'e kadar farklı frekanslarda veri iletimi yapabilir. Burada önemli olan kullanılan veri iletim hızının mikroişlemcinin döngü süresiyle ya da haberleşme protokolünün iletişim hızıyla uyumlu olmasıdır. Aksi takdirde özellikle zamana bağlı uygulamalarda veri kaybı meydana gelebilir. Kamera dengeleme sisteminde mikroişlemci üzerindeki kodun döngüsü 20 ms'de tamamlanacak şekilde düzenlenmiştir. Bu durumda veri iletim hızının 50 Hz'ten az seçilmesi gerekir. Yapılan çalışmada ivmeölçer kısmında zamana bağlı bir uygulama yapılmadığından ölçüm hızı ve kod döngüsü arasında meydana gelebilecek uyumsuzlukların çalışma üzerindeki etkisi çok azdır.

Parametreler ayarlandıktan sonra sensör üzerinden ölçüm yapılabilir. Her eksen için ölçüm verileri sensörün hafızasındaki yazmaçlarda depolanır. Bu alanlar hafıza üzerinde belirli bir adrese sahiptir. Bunun için mikroişlemci ilk olarak sensörden istediği ölçüm verilerini talep eder. Daha sonra bu veriler her eksen için 2 baytlık

paketler halinde mikroişlemciye iletilir. En son olarak mikroişlemci elde ettiği bu verileri çalışılacak uygulamaya göre düzenler ve her eksen için ivme değerleri elde edilir.

3.2.2.2 Jiroskobun özellikleri

Kamera dengeleme sisteminde ITG3200 isimli üç eksenli jiroskop kullanılmıştır. Bu sensör de ADXL345 gibi mikroişlemci ile I²C protokolü üzerinden haberleşir. İvmeölçerdeki programlamaya benzer şekilde jiroskop ile de ölçüm yapılmadan önce çeşitli parametrelerin belirlenmesi ve mikroişlemci üzerinden sensöre iletilmesi gerekmektedir. Jiroskop üzerinde de ölçüm parametreleri ve ölçüm verileri belli adreslere sahip yazmaçlarda saklanır. Burada jiroskoptan ölçüm yapılmadan önce sensöre iletilmesi gereken 3 tane parametre gösterilecektir. Bu bölümde ITG3200 ile ilgili gösterilen bütün veriler sensöre ait veri floyünden alınmıştır [37].

Jiroskop ile çalışmadan önce belirlenmesi gereken parametrelerden ilki örnekleme hızıdır. Burada örnekleme hızı frekans olarak tanımlanmıştır ve bu parametre sensörün ölçüm hızını belirtir. İvmeölçerdeki veri iletim hızında olduğu gibi burada da örnekleme hızının mikroişlemcideki kodun döngü süresine göre ayarlanması gerekir. Aksi takdirde yapılan ölçümlerde veri kaybı meydana gelebilir. Ayrıca jiroskobun kullanıldığı uygulama zamana bağlı olduğundan veri kaybının elde edilen sonuçlar üzerindeki etkisi oldukça fazladır. Bu yüzden örnekleme hızının 20 ms uzunluğundaki kod döngü süresine göre ayarlanması gerekir. Sensör üzerinde 8 kHz'ten 3,9 Hz'e kadar birçok örnekleme frekansı seçeneği mevcuttur. Kamera dengeleme sistemi için 50 Hz'ten düşük değerlerde bir örnekleme frekansı seçilmesi yeterli olacaktır.

İkinci önemli parametre ölçme aralığıdır. İvmeölçerde olduğu gibi burada da sensörün algılayabileceği maksimum ve minimum açısal hız değerleri vardır. ITG3200 için tek bir ölçme aralığı mevcuttur, fakat daha sonra yapılacak revizyonlarda yeni ölçme aralıkları eklenmesi ihtimaline karşın sensör hafızasında farklı ölçme aralıkları için boş adresler bırakılmıştır. Buna göre kullanılan jiroskop sadece ± 2000 °/s aralığında ölçme yapabilir. İvmeölçerde olduğu gibi burada da sensörün hassasiyet değeri ölçme aralığına bağlıdır. ITG3200'e ait veri floyünde jiroskoptan elde edilen verilerin açısal hıza dönüştürülmesinde kullanılacak hassasiyet değeri 14,375 LSB/°/s olarak verilmiştir.

Sensörün içinde çıkış verisini filtrelemek için düşük geçiren filtre kullanılmıştır. Böylece gürültünün veri üzerindeki etkisi azaltılmıştır. Sensör bu filtrenin bant genişliğinin mikroişlemci tarafından programlanmasına olanak sağlar. Bu sayede filtrenin ölçüm üzerindeki etkisi değiştirilebilir. Burada filtrenin bant aralığı 5 Hz ile 256 Hz arasında değiştirilebilir. Yapılan uygulamada elde edilen açısal hız değerlerinde filtre kullanılmasına gerek duyulmamıştır. Bu yüzden sensör üzerindeki filtre için en düşük bant aralığı seçilmiştir.

Bu özelliklere ek olarak sensörün üzerinde zaman sayacı için referans belirleme özelliği ve hızlı çalışma durumlarında kullanılan kesme özelliği de bulunur. Bu özellikler çok hızlı rotasyonlarda sensörün performansını arttırmak için kullanılır. Yapılan çalışmada yüksek hızlarda çalışmadığı için bu özellikler aktive edilmemiştir.

Jiroskop üzerinden veri elde etmek için ivmeölçerde kullanılan yöntemin aynısı kullanılır. Burada da mikroişlemci, haberleşme protokolü üzerinden istediği verileri sensöre bildirir. Bunun sonucunda her eksen için açısal hız verileri 2 baytlık paketler halinde mikroişlemciye iletilir. Gerekli düzenlemelerden sonra mikroişlemci uygulamada kullanmak üzere açısal hız değerlerini elde eder. Kamera dengeleme sisteminde sadece X ve Y eksenindeki açısal hız değerleri kullanılmıştır.

ITG3200 için tanımlanmış sıfır değeri toleransı oldukça fazladır. Bu yüzden jiroskobun bağlı olduğu gövde hareketsiz şekilde dururken bile sensörden okunan veriler 0 °/s'den farklı olur. Bu hatanın giderilmesi için sensör ile çalışmaya başlamadan önce sabit durumda her eksen üzerindeki açısal hız değerleri belirli bir sefer için okunup daha sonra bu değerler her eksen için ayrı ayrı kaydedilir. Belirli bir sayıda ölçüm alındıktan sonra bu değerler kendi aralarında toplanıp ortalamaları alınır. Bu sayede her eksen için sensörün hareketsiz durumda mikroişlemciye ilettiği hatalı açısal hız değeri belirlenir. Daha sonra yapılan ölçümlerde elde edilen veriler bu değerlerden çıkarılır ve gerçek açısal hız değerleri hesaplanır. Denklem (3.6)'da ve Denklem (3.7)'de sensörden ölçülen verilere göre X ve Y ekseninde açısal hız değerlerinin elde edilmesi gösterilmiştir.

$$W_{yz} = (DV_{W_{yz}} - O_x) / 14,375 \quad (3.6)$$

$$W_{xz} = (DV_{W_{xz}} - O_y) / 14,375 \quad (3.7)$$

Burada O_x ve O_y değerleri sıfır değer toleransındaki hatanın düzeltilmesinde kullanılan değerlerdir. Jiroskoptan okunan veriler de DV olarak isimlendirilmiştir. Önceden belirtildiği gibi açısal hızın belirlenmesi için jiroskobun veri föyünde tanımlanmış olan hassasiyet değeri kullanılmıştır.

3.2.3 Motor sürücü kartı

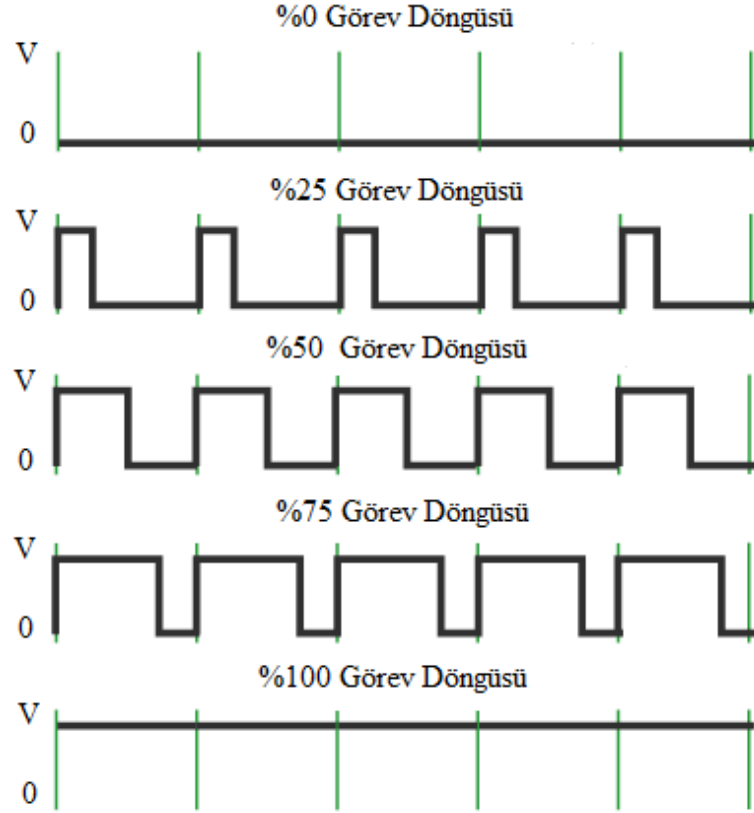
Mobil robotun paletlerine bağlı olan motorların robotun hareketine göre farklı hızlarda ve farklı yönlerde çalışması gerekir. Bu yüzden motorların hareketini kontrol edecek özel bir sürücü devreye ihtiyaç vardır. Çizelge 3.1’de belirtildiği gibi palet motorları bağlı oldukları kaynaktan maksimum yük altında en fazla 2,4 A büyüklüğünde bir akım çekeceklerdir. Buna göre seçilen motor sürücü devresinin de bu akım değerlerinde çalışabilmesi gerekmektedir. Sonuç olarak yapılan çalışmada iki palet motoru için de uygun sürücü devreleri barındıran bir hazır elektronik kart kullanılmıştır. Kartın üzerinde iki adet MC33926 sürücü entegre H-köprüsü ve bu sürücülere ait ek devre elemanları bulunur. Çizelge 3.6’da sürücü entegre devre ile ilgili teknik özellikler verilmiştir [38].

Çizelge 3.6 : MC33926 çift motor sürücü karta ait teknik özellikler.

Kullanılan sürücü	MC33926
Motor çıkış kanalı sayısı	2
Çalışma gerilim aralığı	5 – 28 V
Kanal başına izin verilen sürekli akım	2,5 A
Kanal başına izin verilen maksimum akım	5 A
Akım sensörü hassasiyeti	0,525 V/A
Maksimum PWM frekansı	20 kHz
Minimum lojik gerilimi	2,5 V
Maksimum lojik gerilimi	5 V
Ebatlar	2,8 x 4,6 cm
Ağırlık	7 g

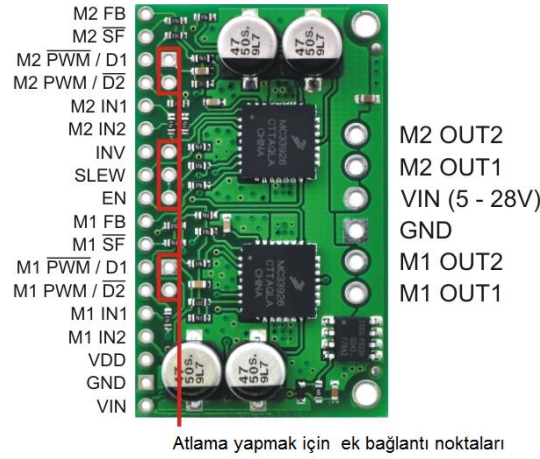
Yapılan uygulamada motorların hız kontrolü için PWM yöntemi kullanılmıştır. PWM yönteminde motorun enerji uçlarına sabit bir gerilim sinyali yerine belirli bir görev döngüsüne sahip kesintili bir gerilim sinyali uygulanır. Bu sayede motora uygulanan ortalama gerilim değeri değiştirilir ve motorun dönüş hızı ayarlanır. PWM yöntemi ile üretilen gerilim sinyalinin frekansı mikroişlemci tarafından belirlenir. Buna göre PWM frekansı azaldıkça uygulanan gerilim azalır ve motorun dönüşü yavaşlar. Benzer şekilde PWM frekansı artırılarak motor hızlandırılabilir. Şekil 3.5’te PWM yöntemi ile elde edilen gerilim sinyalinin görev döngüsüne göre

değişimi gösterilmiştir. Burada V değeri motora uygulanan maksimum gerilimi göstermektedir. PWM frekansına göre değişen görev döngüleri de yüzde cinsinden verilmiştir.



Şekil 3.5 : Farklı görev döngüsü için uygulanan gerilim sinyalleri.

Şekil 3.6’da sürücü kartın lojik ve enerji bağlantı noktaları gösterilmiştir [38]. Burada kartın sol kısmında bulunan giriş/çıkışlar mikroişlemci ile yapılan bağlantılar için kullanılır. Sağ kısımda bulunan bağlantı noktaları ise motor ve enerji kaynağı ile ilgili bağlantılardır. Kartın üzerinde bulunan bu boşluklara uygun konektörler ya da klemensler lehim ile monte edilir. Ayrıca kartın üzerine uygulamada kullanılan kablo sayısını azaltmak için ek boşluklar yerleştirilmiştir. Kartın sol kısmında ikinci bir sıra halinde yerleştirilmiş olan bu boşluklar uygulama sırasında sürekli lojik 1 ya da lojik 0 değerinde bulunması gereken girişler için kullanılır. Buna göre kartın bu bölümünde bulunan yuvarlak boşluklar hep lojik 1 değerine sahiptir. Kare şeklindeki boşluklar ise hep lojik 0 değerinde olur. Bu boşluklar kendileri ile aynı hizada bulunan konektörlere lehim ile bağlanır ve bu sayede uygulamada kullanılması gereken kablo sayısı azaltılabilir. Fakat uygulamanın bu şekilde yapılabilmesi için Şekil 3.6’da sol kısımda gösterilen “VDD” ve “GND” girişlerinin mikroişlemciye uygun şekilde bağlanması gerekir.



Şekil 3.6 : MC33926 çift motor sürücü karta ait bağlantılar.

Kart üzerinde mevcut olan bağlantı noktalarının isimleri ve görevleri Çizelge 3.7’de verilmiştir [38]. Burada her pin için bir başlangıç lojik değeri bulunmaktadır. Buna göre sürücünün istenen şekilde çalışması için bu değerler başlangıç anında ya da motorların hareketi sırasında değiştirilebilir. Ek bağlantı noktalarının kullanılması durumunda ise buralara bağlanan pinler çalışma boyunca aynı lojik değere sahip olacaktır.

Çizelge 3.7 : MC33926 çift motor sürücü kartta bulunan bağlantılar ve görevleri.

Bağlantı	Görevi
VIN	Sürücünün enerji besleme girişidir.
GND	Kartın lojik ve enerji beslemesinde kullanılan toprak bağlantısıdır.
OUT2, OUT1	Motor besleme çıkışlarıdır.
IN1,IN2	Motorun hareket yönünü ve hızını kontrol eden girişlerdir. PWM sinyali bu girişlere uygulanır.
VDD	Lojik bağlantılar için gerilim beslemesidir.
D1, D2	Bu pinler aktivasyon girişleri olarak adlandırılır. Bu girişler istenen şekilde bağlanmadıkça sürücü mikroişlemciden gelen sinyallere tepki göstermez. Genellikle D2 lojik 1, D1 ise lojik 0 olacak şekilde bağlanır.
SF	Normalde lojik 1 durumundadır. Aşırı akım veya ısınma durumlarında lojik 0 değerini alır. Ayrıca D1 veya D2 bağlantıları yanlış yapıldığında da lojik 0 değerini alır.
FB	Sürücü üzerinde bulunan akım sensörüne ait çıkıştır.
EN	Sürücünün çalışır duruma geçmesi için bu bağlantının lojik 1 değerini alması gerekir.
SLEW	Yüksek PWM frekanslarına erişebilmek için kullanılır.
INV	Her lojik 1 değerini aldığı anda motorun çalışma yönünü değiştirir.

Sürücünün çalışır duruma gelmesi için “EN” isimli girişi lojik 1 değerine getirilir. Bu sayede entegre bekleme konumundan çıkar ve çalışmaya başlar. Fakat motorun hareket etmesi için bu girişin aktive edilmesi yeterli değildir. Buna ek olarak “D1” ve “D2” girişlerinin uygun şekilde bağlanması gerekir; aksi takdirde sürücü devre mikroişlemci tarafından gönderilen sinyallere tepki vermeyecektir. Buna göre iki motor için de “D1” girişlerinin lojik 0, “D2” girişlerinin ise lojik 1 değerini alması gerekir. Bu bağlantılar kartın üzerinde aynı hizada bulunan ek boşluklar kullanılarak yapılır. Ek boşluklardan kare şeklinde olanlar kartın sol kısmındaki “GND” isimli bağlantı noktasına, yuvarlak boşluklar ise kartın üzerindeki “VDD” isimli bağlantı noktasına seri şekilde bağlanmıştır. Bu sayede “VDD” ve “GND” girişleri kullanılarak kartın üzerinde gerekli yerlerin lojik 1 ve lojik 0 değerini alması sağlanabilir. Bu uygulamada “EN”, “D1” ve “D2” girişleri bu yöntemle istenen şekilde ayarlanmıştır. Bu sayede mikroişlemci ile sürücü kart arasına bu girişler için herhangi bir kablo bağlanmamıştır. Mobil robotun hareket edebilmesi için bu girişlere ek olarak “IN1” ve “IN2” girişlerinin de kontrol edilmesi gerekmektedir. Bu girişlere uygulanan sinyalin PWM frekansına göre motorun hızı kontrol edilir. Motorun dönüş yönü ise sinyalin uygulandığı girişe bağlıdır. Normal çalışma durumunda iki girişin lojik olarak farklı değerlerde olması gerekmektedir. Sürücü kart ile mikroişlemci arasındaki bağlantılar tamamlandıktan sonra kartın sağ kısmında bulunan enerji girişi ve motor çıkışları ise gerekli yerlere bağlanır. Bağlantılar tamamlandıktan sonra palet motorları mikroışlemciden gelen komutlara göre kontrol edilebilir. Yapılan uygulamada kartın üzerinde bulunan diğer giriş/çıkışlar kullanılmamıştır. Bu yüzden bu pinlere herhangi bir atama yapılmamış ve pinlerin başlangıç lojik durumunda kalması sağlanmıştır.

3.2.4 Kablosuz haberleşme ünitesi

Yapılan çalışmada mobil robotun hareketi uzaktan kablosuz haberleşme yoluyla kontrol edilmiştir. Hazırlanan kumanda devresi ile hem mobil robotun palet motorlarının hareketi kontrol edilmiş hem de robotun üzerinde bulunan kameranın bakış yönü değiştirilmiştir. Burada belirtilen işlem için ZigBee haberleşme protokolünü kullanan XBee isimli modüller kullanılmıştır. Bu modüller genellikle ucuz ve az veri gönderimi gerektiren uygulamalarda yer alırlar. Benzer şekilde bu uygulamada da modüller sadece kumanda verilerinin iletilmesinde kullanılmıştır. Şekil 3.7’de kullanılan XBee modülleri gösterilmiştir.



Şekil 3.7 : XBee kablosuz haberleşme modülünün görünümü.

Kablosuz haberleşmede bu modüllerin kullanılmasının en büyük nedeni pratik yapılarıdır. Önceden de bahsedildiği gibi piyasada Arduino Duemilanove mikroişlemci kartı için özel olarak üretilmiş eklenti kartları bulunur. Bu kartlar Arduino Duemilanove üzerine yerleştirilerek mikroişlemcinin çeşitli ek özellikler kazanmasını sağlar. Kablosuz haberleşme için de Arduino uyumlu XBee eklenti kartı kullanılmıştır. XBee Shield isimli bu kart, üzerinde bulunan XBee modülü ile mikroişlemcinin 0 ve 1 numaralı haberleşme uçları arasında bağlantı kurar ve bu sayede mikroişlemcinin başka bir XBee modülü ile kablosuz olarak haberleşmesi sağlanır. Şekil 3.8’de XBee Shield kartının mikroişlemci üzerine monte edilmiş hali görülmektedir.



Şekil 3.8 : XBee Shield ve Arduino Duemilanove birleşik görünümü.

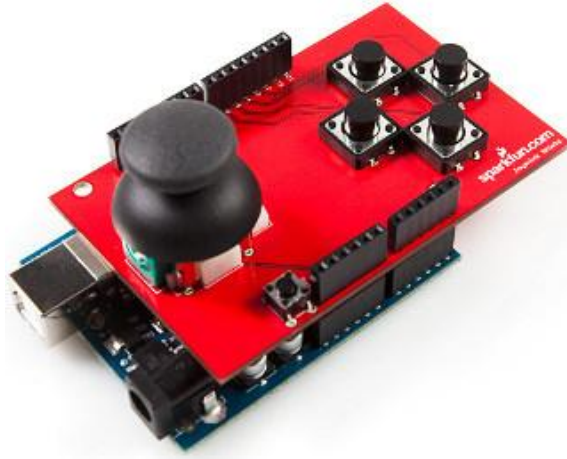
XBee haberleşme modülleri 1 serisi ve 2.5 serisi olarak ikiye ayrılır. Bu modüllerden 1 serisine ait olanlar sadece iki nokta arasında kablosuz haberleşme gerçekleştirebilirler. 2.5 serisine ait olan modüller ise 1 serisine ek olarak kablosuz ağ sistemleri kurmak için de kullanılabilirler. İki seriye ait modüller de XBee Shield ile uyumludur. Bu sayede eklenti kartı farklı ortamlar için kablosuz haberleşme uygulamalarında kullanılabilir.

Kullanıcı ile robot arasında haberleşmeyi sağlamak için hem robot üzerine hem de operatör tarafında birbiriyle haberleşebilen XBee modüllerinin bulunması gerekmektedir. Robot üzerindeki mikroişlemciye XBee modülü XBee Shield kullanılarak monte edilmiştir. Benzer şekilde operatör tarafında hazırlanan kumanda devresinde de bir adet Arduino Duemilanove mikroişlemci kartı kullanılmış ve kartın üzerine yerleştirilen XBee Shield vasıtasıyla iki nokta arasında kablosuz haberleşme sağlanmıştır.

XBee Shield'ın sağladığı en büyük kolaylık kablosuz haberleşmenin başlatılması için kod kısmında ek bir fonksiyona ihtiyaç duyulmamasıdır. Mikroişlemci bu kart sayesinde seri iletişim sırasında kullandığı fonksiyonları kullanarak ZigBee protokolü üzerinden veri alışverişi gerçekleştirebilir. Fakat yapılan uygulamada XBee Shield'ın bağlandığı mikroişlemci üzerinde 0 ve 1 numaralı pinlerine başka bir cihazın bağlanılmasına dikkat edilmelidir. Aksi takdirde kablosuz haberleşmede bozulmalar yaşanabilir, zira bu pinler önceden de belirtildiği gibi seri iletişim gerçekleştirebilen özel pinlerdir.

3.2.5 Kumanda devresi

Yapılan çalışmada mobil robotun ve kamera sisteminin kablosuz şekilde uzaktan kontrol edilebilmesi bir kumanda devresi hazırlanmıştır. Bunun için bir adet Arduino Duemilanove mikroişlemci kartı, bir adet XBee Shield ve kumanda görevini gören bir adet Joystick Shield kullanılmıştır. Kumanda devresine ait mikroişlemci kodu ise Ek A.2 bölümünde verilmiştir. Bu bölümde Joystick Shield'ın yapısından bahsedilecektir. Şekil 3.9'da sadece Joystick Shield'ın Arduino Duemilanove üzerine monte edilmiş bir şekli görülmektedir.



Şekil 3.9 : Joystick Shield ve Arduino Duemilanove birleşik görünümü.

Joystick Shield'in üzerinde robotun kontrolünde kullanılacak çeşitli kumanda ekipmanları bulunur. Buna göre kartın üzerinde 4 adet anlık elektrik düğmesi, 1 adet hareketli kol ve hareketli kolun altında 1 adet normal elektrik düğmesi mevcuttur.

Kartın üzerinde bulunan anlık elektrik düğmelerinin hepsi üzerlerine basılı şekilde duruldukça durumunu değiştirir. Düğmeler mikroişlemci pinleri ile lojik toprak bağlantısı arasına yerleştirilmiştir. Normal çalışma sırasında düğmelerin bağlandığı pinler pull-up direnci kullanılarak lojik 1 durumuna getirilir. Düğmeye basıldığı anda ise pin lojik 0 değerini alır ve bu sayede mikroişlemci tarafından düğmenin durumu algılanır. Joystick Shield üzerinde bulunan 4 elektrik düğmesinden iki tanesi robot üzerindeki kameranın bakış yönünü değiştirmek için kullanılmıştır.

Kartın üzerinde bulunan hareketli kol ise iki farklı eksen üzerinde hareket edebilecek şekilde tasarlanmıştır. Kolun anlık pozisyon bilgisinin elde edilebilmesi için kolun içinde kolun hareketine göre değerini değiştiren iki adet potansiyometre bulunur. Bu potansiyometreler üzerinden gerilim okuması yapılarak kolun anlık pozisyonu belirlenebilir. Bunun için potansiyometrelere mikroişlemci tarafından 5 V gerilim uygulanır ve potansiyometrelerin direnç değerlerine göre değişen gerilim sinyalleri mikroişlemciye ait iki tane analog giriş üzerinden okunur. Bu sayede analog girişler üzerinden kolun X ve Y eksenlerindeki konumu belirlenir. Yapılan uygulamada hareketli kol palet motorlarını kontrol etmek için kullanılmıştır.

Bu parçalara ek olarak hareketli kolun alt kısmında ayrı bir elektrik düğmesi daha bulunur. Bu düğme diğerlerinin tersine kolun üzerine her basıldığında durumunu değiştirir.

Bütün düğmeler ve hareketli koldaki potansiyometre çıkışları mikroişlemci üzerinde uygun yerlere bağlanmıştır. Joystick Shield ürününü satan firmanın internet sayfasından kartın mikroişlemci üzerinde bağlandığı pinler hakkında bilgi edinilip daha sonra uygulamaya göre istenen ekipmanlar kullanılabilir [39].

3.2.6 Kablosuz kamera

Yapılan çalışmada kamera dengeleme sistemi üzerine yerleştirilmek üzere bir adet kablosuz kamera kullanılmıştır. Bu kamera sayesinde hem dengeleme sistemi test edilmiş hem de robot ile operatör arasında görsel veri aktarımı yapılmıştır. Veri aktarımının yapılabilmesi için operatör tarafında bir adet alıcı kullanılmıştır. Bu alıcı üzerinden elde edilen görüntü verileri EasyCAP USB isimli bir dönüştürücü

sayesinde S-Video'dan USB'ye dönüştürülerek bilgisayara iletilmiştir. Şekil 3.10'da kablosuz kamera ve kamera ile birlikte kullanılan alıcı birlikte gösterilmiştir.



Şekil 3.10 : Kablosuz kamera ve alıcı ünitenin görünümü.

Kablosuz kamera sistemi diğer elektronik ekipmanlardan bağımsız çalışmaktadır. Bu yüzden kamera ile mikroişlemci arasında bir bağlantı bulunmamaktadır. Kameranın çalışması için sadece enerji girişine uygun pilin bağlanması yeterli olacaktır.

3.3 Yazılım ve Haberleşme

Bu kısımda ilk olarak hem robot üzerinde hem de kumanda devresinde kullanılan mikroşlemcilerin kodlama yöntemleri anlatılmıştır. Daha sonrasında ise ataletsel ölçüm cihazı ile haberleşmede kullanılan I²C protokolü gösterilmiştir.

3.3.1 Mikroşlemci programlama dili

Arduino firmasına ait mikroşlemci kartları firmanın internet sayfasında bulunan bir ara yüz programı aracılığıyla programlanır. Bu program hem derleyici hem de kod yükleyici olarak görev yapar. Kullanıcı ara yüz programında programlama dili olarak mikroşlemciler için özel fonksiyonlar barındıran basitleştirilmiş bir C++ dili kullanılmaktadır. Bu ek fonksiyonlar sayesinde mikroşlemciye ait birçok işlem tek satırlık kodlarla tanımlanabilir [35]. Yapılan çalışmada kullanılan ek fonksiyonların isimleri ve görevleri aşağıda verilmiştir.

- pinMode(): Çalışmada kullanılacak pinlerin giriş ya da çıkış olarak yapılandırılmasında kullanılır.
- digitalWrite(): Dijital çıkış pinlerine lojik değer ataması yapar. Buna göre dijital çıkışlar lojik 0 ya da lojik 1 değerini alabilir.

- digitalRead(): Dijital giriş olarak tanımlanmış pinin o andaki lojik değerini verir.
- analogReference(): Mikroişlemci üzerindeki analog girişlerin referans gerilimini belirlemek için kullanılır.
- analogRead(): Analog girişlere bağlı 10 bit çözünürlüğe sahip analog-dijital dönüştürücüden elde edilen değerleri okur.
- analogWrite(): PWM özelliği bulunan çıkışlardan PWM yöntemi ile üretilmiş analog sinyaller elde etmek için kullanılır.
- millis(): Mikroişlemci programının çalışmaya başladığından beri geçen zamanı milisaniye cinsinden verir.
- delay(): Belli bir milisaniye için kodun çalışmasını durdurur.
- Serial: Seri iletişim kullanılarak başka bir nokta ile veri alışverişi yapmak için kullanılır. Yapılan çalışmada kullanılan XBee modülleri seri iletişim pinlerine bağlanmış ve bu sayede kablosuz haberleşme seri iletişime ait kodlar kullanılarak gerçekleştirilmiştir.
- Serial.print() ve Serial.println(): Seri iletişim üzerinden bilgi alışverişi için standart Amerikan kodu (ASCII) yapısında veri iletir.
- Serial.read(): Seri iletişim ile mikroişlemciye iletilen veriyi okur.
- Serial.begin(): Seri iletişimi başlatır. Burada kullanılacak bilgi akış hızı (baud rate) tanımlanır. Bilgi akış hızı kablosuz haberleşme için XBee modüllerinde tanımlanmış hız ile aynı olmalıdır.
- Serial.available(): Seri iletişim üzerinden mikroişlemciye ulaşmış bayt sayısını belirtir. Genellikle seri iletişimin aktivitesini kontrol etmek için kullanılır.

Mikroişlemci kodu “setup” ve “loop” olmak üzere iki temel fonksiyondan meydana gelir. Mikroişlemcinin çalışmaya başlarken gerçekleştireceği işlemler “setup” fonksiyonu içinde belirtilir. Buna göre karta enerji verildikten sonra mikroişlemci bir seferlik bu kısımda bulunan kodda belirtilen görevleri yerine getirir. Sonrasında ise mikroişlemci “loop” fonksiyonunda bulunan koda göre çalışmaya devam eder. Burada bulunan kod ise mikroişlemci aktif olduğu sürece sürekli kendini tekrarlar.

Önceden de belirtildiği gibi Arduino mikroişlemci kartlarının farklı sensörler ya da cihazlar ile çalışmasında kolaylık sağlayan çeşitli kütüphane dosyaları mevcuttur. Burada da kamera dengeleme sistemini hareket ettiren servo motorların kontrol edilmesinde ve ataletsel ölçüm cihazı ile haberleşmede “Servo” ve “Wire” isimli iki ayrı kütüphane dosyası kullanılmıştır. Bu kütüphanelerde tanımlanmış ve mikroişlemci kodlarında yer alan özel fonksiyonlar aşağıda verilmiştir.

- attach(): Fonksiyon içinde belirtilen pinin servo motorun sinyal girişine bağlı olduğunu mikroişlemciye bildirir.
- write(): Servo motoru fonksiyon içinde belirtilen açı değerine hareket ettirir.
- Servo: Kod içinde servo motor objesi tanımlar. Sonrasında diğer fonksiyonlar bu obje ile birlikte yazılır.
- Wire.begin(): I²C protokolü üzerinden iletişimi başlatır.
- Wire.beginTransmission(): I²C iletişim hattı üzerindeki cihazla haberleşmeyi başlatır. İletişim hattı üzerindeki cihaz yedi bitten oluşan sabit bir adrese sahiptir. Cihaz ile haberleşmek için fonksiyonun içine bu adresin yazılması gerekir.
- Wire.write(): I²C protokolü üzerinden iletilecek baytlar tanımlanır. Bu veriler haberleşilen cihaz içindeki yazmacın adresi ya da cihazın yazmacına yazılacak bitler olabilir.
- Wire.endTransmission(): Wire.beginTransmission() fonksiyonundan itibaren Wire.write() fonksiyonları içine yazılmış baytları cihaza iletir. Daha sonra da cihaz ile haberleşmeyi durdurur.
- Wire.requestFrom(): Cihazdan belli bir büyüklükte veri talebinde bulunur.
- Wire.read(): Cihaz tarafından mikroişlemciye gönderilen verileri okur.
- Wire.available(): I²C iletişim hattı üzerinden mikroişlemciye gönderilen baytların sayısını verir. Genellikle iletişim hattının aktivitesini kontrol etmek için kullanılır.

“Servo” isimli kütüphane dosyası aracılığıyla herhangi bir ek kontrolör kullanılmadan servo motorların istenen şekilde belirlenen konuma doğru hareket etmesi sağlanabilir. Yapılan uygulamada normal bir doğru akım motoru kullanılsaydı

motorların millerinin konum kontrolü için ayrıca PID (Proportional-Integral-Derivative) gibi bir kontrol algoritması kullanılması gerekirdi. Fakat servo motorlarda motorun milinin açısal konumu direkt olarak PWM yöntemiyle üretilmiş bir sinyal kullanılarak belirlenebilir. Bu konu ile ilgili ayrıntılı bilgi Bölüm 3.1.3'te verilmiştir. Sonuç olarak servo motorların belli bir açı değerine hareket edebilmesi için ayrı bir kontrol algoritmasına gerek yoktur. Mikroişlemci kodunda sadece kütüphanede belirtilen fonksiyonlar kullanılarak servo motorların belli bir açıya doğru hareket etmesi sağlanabilir. Bunun için mikroişlemci kodunda Euler açıları hesaplandıktan sonra servo motorların bu açılara göre hareket ettirilmesi için mikroişlemci kodunda ayrı bir kontrol algoritması kullanılmamıştır.

“Wire” kütüphane dosyasındaki fonksiyonlar ile I²C protokolü üzerinden yapılan haberleşmenin gerçekleşmesi sağlanmıştır. Burada belirtilen ek fonksiyonlar sayesinde I²C protokolünde kullanılan veri paketleri kolayca yaratılmış ve uygun cihazlara iletilmiştir.

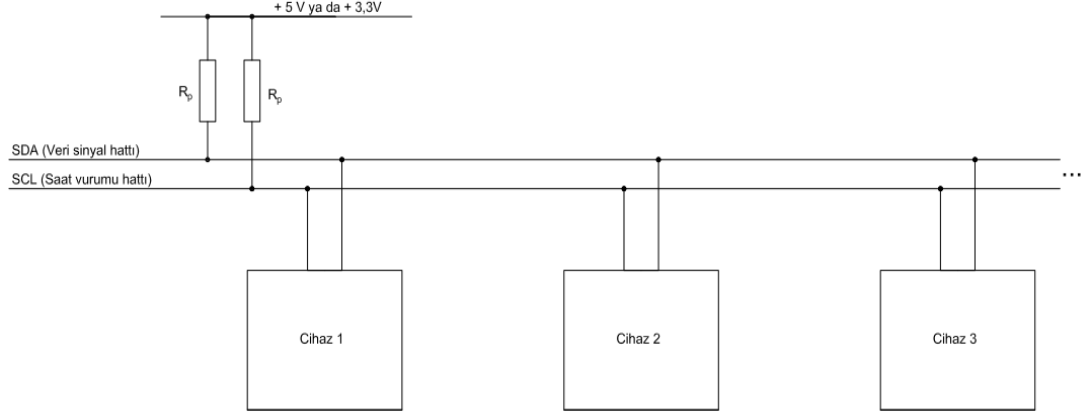
3.3.2 Haberleşme protokolü

Kamera dengeleme sistemi uygulamasında ataletsel ölçüm cihazı ile haberleşmede SPI ve I²C isimli iki farklı haberleşme protokolü kullanılabilir. Burada haberleşme I²C protokolü üzerinden yapılmıştır. Bu kısımda I²C haberleşme protokolü ayrıntılı şekilde anlatılmış [40] ve ataletsel ölçüm cihazındaki sensörlere ait kullanılan yazmaçlar gösterilmiştir [29,37].

I²C protokolü Philips tarafından üretilmiş bir iki kablolu haberleşme protokolüdür. Protokol ilk olarak 1980'lerde mikroişlemciler ile Philips'e ait yan birim entegre devrelerin haberleşmesinde kullanılmıştır. Daha sonra protokol fiziksel yapısının basitliğinden dolayı telekomünikasyon, otomotiv, bilgisayar gibi birçok alanda kullanılmaya başlanmıştır. Günümüzde I²C protokolü sadece Philips'e özel bir protokol olmaktan çıkıp birçok firmanın ürünlerinde haberleşme için kullanılmaktadır.

I²C protokolü fiziksel olarak veri sinyal hattı (SDA) ve saat vurumu hattı (SCL) olarak kullanılan iki kablodan meydana gelir. SDA cihazlar arasında verilerin iletiildiği hattır. SCL hattı ise I²C veri yolu üzerinden yapılan veri transferinin senkronizasyonu için kullanılır. Veri yolu üzerinde kullanılan iki hat da iki yönlü olarak çalışabilir. Haberleşme sırasında hatta bağlı olan cihazlar hattın lojik

durumunu deęiřtirir ve bu sayede retilen veri paketleri hat zerindeki bařka bir cihaza iletilir. Normal alıřmada iki hat da birer pull-up direnci ile 5 V ya da 3,3 V byklęnde pozitif gerilime baęlanır. Kullanılacak olan pull-up direnlerinin byklękleri cihaza ait veri fynde verilir. I²C veri yolunun genel yapısı řekil 3.11’de verilmiřtir. Burada R_p direnleri pull-up direnlerini temsil eder.



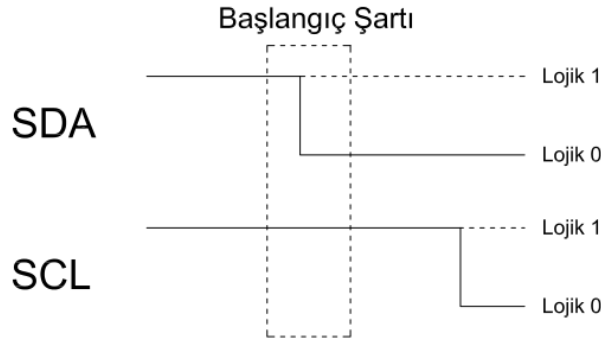
řekil 3.11 : I²C veri yolunun fiziksel yapısı.

Hat zerinde bulunan cihazlar farklı zelliklere sahip olabilir; fakat genel olarak hat zerindeki cihazlar merkez cihaz (master) ve alt istasyon (slave) olarak ikiye ayrılır. SCL hattını yneten cihazlara merkez cihaz denir. Bu cihazlar veri yolu zerindeki haberleřmeyi ynetirler. Bu cihazların verdięi komutlara gre haberleřen cihazlara da alt istasyon denir. Cihaz hangi tip olursa olsun SDA hattı zerinden veri alıřveriři yapabilir; fakat bu hat zerindeki veri transfer iřlemi merkez cihaz tarafından ynetilir. Yapılan uygulamada mikroiřlemci, merkez cihaz grevi grrken jiroskop ve ivmeler alt istasyon olarak alıřmıřtır.

I²C veri yolu zerinden yapılan veri transferinin zamanlaması merkez cihaz tarafından yapılır. Buna gre merkez cihaz veri yolu zerindeki iletiřimin ne zaman bařlayacaęını ya da biteceęini belirler. Bunun iin I²C protokolnde belirli lojik řartlar tanımlanmıřtır. Merkez cihaz iki veri hattını bu řartlarda belirtilen lojik durumlara getirerek haberleřmeyi bařlatabilir ya da bitirebilir.

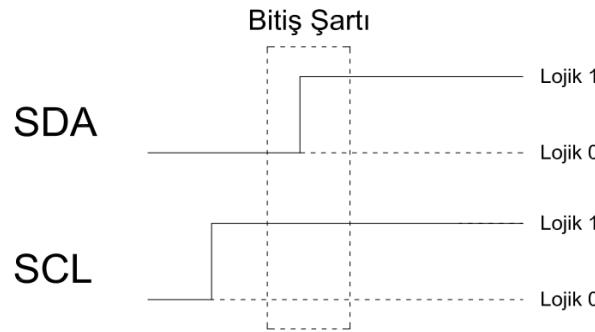
Veri transferinin bařlangı ve bitiř durumları dıřında SCL hattının lojik deęeri 1 olduęu srece veri yolu zerindeki cihazlar SDA hattının lojik deęerini deęiřtiremez. Merkez olarak tanımlanmıř cihaz SDA hattı zerinden veri transferini bařlatmak iin iki hattın lojik durumunu řekilde 3.12’deki gibi deęiřtirir. I²C veri yolu zerinden iletiřimi bařlatmak iin tanımlanmıř bu lojik duruma bařlangı řartı denir. Bařlangı

şartı gerçekleştikten sonra SCL hattı lojik olarak 0 değerini alacağından veri yoluna bağlı cihazlar artık SDA hattının lojik durumunu değiştirebilir.



Şekil 3.12 : Veri transferinin başlaması için gereken lojik değerler.

Veri transferi tamamlandıktan sonra ise merkez cihaz iki hattı Şekil 3.13'te gösterilen bitiş durumuna getirir. Bu andan sonra SCL hattı lojik 1 değerini alacağından tekrar başlangıç şartı yerine gelene kadar veri transferi yapılamaz.

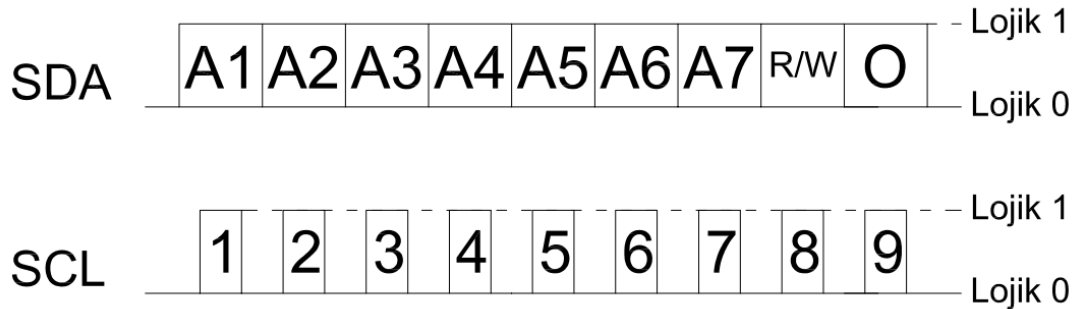


Şekil 3.13 : Veri transferinin durması için gereken lojik değerler.

Sonuç olarak I²C protokolü üzerinden yapılan iletişimlerde cihazlar yukarıda verilen iki durum arasında haberleşebilir. Cihazlar arasında haberleşme süreci genel olarak üç adımda gerçekleşir. İlk olarak merkez cihaz veri yolu üzerinde haberleşeceği alt istasyonu belirler. Daha sonra bu alt istasyonun hafızasında erişilecek yazmacın adresi belirtilir. En son olarak da bu yazmaç üzerinden okuma ya da yazma işlemi gerçekleşir. Burada belirtilen her işlem SDA hattında 8 bit büyüklüğünde bir veri paketi (bayt) olarak tanımlanmıştır.

I²C protokolü üzerinden iletişime başlanırken ilk olarak merkez cihaz alt istasyona ait bilgileri veri yoluna iletir. I²C protokolü ile haberleşen her cihazın 7 bitlik bir adres bilgisi mevcuttur. Buna göre veri yolu üzerine farklı adrese sahip 128 tane cihaz bağlanabilir. Haberleşmenin başlaması için ilk olarak merkez cihaz SDA ve SCL hatlarının Şekil 3.13'te gösterilmiş olan lojik değerleri almasını sağlar ve bu

sayede SDA hattı üzerinde veri transferi başlatılır. Artık SCL hattının lojik değeri 0 olduğu için SDA hattının lojik değeri de değiştirilebilir. Veri transferi başladıktan sonra alt istasyona ait 7 bitlik adres bilgisi teker teker SDA hattına iletilir. Gönderilen veri paketinde her bitin lojik değerine göre SDA hattının da lojik değeri değişir. SDA hattına iletilen her bitten sonra SCL hattı pull-up direnci tarafından tekrar lojik 1 değerine getirilir. Fakat bütün veri paketi iletilene kadar merkez cihaz SCL hattını lojik 0 değerine getirmeye devam eder. 7 bitlik veri paketi iletdikten sonra merkez cihaz tarafından SDA hattına “R/W” isminde bir bit daha gönderilir. Bu bitin aldığı lojik değeri merkez cihazın okuma ya da yazma işlemlerinden hangisini yapacağını belirtir. Buna göre bu bitin lojik değeri 1 ise merkez cihaz haberleşilen alt istasyondan veri okur. “R/W” isimli bitin lojik 0 değerinde olması durumunda ise merkez cihaz alt istasyonda belirli bir yazmaca veri yazar. Merkez cihaz tarafından gönderilen ilk veri paketi haberleşilecek cihaza ait 7 bitlik adres bilgisinden ve okuma/yazma bitinden meydana gelir. Önceden belirtildiği gibi SDA hattında gönderilen her veri paketi 8 bit büyüklüğünde baytlardan meydana gelir. Şekil 3.14’te adres ve okuma/yazma bilgisi barındıran bir baytlık verinin SDA hattı üzerinden transferi sırasında iki hattın aldığı lojik değerler gösterilmiştir. Burada görüldüğü gibi gönderilen verinin durumuna göre SDA hattındaki bitler lojik 1 ya da lojik 0 değerini alırken SCL hattı her bit için lojik değerini değiştirir.

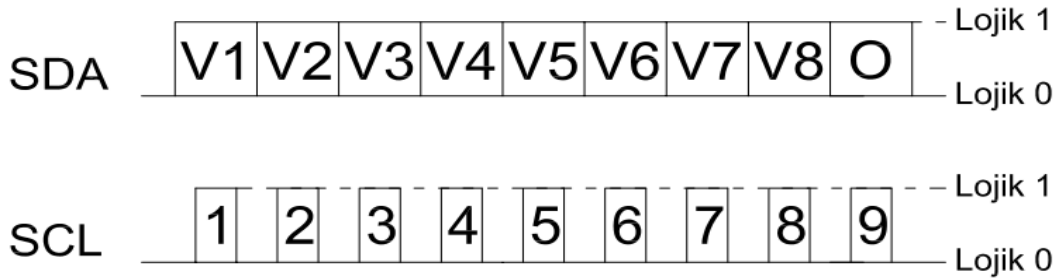


Şekil 3.14 : Adres bilgisi ve okuma/yazma bitini barındıran baytın lojik değerleri.

Yukarıdaki şekilde SDA hattında “R/W” bitinden sonra bir adet bit daha görülmektedir. Bu bit iletilen mesajın durumunu belirten onay bitidir. Buna göre bu bit kullanılarak okuma yapan cihaz tarafından iletilen mesajın alındığı belirtilir. Örnek olarak merkez cihaz alt istasyona veri iletirken gönderilen her bayttan sonra alt istasyon SDA hattına onay bitini gönderir. Böylece merkez cihaz iletilen baytın yerine ulaştığını anlar. Yeni veri paketi onay biti SDA hattına iletilene kadar gönderilmez. Onay biti ayrıca merkez cihazdan okuma yapılırken gönderilecek

verinin bittiğini sensöre bildirmek için de kullanılır. Normal durumlarda onay biti hep lojik 0 değerini alır. Merkez cihazdan okuma yapılan durumlarda ise gönderilecek veri bittiği zaman merkez cihaz SDA hattına onay bitini lojik değeri 1 olacak şekilde iletir. Bu sayede alt istasyonda bulunan cihaz veri yazmayı durdurur. Bu işlemden sonra merkez cihaz da hatları bitiş durumuna getirerek haberleşmeyi durdurur.

Adres ve okuma/yazma bilgisini barındıran bayt SDA hattına gönderildikten sonra haberleşilecek olan cihazın yazmaç adres bilgisi belirtilmelidir. Buradaki adres bilgisi de bir bayt büyüklüğündedir. Merkez cihaz yazmacın adres bilgisini bir önceki işlemdeki gibi teker teker SDA hattına iletir. Şekil 3.15'te yazmaç adres bilgisini barındıran bir baytlık verinin SDA hattı üzerinden transferi sırasında iki hattın aldığı lojik değerler gösterilmiştir.



Şekil 3.15 : Bir baytlık veri transferi sırasında hatların lojik değerleri.

Yazmaç adresi de belirlendikten sonra merkez cihaz tarafından önceden iletilmiş “R/W” bitinin durumuna göre okuma ya da yazma yapılabilir. Burada okunan ya da yazılan veri paketi Şekil 3.15'te gösterilen yapıdadır. Buna göre okunan ya da yazılan veri paketleri de 1 bayt büyüklüğe sahiptir ve paketlerin içerikleri işlem yapılan yazmaca göre değişir. Sensörde bulunan yazmaçların içerikleri sensörlere ait veri föylerinde açıklanmıştır. Haberleşme sırasında okuma ya da yazma işlemi merkez cihaz transfer işlemini durdurana kadar devam eder.

I²C protokolü üzerinden haberleşen cihazlar arasında farklı haberleşme durumları için farklı veri alışveriş sıralamaları mevcuttur. Yazma işleminde hatlar merkez cihaz tarafından başlangıç durumuna getirildikten sonra ilk olarak adres ve okuma/yazma bilgisini barındıran bayt SDA hattına iletilir. Daha sonra ise haberleşilen cihazın yazmaç adres bilgisi belirtilir. En son olarak da yazılacak veri paketi sayısı kadar bayt cihaza gönderilir. İletim işlemi tamamlandıktan sonra hatlar bitiş durumuna

getirilir. Okuma işlemi sırasında ise okumadan önce merkez cihaz tarafından yazma işlemi yapılır. Buna göre ilk olarak cihazın adres bilgisi SDA hattına iletilir. Daha sonra okuma yapılacak yazmacın adresi bildirilir. Bu veri paketi gönderildikten sonra haberleşme durdurulur ve tekrar başlatılır. Bu sefer cihazın adres bilgisinin belirtildiği veri paketinde “R/W” biti lojik 1 değerini alır ve önceden belirtilen yazmaç adresinden veri okunur. Veri okuma işlemi yapıldıktan sonra da merkez cihaz tarafından haberleşme durdurulur.

Önceden belirtildiği gibi kamera dengeleme sisteminde kullanılan ataletsel ölçüm cihazı da mikroişlemci ile I²C protokolü üzerinden haberleşmektedir. Bu yüzden sensör ile çalışma yapılırken bazı yazmaçlar önemli yer tutar. Burada uygulamada kullanılan yazmaçlar ve yazmaçlara yazılan değerler gösterilmiştir.

3.3.2.1 ADXL345 I²C yazmaçları

ADXL345 üzerinde birçok farklı fonksiyon mevcuttur. Burada sensöre ait ek fonksiyonlar kullanılmamış sadece ölçüm parametreleri ayarlanmıştır. Aşağıda sistem içinde kullanılan ölçüm parametrelerinin yazmaç adresleri ve ayarları gösterilmiştir. Burada kullanılan bütün adresler ve yazmaçlara yazılan değerler sensöre ait veri füyünden alınmıştır [29].

- Yazmaç 0x2C: Bu yazmaç adresi düzenlenerek ivmeölçerin veri iletim hızı belirlenebilir. Önceden belirtildiği gibi kullanılan kodun döngü süresi 20 ms olduğundan veri iletim frekansının 50 Hz’ten az olması gerekmektedir. Bunun için 0x2C adresindeki yazmaçta en sağda kalan 4 bitin uygun şekilde düzenlenmesi gerekir. Sensöre ait veri füyünde bu bitlere sırasıyla lojik 1–0–0–1 değerlerinin yazılması sonucunda sensörün veri iletim hızının 50 Hz olacağı belirtilmiştir. Bunun için yapılan uygulamada 0x2C adresine sahip olan yazmaca 0x09 değeri yazılmıştır.
- Yazmaç 0x2D: Bu yazmaç sensörü ölçüm durumuna getirmek için kullanılır. Burada sağdan 4. bitin lojik olarak 1 değerini alması sonucunda sensör ölçüm durumuna geçecektir. Bu sayede veri paketleri mikroişlemci tarafına iletilecektir. Sensörü aktive etmek için yapılan uygulamada 0x2D adresine sahip olan yazmaca 0x08 değeri yazılmıştır.
- Yazmaç 0x31: Burada yazmacın en sağdaki iki bitine yazılacak değerler ile sensörün ölçme aralığı belirlenir. Önceden belirtildiği gibi yapılan çalışmada

ölçüm aralığı ± 2 g olarak alınmıştır. Bu yüzden burada 0x31 adresindeki yazmaca 0x00 değeri yazılmıştır.

- 0x32 ile 0x37 adresleri arasındaki yazmaçlar sensörden elde edilen ivme değerlerini barındırır. Her eksen için ivme değeri iki yazmaç içinde toplanmıştır. Bu değerler ilk olarak mikroişlemci tarafından okunur. Daha sonra ise kod üzerinde iki yazmaçtan elde edilen veriler birleştirilir. Bunun için iki tane veri paketi arka arkaya toplanmıştır.

3.3.2.2 ITG3200 I²C yazmaçları

ITG3200 içinde farklı ölçüm parametreleri mevcuttur. Bu kısımda bu ölçüm parametrelerinin yazmaç adresleri ve ayarlama şekilleri gösterilmiştir. Burada kullanılan yazmaç adresleri ve yazmaçlara yazılan değerler sensöre ait veri füyünden alınmıştır [37].

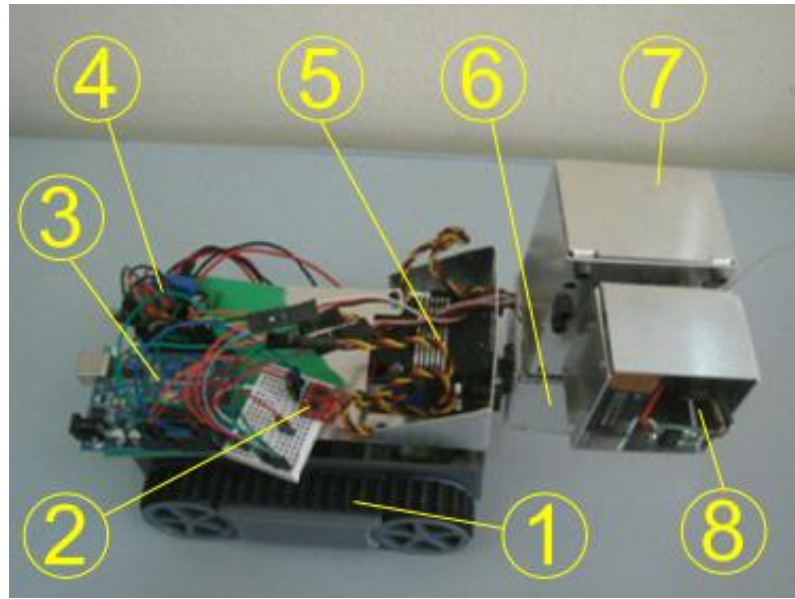
- Yazmaç 0x15: Jiroskop içinde her eksen için iki kere ölçüm yapılır. İlk olarak sensör açısal hız ölçümlerini 1 kHz ya da 8 kHz büyüklüğünde bir örnekleme frekansında yapar. Daha sonra bu veriler filtrelenerek entegre devrenin çıkışına iletilir. Burada örnekleme frekansının büyüklüğü bu filtrenin bant aralığına göre belirlenir. Söz konusu yazmaçta filtreden elde edilen çıkışlara ait veri gönderim hızı belirlenir. Mikroişlemci kodu 20 ms'lik bir sabit döngü süresine sahip olduğu için sensörden elde edilen verilerin her ölçüm adımı için 20 ms'den daha uzun bir örnekleme zamanına sahip olması gerekir. Bunun için sensörün veri gönderim hızı 50 Hz'ten daha az olacak şekilde ayarlanmalıdır. Jiroskoba ait veri füyüne göre bu frekans değeri Denklem (3.8) yardımıyla hesaplanır. Burada $F_{\text{örnek}}$ sensörün çıkışındaki örnekleme frekansını, $F_{iç}$ jiroskoptan elde edilen ölçümlerin örnekleme frekansını ve $D_{\text{yazmaç}}$ ise veri gönderim hızını belirleyen değişkeni gösterir. Burada veri gönderim hızı için yazmacın tamamına değer yazılabilmektedir. Bu sayede $D_{\text{yazmaç}}$ değeri 0 ile 255 arasında herhangi bir değer alabilir. Yapılan uygulamada bu değer 50 Hz'e olacak şekilde ayarlanmıştır. Bu yüzden 0x15 adresindeki yazmaca 0x13 değeri yazılmıştır.

$$F_{\text{örnek}} = \frac{F_{iç}}{(D_{\text{yazmaç}} + 1)} \quad (3.8)$$

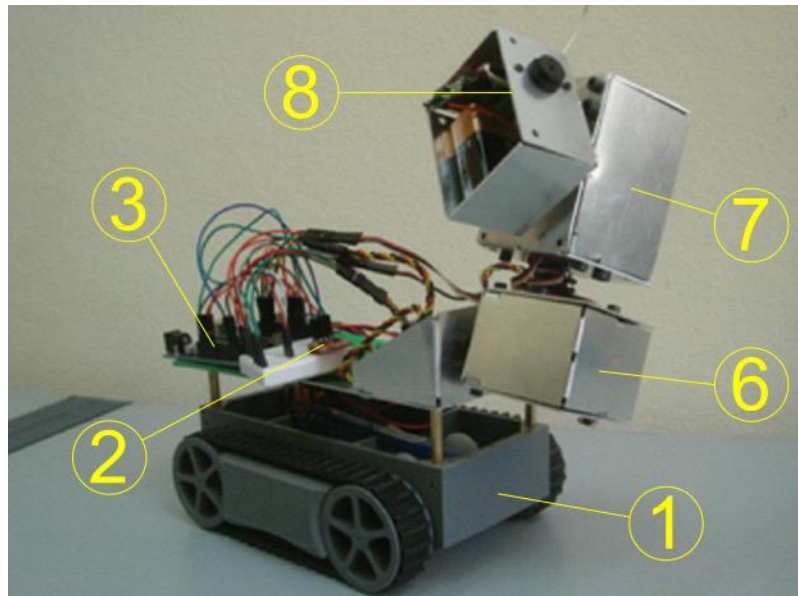
- Yazmaç 0x16: Bu yazmaç iki kısımdan meydana gelir. Birinci kısımda jiroskobun ölçme aralığı belirlenir. İkinci kısımda ise sensör üzerinde bulunan düşük geçiren filtrenin bant aralığı belirlenir. Yazmacın sağdan 4. ve 5. bitlerinin aldığı lojik değere göre sensörün ölçme aralığı belirlenir. Jiroskop üzerinde bir adet ölçme aralığı tanımlanmıştır. Bu yüzden yazmacın bitleri buna uyacak şekilde düzenlenmiştir. Yapılan uygulamada jiroskoptan elde edilen veriler için bir düşük geçiren filtre uygulaması yapılmamıştır. Bunun için jiroskoptan elde edilen verilere uygulanan filtrenin bant aralığı en düşük olacak şekilde ayarlanmıştır. Filtrenin bant aralığının belirlenmesi için yazmaca ait ilk üç bitin düzenlenmesi gerekir. Yukarıda belirtilen iki parametreye göre 0x16 adresindeki yazmaca 0x1E değeri yazılmıştır.
- 0x1B ile 0x22 adresleri arasındaki yazmaçlar jiroskoba ait sensör verisini barındırır. Bu yazmaçlar içinde hem ortamın sıcaklık verisi hem de 3 eksen için açısal hız verileri saklanır. Yapılan uygulamada 8 yazmaçtan da okuma yapılmıştır; fakat sadece X ve Y eksenindeki açısal hız değerlerini veren yazmaç verileri kullanılmıştır.

4. ROBOTUN GERÇEKLEMESİ VE KONTROLÜ

Yapılan çalışmada tasarlanıp imal edilen mobil robot bir önceki bölümde anlatılan elektronik ve mekanik donanım kullanılarak gerçekleştirilmiştir. Söz konusu donanım kullanılarak üretilen robotun son hali Şekil 4.1’de ve Şekil 4.2’de verilmiştir.



Şekil 4.1 : Robotun üstten görünümü.



Şekil 4.2 : Robotun karşıdan görünümü.

Şekillerde verilen numaralar aşağıda açıklanmıştır.

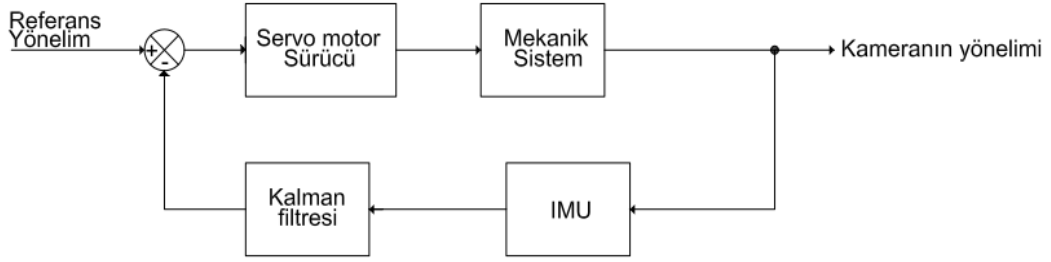
1. Mobil robot gövdesi
2. Ataletsel ölçüm cihazı
3. Arduino Duemilanove mikroişlemci kartı
4. Sürücü devre kartı
5. Yuvarlanma açısını kontrol eden servo motor
6. Sapma açısını kontrol eden birinci servo motoru muhafaza eden kutu
7. Sapma açısını kontrol eden ikinci servo motoru ve yunuslama açısını kontrol eden servo motoru muhafaza eden kutu
8. Kablosuz kamera

Bu bölümde ilk olarak kamera dengeleme sisteminin çalışma şekli anlatılmıştır. Daha sonra da robotun üzerindeki elektriksel donanıma ait bağlantı şeması verilmiş ve son olarak da kullanılan mikroişlemci kodları anlatılmıştır.

4.1 Kamera Dengeleme Sisteminin Çalışma Prensibi

Kamera dengeleme sisteminde ataletsel ölçüm cihazından elde edilen yönelim bilgisi kullanılarak robotun üzerindeki kablosuz kameranın bakış yönü sabit tutulmaya çalışılmıştır.

Yapılan uygulamada ilk olarak robotun üzerindeki kameranın bağlı olduğu gövdenin yeryüzüne göre yönelimi hesaplanmıştır. Bunun için robotun üzerine bir ataletsel ölçüm cihazı bağlanmış ve burada bulunan sensörler aracılığıyla gövdenin yönelimi Euler açısı cinsinden tanımlanmıştır. Sensörden elde edilen verilerdeki hatalar Kalman filtresi kullanılarak düzeltilmiş ve daha sonra buradan elde edilen Euler açıları mikroişlemciye kütüphane dosyaları aracılığıyla kamera dengeleme sistemi üzerinde bulunan servo motorlara iletilmiştir. Servo motorlar içinde bulunan kontrol devreleri elde edilen sinyallere göre hareket ederek konumlarını değiştirmiş ve bu sayede kameranın yeryüzüne göre yönelimi düzenlenmiştir. Her döngüden sonra gövdenin yeryüzüne göre yönelimi tekrar hesaplanmış ve burada meydana gelen değişikliklere göre kameranın konumu sabit tutulmuştur. Şekil 4.3'te yukarıda anlatılan çalışma prensibi gösterilmiştir.



Şekil 4.3 : Kamera dengeleme sisteminin çalışma prensibi.

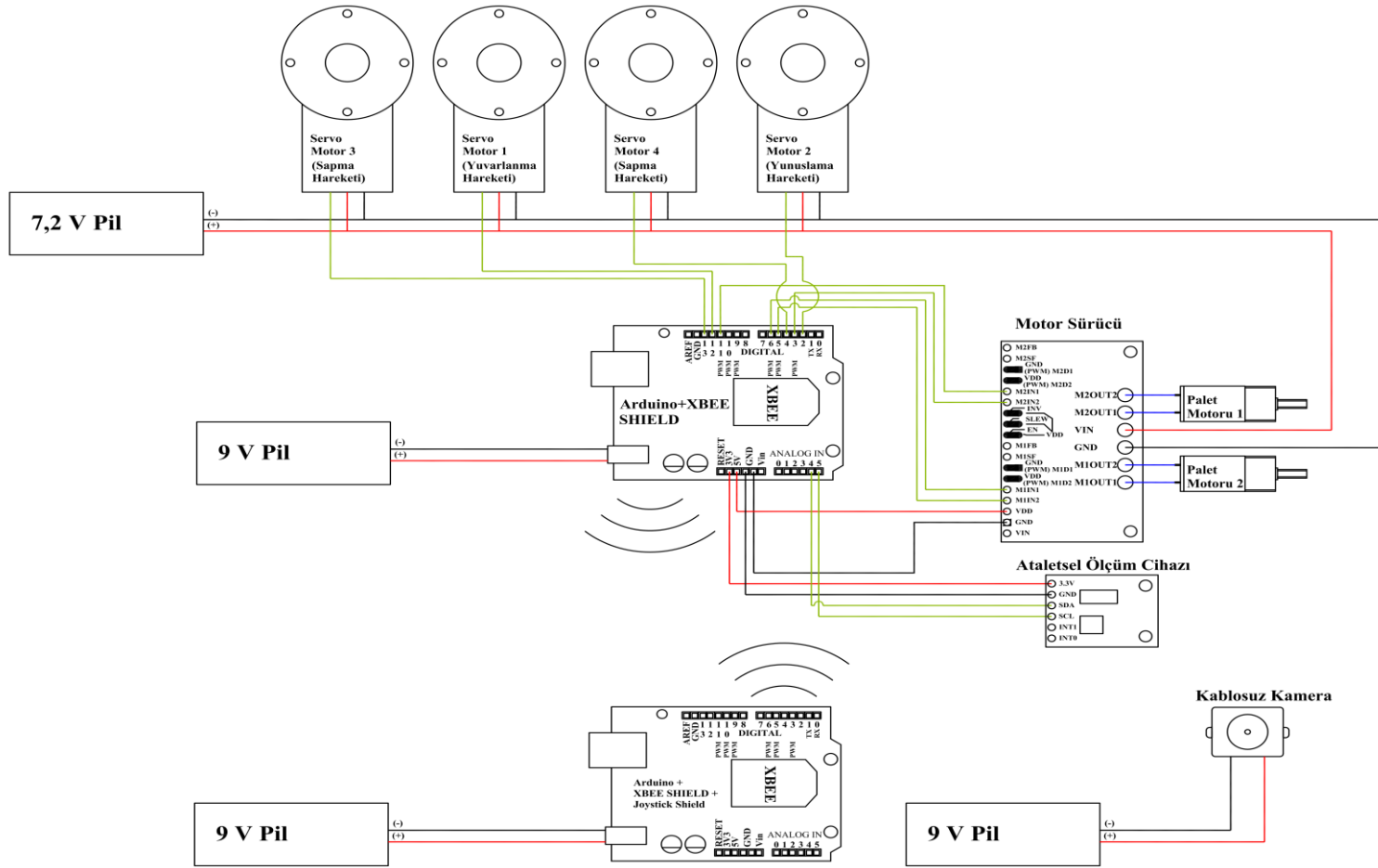
Kameranin sapma açısını yani bakış yönünü değiştirmek için ise farklı iki tane servo motor kullanılmıştır. Bu motorlar uzaktan bir kumanda devresi üzerinden kontrol edilmiştir. Kumanda devresi üzerinde tanımlanmış iki elektrik düğmesine sapma açısını kontrol eden servo motorları hareket ettirme özelliği eklenmiştir. Bu sayede motorlar kumanda üzerinde düğmeye basıldıkça bir yöne doğru hareket etmiştir. Burada kameranin 360 derece dönebilmesini sağlamak için iki servo motorun milleri birbirine bağlı şekilde monte edilmiştir. Bu sayede ilk olarak birinci servo motor 0 ile 180 derece arasında hareket eder daha sonra ikinci servo motorun hareketi başlar.

Önceden belirtildiği gibi kameranin yuvarlanma ve yunuslama hareketlerinin istenen şekilde kontrol edilebilmesi için robotun hareketi esnasında kameranin sapma açısı sıfıra eşit olmalıdır. Hareket sırasında sapma açısını kontrol eden servo motorların başlangıç konumundan farklı bir konumda bulunması durumunda diğer servo motorlar kameranin yunuslama ve yuvarlanma hareketlerini tam olarak kontrol edemez. Bunun için robotun hareketi sırasında sapma açısını değiştiren servo motorlar hep başlangıç konumuna getirilmiştir.

4.2 Bağlantı Şeması

Robot üzerindeki elektronik ekipmana ait bağlantı şeması Şekil 4.4'te verilmiştir. Burada bağlantılar enerji ve sinyal olmak üzere iki parçaya ayrılmıştır. Şekilde kırmızı ve siyah renkli çizgiler enerji bağlantılarını temsil ederken yeşil çizgiler ise robot üzerinde farklı yerlere bağlanmış olan sinyal bağlantılarını gösterir. Motorlara iletilen gerilim ise motorların hareket yönüne göre değiştiğinden mavi çizgiyle gösterilmiştir.

Kumanda devresi üst üste takılmış eklenti kartlarından meydana geldiği için sinyal bağlantılarında herhangi bir kablo kullanılmamıştır. Bu yüzden burada sadece mikroişlemciyi besleyen pile ait enerji kablosu gösterilmiştir.



Şekil 4.4 : Devre bağlantı şeması.

4.2.1 Enerji bağlantıları

Yapılan çalışmada robot üzerinde 3 ve kumanda devresinde 1 adet olmak üzere 4 tane farklı pil kullanılmıştır. Şekil 4.4'te gösterilen kırmızı ve siyah çizgiler pillere ait negatif ve pozitif uçları temsil etmektedir.

Önceden belirtildiği gibi Arduino Duemilanove mikroişlemci kartlarına farklı şekillerde enerji verilebilmektedir. Burada iki mikroişlemci kart için de adaptör girişi kullanılmıştır. Buna göre birer adet 9 V'luk pil orta kısmı pozitif olan adaptör uçları üzerinden kartlara bağlanmış ve bu şekilde mikroişlemcilere enerji verilmiştir. Robotun çalışmadığı zamanlarda bu iki pil yerinden çıkarılarak robot durdurulmuştur; zira hem kamera dengeleme sistemi hem de palet motorları sistemdeki mikroişlemci kartlar ile kontrol edilir. Bu kartlara gönderilen enerjinin kesilmesi durumunda robot çalışmaz.

Dengeleme sistemi üzerinde bulunan kamera diğer ekipmanlardan bağımsız çalışır. Buna ek olarak enerji beslemesi de diğer ekipmanlardan ayrı şekilde yapılmıştır. Mekanik kısımda belirtildiği gibi kamera dengeleme sistemi birbirine bağlı 3 adet metal kutudan meydana gelmektedir. Kablosuz kamera ise bu sistemin en uç kısmında bulunur. Normal çalışma sırasında dengeleme sistemi sürekli hareket edeceği için kutuların içinden geçirilen herhangi bir kablounun kopma riski kablo sayısı arttıkça artar. Bu yüzden kameraya enerji veren pil kameranın bulunduğu yere yakın şekilde monte edilmiştir.

Servo motorlar ve palet motorları aynı pil üzerinden beslenmiştir. Bunun için 7,2 V çıkış gerilimine sahip Lityum-Polimer pil kullanılmıştır. Bütün ortak besleme noktalarının bir araya getirebilmesi için robotun üzerine bir adet elektronik devre tahtası yerleştirilmiştir.

Yapılan çalışmada ataletsel ölçüm cihazı ve sürücü kartının enerji beslemesi ise Arduino Duemilanove üzerinden yapılmıştır. Bunun için ataletsel ölçüm cihazı mikroişlemci kart üzerindeki 3,3 V çıkışına, sürücü kart ise aynı kart üzerindeki 5 V çıkışına bağlanmıştır. Bu devrelerin çalışma esnasında çektikleri akımlar çok düşük olduğu için mikroişlemci üzerinden enerji verilmesinde bir sakınca yoktur.

Yapılan projede esas amaç kameranın hareket esnasında dengelenmesi olduğundan normalde arama kurtarma robotlarında gerek duyulan batarya çalışma süresi hesabı burada yapılmamıştır.

4.2.2 Sinyal bağlantıları

Mikroişlemci ile robot üzerindeki çeşitli elektronik ekipmanlar arasında sinyal bağlantıları bulunmaktadır. Yapılan çalışmada mikroişlemciyle ataletsel ölçüm cihazı, servo motorlar ve sürücü devre arasındaki sinyal alışverişi için bağlantı kablosu kullanılmıştır. Kullanılan eklenti kartları da üzerlerine yerleştirildikleri mikroişlemci üzerindeki çeşitli pinlere bağlanarak ara bir kablo kullanmadan sinyal bağlantısının gerçekleşmesine olanak sağlarlar. Bu bölümde mikroişlemci ile sinyal alışverişi yapan devre ve cihazlar anlatılacaktır.

Servo motorların kontrolü için önceden belirtildiği gibi PWM yöntemi kullanılmaktadır. Buna göre motorlar mikroişlemci tarafından gönderilen sinyalin uzunluğuna göre belli bir açı değerine doğru hareket ederler. Arduino Duemilanove mikroişlemcisi içinde servo motorların kontrolü için özel bir kütüphane mevcuttur. Bu sayede herhangi bir dijital pin kullanılarak servo motorların kontrol edilmesi mümkün olur. Şekil 4.4'te görüldüğü gibi kullanılan her servo motorun mikroişlemci kart ile arasında sinyal bağlantısı bulunmaktadır.

Palet motorlarının hareketinin kontrolü sürücü kart üzerindeki ek bağlantı noktaları sayesinde sadece 4 sinyal bağlantısı kullanılarak yapılmıştır. Bu kablolar ile hareket yönü ve hareket hızını belirlemek için kullanılan PWM sinyalleri mikroişlemciden sürücü karta iletilmiştir. Burada kullanılan bağlantı kabloları mikroişlemci üzerinde PWM özelliği olan çıkışlara bağlanabilir.

Ataletsel ölçüm cihazı mikroişlemci ile haberleşmek için I²C ya da SPI protokolünü kullanır. Yapılan çalışmada iletişim I²C protokolü üzerinden yapılmıştır. Bunun için kod kısmında özel bir kütüphane dosyası kullanılmış ve bu sayede sensör ile mikroişlemcinin kolayca haberleşmesi sağlanmıştır. Önceden de belirtildiği gibi Arduino Duemilanove üzerindeki bazı pinler kütüphane dosyalarında özel olarak tanımlanır ve bu dosyaların sağlayacağı ek özellikler için bu pinlerin kullanılması gerekir. I²C protokolünü kullanabilmek için de mikroişlemci üzerindeki 4 ve 5 numaralı analog girişler bağlantı şemasında gösterilen şekilde sensöre bağlanmıştır.

Geriye kalan sinyal bağlantıları ise ara kablo kullanmadan eklenti kartları üzerinden yapılmıştır. Bu kartlar Arduino Duemilanove mikroişlemci kartının konektörlerine uyumlu olduğundan gerekli bağlantının yapılması için eklenti kartının Arduino Duemilanove üzerine yerleştirilmesi yeterli olacaktır. Hem kumanda devresinde hem

de robot üzerinde kullanılan mikroşlemcilerde kablosuz haberleşme için XBee Shield isimli eklenti kartı kullanılmıştır. Bu kart mikroşlemcinin 0 ve 1 numaralı seri iletişim özelliği olan pinlerini kullanarak mikroşlemciye kablosuz haberleşme özelliği katar. Ayrıca kumanda devresinde kumanda işlemlerini yerine getirmek için mikroşlemci kart üzerine bir adet Joystick Shield bağlanmıştır. Joystick Shield üzerindeki her düğme ya da kol mikroşlemci üzerinde belli pinlere bağlıdır. Bu pinlere kodlama kısmında gerekli atamalar yapıldıktan sonra Joystick Shield üzerindeki ekipmanlar kullanılabilir.

4.3 Mikroşlemci Kodları

Yapılan çalışmada hem mobil robot üzerinde hem de kumanda devresinde birer adet mikroşlemci bulunmaktadır. Bu mikroşlemcilere ait kodlar Ek A.1 ve Ek A.2 bölümlerinde verilmiştir. Bu kısımda ise kullanılan kodların çalışma şekilleri anlatılacaktır. Böylece mikroşlemcilerin yapılan programlama sonucunda hangi işlemleri yerine getirdiği anlaşılabilecektir. Bu kısımda verilen açıklamalara ek olarak kodun içinde de birçok satırdan sonra açıklama eklenmiştir. Bu açıklamalarda da kodun çalışma şekli ayrıntılı olarak anlatılmıştır.

4.3.1 Mobil robot kodu

Mobil robot üzerinde bulunan mikroşlemci hem palet motorlarının hareketinin kontrolünden hem de kamera dengeleme sisteminin çalışmasından sorumludur. İki işlem kod olarak birbirinden bağımsızdır. Bu yüzden mikroşlemci kodunda ayrı şekilde yazılmışlardır. Genel olarak mobil robot üzerindeki mikroşlemcinin kodu üç farklı bölüm halinde ele alınacaktır.

İlk bölüm başlangıç kısmı olarak adlandırılmıştır. Burada kod içinde kullanılacak olan kütüphane dosyaları eklenmiştir. Ayrıca bu kısımda sistemde bulunan sabit ve değişkenlerin atamaları yapılmıştır. Değer atamaları ve değişken tanımlamaları bu bölüm dışında ayrıca fonksiyonlar içinde de yapılmıştır. Başlangıç kısmında tanımlanan değişken ve sabitler kodun genelinde hitap eder ve kodun farklı yerlerinde kullanılabilir. Fakat fonksiyon içinde yapılan tanımlamalar sadece o fonksiyon için geçerlidir. Ayrıca kod içinde tanımlanmadan kullanılan değişken ya da sabitler kodun derleyicisinin hata vermesine neden olur. Bunun sonucunda ise mikroşlemci üzerine kod iletilemez.

İkinci bölümde kodu basitleştirmek için kullanılan fonksiyonlar gösterilmiştir. Bu fonksiyonların tamamı ataletsel ölçüm cihazı ile ilgilidir. Buradaki fonksiyonların isimleri ve görevleri aşağıda verilmiştir.

- `initAcc()`: İvmeölçerin ölçüm parametrelerini sensöre iletir.
- `getAccelerometerData()`: İvmeölçerden her eksen için ivme değerlerini bulunduran veri paketlerini alır. Daha sonra bu paketleri uygun şekilde birleştirir.
- `rawAccToG()`: İvmeölçerden elde edilen verileri hassasiyet değerini kullanarak g cinsinden ivme büyüklüğüne çevirir.
- `normalize3DVec()`: İvmeölçerden elde edilen R vektörünün bileşenlerini normalize eder. Bunun için her ekseninde elde edilen vektör bileşenleri R vektörünün büyüklüğüne bölünür. Bu sayede vektörün büyüklüğünün etkisi giderilmiş olur.
- `initGyro()`: Jiroskobun ölçüm parametrelerini sensöre iletir.
- `getGyroscopeData()`: Jiroskoptan her eksen için açısal hız değerlerini ve sıcaklık değerini bulunduran veri paketlerini alır. Daha sonra bu paketleri uygun şekilde birleştirir.
- `rawGyroToDegSec()`: Jiroskoptan elde edilen verileri hassasiyet değerini kullanarak °/s cinsinden açısal hız değerine çevirir.
- `Zero()`: Jiroskopta sıfır değeri toleransından meydana gelen hataları giderir. Buna göre jiroskoptan her eksen için 100 kere ölçüm alınır. Daha sonra bu değerler her eksen için ayrı şekilde toplanıp ortalamaları alınır. Bu ortalama değerleri daha sonraki ölçümlerde düzeltme faktörü olarak kullanılır. Bunun için yapılan her ölçümde elde edilen veriler burada hesaplanan değerlerden çıkarılır.
- `kalmanCalculateX()`: X eksenini için Kalman filtresinden elde edilen Euler açısını hesaplar. Burada Kalman filtresi denklemleri fonksiyonun içine yerleştirilmiştir.
- `kalmanCalculateY()`: Y eksenini için Kalman filtresinden elde edilen Euler açısını hesaplar. Burada Kalman filtresi denklemleri fonksiyonun içine yerleştirilmiştir.

- writeTo(): I²C protokolünde haberleşilen cihaza ait bir yazmacın değerini değiştirmek için kullanılır. Bunun için I²C iletişim hattı üzerinden ilk olarak yazmacın adresi gönderilir daha sonra da yazmaca yazılacak değer cihaza iletilip haberleşme durdurulur.
- readFrom(): I²C protokolünde cihaz üzerinden okuma yapmak için kullanılır. Bunun için mikroişlemci ilk olarak I²C iletişim hattı üzerinden sensör verilerini bulunduran birinci yazmacın adresi cihaza iletir. Sonrasında ise mikroişlemci sensörden gönderilen adresteki yazmaçtaki ve belirlenen veri sayısına göre ardışık yazmaçlardaki sensör verilerini talep eder.

Mikroişlemci kodunun son kısmı ise kodun temelini oluşturan “setup” ve “loop” fonksiyonlarıdır. Burada önceki kısımlarda belirlenmiş değişkenler, sabitler ve fonksiyonlar kullanılarak sistemin istenen şekilde çalışması sağlanır.

Mikroişlemci başlangıç ayarlarının belirlendiği “setup” fonksiyonuna göre karta enerji verildiği anda ilk olarak seri iletişim ve I²C protokolü üzerinden iletişim başlatılacaktır. Sonrasında ise sırasıyla sensörlere kendi ölçüm parametreleri iletilmiş, jiroskop sıfır değer toleransındaki hatayı giderecek değişkenler hesaplanmış ve mikroişlemci pinlerinin yapılandırılmaları yapılmıştır. Kısacası “setup” kısmında mikroişlemci yapacağı çalışma için uygun şekilde düzenlenmiştir.

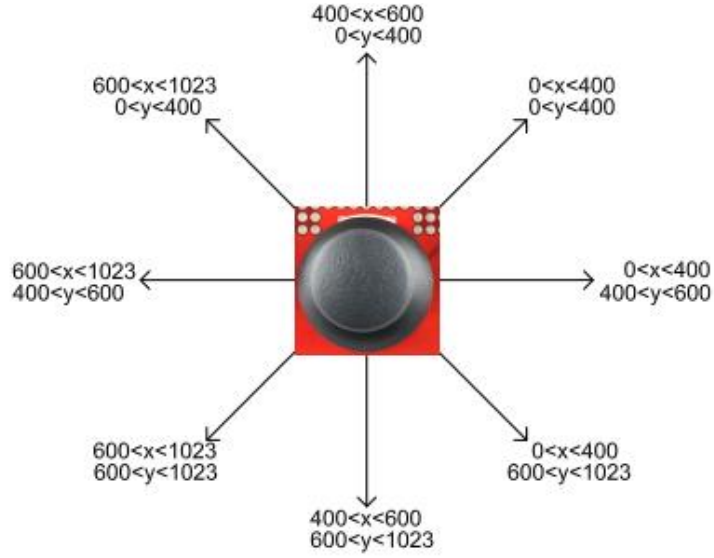
Kod kısmında mobil robotun hareket kontrolü ve kamera dengeleme sistemi birbirinden bağımsız olarak yazılmıştır. “loop” fonksiyonu içinde ilk olarak XBee modülleri aracılığıyla kumanda devresinden gönderilen veriler okunmuştur. Kumanda devresi hem palet motorlarının kontrolünde hem de kamera sapma açısının değiştirilmesinde kullanılmıştır. Bu yüzden buradan mikroişlemciye ulaşan veriler iki hareketten birisi ile ilgili bilgiyi bulundurur. Fakat hazırlanan kodda iki hareketin aynı anda gerçekleşmesini sağlayacak bir durum tanımlanmamıştır. Bu yüzden kameranin sapma açısı değiştirilirken robotun hareket etmesi mümkün değildir; zira kameranin sapma açısı sıfır derece olmadığı zaman kameranin dengeleme hareketi istenen şekilde yapılamaz. Döngü kodu içindeki ikinci kısım kamera dengeleme sisteminin kontrolü ile ilgilidir. Burada ilk olarak ivmeölçer ve jiroskoptan sensör verileri elde edilmiş ve sonrasında ise bu veriler işlenerek her eksen için ivme ve açısal hız değerleri hesaplanmıştır. İvmeölçerden elde edilen vektör bileşenleri Denklem (2.8) ve Denklem (2.9) kullanılarak gövdeye ait yönelimi veren Euler

açısına çevrilmiştir. Benzer şekilde Denklem (2.13) ve Denklem (2.14) kullanılarak jiroskoptan elde edilen açısal hız değerleri rotasyon açısındaki değişim değerlerine çevrilmiştir. Bu değerler daha sonra Kalman filtresi denklemlerinde kullanılarak gövdenin yeryüzüne göre yönelimini tanımlayan Euler açıları hesaplanmıştır. Burada her eksen için ayrı Kalman filtresi kullanılmıştır ve ataletsel ölçüm cihazı ters olarak yerleştirildiğinden X ekseninde gönderilen açı değeri de diğer eksene göre farklı yöndedir. Euler açıları elde edildikten sonra kameranın yunuslama ve yuvarlanma hareketlerini kontrol eden servo motorlara gerekli sinyaller iletilmiştir. Mikroişlemci içinde yapılan hesaplamalar sonucunda servo motorların çalışma sınırlarının dışına çıkan bir sinyalin gönderilmemesi için motorların dönebileceği limit değerleri tanımlanmıştır. Buna göre kod içinde iki motora da 180 dereceyi aşacak bir hareket sinyali gönderilemeyecek şekilde düzenleme yapılmıştır. “loop” kısmında son olarak zamanlayıcı hesaplaması yapılmıştır. Burada her döngü için kodun ne kadar sürede tamamlandığı hesaplanmış ve 20 ms’den kısa süren durumlarda mikroişlemci kod döngü süresi 20 ms olana kadar bekletilmiştir. Bu sayede yapılan hesaplarda sabit bir kod döngü süresi elde edilmiştir.

4.3.2 Kumanda devresi kodu

Kumanda devresi robot üzerindeki kameranın bakış yönünün değiştirilmesinde ve mobil robotun hareket ettirilmesinde kullanılır. Bunun için Joystick Shield üzerinde bulunan hareketli kol ve iki adet anlık elektrik düğmesi kullanılmıştır. Joystick Shield üzerindeki her ekipman altında bulunan mikroışlemcide ayrı bir pine bağlanmıştır. Eklenti kartının üzerinde bulunan hareketli kola bağlı potansiyometrelere ait gerilim değerleri mikroişlemci kartının üzerindeki 0 ve 1 numaralı analog girişlerden okunur. Servo motorları hareket ettirecek olan anlık elektrik düğmeleri ise mikroışlemcinin 3 ve 5 numaralı dijital pinlerine bağlıdır.

Buradaki hareketli kol iki ekseninde ileri ve geri şekilde hareket edebilmektedir. Ayrıca kolun içinde bulunan potansiyometreler kolun durumuna göre direnç değerlerini değiştirmektedir. Şekil 4.5’te kolun hareketi ile Arduino mikroişlemci kartının analog girişinden okunan değerler arasındaki ilişki gösterilmiştir. Burada verilen değerler 10 bit çözünürlüğe sahip bir analog-dijital dönüştürücünden alınmıştır. Bu yüzden iki eksen için de hareketli kol üzerinden elde edilen değerler 0 ile 1023 arasında değişmektedir.



Şekil 4.5 : Hareketli koldan farklı yönler için elde edilen veriler.

Kumanda devresinde kullanılan mikroişlemcide yukarıda belirtilen her durum için bir bayt tanımlanmıştır. Daha sonra bu baytlar kablosuz haberleşme ile robot üzerindeki mikroişlemciye iletilmiştir. Bu sayede palet motorlarının hareketi uzaktan kontrol edilmiştir.

Kumanda devresi üzerinde palet motorlarına ek olarak kamera bakış yönünü değiştiren servo motorlar da kontrol edilmiştir. Bunun için kumanda devresi üzerinde mikroişlemcinin 3 ve 6 numaralı pinlerine bağlanmış olan elektrik düğmeleri kullanılmıştır. Bu düğmelere basıldıkça kumanda devresi servo motorları hareket ettirecek veriyi kablosuz haberleşme üzerinden robota iletecektir.

Kumanda devresi üzerinden mobil robota gönderilen verilerde yığılma olmaması için veri iletimi saniyede 1 kez olabilecek şekilde düzenlenmiştir. Buna göre kumanda devresi üzerinden gönderilen kontrol sinyalleri robota her saniyede bir defa ulaştırılacaktır.

5. TESTLER VE SONUÇLAR

Kamera dengeleme sisteminde yapılan testler üç parça halinde ele alınacaktır. Bu kısımda ilk olarak sadece ivmeölçerden elde edilen ölçümlere göre elde edilen Euler açıları gösterilecektir. Daha sonra ise aynı testler Kalman filtresi kullanılarak yapılacak ve elde edilen sonuçlar karşılaştırılacaktır. Burada ayrıca Kalman filtresi denklemlerinde tanımlanan gürültü kovaryans değerlerinin çıktılar üzerindeki etkisi de incelenecektir. Yapılacak testlerde en son olarak kamera üzerinden elde edilen görüntüler incelenerek sistemin çalışması gösterilecektir.

Test sonuçları hakkında analiz yapılabilmesi için robota ait bazı performans ölçütleri belirlenmiştir. Robotun mekanik ve elektronik tasarımında burada belirtilen ölçütlere dikkat edilmiş ve buna göre gerekli düzenlemeler yapılmıştır. Robota ait performans ölçütleri ve kabuller aşağıda sıralanmıştır:

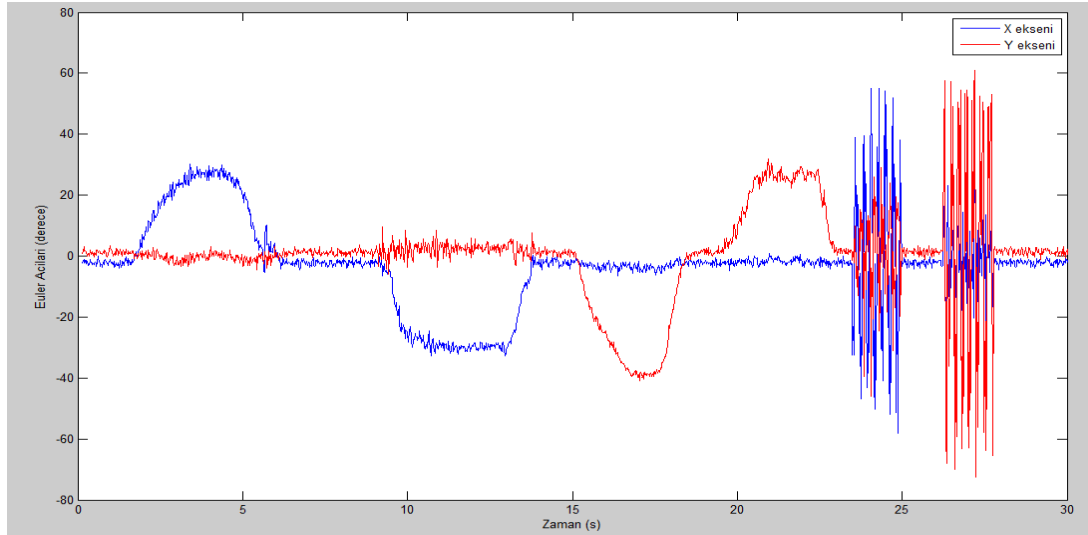
- Robotun ilerleyebileceği en büyük eğim değeri için arazi araçları için belirlenmiş değerler kullanılmıştır [41]. Buna göre robot en fazla 30 derecelik bir eğimde hareket edecek şekilde tasarlanmıştır. Sonuç olarak yunuslama ve yuvarlanma hareketini kontrol eden servo motorların en fazla ± 30 derecede hareket etmesi beklenmektedir.
- Yapılan uygulamada robotun yönelimindeki değişikliklere karşı servo motorların en fazla 2 saniye içinde dengeleme işlemini tamamlaması istenmektedir. Bu sayede kameradan elde edilen görüntüde meydana gelecek salınım hareketleri azaltılacaktır.
- Yapılan uygulamada Kalman filtresinin robot üzerinde meydana gelecek titreşim ya da ivme değişikliği durumlarına karşı Euler açılarını sabit tutması istenmektedir.
- Ataletsel ölçüm cihazında bulunan ivmeölçerin çalışma koşullarının uygulama boyunca değişmediği kabul edilmiştir. Bu yüzden yapılan çalışmada Kalman filtre denklemlerinde kullanılacak ölçüm gürültü kovaryansı sabit alınmıştır.

- Robotun hareketi sırasında meydana gelen titreşim hareketlerinin ve ivme değişikliklerinin belirli bir aralıkta olduğu varsayılmıştır. Bunun sonucunda Kalman filtrelerindeki işlem gürültü kovaryansı sabit alınmıştır.
- Yapılan çalışma prototip uygulaması olduğu için bataryaların ömrü hakkında bir ölçüt belirlenmemiştir.

Robot üzerindeki testler tamamlandıktan sonra elde edilen sonuçlara göre gelecekte yapılabilecek yeni çalışmalar açıklanacaktır.

5.1 İvmeölçer Sonuçları

Bölüm 2.1’de ivmeölçerden elde edilen ivme verileri kullanılarak gövdenin yönelimine ait Euler açılarının elde edilmesi gösterilmişti. Burada da Denklem (2.8)’de ve Denklem (2.9)’da gösterilen yöntem kullanılarak gövdenin yönelimini gösteren Euler açıları belirlenecektir. Bunun için gövdenin üzerindeki ataletsel ölçüm cihazı kullanılmış ve farklı eksenlerde yapılan rotasyonlar için sensörde bulunan ivmeölçerden elde edilen veriler incelenmiştir. Şekil 5.1’de ivmeölçerden elde edilen verilere göre hesaplanan Euler açılarının değişimi verilmiştir.



Şekil 5.1 : X ve Y eksenleri için ivmeölçerden elde edilen Euler açıları.

Burada yapılan testte ilk 15 saniyelik kısımda ivmeölçerin bağlı olduğu devre tahtası sırasıyla X eksenini etrafında $+30^\circ$ ve -30° kadar çevrilmiştir. Bu yüzden bu kısımda X eksenini için ivmeölçerden alınan veriler değişirken Y eksenini için alınan veriler sabit kalmıştır. Daha sonra ivmeölçer tekrar başlangıç konumuna getirilmiş ve aynı rotasyon hareketleri bu sefer Y eksenini etrafında yapılmıştır. En son olarak ivmeölçer

sırasıyla X ve Y eksenleri üzerinde de ileriye ve geriye doğru hızlıca hareket ettirilmiştir. Burada titreşim hareketlerinin ivmeölçer üzerindeki etkisi görülebilmektedir.

5.2 Kalman Filtresi Sonuçları

İvmeölçerden elde edilen Euler açılarının daha istikrarlı olması için hesaplamalarda Kalman filtresi denklemleri kullanılmıştır. Kalman filtresi denklemlerinin daha iyi anlaşılabilmesi için Denklem (2.26)'da ve Denklem (2.27)'de verilen durum ve ölçüm denklemleri tekrar gösterilmiştir.

$$x_k = \begin{bmatrix} a \\ b \end{bmatrix}_k = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} T \\ 0 \end{bmatrix} u_k + w_{k-1} \quad (5.1)$$

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \quad (5.2)$$

Yukarıdaki denklemlerde görüldüğü gibi kullanılan modele ait sistem durumları jiroskoptaki kayma değeri ve dönme eksenine ait Euler açısı olarak seçilmiştir. Ayrıca Denklem (5.2)'de verilen ölçüm denklemine göre kamera dengeleme sistemine monte edilen bir sensör ile kameraya ait Euler açıları elde edilmiştir. Sonuç olarak burada verilen iki denklem kullanılarak sisteme uygun bir Kalman filtresi tasarlanmış ve bu sayede her ölçüm adımı için düzeltilmiş Euler açıları ve jiroskop kayma değerleri tahmin edilmiştir.

Denklem (5.1)'de ve Denklem (5.2)'de gösterilen w ve v değerleri sırasıyla sisteme ve sensöre ait gürültü değerleridir. Bu değerler Kalman filtre denklemlerinde R ve Q kovaryanslarına sahip beyaz gürültüler olarak kabul edilir. Buna göre iki gürültü değerinin de büyüklükleri birbirinden bağımsızdır. Yapılan uygulamalarda burada belirtilen kovaryans değerleri değiştirilerek sistemin ve sensörün çalışma koşulları değiştirilebilir. Ayrıca yapılan çalışmaya göre Kalman filtre denklemlerindeki gürültü kovaryans değerleri sabit ya da değişken olarak alınabilir. R ve Q değerlerinin sabit alınması durumunda Kalman filtre denklemlerinde gösterilen Kalman kazancı ve hata kovaryansları her ölçüm adımında hızlıca azalır en sonunda sabitlenirler. Bu yüzden çalışma koşullarının çok değiştiği uygulamalarda gürültü kovaryans değerlerinin sabit seçilmesi Kalman filtresinin yanlış sonuçlar vermesine neden olur.

Kalman filtresi ile hesaplama yapılmadan önce başlangıç anındaki sistem durum tahminlerinin değerlerinin ve gürültü kovaryanslarının belirlenmesi gerekir. Bu değerler filtrenin başlangıç ve çalışma koşullarını belirtir.

Başlangıç anında ilk olarak sistemin durum tahminlerinin değerleri ve bu değerlere ait hata kovaryansları belirlenir. Bu değerler sayesinde Kalman filtresinin başlangıç koşulları düzenlenir. Yapılan uygulamada Euler açıları ile jiroskop kayma değerlerinin ilk ölçüm adımı için başlangıç değerleri belirtilmiştir. X ve Y eksenine ait Euler açılarının başlangıç değerleri ivmeölçerden yapılan ilk ölçümlere göre hesaplanmıştır. Ayrıca jiroskoptan elde edilen rotasyon açısı verilerinde başlangıç anında herhangi bir hata bulunmadığı varsayılmış ve başlangıç durumu için kayma değeri sıfır kabul edilmiştir. Durum tahminlerinin başlangıç değerleri seçildikten sonra bu değerlere ait hata kovaryansları belirlenmiştir. Bu çalışmada başlangıç anında robotun durmakta olduğu kabul edilmiştir. Bu yüzden ataletsel ölçüm cihazından elde edilen değerlerdeki hatanın az olduğu varsayılmıştır. Buna göre ilk ölçüm adımında Euler açısına ve jiroskobun kayma değerlerine ait durum tahminleri için küçük bir hata değeri seçilmiştir.

$$P_0^- = \begin{bmatrix} P_{00}^- & P_{01}^- \\ P_{10}^- & P_{11}^- \end{bmatrix}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.3)$$

Filtre denklemlerinin oluşturulabilmesi için belirlenmesi gereken ikinci parametre grubu gürültü kovaryans değerleridir. Kalman filtre denklemlerinde bulunan ölçüm ve işlem gürültü kovaryans değerleri filtrenin çalışma performansını belirler. Buna göre buradaki kovaryans değerleri değiştirilerek Kalman filtresinin çalışma hızı ve hassasiyeti ayarlanabilir. Önceden belirtildiği gibi yapılan uygulamada sensör için çalışma koşullarının değişmediği kabul edilmiştir. Ayrıca bu çalışmada sistem durumlarına ait standart sapma değerleri de sabit seçilmiştir. Bu yüzden filtre denklemlerinde kullanılan gürültü kovaryans değerleri her ölçüm adımı için sabit alınmıştır. Bu değerlerin Kalman filtresi sonuçları üzerindeki etkileri yapılan testlerde gösterilmiştir. Buradaki testlerde sistem için uygun gürültü kovaryansları belirlenirken bir kovaryans değeri sabit tutulup diğer kovaryans değeri değiştirilmiştir. Bu sayede iki kovaryans değerindeki değişimlerin elde edilen sonuçlar üzerindeki etkisi ayrı ayrı incelenebilmiştir. Yapılan testlerde her 20 ms’de bir uygulamada kullanılacak Kalman filtresi çıktıları ele alınmış ve bu değerlerin farklı durumlardaki değişimi gösterilmiştir.

5.2.1 Ölçüm gürültü kovaryansının Euler açıları üzerindeki etkisi

Ölçüm gürültü kovaryansı sensörden elde edilen verilerin ne kadar güvenilir olduğunu gösterir. Buna göre bu kovaryans değeri küçüldükçe tahmin değerlerinin güncellenmesinde daha çok sensör verisine güvenilmeye başlanır.

Bu kısımda yapılan testlerde işlem gürültü kovaryans değeri sabit tutulmuş ve sadece ölçüm gürültü kovaryansındaki değişikliklerin Kalman filtresi çıktıları üzerindeki etkileri incelenmiştir. Buna göre yapılan testlerde işlem gürültü kovaryansı Denklem (5.4)'teki gibi seçilmiştir.

$$Q = \begin{bmatrix} Q_a & Q_{ab} \\ Q_{ba} & Q_b \end{bmatrix} = \begin{bmatrix} 0,01 & 0 \\ 0 & 0,01 \end{bmatrix} \quad (5.4)$$

Ölçüm gürültü kovaryansına ait büyüklük değeri sensör üzerinden ölçüm yapılarak elde edilebilir. Bunun için sensörden elde edilen verilere ait standart sapma değeri belirlenmelidir. Kullanılacak ölçüm gürültü kovaryansı bu değerın karesine eşit olacaktır. Buna göre ölçüm gürültü kovaryansı Denklem (5.5)'teki gibi hesaplanabilir. Burada verilen σ_z değeri sensöre ait ölçüm verilerinin standart sapmasıdır.

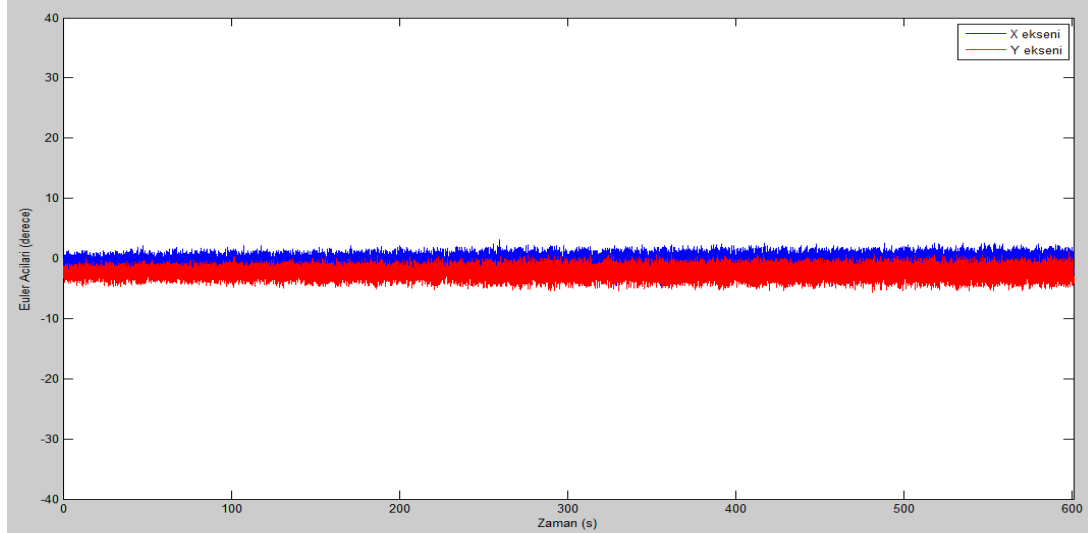
$$R = (\sigma_z^2) \quad (5.5)$$

Yapılan uygulamada sistem durum değerlerinin tahmin edilmesinde sadece ivmeölçere ait ölçüm verileri kullanılmıştır. Bu yüzden ölçüm gürültü kovaryansı içinde sadece bu sensöre ait standart sapma değerleri bulunur. Denklem (5.5)'te verilen standart sapma değerinin hesaplanması için ilk olarak ivmeölçerden hareketsiz durumda 1000 kere ölçüm alınmıştır. Filtre denklemlerinde sensör verileri Euler açısı olarak tanımlandığından elde edilen ivme değerleri Euler açısına çevrilmiştir. Daha sonra bu açı değerleri kullanılarak her eksen için açılara ait standart sapmalar hesaplanmıştır. Burada yapılan hesaplamada Denklem (5.6)'da gösterilen yöntem kullanılmıştır.

$$\sigma_k = \sqrt{\frac{1}{N} \sum_{i=1}^N (v_i - \mu)^2} \quad (5.6)$$

Yukarıdaki denklemde N sensörden yapılan toplam ölçüm sayısını, v_i Euler açılarının büyüklüğünü ve μ ise Euler açılarının ortalamasını temsil eder.

Şekil 5.2’de hareketsiz durumda ivmeölçerden 10 dakika boyunca elde edilen veriler ile hesaplanmış Euler açıları gösterilmiştir. Burada görüldüğü gibi hareketsiz durumda sensörden elde edilen veriler sabit bir standart sapmaya sahiptir. Bu yüzden sensöre ait ölçüm gürültü kovaryansı da sabit alınabilir.



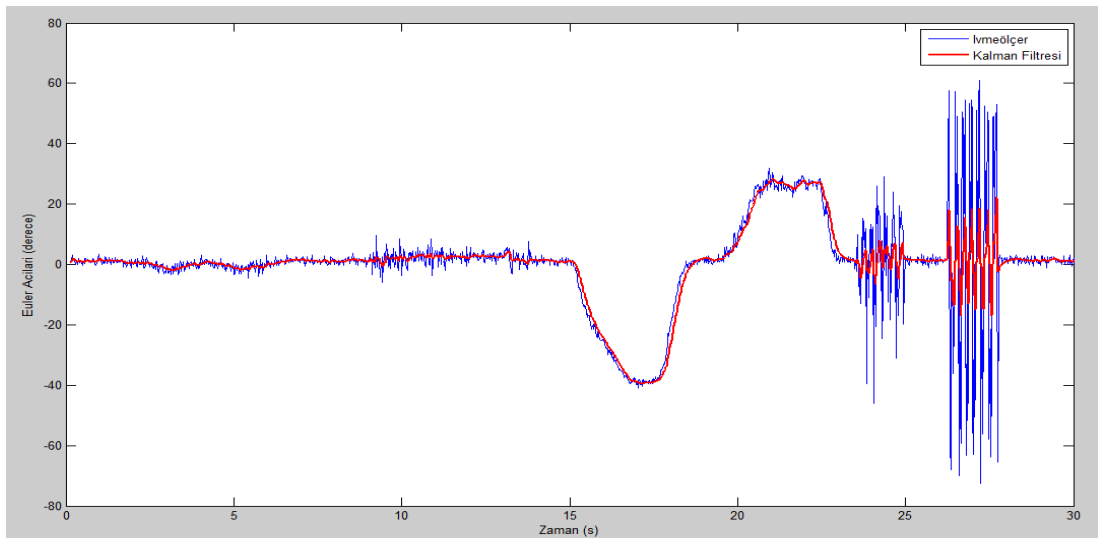
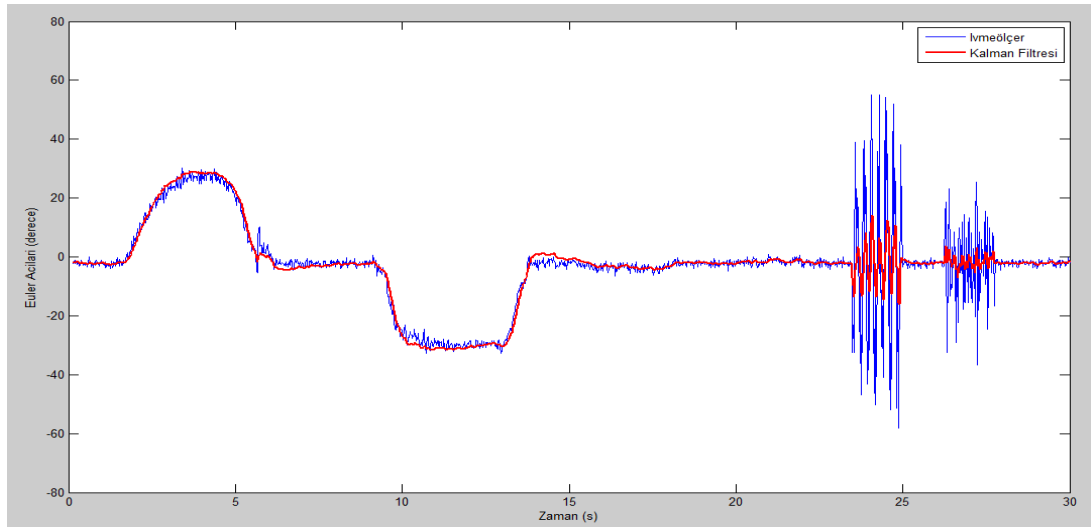
Şekil 5.2 : X ve Y eksenleri için ivmeölçerden hareketsiz durumda ölçülen veriler.

Denklem (5.6)’da verilen hesaplama kontrol edilen her eksen için 5 kere yapılmış ve daha sonra elde edilen değerlerin ortalaması alınarak standart sapma değerleri bulunmuştur. Sonuç olarak her eksen için bulunan standart sapma değerleri Çizelge 5.1’de verilmiştir. Çizelgede standart sapma değerlerinin hepsi derece cinsinden gösterilmiştir.

Çizelge 5.1 : Euler açılarına ait standart sapma değerleri.

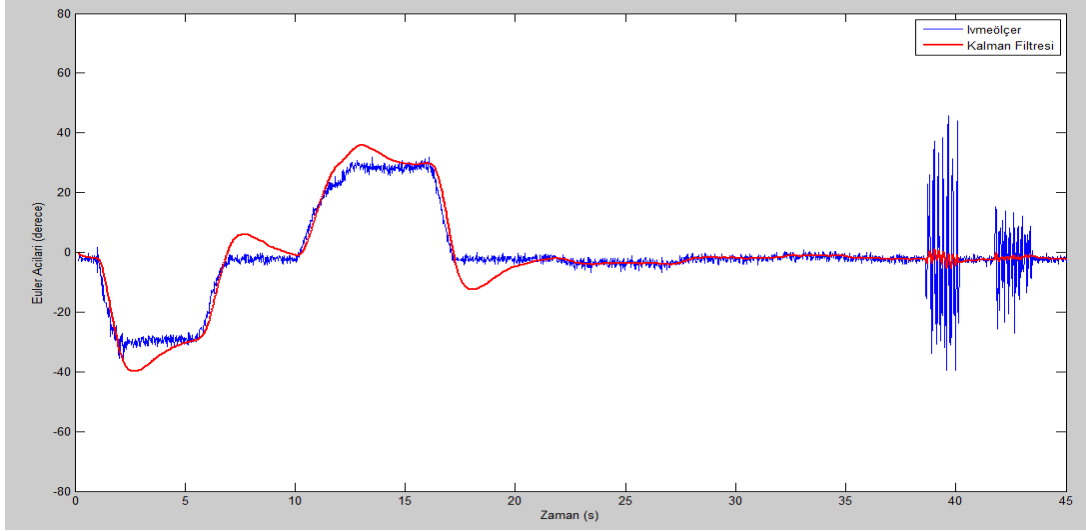
Hesaplama sırası	X eksen	Y eksen
1	0,6506°	0,6598°
2	0,6860°	0,6601°
3	0,6534°	0,6410°
4	0,6868°	0,6876°
5	0,6679°	0,6564°
Ortalama	0,6689°	0,6610°

Ölçüm gürültü kovaryansının hesaplanabilmesi için testlerden elde edilen değerlerin karesi alınmalıdır. Buna göre yapılan hesaplamalar sonucunda sensöre ait ölçüm gürültü kovaryans değerleri X eksen için 0,4461 ve Y eksen için 0,4369 olarak alınmıştır. Burada hesaplanan kovaryans değerleri kullanılarak filtreden elde edilen Euler açıları X eksen için Şekil 5.3’te ve Y eksen için ise Şekil 5.4’te verilmiştir.

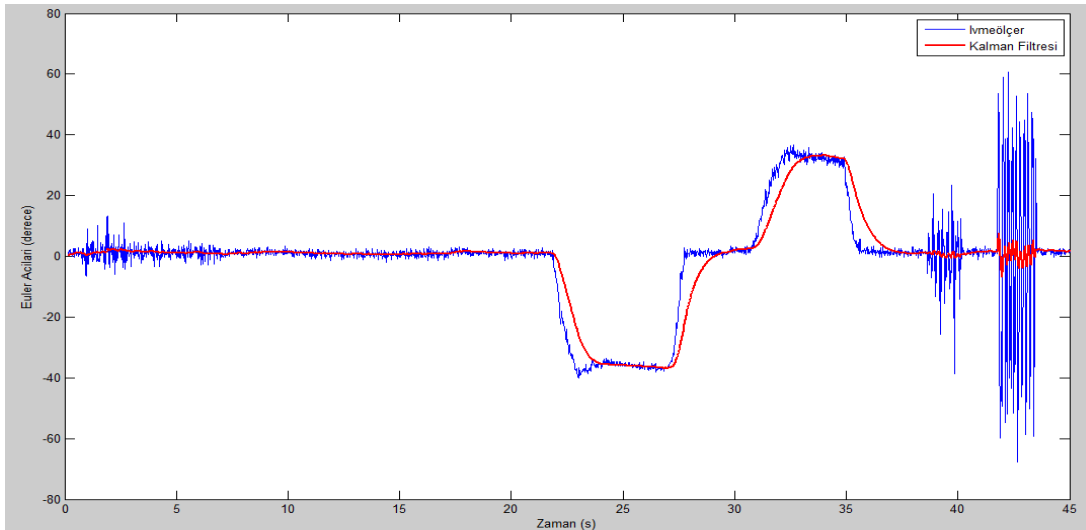


Yukarıdaki şekillerde bir önceki testte olduğu gibi ivmeölçere sırasıyla X ve Y eksenleri etrafında 30’ar derecelik rotasyon hareketleri yaptırılmış ve daha sonra da sensör yine sırasıyla X ve Y eksenleri üzerinde titreşime maruz bırakılmıştır. Şekil 5.3’te ve Şekil 5.4’te verilmiş olan grafiklerde Kalman filtresi kullanılarak elde edilen Euler açılarının ivmeölçerden elde edilen sonuçlara göre daha düzgün şekilde ilerlediği görülmektedir. Burada Euler açıları üzerindeki küçük salınımlar giderilmiş ve servo motorlara iletilen sinyalin daha sabit şekilde olması sağlanmıştır. Ayrıca sensörün iki eksen üzerinde maruz kaldığı titreşimlerin Euler açısı üzerindeki etkisi azaltılmıştır. Fakat Kalman filtresinden elde edilen Euler açılarında yapılan düzeltme yeterli değildir; zira titreşim durumlarında 20 dereceye yakın bir hata meydana gelmektedir. Bunun için işlem gürültü kovaryansının da düzenlenmesi gerekir.

Bu kısımda ayrıca ölçüm gürültü kovaryansındaki değişikliklerin sonuçlar üzerindeki etkisi gösterilecektir. Bunun için ilk olarak R değeri 10 alınmış ve ivmeölçer üzerinde yapılan testler tekrarlanmıştır. Bu testlere ait çıktılar Şekil 5.5'te ve Şekil 5.6'da verilmiştir.



Şekil 5.5 : X eksenini Kalman filtresi ve ivmeölçer sonuçları (R=10).

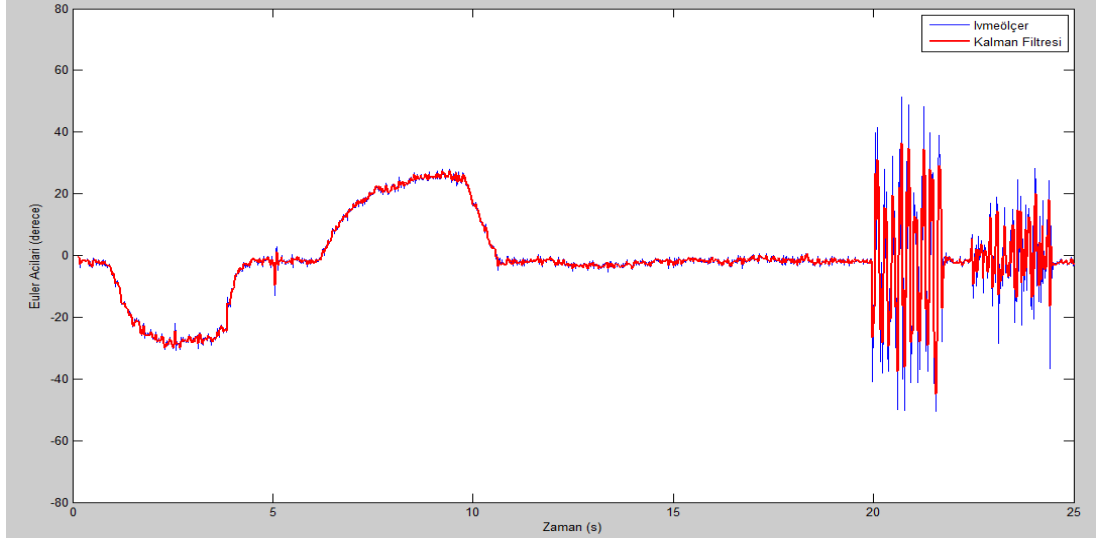


Şekil 5.6 : Y eksenini Kalman filtresi ve ivmeölçer sonuçları (R=10).

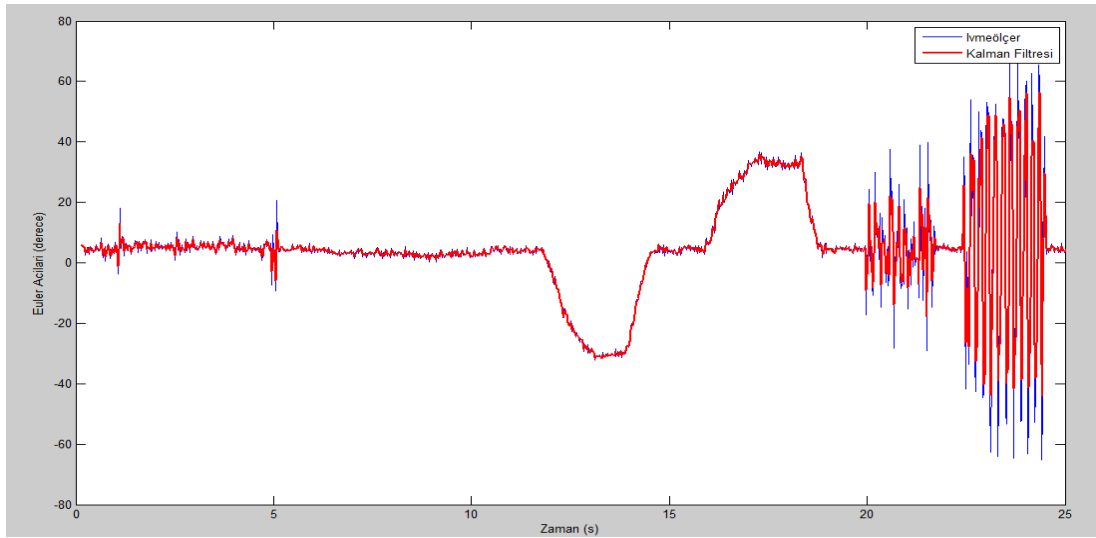
Şekil 5.5'te ve Şekil 5.6'da görüldüğü gibi işlem gürültü kovaryans değeri sabit iken R değerinin artırılması Kalman filtresinin daha yavaş çalışmasına neden olur. Bunun sonucunda filtreden elde edilen Euler açıları gerçek değerlerine daha geç yakınsar; fakat bu durumda gövdenin maruz kalacağı titreşim hareketlerinin Euler açıları üzerindeki etkisi azaltılır.

Son olarak ölçüm gürültü kovaryansının azaltılması sonucunda Kalman filtresi çıktılarına meydana gelen değişiklikler gösterilmiştir. Burada işlem gürültü

kovaryansı yine Denklem (5.4)'teki gibi alınmış ve R değeri 0,01'e düşürülmüştür. Şekil 5.7'de ve Şekil 5.8'de yapılan testlerin sonuçları görülmektedir.



Şekil 5.7 : X eksenli Kalman filtresi ve ivmeölçer sonuçları (R=0,01).



Şekil 5.8 : Y eksenli Kalman filtresi ve ivmeölçer sonuçları (R=0,01).

Yukarıdaki şekillerde görüldüğü gibi R değerinin azalması sonucunda Kalman filtresi çıktıları sensörden elde edilen verilere yakınsayacaktır. Bunun nedeni R değerinin azalması sonucunda Kalman filtresinin tahmin hesaplamalarında sensör verisini temel almasıdır. Buna göre R değeri azaldıkça Kalman kazancı Denklem (2.23)'teki gibi olacaktır ve bir sonraki ölçüm adımı için tahmin değeri hesaplanırken sadece sensör verisi kullanılacaktır.

Sonuç olarak R değeri her iki eksen için de ayrı şekilde belirlenmiş ve Kalman filtresinden elde edilen Euler açılarında istenen ayarlamalar yapılmıştır. Ayrıca diğer testlerde R değeri değiştirilerek filtrenin hassasiyeti ve çalışma hızı da ayarlanmıştır.

5.2.2 İşlem gürültü kovaryansının Euler açıları üzerindeki etkisi

İşlem gürültü kovaryansı Kalman filtre çıktıların sistem durumlarında meydana gelen değişimlere ayak uydurabilme yeteneği olarak tanımlanabilir. Bu değerin küçülmesi sonucunda Kalman filtresi sistem durumu tahmin değerlerini güncellerken bir önceki ölçümden gelen tahmin değerlerini temel alır. Böylece sistem çalışmasını kararlılığa zorlar ve filtrelenmiş değerler, gerçek durum değerlerinde meydana gelen değişiklikleri iyi takip edemez; fakat bununla birlikte filtreden elde edilen değerlerde üstün bir stabilizasyon sağlanır. İşlem gürültü kovaryansı büyük alındığında ise sistem çalışmasını tam olarak sabitlemediği için gerçek durum değişkenlerindeki değişiklikler yakından takip edilir, fakat bu durum da stabilizasyon performansı düşer. Sağlıklı bir düzgünleştirme için işlem gürültü kovaryansı tipik olarak 0 ile 1 arasında tanımlanmalıdır [42]. Burada belirtilen işlem gürültü kovaryansı Denklem (5.7)'deki gibi yazılabilir.

$$Q = E[w_k w_k^T] \quad (5.7)$$

Yukarıdaki denklemde gösterilen w_k değeri Denklem (5.1)'de verilmiş işlem gürültüsüdür. Yapılan uygulamada sistem durumları olarak jiroskobun kayma değeri ve dönme eksenine ait Euler açısı seçilmiştir. Burada iki sistem durumu olduğundan işlem gürültü kovaryansı matris yapısında yazılır.

$$Q = \begin{bmatrix} Q_a & Q_{ab} \\ Q_{ba} & Q_b \end{bmatrix} \quad (5.8)$$

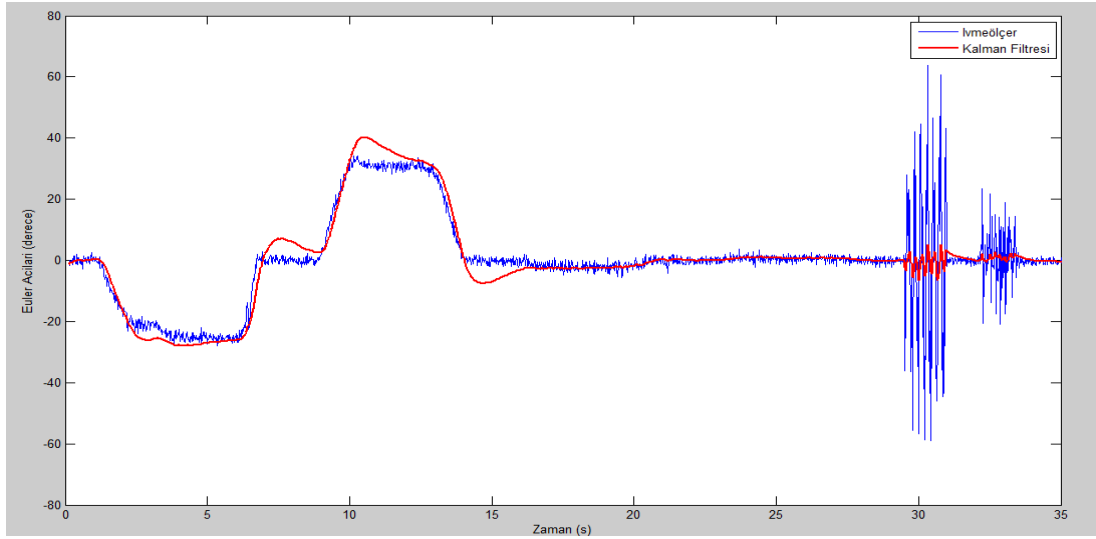
Denklem (5.8)'de verilen Q_a ve Q_b değerleri sistem durumlarına ait standart sapma değerlerinin karesidir. Q_{ba} ve Q_{ab} değerleri ise iki durumun birbirine göre standart sapmalarını verir. Burada iki duruma ait gürültü değeri birbirinden bağımsız olduğu için Q_{ab} ve Q_{ba} değerleri sıfır alınmıştır. Ayrıca çalışma boyunca jiroskoptaki kayma değerinin ve Euler açılarının standart sapma değerleri sabit kaldığı kabul edilmiştir.

$$Q = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix} \quad (5.9)$$

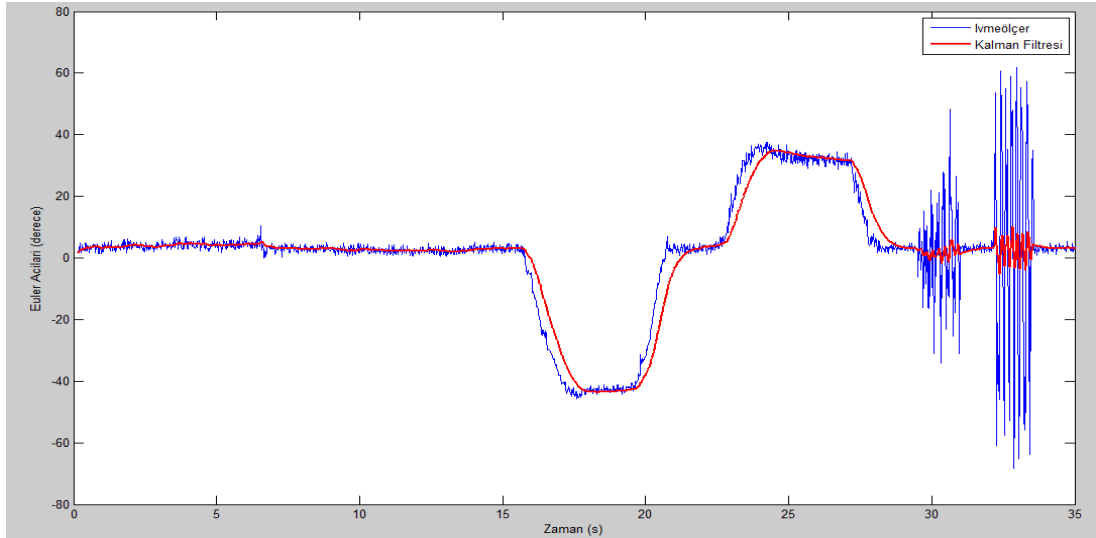
Kalman filtre denklemlerinde hesaplanan sistem durum tahminleri ölçülemez büyüklüklerdir. Bu yüzden işlem gürültü kovaryansının değerleri ölçüm gürültü kovaryansındaki gibi sensör üzerinden ölçüm yapılarak belirlenemez. Bunun için

işlem gürültü kovaryans değerlerinin hesaplanmasında genellikle farklı tahmin yöntemleri kullanılır [43]. Bu sayede işlem gürültü kovaryansına ait değerler yaklaşık olarak belirlenebilir.

Yapılan testlerde R değeri sabit alınmış ve işlem gürültü kovaryansı değiştirilmiştir. Bunun sonucunda da işlem gürültü kovaryansının Kalman filtresi sonuçları üzerindeki etkisi gösterilmiştir. Şekil 5.9'da ve Şekil 5.10'da Q_a ve Q_b değerleri 0,001 olarak alınmıştır.



Şekil 5.9 : X eksenli Kalman filtresi ve ivmeölçer sonuçları (Q_a ve $Q_b=0,001$).

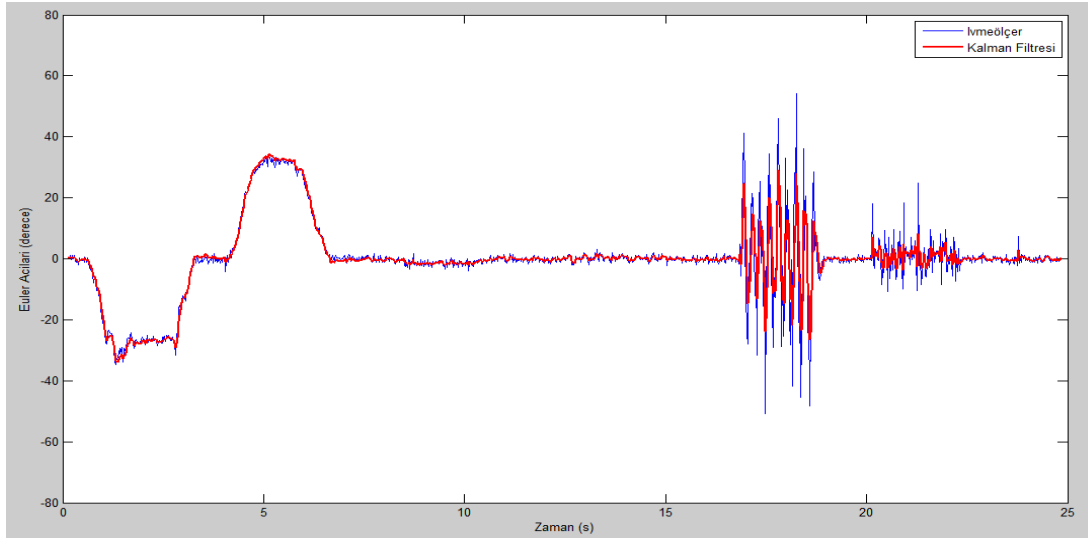


Şekil 5.10 : Y eksenli Kalman filtresi ve ivmeölçer sonuçları (Q_a ve $Q_b=0,001$).

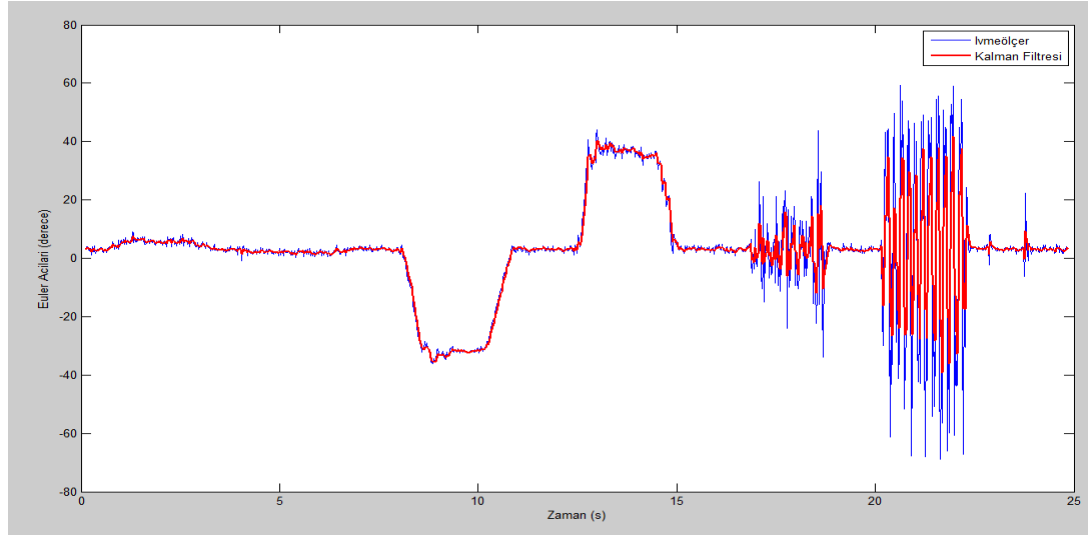
Yukarıdaki grafikler ile Şekil 5.3'te ve Şekil 5.4'te verilen grafikler karşılaştırdığında sabit R değerleri için işlem gürültü kovaryansı azaltıldığında Kalman filtresi çıktılarının daha kararlı şekilde ilerlediği görülmektedir. Burada hem filtrenin çalışma hızı istenen aralıkta tutulmuş hem de meydana gelen titreşimlerin

sonular zerindeki etkisi azaltılmıřtır. Buna gre ivmelerin eksen zerinde hızlı bir řekilde ileriye ve geriye doėru hareket ettirilmesi sonucunda elde edilen verilerde ± 30 – 40 derecelik hatalar meydana gelirken Kalman filtresinden elde edilen Euler aısı sadece ± 5 derecelik bir deėiřim gstermiřtir.

Son olarak iřlem grlt kovaryansının arttırılması durumunda Kalman filtresine ait sonulardaki deėiřimler incelenmiřtir. Bunun iin Q_a ve Q_b deėerleri 0,1 olarak alınmıřtır. Bu testlere ait grafikler řekil 5.11’de ve řekil 5.12’de verilmiřtir.



řekil 5.11 : X eksenı Kalman filtresi ve ivmeler sonuları (Q_a ve $Q_b=0,1$).



řekil 5.12 : Y eksenı Kalman filtresi ve ivmeler sonuları (Q_a ve $Q_b=0,1$).

Yukarıdaki řekillerde verilen grafiklerde Kalman filtresi ıktıları beklendiėi gibi sensr verilerini takip etmeye bařlamıřtır. nceden de belirtildiėi gibi Q matrisinin iindeki deėerler arttıca sistem kararlı olmaktan uzaklařır ve gerek sistem durumlarını takip etmeye bařlar.

5.2.3 Kalman filtresi sonuçlarının analizi

Yukarıda yapılan testler sonucunda Kalman filtresine ait en uygun gürültü kovaryans değerleri belirlenmiştir. Buna göre uygulamada ölçüm gürültü kovaryansı olarak X eksenini için $R_x = 0,4461$ ve Y eksenini için $R_y=0,4369$ kullanılmıştır. Ayrıca yapılan testler sonunda elde edilen en uygun kullanılan işlem gürültü kovaryans değerleri aşağıda verilmiştir.

$$Q_x = \begin{bmatrix} 0,001 & 0 \\ 0 & 0,001 \end{bmatrix} \quad (5.10)$$

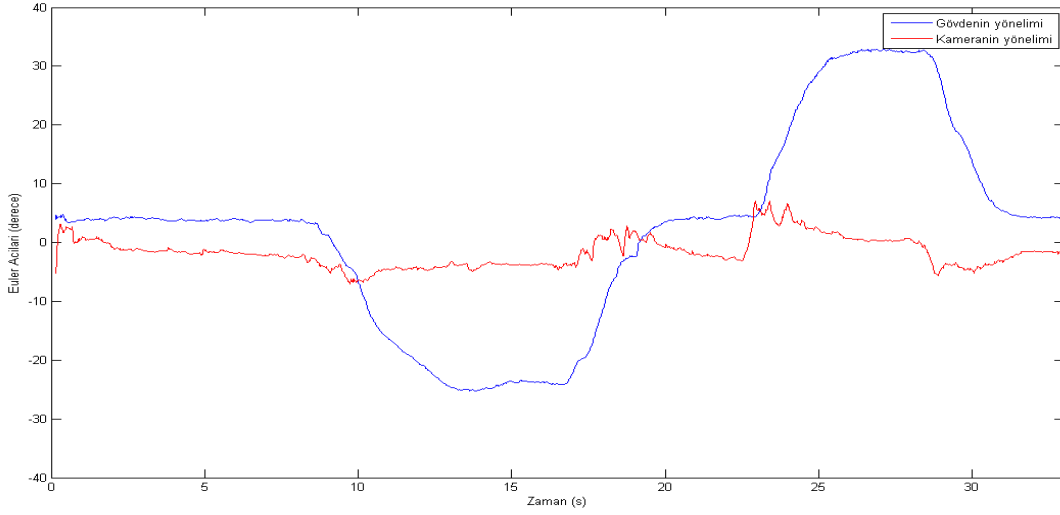
$$Q_y = \begin{bmatrix} 0,001 & 0 \\ 0 & 0,001 \end{bmatrix} \quad (5.11)$$

Bu kısımda elde edilen sonuçların önceden belirtilen performans ölçütlerine uygunluğu test edilecektir. Kalman filtresinin performansını test etmek için iki ölçüt belirlenmiştir. Bunlardan ilki sistemin tepki hızı, ikincisi ise titreşim ve ivme değişikliklerinin Euler açıları üzerindeki etkisinin giderilmesidir.

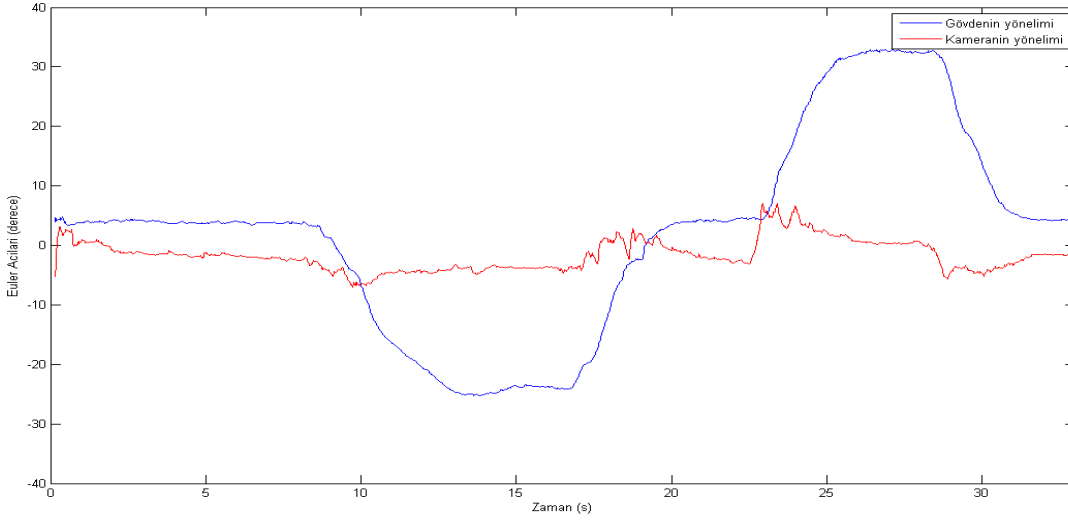
5.2.3.1 Tepki hızının incelenmesi

Performans ölçütlerinde kamera dengeleme sistemindeki servo motorların dengeleme işlemini 2 saniye içinde tamamlaması gerektiği belirtilmiştir. Burada amaçlanan gövdenin yönelimindeki değişikliklerin kamera görüntüsü üzerindeki etkisini azaltmaktır. Bunun için de sistemin robotun hareketi sırasında meydana gelen yönelim değişikliklerine karşı tepki süresinin olabildiğince kısa tutulmasıdır. Tepki süresinin uzaması durumunda kameranın kendini dengelemesi daha uzun sürede gerçekleşecek ve görsel verideki bozulmalar artacaktır.

Mikroişlemci üzerindeki kod döngüsünde Kalman filtresinden elde edilen Euler açıları 20 ms'lik aralıklar ile servo motorlara iletilmiştir. Ayrıca Çizelge 3.2'de verildiği gibi seçilen servo motorlar 180 derecelik hareketi 1,5 saniyede gerçekleştirebilmektedir. Buradaki mikroişlemci kodunun ve servo motorların çalışma hızları ele alınarak sistemin değişikliklere karşı 2 saniyede kendini dengelemesi istenmiştir. Bu işlemin test edilebilmesi için kamera üzerine ayrı bir ataletsel ölçüm cihazı yerleştirilmiş ve kameranın yönelimindeki değişiklikler gövdedeki değişiklikler ile karşılaştırılmıştır. Yapılan testin sonuçları Şekil 5.13'te ve Şekil 5.14'te gösterilmiştir.



Şekil 5.13 : Kamera ve gövde yöneliminin karşılaştırılması (X eksen).



Şekil 5.14 : Kamera ve gövde yöneliminin karşılaştırılması (Y eksen).

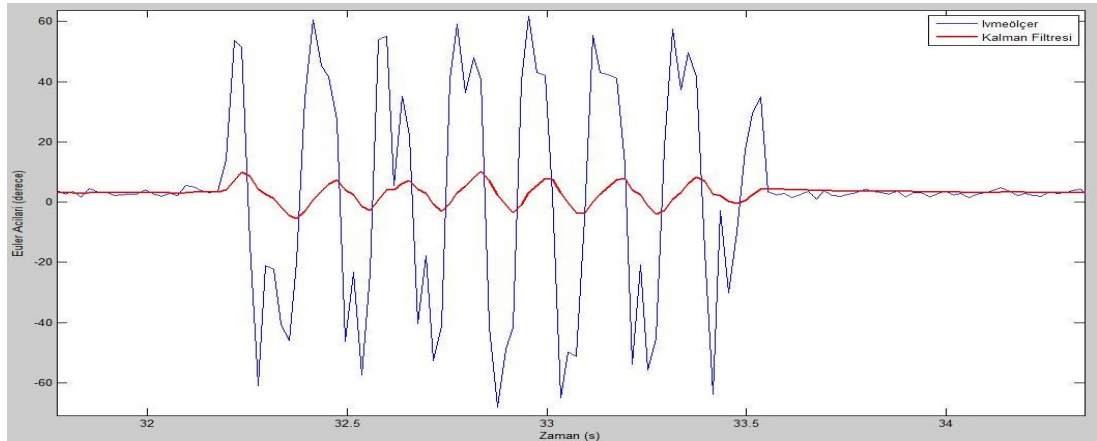
Yukarıdaki şekillerden de görülebileceği gibi yapılan testlerde robotun gövdesi X ve Y eksenleri üzerinde $\pm 30^\circ$ kadar hareket ettirilmiş ve bu hareketler sonucunda iki ataletsel ölçüm cihazından elde edilen veriler karşılaştırılmıştır. Burada hareketsiz durumda sensörlerden elde edilen değerler arasındaki farklar sensörlerin devre tahtası üzerine yerleştirme şekillerindeki farklılıklardan kaynaklanmaktadır. Yapılan testlerde hareket sırasındaki yönelim değişikliklerinin bu değerlerde meydana getirdiği bozulmalar incelenmiştir. Kamera dengeleme sisteminin ideal şekilde çalışması durumunda grafiklerde kırmızı ile gösterilen çizgiler hareket sırasında meydana gelen değişikliklerden etkilenmeden düz olarak ilerleyecektir. Buna göre ideal durumda hareket esnasında robotun gövdesinin yönelimi değişse bile kameranın yönelimi sabit kalacaktır. Fakat burada görülebildiği gibi iki eksen için de yapılan rotasyon hareketleri kameranın yöneliminde bozulmalara yol açmıştır.

Şekil 5.13'ten ve Şekil 5.14'ten elde edilen bilgilerden yola çıkılarak kamera dengeleme sisteminin tepki hızı anlaşılabilir. Yapılan testler sırasında robotun gövdesine başlangıç konumundan itibaren yaklaşık 5 saniye boyunca rotasyon hareketleri yaptırılmıştır. Önceden belirtildiği gibi testlerde sınır değeri olarak $\pm 30^\circ$ alındığından burada yapılan hareketlerin hızının 6 °/s olduğu kabul edilebilir. Elde edilen sonuçlarda kamera dengeleme sistemi aktifken bu hızda bir hareketin kameranın yöneliminde meydana getirdiği bozulmalar görülmektedir. Her rotasyon hareketi sırasında kameranın yöneliminde yaklaşık 5 derecelik bozulmalar meydana gelmiştir. Fakat rotasyon hareketleri sonlandığında dengeleme sistemi kamerayı hızlıca eski konumuna getirebilmiştir. Burada yapılan testlerde rotasyon hareketlerinin hızının artırılması sonucunda kamera görüntüsünde meydana gelen bozulmalar artacaktır.

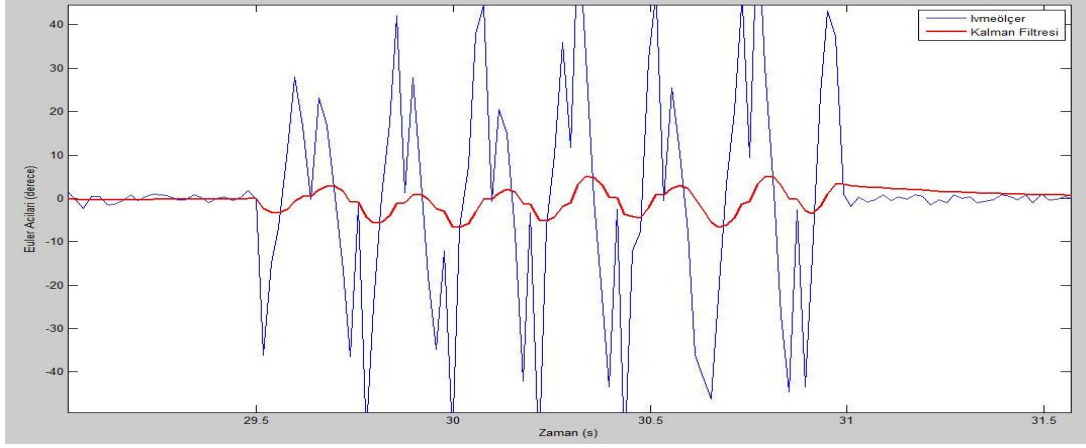
Kameranın yöneliminde meydana gelen bozulmaların azaltılması için servo motorların ve Kalman filtresinin çalışma hızının artırılması gerekir.

5.2.3.2 Titreşimin etkilerinin incelenmesi

Kalman filtresinin sisteme uygunluğunu test ederken kullanılacak ikinci ölçüt titreşim etkilerinin giderilmesidir. Şekil 5.1'de ivmeölçerin bağlı olduğu gövdede meydana gelen titreşimlerin sensör çıktıları üzerindeki etkisinin oldukça fazla olduğu gözlemlenmiştir. Kalman filtresinin en önemli görevlerinden biri bu etkinin belirli bir sınıra kadar azaltılmasıdır. Bunun için Kalman filtresi çıktıları üzerinde yapılan testlerde robotun gövdesi hızlı şekilde farklı yönlerde hareket ettirilmiş ve bunu sonucunda Euler açısında meydana gelen değişiklikler incelenmiştir. Sonuç olarak iki durum için elde edilen Euler açıları Şekil 5.15'te ve Şekil 5.16'da gösterilmiştir.



Şekil 5.15 : Titreşim sırasında X eksenine ait Euler açılarının değişimi.



Şekil 5.16 : Titreşim sırasında Y eksenine ait Euler açılarının değişimi.

Yukarıdaki şekillerden görüldüğü gibi titreşim sırasında ivmeölçerden elde edilen veriler $\pm 50^\circ$ arasında değişmektedir. Kalman filtresinin kullanılması sonucunda buradaki hata değerleri $\pm 5^\circ$ mertebelerine indirilmiştir.

5.3 Kamera Görüntüleri

Bu kısımda yapılan testlerde robot üzerindeki kameradan alınan görüntülerin kamera dengeleme sistemi tarafından düzeltilmesi gösterilmiştir. Buna göre ilk olarak robot kamera dengeleme sistemi aktif değilken farklı yönlerde çevrilmiş ve kamera görüntüleri alınmıştır. Daha sonra ise kamera dengeleme sistemi çalıştırılarak aynı hareketler tekrarlanmıştır. Böylelikle iki durum için de kameradan elde edilen görüntüler karşılaştırılmıştır. Yapılan testlerde robotun yaptığı rotasyon hareketlerine karşılık gelen Euler açıları mikroişlemci kart ile seri iletişim yapılarak bilgisayar üzerinden okunmuştur.

Kamera görüntüleri alınırken robota X ekseninde ± 30 derecelik ve Y ekseninde ± 15 derecelik rotasyon hareketleri yaptırılmıştır. Y ekseninde yapılan hareketlerin daha az olmasının nedeni 15 dereceden fazla olan rotasyonlarda kameradan elde edilen görüntünün başlangıç durumu ile karşılaştırılamamasıdır. Burada sol kısımda verilen kamera görüntüleri dengeleme sistemi aktifken, sağ kısımda bulunan görüntüler ise kamera dengeleme sistemi devrede değilken alınmıştır. Burada görüldüğü gibi farklı yönelim durumlarına rağmen kamera dengeleme sistemi kamera görüntüsünü sabit tutabilmiştir.

Robottan farklı yönelim durumları için elde edilen kamera görüntüleri Şekil 5.17’de gösterilmiştir.

Kamera Dengeleme Sistemi Açık

Kamera Dengeleme Sistemi Kapalı



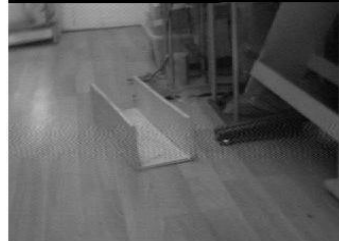
$X = 0, Y=0$



$X = 30, Y=0$



$X = -30, Y=0$



$X = 0, Y=15$



$X=0, Y=-15$



$X=-30, Y=15$

Şekil 5.17 : Robot üzerinden elde edilen kamera görüntüleri.

5.4 Gelecek Çalışmalar

Bu kısımda elde edilen sonuçların analizi ve kamera dengeleme sistemi üzerinde yapılabilecek iyileştirme çalışmaları açıklanacaktır.

Yapılan çalışmada robotun üzerinde bulunan kamera başarıyla dengelenmiştir. Burada yapılan çalışmanın sonucunda kameradan elde edilen veriler Şekil 5.17’de verilmiştir. Aynı zamanda Kalman filtresi algoritmasında yapılan düzenlemelerle servo motorların yönelim değişikliklerine istenen hızda cevap verebilmesi sağlanmıştır. Ayrıca yapılan elde edilen testlerde farklı çalışma koşulları için Kalman filtresinin çalışma hızı ve hassasiyeti düzenlenebilmiştir. Bu sayede robotun farklı koşullarda çalışabilmesi sağlanmıştır.

Mobil robot üzerindeki kameranın yönelimi sadece servo motorlar ile değiştirilmiş ve kameradan elde edilen görüntüler üzerinde herhangi bir görüntü işleme yapılmamıştır; zira bu uygulamada kamera görüntüsünün görüntü işleme yöntemi ile dengelenmesi için gereken görsel veri yetersizdir. Bu tip bir uygulamada düzeltme işlemi yapıldıktan sonra elde edilen görüntüde veri kayıpları meydana gelecektir ve ilk durumdaki görüntünün elde edilmesi mümkün olmayacaktır. Kamera gövdesinin hareket ettirerek düzeltme yapılan uygulamalarda ise işlem kameradan elde edilen görüntüden bağımsızdır. Bu yüzden bu tür uygulamalarda tek bir kameranın kullanılması yeterli olacaktır. Şekil 5.17’de bu yöntem kullanılarak yapılan bir dengeleme işleminin sonuçları verilmiştir.

Bu çalışmadan elde edilen sonuçlara göre benzer sensör ve filtre algoritmaları kullanılarak yeni platformlar için dengeleme sistemleri geliştirilebilir. Benzer şekilde ataletsel ölçüm cihazından elde edilen verilerde yapılan hesaplamalar ile çeşitli gövdelerin yönelimi belirlenebilir ve bu gövdelerin çeşitli çalışmalar sırasında dengelenmesi sağlanabilir. Bunlara örnek olarak insansız hava araçlarında kameranın dengelenmesi, insansı robotlarda herhangi robotun bir parçasının yöneliminin belirlenmesi ve tıpta kullanılan kamera robotlarının insan vücudu içerisinde ilerlerken kameranın dengelenmesi verilebilir. Benzer şekilde sistemde mevcut olan sensörler kullanılarak çalışma sırasında gövdenin yöneliminde meydana gelen değişimlerden kaydedilebilir ve bu sayede robotun ilerlediği yüzeyin topografyası çıkarılabilir. Bunun sonucunda kullanıcıya robotun çalışma alanı hakkında ek bilgiler iletilebilir.

Yapılmış olan uygulamada elde edilen sonuçlardan yola çıkılarak dengeleme sistemine ve mobil robota ait çeşitli eksiklikler tespit edilmiştir. Burada tespit edilen eksikliklerden ve bunlara ait çözüm önerilerinden kısaca bahsedilecektir.

Hazırlanan sistemde dengeleme işlemi sadece gövdenin yönelimindeki değişiklikler temel alınarak gerçekleştirilmiştir ve burada kullanılan kontrol algoritması açık çevrim yapısındadır. Yapılan testlerde kamera muhafaza kutusunun üzerine yerleştirilen ek ataletsel ölçüm cihazı farklı bir mikroişlemciye bağlanmış ve bu sensör üzerinden kameranın yönelimine ait veriler ölçülmüştür. Fakat bu sensör farklı bir mikroişlemciyle haberleştirilmiş ve bu mikroişlemci üzerinden sistemin çalışma performansı incelenmiştir. Buna göre kullanılan sistemde kameranın yönelimine ait veriler sisteminin kontrol algoritmasında kullanılmamıştır. Bu yüzden kontrol algoritmasında kameranın yönelimine ait bir geri besleme mevcut değildir. Kamera dengeleme sisteminin çalışmasında sistemin daha hassas kontrol edilebilmesi kameranın üzerine yerleştirilen sensörün de kontrol algoritmasına eklenmesi gerekir. Bu sayede dengeleme işlemi elde edildikten sonra görüntüde meydana gelebilecek kalıcı hatalar giderilir. Bu uygulamada bu yöntemin kullanılmama nedeni ise hareketli dengeleme sistemi üzerinde meydana gelen kablolama sorunlarıdır. Önceden de bahsedildiği gibi kamera dengeleme sisteminden mikroişlemci üzerine giden kablo sayısının artmasıyla buradaki hareketli parçaların kabloları zarar verme riski de artacaktır. Bunun için yapılan tasarımda servo motorların bulunduğu kısımda olabildiğince az kablo bulundurulmaya çalışılmıştır.

Yapılan uygulamada servo motorların kameranın yönelimini başarılı şekilde sabit tutabildiği görülmektedir. Fakat bu dengeleme işlemi sonucunda kullanıcının sadece kamera görüntülerine bakarak robotu hareket ettirmesi istenmeyen durumlara neden olabilir, zira kullanıcının robotu düzgün şekilde hareket ettirebilmesi için gövdenin yönelimi hakkında bilgi sahibi olması gerekir. Bu yüzden gelecek çalışmalarda gövdenin yönelimine ait bilgilerin de kullanıcı tarafına iletilmesi gerekir.

Kalman filtresi uygulamasında gürültü kovaryanslarının değerleri sabit seçilmiştir. Bunun için de sistem durumlarının standart sapma değerlerinin sabit olduğu ve sensör çalışma koşullarının değişmediği kabul edilmiştir. Fakat bu değerlerin belli bir kontrol algoritması ile değişken halde getirilmesi sonucunda sistemin farklı koşullarda istenen şekilde çalışması sağlanabilir. Ayrıca bu uygulamada gürültü kovaryans değerlerinin belirlenmesi için filtre çıktılarından yararlanılmıştır. Buradaki

hesaplamalarda literatürde bulunan tahmin yöntemleri kullanılarak gürültü kovaryansları gerçek değerlere daha yakın şekilde belirlenebilir [43]. Bu sayede Kalman filtresinin çalışma performansı arttırılabilir. Bunun sonucunda dengeleme sisteminin daha küçük aralıktaki rotasyon hareketlerine karşı tepki vermesi ve görüntüde titreşimden meydana gelen bozulmaların azaltılması sağlanabilir.

KAYNAKLAR

- [1] **Zhao, J., Liu, G., Liu, Y. ve Zhu, Y.** (2008). Research on the application of a marsupial robot for coal mine rescue, *Lecture Notes in Computer Science*, Cilt **5315**, Sf. 1127–1136.
- [2] **Miyasaka, K. ve Hatano, M.** (2008). Research on rescue robots with force sensors on the fingertips for rubble withdrawal works, *Proceedings of the SICE Annual Conference*, Sf. 1947–1950.
- [3] **Sahashi, T., Sahashi, A., Uchiyama, H. ve Fukumoto, I.** (2011). A study of operational liability of the medical rescue robot under disaster, *2011 IEEE/SICE International Symposium on System Integration*, Sf. 1281–1286.
- [4] **Davids, A.** (2002). Urban search and rescue robots: From tragedy to technology, *IEEE Intelligent Systems and Their Applications*, Cilt **17**, Sayı. 2, Sf. 81–83.
- [5] **Siciliano, B. ve Khatib, O.** (2008). *Springer handbook of robotics*, Springer-Verlag, Berlin.
- [6] **Sun, H., Yang, P., Liu, Z., Zu, L. ve Xu, Q.** (2011). Microphone array based auditory localization for rescue robot, *Proceedings of the 2011 Chinese Control and Decision Conference*, Sf. 606–609.
- [7] **Ko, A. W. Y. ve Lau, H. Y. K.** (2009). Intelligent robot-assisted humanitarian search and rescue system, *International Journal of Advanced Robotic Systems*, Cilt **6**, Sayı. 2, Sf. 121–128.
- [8] **Stopforth, R.** (2010). *Contractible arms elevating search and rescue (CAESAR) robot – Improvements and modifications for urban search and rescue (USAR) robots*, (doktora tezi), University of KwaZulu Natal, Durban.
- [9] **Chiu, Y., Shiroma, N., Igarashi, H., Sato, N., Inami, M. ve Matsuno, F.** (2005). FUMA: Environment information gathering wheeled rescue robot with one-DOF arm, *Proceedings of the 2005 IEEE International Workshop on Safety, Security and Rescue Robotics*, Sf. 81–86.
- [10] **Ruangpayoongsak, N., Roth, H. ve Chudoba, J.** (2005). Mobile robots for search and rescue, *Proceedings of the 2005 IEEE International Workshop on Safety, Security and Rescue Robotics*, Sf. 212–217.
- [11] **Tunwannarux, S. ve Tunwannarux, A.** (2008). The CEO Mission IV Rescue Robot for World RoboCup and USAR field, *Proceedings of the 10th WSEAS International Conference on Automatic Control, Modelling and Simulation*, Sf. 439–444.

- [12] **Marques, C., Cristóvão, J., Lima, P., Frazão, J., Ribeiro, I. ve Ventura, R.** (2006). RAPOSA: Semi-autonomous robot for rescue operations, *IEEE International Conference on Intelligent Robots and Systems*, Sf. 3988–3993.
- [13] **Edlinger, R., Pölzleithner, A. ve Zauner, M.** (2011). Mechanical design and system architecture of a tracked vehicle robot for urban search and rescue operations, *Communications in Computer and Information Science*, Cilt **156**, Sf. 46–56.
- [14] **Miyauchi, R., Shiroma, N. ve Matsuno, F.** (2008). Compact image stabilization system using camera posture information, *Journal of Field Robotics*, Cilt **25**, Sayı. 4–5, Sf. 268–283.
- [15] **Ryu, Y. G., Roh, H. C., Kim, S. J., An, K. H. ve Chung, M. J.** (2009). Digital image stabilization for humanoid eyes inspired by human VOR system, *2009 IEEE International Conference on Robotics and Biomimetics*, Sf. 2301–2306.
- [16] **Luo, Q. ve Khoshgoftaar, T. M.** (2007). An empirical study on estimating motions in video stabilization, *2007 IEEE International Conference on Information Reuse and Integration*, Sf. 360–366.
- [17] **Smith, M. J., Boxerbaum, A., Peterson, G. L. ve Quinn, R. D.** (2010). Electronic image stabilization using optical flow with inertial fusion, *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems*, Sf. 1146–1153.
- [18] **Sato, K., Ishizuka, S., Nikami, A. ve Sato, M.** (1993). Control techniques for optical image stabilizing system, *IEEE Transactions on Consumer Electronics*, Cilt **39**, Sayı. 3, Sf. 461–466.
- [19] **Cardani, B.** (2006). Optical image stabilization for digital cameras, *IEEE Control Systems Magazine*, Cilt **26**, Sayı. 2, Sf. 21–22.
- [20] **Hurák, Z. ve Řezáč, M.** (2009). Combined line-of-sight inertial stabilization and visual tracking: Application to an airborne camera platform, *Proceedings of the IEEE Conference on Decision and Control*, Sf. 8458–8463.
- [21] **Lee, D. B., Chitrakaran, V., Burg, T., Dawson, D. ve Xian, B.** (2007). Control of a remotely operated quadrotor aerial vehicle and camera unit using a fly-the-camera perspective, *Proceedings of the IEEE Conference on Decision and Control*, Sf. 6412–6417.
- [22] **Nickels, K. ve Huber, E.** (2001). Inertially assisted stereo tracking for an outdoor rover, *Proceedings - IEEE International Conference on Robotics and Automation*, Cilt **3**, Sf. 3078–3083.
- [23] **Salzmann, D. C.** (2007). Ampelmann prototype-developing a motion compensating platform for offshore access, *European Wind Energy Conference*.
- [24] **Kim, K. H., Lee, J. K., Park, B. S. ve Kang, M. S.** (2012). Chatter-free sliding mode control for inertial stabilization of OTM (On-the-move) antenna driven by gear and flexible shaft, *International Journal of Precision Engineering and Manufacturing*, Cilt **13**, Sayı. 8, Sf. 1317–1325.

- [25] **Koruba, Z., Dziopa, Z. ve Krzysztofik, I.** (2010). Dynamics and control of a gyroscope-stabilized platform in a self-propelled anti-aircraft system, *Journal of Theoretical and Applied Mechanics*, Cilt **48**, Sayı. 1, Sf. 5–26.
- [26] **Tsai, L.** (1999). *Robot analysis: the mechanics of serial and parallel manipulators*, John Wiley & Sons, New York.
- [27] **Curtis, H. D.** (2009). *Robot analysis: the mechanics of serial and parallel manipulators*, Elsevier Butterworth-Heinemann, Amsterdam.
- [28] **Url-1** <http://www.starlino.com/imu_guide.html>, alındığı tarih: 14.05.2012.
- [29] **Analog Devices**, ADXL345 3-Axis, ± 2 g / ± 4 g / ± 8 g / ± 16 g Digital Accelerometer Datasheet.
- [30] **Borenstein, J., Ojeda, L. ve Kwanmuang, S.** (2009). Heuristic reduction of gyro drift for personnel tracking systems, *Journal of Navigation*, Cilt **62**, Sayı. 1, Sf. 41-58.
- [31] **Url-2** <web.mit.edu/scolton/www/filter.pdf>, alındığı tarih: 04.06.2012.
- [32] **Welch, G. ve Bishop, G.** (2006). An Introduction to the Kalman Filter, *UNC-Chapel Hill*.
- [33] **Url-3** <<http://www.pololu.com/catalog/product/1060>>, alındığı tarih: 15.06.2012.
- [34] **Parallax**, Standart Servo Datasheet.
- [35] **Url-4** <<http://www.arduino.cc/>>, alındığı tarih: 12.05.2012.
- [36] **Url-5** <<http://www.sparkfun.com/products/10121/>>, alındığı tarih: 25.05.2012.
- [37] **InvenSense**, ITG-3200 Product Specification Datasheet, Revision 1.4.
- [38] **Url-6** <<http://www.pololu.com/catalog/product/1213>>, alındığı tarih: 25.05.2012.
- [39] **Url-7** <<http://www.sparkfun.com/tutorials/171>>, alındığı tarih: 15.06.2012.
- [40] **Paret, D. ve Fenger, C.** (1997). *The I²C bus: from theory to practice*. John Wiley & Sons, Chichester.
- [41] **Gao, Y. ve Ehsani, M.** (2006). Parametric design of the traction motor and energy storage for series hybrid off-road and military vehicles, *IEEE Transactions on Power Electronics*, Cilt **21**, Sayı. 3, Sf. 749–755.
- [42] **Yaman, E. ve Ertürk, S.** (2002). 'Görüntü stabilizasyonu için paralel işlev gören iki Kalman filtresiyle işlem gürültü varyansının adaptifleştirilmesi", 10. *IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU'2002) Bildirileri Kitabı*, Sf. 506–511.
- [43] **Salau, N. P. G., Trierweiler, J. O., Secchi, A. R. ve Marquardt, W.** (2009). A new process noise covariance matrix tuning algorithm for Kalman based state estimators, *International Symposium on Advanced Control of Chemical Processes*, Cilt **7**, Sayı. 1, Sf. 572–577.

EKLER

EK A : Mikroişlemci kodları

EK A.1: Mobil robot mikroişlemci kodu

EK A.2: Kumanda devresi mikroişlemci kodu

EK A.1

```
//***** //  
// İsim      : Mobil Robot Mikroişlemci Kodu      //  
// Yazan     : Mehmet Volkan BÜKEY              //  
// Tarih     : 8 Haziran 2012                    //  
// Versiyon  : 1.0                              //  
//***** //  
  
/*****  
BAŞLANGIÇ  
*****/  
  
// Kütüphane dosyaları  
#include <Wire.h>           // I2C kütüphanesi  
#include <Servo.h>          // Servo motor kütüphanesi  
  
// İsim atamaları  
#define ACC (0x53)          // ADXL345 I2C cihaz adresi  
#define GYRO 0x68           // ITG3200 I2C cihaz adresi  
#define A_TO_READ (6)      // Her döngüde ivmeölçerden alınacak veri paketi sayısı  
#define G_TO_READ 8        // Her döngüde jiroskoptan alınacak veri paketi sayısı  
#define BW_RATE 0x2C        // İvmeölçer I2C yazmaçları: Veri iletim hızı  
#define DATA_FORMAT 0x31   // İvmeölçer I2C yazmaçları: Ölçme aralığı  
#define POWER_CTL 0x2D      // İvmeölçer I2C yazmaçları: Çalışma modu  
#define G_SMPLRT_DIV 0x15    // Jiroskop I2C yazmaçları: Örneklem hızı  
#define G_DLPF_FS 0x16      // Jiroskop I2C yazmaçları: Düşük geçiren filtre  
#define G_INT_CFG 0x17       // Jiroskop I2C yazmaçları: Kesme ayarı  
#define G_PWR_MGM 0x3E       // Jiroskop I2C yazmaçları: Güç ayarı  
#define motor1yon1 5         // Motor sürücü kartına ait pinlerin atamaları  
#define motor2yon1 11  
#define motor1yon2 6  
#define motor2yon2 3  
  
// Servo motor kütüphaneleri atamaları  
Servo serv1;                // Yuvarlanma açısı  
Servo serv2;                // Yunuslama açısı  
Servo serv3;                // Sapma açısı 1  
Servo serv4;                // Sapma açısı 2  
  
int limitx,limity;          // Servo motorlar için limit değerleri  
int pos1=120;               // Sapma açısını kontrol eden servo motorların  
int pos2=180;               // başlangıç değerleri  
  
// Zamanlayıcı atamaları  
int STD_LOOP_TIME = 20;     // 20 ms sabit döngü süresi  
int lastLoopTime = STD_LOOP_TIME; // En son döngü süresi başlangıç değeri  
int lastLoopUsefulTime = STD_LOOP_TIME; // Kullanılan döngü süresi başlangıç  
// değeri  
unsigned long loopStartTime = 0; // Döngü başlangıç süresi
```

// Ataletsel ölçüm cihazı atamaları

```
float tmpf=0.0;      // Her döngü için açış deęerleri
float g_offx=0;      // Jirokobun sıfır deęer toleransı hatasını düzelten deęişkenler
float g_offy=0;
float g_offz=0;
```

```
float Accel[2];      // İvmeölçerden elde edilen yunuslama ve yuvarlanma açısı
float Gyro [2];      // Jiroskoptan elde edilen yunuslama ve yuvarlanma açısı
```

```
float Gyro_ds[3];    // Düzenlenmiş jirooskop verilerini bulunduran dizi
float RwAcc[3];      // Düzenlenmiş ivmeölçer verilerini bulunduran dizi
```

// Kalman filtresi atamaları

```
float xAngle;        // Kalman filtresinden elde edilen yuvarlanma açısı
float yAngle;        // Kalman filtresinden elde edilen yunuslama açısı
float Q_aa= 0.001;   // İşlem gürültü kovaryansları
float Q_ga= 0.001;
float Q_ag= 0;
float Q_ga= 0;
```

```
float RX_angle= 0.4461; // Ölçüm gürültü kovaryansı
float Ry_angle= 0.4369; // Ölçüm gürültü kovaryansı
```

```
float x_angle= 0;      // X eksen Euler açısı
float x_bias = 0;      // X eksen jirooskop kayma deęeri
float PX_00 = 1, PX_01 = 0, PX_10 = 0, PX_11 =1; // X eksen için hata
// kovaryansları
```

```
float y_angle = 0;     // Y eksen Euler açısı
float y_bias = 0;      // Y eksen Jirooskop kayma deęeri
float PY_00 = 1, PY_01 = 0, PY_10 = 0, PY_11 = 1; // Y eksen için hata
// kovaryansları
```

```
float dt, yX, SX,yY,SY; // İki eksen için Kalman filtresi denklemlerinde kullanılan
// parametreler
```

```
float KX_0, KX_1,KY_0, KY_1; //İki eksen için Kalman kazancı vektörü bileşenleri
```

//Seri iletişim atamaları

```
int incomingByte = 0; // Seri iletişim üzerinden alınan bayt
```

*/*******

EK FONKSİYONLAR

******/*

// İvmeölçer ölçüm parametrelerini sensöre ileten fonksiyon

```
void initAcc() {
    writeTo(ACC, BW_RATE,0x09); // Veri iletim hızı 50 Hz yapılır.
    writeTo(ACC, DATA_FORMAT, 0x00); // Ölçme aralığı ± 2 g yapılır.
    writeTo(ACC, POWER_CTL, 0x08); // İvmeölçer ölçüm moduna getirilir.
}
```

```

// İvmeölçerden her eksen için ivme değerlerini alan fonksiyon
void getAccelerometerData(int * result) {
    int regAddress = 0x32;    // İvme verilerinin sensör hafızasındaki ilk adresi
    byte buff[A_TO_READ];    // Sensör verilerinin kaydedileceği dizi

    readFrom(ACC, regAddress, A_TO_READ, buff); // İvmeölçerden veri paketleri
                                                // alınır ve her eksene ait iki
                                                // veri paketi birleştirilir.
    result[0] = (((int)buff[1]) << 8) | buff[0];    // X eksen ivme değerleri
    result[1] = (((int)buff[3]) << 8) | buff[2];    // Y eksen ivme değerleri
    result[2] = (((int)buff[5]) << 8) | buff[4];    // Z eksen ivme değerleri
}

// İvmeölçerden elde edilen verileri g cinsinden hesaplayan fonksiyon
void rawAccToG(int * raw, float * RwAcc) {
    RwAcc[0] = ((float) raw[0]) / 256.0;
    RwAcc[1] = ((float) raw[1]) / 256.0;
    RwAcc[2] = ((float) raw[2]) / 256.0;
}

// İvme değerlerini normalize eden fonksiyon
void normalize3DVec(float * vector) {
    float R;    // Denklem 2.1'de tanımlanmış R vektörü
               // R vektörünün büyüklüğü
    R = sqrt(vector[0]*vector[0] + vector[1]*vector[1] + vector[2]*vector[2]);
    vector[0] /= R;
    vector[1] /= R;
    vector[2] /= R;
}

// Jiroskop ölçüm parametrelerini sensöre ileten fonksiyon
void initGyro()
{
    writeTo(GYRO, G_SMPLRT_DIV, 0x13); // Örnekleme frekansı 50 Hz alınır.
    writeTo(GYRO, G_DLPF_FS, 0x1E);    // Ölçme aralığı ± 2000 derece/san olarak
                                        // belirlenir ve düşük geçiren filtrenin bant
                                        // aralığı 5 Hz olarak alınır.
}

void getGyroscopeData(int * result)
{
    int regAddress = 0x1B; //Açısal hız verilerinin sensör hafızasındaki ilk adresi
    byte buff[G_TO_READ]; // Sensör verilerinin kaydedileceği dizi
    readFrom(GYRO, regAddress, G_TO_READ, buff); // Jiroskoptan veri paketleri
                                                // alınır ve her eksene ait veri
                                                // paketleri birleştirilir.
    result[0] = ((buff[2] << 8) | buff[3]);    // X ekseninde açısal hız
    result[1] = ((buff[4] << 8) | buff[5]);    // Y ekseninde açısal hız
    result[2] = ((buff[6] << 8) | buff[7]);    // Z ekseninde açısal hız
    result[3] = (buff[0] << 8) | buff[1];    // Sıcaklık
}

```

// Jiroskoptan elde edilen verileri g cinsinden hesaplayan fonksiyon

```
void rawGyroToDegsec(int * raw, float * gyro_ds) {  
    gyro_ds[0] = (((float) raw[0]) - g_offx) / 14.375;  
    gyro_ds[1] = (((float) raw[1]) - g_offy) / 14.375;  
    gyro_ds[2] = (((float) raw[2]) - g_offz) / 14.375;  
}
```

// Jiroskoptan hareketsiz durumda elde edilen verileri hesaplayan fonksiyon

```
void Zero(){  
    int result[4];          // Jiroskoptan hareketsizken 100 kere ölçüm alınır.  
    int regAddress = 0x1B; // Bu değerler her eksen için toplanır ve ortalaması alınır.  
  
    byte buff[G_TO_READ];  
    int turn=100;  
    for(int i=0;i<turn;i++){  
        readFrom(GYRO, regAddress, G_TO_READ, buff);  
  
        result[0] = ((buff[2] << 8) | buff[3]);  
        result[1] = ((buff[4] << 8) | buff[5]);  
        result[2] = ((buff[6] << 8) | buff[7]);  
        result[3] = (buff[0] << 8) | buff[1];  
  
        g_offx += result[0];  
        g_offy += result[1];  
        g_offz += result[2];  
    }  
    g_offx = g_offx/turn;  
    g_offy = g_offy/turn;  
    g_offz = g_offz/turn;  
}
```

//Y ekseninde Euler açısını düzelten Kalman filtresi

```
float kalmanCalculateY(float newAngle, float newRate,int looptime) {
```

```
    dt = float(looptime)/1000;          // Kalman filtresinde kullanılan zaman döngüsü  
    y_angle += dt * (newRate - y_bias); // Yeni ölçüm adımı için durum tahmini  
                                         // hesaplanır.
```

// Yeni durum tahminine göre hata kovaryansı güncellenir.

```
PY_00 += - dt * (PY_10 + PY_01) + PY_11*(dt*dt)+ Q_aa;  
PY_01 += - dt * PY_11+Q_ag;  
PY_10 += - dt * PY_11+Q_ga;  
PY_11 += + Q_gg;
```

```
yY = newAngle - y_angle;          // Kalman kazancı hesaplanır.  
SY = PY_00 + RY_angle;  
KY_0 = PY_00 / SY;  
KY_1 = PY_10 / SY;
```

```
y_angle += KY_0 * yY;          // Yeni tahmin değeri hesaplanır  
y_bias += KY_1 * yY;
```

```

PY_00 -= KY_0 * PY_00;      // Gelecek ölçüm için hata kovaryansı güncellenir.
PY_01 -= KY_0 * PY_01;
PY_10 -= KY_1 * PY_00;
PY_11 -= KY_1 * PY_01;

```

```

return y_angle;      // Kalman filtresinden düzeltilmiş Euler açısı elde edilir.
}

```

// X ekseninde Euler açısını düzelten Kalman filtresi

```

float kalmanCalculateX(float newAngle, float newRate, int looptime) {
    dt = float(looptime)/1000;      // Kalman filtresinde kullanılan zaman döngüsü

```

```

    x_angle += dt * (newRate - x_bias); // Yeni ölçüm adımı için durum tahmini
                                         // hesaplanır.

```

// Yeni durum tahminine göre hata kovaryansı güncellenir.

```

PX_00 += - dt * (PX_10 + PX_01) + PX_11*(dt*dt)+Q_aa;
PX_01 += - dt * PX_11+Q_ag;
PX_10 += - dt * PX_11+Q_ga;
PX_11 += + Q_gg;

```

```

yX = newAngle - x_angle;      // Kalman kazancı hesaplanır.
SX = PX_00 + RX_angle;
KX_0 = PX_00 / SX;
KX_1 = PX_10 / SX;

```

```

x_angle += KX_0 * yX;      // Yeni tahmin değeri hesaplanır
x_bias += KX_1 * yX;

```

```

PX_00 -= KX_0 * PX_00;      // Gelecek ölçüm için hata kovaryansı güncellenir.
PX_01 -= KX_0 * PX_01;
PX_10 -= KX_1 * PX_00;
PX_11 -= KX_1 * PX_01;

```

```

return x_angle;      // Kalman filtresinden düzeltilmiş Euler açısı elde edilir.
}

```

// I2C veri yazma fonksiyonu

```

void writeTo(int DEVICE, byte address, byte val) {
    Wire.beginTransmission(DEVICE);      // Cihazla iletişime başlanır.
    Wire.write(address);                  // Yazmaç adresi belirlenir.
    Wire.write(val);                      // Yazmaça yazılacak değer belirlenir.
    Wire.endTransmission(); // Yazmaç adresi ve yazmaça yazılacak değer cihaza
                                // gönderilir. Sonrasında iletişim durdurulur.
}

```

//I2C veri okuma fonksiyonu

```

void readFrom(int DEVICE, byte address, int num, byte buff[]) {
    Wire.beginTransmission(DEVICE);      // Cihazla iletişime başlanır.
    Wire.write(address);                  // Cihazdan okunacak değer yazmaç adresi belirlenir.
    Wire.endTransmission(); // Yazmaç adresi cihaza gönderilir ve iletişim durdurulur.
}

```

```

Wire.beginTransmission(DEVICE);    // Cihazla iletişime tekrar başlanır.
Wire.requestFrom(DEVICE, num);    // Cihazdan belirli sayıda veri paketi istenir.

int i = 0;
while(Wire.available())            // Cihazdan veri geldikçe okunup kaydedilir.
{
    buff[i] = Wire.read();
    i++;
}
Wire.endTransmission();            // Cihazla iletişim durdurulur.
}

/*****
KOD DÖNGÜSÜ VE MİKROİŞLEMCİ AYARLARI
*****/

void setup()                       // Ayarlar
{
    serv1.attach(12);              // Servo motor pinleri belirtilir.
    serv2.attach(2);
    serv3.attach(13);
    serv4.attach(4);

    Serial.begin(115200);          // Seri iletişim başlatılır.
    Wire.begin();                 // I2C iletişimi başlatılır.

    Zero();                       // Jiroskoptan hareketsiz durumda elde edilen veriler hesaplanır.
    initAcc();                    // İvmeölçer ölçüm parametreleri girilir.
    initGyro();                   // Jiroskop ölçüm parametreleri girilir.

    serv1.write(92);              // Servo motorlar başlangıç konumuna getirilir.
    serv2.write(73);
    serv3.write(120);
    serv4.write(180);
    delay(10);                    // 10 ms beklenir.
    pinMode(motor1yon1,OUTPUT);    //Sürücü kartın pinleri çıkış olarak tanımlanır.
    pinMode(motor1yon2,OUTPUT);
    pinMode(motor2yon1,OUTPUT);
    pinMode(motor2yon2,OUTPUT);
}

void loop()                       // Kod döngüsü
{

//Kumanda devresi ile kontrol
if (Serial.available() > 0) {    // Seri iletişim mevcut ise alınan baytlar kaydedilir.
    incomingByte=Serial.read();  // Sonrasında alınan byte değerlerine göre palet
                                // motorları ya da sapma açısını kontrol eden servo
                                // motorlar kontrol edilir. Palet motorları hareket
                                // ederken sapma açısını kontrol eden servo motorlar
                                // başlangıç konumuna getirilir.

```

```

} if (incomingByte == '5') {                                // Robot durur.
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,LOW);
}
else if (incomingByte=='6'){                                // Robot saat yönünde hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,HIGH);
    digitalWrite(motor2yon1,HIGH);
    digitalWrite(motor2yon2,LOW);
}
else if (incomingByte=='4'){                                // Robot saat yönünün tersine doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,HIGH);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,HIGH);
}
else if (incomingByte=='1'){                                // Robot geriye ve sola doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    analogWrite(motor1yon1,150);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,HIGH);
    digitalWrite(motor2yon2,LOW);
}
else if (incomingByte=='3'){                                // Robot geriye ve sağa doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,HIGH);
    digitalWrite(motor1yon2,LOW);
    analogWrite(motor2yon1,150);
    digitalWrite(motor2yon2,LOW);
}
else if (incomingByte=='2'){                                // Robot geriye doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,HIGH);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,HIGH);
    digitalWrite(motor2yon2,LOW);
}
else if (incomingByte=='7'){                                // Robot ileriye ve sola doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,LOW);

```

```

    analogWrite(motor1yon2,150);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,HIGH);
}
else if (incomingByte=='8'){ // Robot ileriye doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,HIGH);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,HIGH);
}
else if (incomingByte=='9'){ // Robot ileriye ve sağa doğru hareket eder.
    serv3.write(120);
    serv4.write(180);
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,HIGH);
    analogWrite(motor2yon2,150);
    digitalWrite(motor2yon1,LOW);
}
else if( incomingByte=='a'){ // Sapma açısını kontrol eden servo motorlar hareket
    // eder.
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,LOW);
    if(pos1<170){ // Harekete ilk olarak altta bulunan motordan başlanır.
        pos1 += 1;
        serv3.write(pos1);
    }
    else{ // Daha sonra yukarıda bulunan motor hareket ettirilir.
        pos1=170;
        pos2 -=1;
        serv4.write(pos2);
        if(pos2<1){
            pos2=0;
        }
    }
}
else if( incomingByte=='d'){ // Sapma açısını kontrol eden servo motorlar ters
    // yönde hareket eder.
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,LOW);
    if(pos1<170){
        pos1 -= 1;
        serv3.write(pos1);
        if(pos1<20){
            pos1=20;
        }
    }
}

```

```

else if( pos1==170 && pos2==180){
    pos1 -= 1;
}
else{
    pos2 +=1;
    serv4.write(pos2);
}
}
else{ // Yukarıda belirtilen baytlar dışında bir bayt gelmesi durumunda robot
    // hareket etmez.
    digitalWrite(motor1yon1,LOW);
    digitalWrite(motor1yon2,LOW);
    digitalWrite(motor2yon1,LOW);
    digitalWrite(motor2yon2,LOW);
}

//Kamera dengeleme sistemi
int acc[3]; // İvmeölçer verileri bu dizide kaydedilir. (x,y,z)
int gyro[4]; // Jiroskop verileri bu dizide toplanır. (x,y,z,sıcaklık)
getAccelerometerData(acc); // İvmeölçer verileri alınır.
rawAccToG(acc, RwAcc); // Veriler hassasiyet değerine bölünerek g
// cinsinden elde edilir.
normalize3DVec(RwAcc); // Elde edilen ivme vektörü normalize edilir.
getGyroscopeData(gyro); // Jiroskop verileri alınır.
rawGyroToDegsec(gyro, Gyro_ds); // Veriler hassasiyet değerine bölünerek
// derece/san cinsinden elde edilir.

for(int w=0;w<=1;w++){ // Jiroskop verileri açısal hızdan açığa çevrilir.
    tmpf = Gyro_ds[w];
    tmpf *= STD_LOOP_TIME / 1000.0f;
    Gyro[w] += tmpf;
}

// İvmeölçerden elde edilen yunuslama açısı
Accel[0]= atan2 (RwAcc[0] , RwAcc[2]) * 180 /PI;
// İvmeölçerden elde edilen yuvarlanma açısı
Accel[1]= atan2 (RwAcc[1] , RwAcc[2]) * 180 /PI;
// Kalman filtresi ile düzeltilen yuvarlanma açısı
xAngle = kalmanCalculateX(Accel[1], Gyro[0],STD_LOOP_TIME);
// Kalman filtresi ile düzeltilen yunuslama açısı
yAngle = kalmanCalculateY( Accel[0], Gyro[1],STD_LOOP_TIME);

limitx=92+xAngle; // Yuvarlanma açısını kontrol eden servo motorun limit değeri
limity=73+yAngle; // Yunuslama açısını kontrol eden servo motorun limit değeri
if(limity >= 180){ // İki servo motor için de limit değeri aşılması durumunda
    // motorlar durur.
    serv2.write(179);
}
else if (limity <=0){
    serv2.write(1); }

```

```

else{
    serv2.write(73-yAngle);} // Kalman filtresinden elde edilen açı değeri servo
                             // motorun sıfır konumundan çıkarılarak servo motora
                             // iletilir. Bu sayede robotun gövdesi yönelimini
                             // değiştirse bile servo motor ters yönde hareket ederek
                             // kamerayı sabit tutar.

    if(limitx >= 180){
        serv1.write(179);
    }
    else if ( limitx <=0){
        serv1.write(1);
    }else{
        serv1.write(92+xAngle); // Sensör ters monte edildiği için X eksenini üzerindeki
                                // açı çıkarılmak yerine toplanmıştır.
    }

//Zamanlayıcı kontrolü
lastLoopUsefulTime = millis()-loopStartTime; // En son döngü süresi hesaplanır.
if(lastLoopUsefulTime<STD_LOOP_TIME) { // Eğer bu değer sabit döngü
                                         // süresinden az ise
        delay(STD_LOOP_TIME-lastLoopUsefulTime); // döngü süresi 20 ms'e eşit
                                                  // olana kadar kod bekletilir.
    }
    loopStartTime = millis(); // Yeni döngü için zamanlayıcı tekrar başlatılır.
}

```

EK A.2

```
/**
// İsim      : Kumanda Devresi Mikroişlemci Kodu
// Yazan     : Mehmet Volkan BÜKEY
// Tarih      : 13 Temmuz 2012
// Versiyon   : 1.1
**/

//Joystick Shield pin atamaları
const byte PIN_BUTTON_SELECT = 2; // Hareketli kolun altında bulunan düğme
const byte PIN_BUTTON_RIGHT = 3;  // Elektrik düğmesi pinleri
const byte PIN_BUTTON_UP = 4;
const byte PIN_BUTTON_DOWN = 5;
const byte PIN_BUTTON_LEFT = 6;
const byte PIN_ANALOG_X = 0; // Hareketli kolun altındaki
const byte PIN_ANALOG_Y = 1; // potansiyometrelerin pinleri

byte s=LOW; // Servo motorları kontrol eden düğmelerin durumunu gösterir.
int x,y; // Hareketli kola ait analog giriş değerleri
byte a,d; // Servo motorları kontrol eden düğmelerin lojik değerleri
void setup() { // Ayarlar
  Serial.begin(115200); // Seri iletişim başlatılır.
  pinMode(PIN_BUTTON_RIGHT, INPUT); // Bütün düğme pinleri Joystick
  // Shield'a göre ayarlanır.

  digitalWrite(PIN_BUTTON_RIGHT, HIGH);
  pinMode(PIN_BUTTON_LEFT, INPUT);
  digitalWrite(PIN_BUTTON_LEFT, HIGH);
  pinMode(PIN_BUTTON_UP, INPUT);
  digitalWrite(PIN_BUTTON_UP, HIGH);
  pinMode(PIN_BUTTON_DOWN, INPUT);
  digitalWrite(PIN_BUTTON_DOWN, HIGH);
  pinMode(PIN_BUTTON_SELECT, INPUT);
  digitalWrite(PIN_BUTTON_SELECT, HIGH);
}

void loop() { // Kod döngüsü

  x= analogRead(PIN_ANALOG_X); // Joystick Shield üzerinde kullanılan
  // ekipmanlardan okuma yapılır.

  y= analogRead(PIN_ANALOG_Y);
  a= digitalRead(PIN_BUTTON_LEFT);
  d= digitalRead(PIN_BUTTON_RIGHT);

  if(a==LOW || d==LOW){ // Eğer servo motoru kontrol eden
  // düğmelerden bir tanesi basılı ise s değişkeni
  // lojik değerini değiştirir.

  s=HIGH;

}
}
```

```

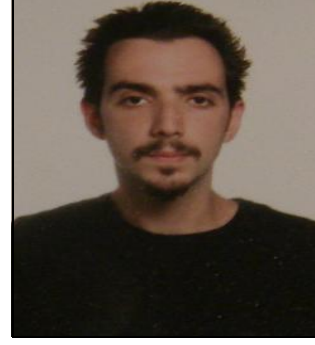
else {
s=LOW;
}
// Hareketli kolun ve düğmelerin durumuna göre kablosuz haberleşme üzerinden veri
// gönderilir.

// Robot durur.
if (x>400 && x<=600 && y>400 && y<=600 && s==LOW){
Serial.print('5');
}
// Robot saat yönünde hareket eder.
else if(x>=0 && x<=400 && y>400 && y<=600 && s==LOW){
Serial.print('6');
}
// Robot saat yönünün tersine doğru hareket eder.
else if(x<=1023 && x>600 && y>400 && y<=600 && s==LOW){
Serial.print('4');
}
// Robot geriye doğru hareket eder.
else if( x>400 && x<=600 && y<=1023 && y>600 && s==LOW){
Serial.print('2');
}
//Robot ileriye doğru hareket eder.
else if(x>400 && x<=600 && y>=0 && y<=400 && s==LOW){
Serial.print('8');
}
// Robot geriye ve sola doğru hareket eder.
else if(x<=1023 && x>600 && y<=1023 && y>600 && s==LOW){
Serial.print('1');
}
// Robot geriye ve sağa doğru hareket eder.
else if(x>=0 && x<=400 && y<=1023 && y>600 && s==LOW){
Serial.print('3');
}
// Robot ileriye ve sola doğru hareket eder.
else if (x<=1023 && x>600 && y>=0 && y<=400 && s==LOW){
Serial.print('7');
}
// Robot ileriye ve sağa doğru hareket eder.
else if (x>=0 && x<=400 && y>=0 && y<=400 && s==LOW){
Serial.print('9');
}
// Sapma açısını kontrol eden servo motorlar sola doğru döner.
else if ( s==HIGH && a==LOW){
Serial.print('a');
}
// Sapma açısını kontrol eden servo motorlar sağa doğru döner.
else if ( s==HIGH && d==LOW){
Serial.print('d');
}
}

```

```
delay(1000);           // Kablosuz haberleşme üzerinde yığılma  
                        // olmaması için veriler her saniyede bir defa gönderilir.  
}
```

ÖZGEÇMİŞ



Ad Soyad: Mehmet Volkan BÜKEY

Doğum Yeri ve Tarihi: İstanbul 1986

Adres: Nispetiye Cad. Tezeren Çıkmaşı Gülan Sit. C Blok D:2 Etiler/İstanbul

E-Posta: wolkanb@gmail.com

Lisans: İTÜ Elektrik Mühendisliği