**T.C.**

**TRAKYA ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

# TANK NAMLUSU STABİLİZASYON SİSTEMİNİN ARDUİNO İLE UYGULANMASI VE DENEYSEL DÜZENEĞİNİN HAZIRLANMASI

**SALİM SONGÜL**

**YÜKSEK LİSANS TEZİ**

**MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI**

**Tez Danışmanı: YRD. DOÇ. DR. HİLMİ KUŞÇU**

**EDİRNE-2014**

T.Ü. Fen Bilimleri Enstitüsü onayı

Prof. Dr. Mustafa ÖZCAN

Fen Bilimleri Enstitüsü Müdürü

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

Prof. Dr. Taner TIMARCI

Anabilim Dalı Başkanı

Bu tez tarafımca okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Hilmi KUŞÇU

Tez Danışmanı

Bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Makine Mühendisliği Anabilim Dalında bir Yüksek Lisans tezi olarak oy birliği ile kabul edilmiştir.

Jüri Üyeleri İmza

Yrd. Doç. Dr. Hilmi KUŞÇU

Prof. Dr. Selim KARA

Doç. Dr. Oktay HACIHAFIZOĞLU

Tarih: 03/06/2014

**T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ**

**MAKİNE MÜHENDİSLİĞİYÜKSEK LİSANS PROGRAMI**

**DOĞRULUK BEYANI**

İlgili tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin kaynak gösterilerek ilgili tezde yer aldığını beyan ederim.

03/06/2014

Salim SONGÜL

Yüksek Lisans Tezi
SALİM SONGÜL
T.Ü. Fen Bilimleri Enstitüsü
Makine Mühendisliği Anabilim Dalı

## ÖZET

Günümüzde veya gelecekte icra edilecek olan muharebelerde duran birliklerin yerine yüksek hareket kabiliyetine sahip birliklere ihtiyaç duyulacaktır. Birliklerin hareketli olması hem kolay hedef olmalarını engelleyecek hem icra edilecek olan harekâtın baskın etkisini arttıracaktır. Hareket halinde olan bir silah sistemi değişik/bozuk yol ve arazi şartlarında sürekli sarsılmakta veya konum değiştirmektedir. Bu durumda silah sistemlerini kullanan nişancıların vuruş yüzdelerinin yüksek olması son derece zorlaşmaktadır. Hareket halinde olan bir silah sistemini kullanan nişancının vuruş yüzdesi yüksek atışlar yapmasını sağlamak maksadıyla silahın sürekli hedef bölgesinde ve sabit bir şekilde durması gerekmektedir.

Yıl : 2014
Sayfa Sayısı : 57
Anahtar Kelimeler : Silah Stabilizasyonu, Tank, Otomatik Kontrol

Master's Thesis
SALİM SONGÜL
Trakya UniversityInstitute of Natural Sciences
Department of Mechanical Engineering

## ABSTRACT

Nowadays and in the future, high mobility combat troops will be needed instead of the steady troops in the battlefields. The mobility of troops improves the effect of incursion and prevent from becoming an easy target. Mobile gun systems move and vibrate easily because of rough land. Thus, shooting the target and increase the success of hitting target becomes more difficult for tank gunners. In order to improve the hitting target and shoot robust, the gun barrel on the turret of a mobile tank has to be stable in the target region.

Year : 2014
Number of Pages : 57
Keywords : Gun Stabilization, Tank, Automatic Control

## TEŞEKKÜR

Bu seminer çalışmasının ortaya çıkmasında fikirleriyle bana ışık tutan, tecrübesini benden esirgemeyen ve bana yol gösteren saygıdeğer hocam ve danışmanım Sayın Yrd. Doç. Dr. Hilmi KUŞÇU'ya tüm katkılarından dolayı teşekkür ederim.

Ayrıca yoğun zamanlarımda bana karşı anlayışlı davranan ve manevi desteğini hiç esirgemeyen aileme teşekkür ederim.

# İÇİNDEKİLER

ÖZET ..... iv

ABSTRACT ..... v

TEŞEKKÜR ..... vi

ŞEKİL LİSTESİ ..... ix

1. GİRİŞ ..... 1

2. KAYNAK ARAŞTIRMASI ..... 2

2.1 Tankın Genel Tanımı ..... 2

2.2 Otomatik Kontrol ..... 9

2.2.1 Genel Bilgiler ..... 9

2.2.2 Temel Kavramlar ve Tanımlar ..... 9

2.2.3 Denetim Sistemleri ve Türleri ..... 14

2.2.4 Denetim Sistemlerinin Yapısı ve Kullanım Alanları ..... 17

2.2.5 Otomatik Denetim Sistemi Uygulamaları ..... 18

3. MATERYAL VE TEORİ ..... 20

3.1 Tank Namlusu Stabilizasyon Sisteminin Tasarımı ..... 20

3.2 Tank Namlusu Stabilizasyon Sisteminin Tasarımında Kullanılan Malzemeler ..... 21

3.2.1 Arduino Uno Mikroişlemci Kartı ..... 21

3.2.2 MPU-6050 Sensörü ..... 23

3.2.3 Doğru Akım (DC) Motoru ..... 31

3.2.4 Motor Kontrol Kartı (L298 destekli) ..... 32

3.2.5 Model Tank ..... 35

3.3. Tank Namlusu Stabilizasyon Sisteminin Modellenmesi ..... 37

4. SONUÇLAR VE TARTIŞMA ..... 40

EKLER ..... 41

EK-A ..... 41

KAYNAKLAR .....55
ÖZGEÇMİŞ .....57

# ŞEKİL LİSTESİ

Şekil 2.1 Temel Tanımları Gösterilen Genelleştirilmiş Geri Beslemeli (Kapalı Döngü) Sistem Blok Şeması ..... 10
Şekil 2.2 Açık Döngü Denetim Sistemi ..... 10
Şekil 2.3 Kapalı Döngü Denetim Sistemi ..... 11
Şekil 3.1 Tank Namlusu Stabilizasyon Sistemi Blok Diyagramı ..... 21
Şekil 3.2 Arduino Uno Mikroişlemci Kartı (Ön ve Arka Yüzü) ..... 21
Şekil 3.3 Arduino Uno Mikroişlemci Kartı ..... 22
Şekil 3.4 MPU-6050 Sensörü ..... 23
Şekil 3.5 İvmeölçerin Yerçekimsiz Ortam Modellemesi ..... 24
Şekil 3.6 İvmeölçerin Yerçekimsiz Ortamda Hareket Modellemesi ..... 24
Şekil 3.7 İvmeölçerin Yerçekimi Ortamında Modellemesi ..... 24
Şekil 3.8 İvmeölçerin Yerçekimi Ortamında $45^0$ Sağa Doğru Çevrilmesi ..... 25
Şekil 3.9 R Vektörünün Koordinat Sistemi Üzerinde Gösterimi ..... 25
Şekil 3.10 R Vektörünün Koordinat Sistemi Üzerinde Açısal Gösterimi ..... 27
Şekil 3.11 Örnek Bir Jiroskop Modeli ..... 27
Şekil 3.12 R Vektörünün Koordinat Sistemi Üzerindeki İz Düşümüleri ..... 27
Şekil 3.13 Jiroskop ve İvmeölçer (Accelerometer) Çıkış Sinyalleri ..... 29
Şekil 3.14 Kalman Filtresi Akış Diyagramı ..... 29
Şekil 3.15 Doğru Akım (DC) Motoru ..... 31
Şekil 3.16 Motor Kontrol Kartı (L298 destekli) ..... 33
Şekil 3.17 L298 Motor Sürücü Entegresi ve Bacak Bağlantıları ..... 33
Şekil 3.18 Model Tank Elektronik Kontrol Devresi (Ön ve Arka Yüz) ..... 35
Şekil 3.19 Model Tank Elektronik Kontrol Devresi (Ön Yüz) ..... 36
Şekil 3.20 Model Tank ..... 36
Şekil 3.21 Tank Namlusu Stabilizasyon Sistemi Elektronik Modeli ..... 37
Şekil 3.22 Yazılım Algoritmasına Ait Akış Diyagramı ..... 38

# BÖLÜM 1

# GİRİŞ

Hareket halinde olan bir silah sisteminin değişik/bozuk yol ve arazi şartlarında vuruş yüzdesi yüksek atışlar yapabilmesi için namlunun hedef bölgesinde sabit kalması gerekmektedir. Bunun için silah sisteminin konumunun belirlenmesi, geçen zaman içerisinde konumdaki değişiklikler tespit edilerek namluya ters yönde otomatik hareket verilmesi ve böylece namlunun hedef bölgesinde sabit kalması sağlanmalıdır.

Bu çalışmada bir jiroskop ve ivmeölçer vasıtası ile silah sisteminin konumundaki değişiklikler tespit edilecek, namluya yön veren motorlar aracılığıyla konumda tespit edilen değişikliğin tersi yönde namluya hareket vermesi sağlanacaktır.

Ayrıca namlunun hedef bölgesinde sabit kalması sağlanarak hareketli silah sistemlerinin değişik/bozuk yol ve arazi şartlarında vuruş yüzdesi yüksek olan atışlar yapabilen sistemler geliştirilmesi için otomatik kontrol algoritmalarından yararlanılıp, blok diyagramları oluşturulacaktır.

# BÖLÜM 2

# KAYNAK ARAŞTIRMASI

## 2.1 Tankın Genel Tanımı

**Tank**, ana görevi doğrudan ateş gücünü kullanarak düşman kuvvetlerine saldırmak olan, paletli ve zırhlı bir savaş aracıdır. Tankı diğer savaş araçlarından ayıran en önemli özellikleri ağır bir zırha, yüksek ateş gücüne ve her türlü arazide hızlı gidebilecek şekilde tasarlanmış sürüş takımlarına sahip olmasıdır. Her ne kadar masraflı ve lojistik açıdan desteklenmeleri zor olsa da, karada bulunan hedeflerine saldırma ve piyadelerin moralini çökertme yeteneği sayesinde modern orduların vazgeçilmez bir unsurudur[1].

Bir tankın etkisini belirleyen üç ana özellik vardır: Tankın ateş gücü, hareketliliği ve korunması.

**Ateş gücü**, bir hedefi yenme veya yok etme yeteneğidir. Bununla söylenmek istenen şudur: Bir tankın belirlenen hedefe saldırabileceği maksimum uzaklık, hareket eden hedeflere saldırı yeteneği, birden çok hedefe arka arkaya ateş edebilme hızı ve diğer zırhlı araçlar ile sipere girmiş piyadeyi yenebilme yeteneğidir[1].

Modern tankların ana silahı bir adet büyük çaplı toptur. Tank topları karada kullanılan yüksek kalibreli silahlardandır. II. Dünya Savaşı'ndan sonra tank toplarının kalibrelerinde büyük değişiklikler yapılmamış olsa da modern tank topları teknolojik olarak eski tank toplarından daha üstündür. Günümüzde yaygın olarak kullanılan tank topu boyutları, Batı tankları için 120 mm, Doğu tankları (Eski Sovyet ve Çin) için ise 125 mm'dir. Tank topları değişik özelliklerdeki mermilerle atış yeteneğine sahip olsalar da genel olarak zırh delici ve patlayıcı mermiler kullanmaktadır. Bazı tanklar top namlusundan füze de ateşleyebilmektedir. Günümüzde yaygın olarak kullanılan tank topu namluları yiv-set içermezler[1].

Tanklar, piyadeye karşı ana silahın etkisiz kalabileceği diğer hedeflere veya top mermisinin gereğinden fazla güçlü gelip savurganlık yaratacağı hedeflere karşı ve kısa menzilli savunma için başka silahlar da taşırlar. Bu silahlar, ana silah ile aynı eksene sabitlenmiş olan küçük kalibreli (7,62'den 12,7 mm ye) bir makineli tüfek, yakın çatışma ve sınırlı hava savunması için tavana ya da komutan kubbesine takılı makineli tüfektir[1].

Tank silahları geçmişte basit optik nişangâhlarla nişan alınarak hedefe doğrultulur, rüzgârın etkisi tahmin edilir ve bir gösterge çizelgesinden (nişan aleti içinde bilinen büyüklükteki bir cismi -bu durumda bir tankı- çerçeveleyecek şekilde dizilmiş işaretler topluluğu) yararlanılarak ateş edilirdi. Yani hedefe olan mesafe de gösterge çizelgesi yardımıyla tahmin edilirdi. Sonuç olarak uzun mesafede vuruş olasılığı çok düşüktü ve hareket halindeyken nişan alıp ateş edebilmek büyük oranda imkânsızdı. Çünkü arazide hareket eden bir tank mutlaka sarsılacak ve sürücünün yaptığı manevralar ile nişancının kullandığı tank topu ve nişan aleti bir anda hedeften faklı bir yere dönecektir veya hedef üzerinde kalsa bile sürekli sarsılan ve titreyen bir görüntü oluşturacaktır. Teknolojinin gelişmesiyle birlikte bu nişangâhlar lazer mesafe ölçerlerle değiştirilmiştir[1,2].

Sanayileşmiş ülkelerin çoğu modern tanklarda elektro optik nişan aletleri ve lazer mesafe ölçerler kullansalar bile eski ve az gelişmiş araçlarda hâlâ optik gösterge çizelgeli mesafe ölçerler kullanılmaktadır. Modern tankları daha doğru atış yapar hale getiren bir dizi gelişmiş atış kontrol sistemi vardır. Tank topunu dengede tutmak için jiroskoplar kullanılır. Bilgisayar; çeşitli sensörlerden gelen rüzgâr hızını, hava sıcaklığını, nemi, hedefin hızını, kendi hareketini v.b. verileri toparlar ve bu verilerle uygun yükseklik ve hedef noktasını hesaplar. Ayrıca kızılötesi, gece görüş ya da termik gece görüş sistemleri de modern tanklara ilave edilmektedir. Söz konusu teknolojik yeniliklerin tanklara ilave edilmesi ile birlikte günümüz tankları hareket halinde iken bile uzun mesafelere dahi vuruş yüzdesi yüksek atışlar yapabilmektedir[1,2].

**Hareketlilik** şu hususları kapsar; arazi üzerinde hız ve çeviklik, geçilebilen değişik arazi çeşitleri, geçilebilen engellerin ve siperlerin boyutları, su engelinin derinliği ve genişliği, köprülerden geçebilecek ağırlığı ve genişliği, yakıt ikmali yapılmadan gidilebilecek azami mesafe, hangi araçlarla taşınabileceği, dönüşler ve

manevra kabiliyeti. Savaş alanı hareketliliği ise motor performansı ve hareket yeteneği ile belirlenir ve hızlanma, dik engel aşabilme yeteneği gibi faktörlerle değerlendirilir[1].

Bir tank değişik arazilerde kullanılabilecek, kolay bir hedef olmayacak ve baskın etkisi yaratabilmesi için çok hareketli olabilecek şekilde tasarlanır. Geniş paletleri aracın yüksek ağırlığını daha geniş alana yayarak yere uygulanan basıncı azaltmaya ve zemine daha iyi tutunmaya yaramaktadır. Sorun yaratan arazi tipleri daha çok, ya bataklık gibi yumuşak zemine sahip araziler ya da büyük kayalı taşlıklı arazilerdir. Normal sayılan arazi tiplerinde bir tankın 30 ila 50 km/saat hızla hareket etmesi beklenir. Yol üzerindeki hızı 70 km/saat'e kadar çıkabilir[1].

Geleneksel olarak zırhlı savaş araçlarının hareketliliği aşağıdaki ölçütlerle belirlenir:

- Motor gücü ve torku
- Güç-ağırlık oranı
- Yol ve arazideki hızı (olası değişkenlik göz önüne alındığında geniş aralıkta bir değer)
- Yol ve arazideki menzili
- Zemin basıncı ve ağırlık (köprü sınıflandırması)
- Geçilebilen siper genişliği
- Tırmanılan dik basamak/dik engel
- Tırmanılabilen eğim açısı
- Rahatça geçilebilen yan eğim açısı
- Yerden yükseklik
- Hazırlıklı / hazırlıksız geçilebilen su derinliği

**Korunma**; hangi tür ve ne kadar zırh kullanıldığını, bu zırhın nasıl düzenlendiğini, zırhın yapısını, hangi alanlara daha kalın zırh ve hangi alanlara daha ince zırh konulduğunu ifade etmektedir. Bunun yanında korunma alçak profili, düşük ses ve termal izi, düşman ateşinden kaçınabilmek için aktif önlemleri ya da diğer yöntemleri ve zarar gördükten sonra da savaşa devam edebilme yeteneğini de içerir[1].

Tank tasarımında geleneksel olarak bu üç etmen maksimum düzeyde ve birbiri ile koordineli ve birbirini destekler şekilde kullanılması asıl hedef olmaktadır. Dolayısı ile tank hem çok iyi bir zırh yapısına sahip olacak, hem ağır zırhı ile hareket kabiliyeti yüksek olacak, hem de bu yüksek hareket kabiliyetinin yanında isabetli ve etkili atışlar yapabilecek şekilde tasarlanacaktır. Tanklara; yüksek hızlarda, değişik ve engebeli

arazide hareket ederken uzak mesafelere ve de vuruş yüzdesi yüksek atışlar yapabilme yeteneği namlu stabilizasyon sistemi sayesinde kazandırılmaktadır.

Fransızca kökenli olan "stabilizasyon" kelime anlamı olarak istikrar, dengeleme, sabitleme anlamlarına gelmektedir. Stabilizasyon günümüzde otomotiv, tıp, bilişim, havacılık, denizcilik, savunma teknolojileri gibi çok çeşitli alanlarda kullanılmaktadır. Bunlara örnek vermek gerekirse; elektronik stabilizasyon programı (ESP), hareket algılayıcı sensörlerle desteklenmiş kamera sistemleri, insansız hava araçları, otomatik pilot uygulamaları, kıtalar arası füze sistemleri, uydu sistemleri, hareketli silah platformları gibi pek çok alanda etkin olarak kullanılmaktadır.

**Stabilizasyon** tanklarda Atış Kontrol Sisteminin (AKS) bünyesinde değerlendirilir. Atış kontrol sistemi de dört bölümde incelenir[3]. Bunlar;

1. Nişan Aletleri
    a. Optik nişan aleti (Teleskop)
    b. Elektro optik nişan aleti
    c. Lazer mesafe ölçme sistemi
    d. Termal görüş aleti
    e. Tank komutanı tepe dürbünü
2. Algılayıcılar (Sensörler)
    a. Eğim algılayıcı
    b. Rüzgâr algılayıcı
    c. Isı algılayıcı
    d. Hız algılayıcı
    e. Mesafe algılayıcı
    f. Yan durum algılayıcı
    g. Top yükseliş algılayıcı
3. Top Yöneltme Sistemi
    a. Elle yöneltme sistemi (Yan ve yükseliş)
    b. Güç grubu (Hidrolik / Elektrikli)
    c. Nişancı ve tank komutanı kontrol kolları
4. Merkezi İşlem Birimi (Elektronik Üniteler ve İşlemciler)
    a. Güç grubu elektroniği ve işlemcisi
    b. Atış kontrol elektroniği ve işlemcisi

Yukarıda belirtilen Atış Kontrol Sistemi (AKS) parçalarının tanımı ve açıklamaları şu şekildedir;

1. Nişan Aletleri:
    a. Optik nişan aleti (Teleskop)

Optik nişan aleti genellikle nişancı bölümünün sol önünde topun yanına, tank topu namlusu ile aynı anda hareket edecek şekilde yerleştirilir. Optik nişan aletinin sabit bir büyütmesi olup içerisine her mühimmat tipi için ayrı ayrı taksimat çizilmiştir. Optik teleskopta tahmin edilen ya da ölçülen mesafe değerleri sisteme elle girilir ve bu mesafe bilgileri sayesinde nişan taksimatı alçalış ya da yükseliş alır. Nişancı görmüş olduğu gerçek görüntüyü taksimat üzerine yerleştirmek sureti ile atışını gerçekleştirir. Optik nişan aleti manüel olarak kullanıldığı için günümüz tanklarında atış kontrol sisteminin arızalanması durumlarında yedek nişan aleti olarak işlev görmektedir[1-3].

    b. Elektro optik nişan aleti

Elektro optik nişan aleti tankın esas nişan aletidir. Nişancı bölümünün ön tarafına yerleştirilmiş olup; gözetleme, nişan alma ve mesafe ölçme maksadıyla nişancı tarafından kullanılır. Bazı tank modellerinde tank komutanı tarafından da kullanılabilmektedir[1-3].

Elektro optik nişan aleti olarak adlandırılmasının nedeni; objektifinin üzerine yerleştirilen jiroskop ve motorlar sayesinde elektro optik nişan aletinin de tank topu gibi stabilize edilmiş olmasıdır. Elektro optik nişan aleti stabilizasyon elektroniği ile bağlantılı olarak çalışır[1-3].

Elektro optik nişan aleti tankın esas nişan aleti olduğu için objektifinin korunması ve çalışır durumda tutulması önemlidir. Bu maksatla objektifinin önüne yerleştirilen bir zırh kapağı sayesinde gerekli durumlarda objektifin zarar görmemesi için nişancı tarafından zırh kapağı kapatılmak sureti ile objektif koruma altına alınabilmektedir. Aynı zamanda objektifin kirlenmesi durumunda silecek sistemi ile temizliği sağlanabilmektedir[1-3].

    c. Lazer mesafe ölçme sistemi

Elektro optik nişan aleti üzerine yerleştirilen lazer sistemi sayesinde lazer ile mesafe ölçüp, ölçtüğü mesafeyi atış kontrol işlemcinse göndererek ilave atış değerlerinin otomatik olarak hesaplanmasına katkıda bulunur. Lazer mesafe ölçme

sistemi ile ölçülen mesafe nişan aletinin göz camında gösterilebilmekte ve nişancıya otomatik büyütme sağlamaktadır[3].

d. Termal görüş aleti

Termal görüş aleti genellikle elektro optik nişan aletinin içine yerleştirilmiş olup nişancının göz camı ile bağlantı halindedir. Bu alet ısı farklılıklarını algılama prensibi ile çalışan bir gözetleme ve nişan aletidir. Soğutulmuş ekran üzerine gerçek görüntüden gelen farklı ısılara sahip nesnelerin görüntülerini yansıtır[3].

Gözetlenen arazi ya da nesneden alınan görüntüde ısı değişikliği ne kadar büyük ise görüş de o kadar net olur. Çıplak gözle görülemeyen nesneler termal ekrandan görülebilen bir termal resme dönüşür. Sistemin gerekli algılama hassasiyetine ulaşması için sensörlerin ısıları çok düşük sıcaklıklara (örn.:-200$^0$C) ulaşmış olmalıdır[3].

Nişancı, termal görüş aleti üzerinde bulunan tuşlar vasıtası ile gözetlemek istenilen alanın görüş açısını belirleyebilmektedir. İç taksimatlar bu durumda seçilen görüş alanına göre değişiklik gösterir[3].

Termal görüş aleti gece, görüş şartlarının kısıtlı olduğu gündüz durumlarında ve gizlenmiş hedeflere karşı kullanılabilir. Bazı tank modellerinde tank komutanı tarafından da kullanılabilmektedir[3].

e. Tank Komutanı Tepe Dürbünü

Tank komutanı tepe dürbünü kule tavanına yerleştirilmiştir. Çepeçevre gözetleme, atış ve atış tanzimi maksadıyla kullanılır. Tank komutanın gözetleme ve nişan aleti olup, nişancının elektro optik nişan aletlerinden bağımsız çalışan ancak tank komutanı tarafından istenildiğinde silahları kullanabilme özelliği bulunan, kendi iç taksimatına sahip sağa/sola dönebilen, büyütme özelliğine sahip bir nişan aletidir[3].

Tank komutanı tepe dürbünü objektifi üzerine elektro optik nişan aletlerindeki gibi jiroskop ve motorlar yerleştirilebilir. Bazı tanklarda periskop zırh kapağı ile balistik etkilerden korunur.

2. Algılayıcılar (Sensörler)

Ölçülen ve dolayısıyla denetlenen niceliğin değerini algılayan veya seçen aygıtlardır. Tanklarda atış kontrol sistemini etkileyen birden fazla dış etken olup bunlar algılayıcılar tarafından tespit edilip elektronik üniteler ve işlemcilere gönderilerek hesaplamalarda otomatik olarak kullanılırlar. Bu algılayıcılar ihtiyaca göre tanklarda

çeşitlilik gösterebilir. Örneğin; eğim algılayıcı, rüzgâr algılayıcı, ısı algılayıcı, hız algılayıcı, mesafe algılayıcı, v.s. Bu algılayıcılar daha da geliştirilip çoğaltılabilir[2,3].

3. Top Yöneltme Sistemi
    a. Elle yöneltme sistemi (yan ve yükseliş)

Atış kontrol sisteminin veya sistemin içerisinden herhangi bir grubun arızalanması durumunda tankın savaş dışı kalmaması için nişancı tarafından kule ve tankın namlusu mekanik olarak hedefe yönlendirilebilmelidir. Bunun için yan el pompası ve yükseliş el pompası ile yağ basıncının gücünden istifade ederek (kriko sistemi gibi) kule ve namlu mekanik olarak kontrol edilebilmektedir[4,5].

    b. Güç grubu (Hidrolik / Elektrikli)

Kulenin sağa ve sola hareket etmesi için gerekli gücü sağlayan sistemdir. Tanklarda değişik kullanım şekilleri mevcuttur. Bunlar hidrolik sistemler ve elektrikli sistemlerdir. Güç grubu manüel kullanımlarda nişancının veya tank komutanının kontrol kollarını yönlendirmesi ile devreye girip çalışır. Aynı zamanda stabilizasyon devreye sokulduğunda nişancı yönlendirme yapmasa bile namlunun sürekli olarak hedef bölgesinde sabit kalması için jiroskoplardan alınan verilere uygun olarak otomatik devreye girip çalışır[4,5].

    c. Nişancı ve tank komutanı kontrol kolları

Nişancı kontrol kolları ve tank komutanı kontrol kolları ile güç grubuna komutlar verilerek tank topunun ve kulenin istenilen yere hareket etmesi sağlanır. Nişancı ve tank komutanı kontrol kolları sadece tank topu ve kulenin yönlendirilmesinde kullanılmaz. Üzerlerine yerleştirilen butonlar sayesinde mesafe ölçme, ateş etme, otomatik takip gibi değişik işlevlerin yerine getirilmesinde de kullanılır[4,5].

4. Merkezi İşlem Birimi (Elektronik Üniteler ve İşlemciler)
    a. Güç grubu elektroniği ve işlemcisi

Sistemin manüel olarak çalışması veya tank topunun sürekli olarak hedef bölgesinde kalması gibi isteklerin yerine getirilmesi bu bölümde gerçekleştirilen otomatik kontrol sayesinde olur. Algılayıcılar ve kontrol kollarından gelen veriler işlendikten sonra güç grubu elektronik ünitesine iletilerek burada güç grubu tarafından ihtiyaç olunan gücün üretilmesi sağlanır[3].

b. Atış kontrol elektroniği ve işlemcisi

Nişancının tespit ettiği hedefe daha sağlıklı ve doğrulukla atış yapabilmesi için nişancı tarafından tespit edilemeyen ancak atışı olumsuz etkileyebilecek etkilerin tespit edilip bunların merminin uçuş yoluna yapacağı etkiler belirlenmelidir. Atışı bu olumsuz etkilerden arındırmak içinde mermi uçuş yolu tekrar hesaplanarak mermi ateşlenmeden hemen önce sisteme ilave düzeltmeler verilmelidir. Atış kontrol elektroniği ve işlemcisi, bünyesinde bulunan algılayıcılardan ve nişancının tespit edip kontrol kollarından gönderdiği verileri otomatik kontrol elemanları vasıtası ile işleyerek hedef için gerekli ilave hesaplamaları yapar ve ilgili birime iletir[3-5].

## 2.2 Otomatik Kontrol

### 2.2.1 Genel Bilgiler

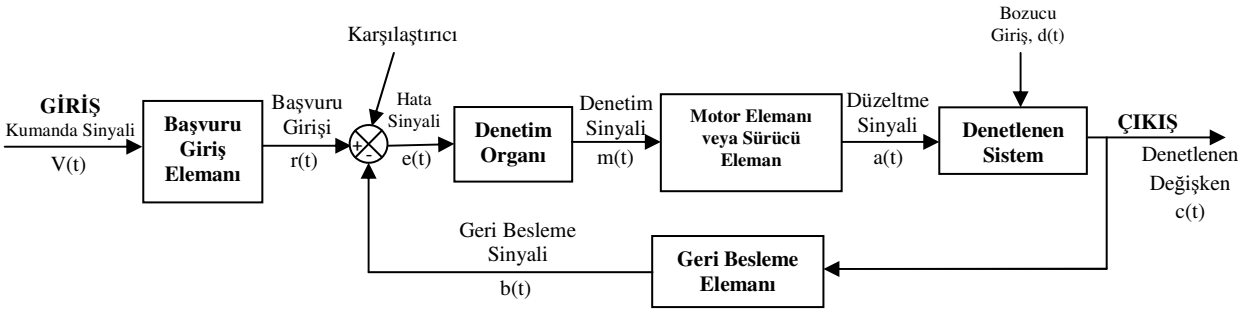
Otomatik denetim sistemleri ya da kısa adı ile denetim sistemleri, çağımızda gelişmiş toplumların günlük yaşamına girmiş ve hemen hemen her alanda kullanılmaktadır. Herkesin kullandığı otomatik çamaşır makinesi, otomatik bulaşık makinesi, termostatlı fırınlar, ütüler, endüstriyel ve araştırma alanında kullanılan robotlar, mikroişlemciler, bilgisayarlar, uzay taşıtları v.b. denetim sistemleri üretim ve üretim kalitesini sürekli olarak geliştirmekte olup, yaşamımıza etki etmektedirler. Denetim sistemleri herhangi bir endüstri toplumunun tamamlayıcı bir parçası olup artan nüfusun sınırsız ihtiyaçlarını üretmek için gereklidirler[6,7].

Denetim sistemleri; en kısa şekli ile enerji, malzeme ve diğer kaynakların akışını düzenleyen aygıtlar olarak tanımlanır. Denetim sistemlerinin düzenlenmesi, sistemin karmaşıklığı, görünüşü, kullanım amaçları ve işlevlerine göre değişiklik gösterir. Denetim sistemleri denetlenen niceliklerin değerlerini sabit tutar veya bu değerlerin daha önceden belirlenmiş biçimde değişimini gerçekleştirir[6,7].

### 2.2.2 Temel Kavramlar ve Tanımlar

Bir denetim sistemi çeşitli elemanların karşılıklı bir şekilde birbirine bağlanmasından meydana gelir ve bu bağlantılar giriş - çıkış yollarıyla gerçekleştirilir. Sistem elemanlarının işlevleri, bireysel giriş ve çıkışları, sistem elemanları arasındaki

bilgi akışı, işlevsel blok şemaları ile gösterilir. İşlevsel blok şemaları; sistemdeki elemanların neden-sonuç ve etki ilişkilerine göre sıralanmalarının yanı sıra sistemin yapısının incelenmesini sağlar. İşlevsel blokların davranışı, giriş çıkış bağlantısı ile belirlenir. Burada "giriş"i neden olarak ele alırsak, "çıkış'"da girişin neden olduğu sonuçtur. Yani giriş - çıkış bağlantısı neden - sonuç davranışı olarak ifade edilir[6,8].



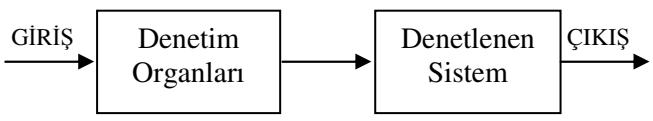

**Şekil 2.1** Temel Tanımları Gösterilen Genelleştirilmiş Geri Beslemeli (Kapalı Döngü) Sistem Blok Şeması

**Blok şeması:** İşlevsel şemadaki elemanların işlevlerini matematiksel ifadelerle gösteren şemalardır.

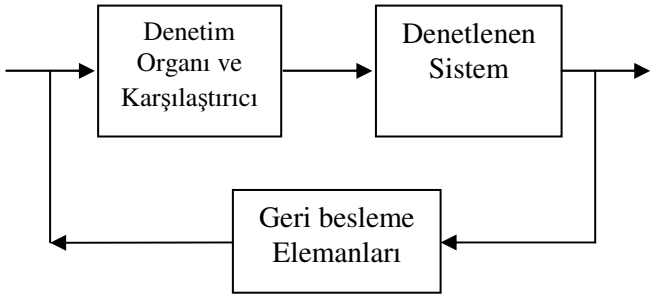
**Sistem:** Genel anlamda; bir araya geldiklerinde bir bütün oluşturacak şekilde karşılıklı olarak birbirine bağlı elemanların tamamıdır. Fiziksel anlamda; bir amacı gerçekleştirmek için bir araya getirilerek düzenlenen ve bütün bir birim olarak hareket etmek üzere birleştirilen etkileşimli ya da ilişkili fiziksel elemanların tamamıdır.

**Denetim (Kontrol):** Bir değişken niceliğin ya da değişken nicelikler kümesinin önceden belirlenmiş bir koşula uyumunu sağlamaya yönelik olarak gerçekleştirilen işlemler bütünüdür.

**Denetim Sistemi:** Kendisini veya diğer bir sistemi kumanda etmek, yönlendirmek veya ayarlamak üzere birleştirilen fiziksel organlar kümesidir. Denetim sistemi, denetlenen niceliklerin değerlerini sabit tutar ya da bu değerleri, önceden belirlenmiş biçimde değişmesini sağlar.




**Şekil 2.2** Açık Döngü Denetim Sistemi




**Şekil 2.3** Kapalı Döngü Denetim Sistemi

**Otomatik Denetim (Otomatik Kontrol):** Bir sistemde denetim faaliyetlerinin insan girişimi olmaksızın önceden belirlenen bir amaca göre denetimi ve yönlendirilmesidir. Otomatik denetim aynı zamanda doğrudan insan girişimi olmaksızın çalışan aygıtların, makinelerin ve sistemlerin çalışması ve gelişmesi ile ilgilenen bilim dalıdır.

**Tesisat veya Denetlenen Sistem:** Amacı özel bir işlemi yerine getirmek olan ve birlikte çalışan makine parçaları bütünü (cihaz), tesisat veya sistem adını alır (Isıtma fırını, kimyasal reaktör, damıtma tesisleri gibi denetlenmesi gerekli fiziksel nesneler).

**Giriş:** Sistemden belli bir cevap almak üzere bir dış enerji kaynağından sisteme uygulanan uyarıdır.

**Çıkış:** Sistemden sağlanan gerçek cevaptır. Çıkış, girişin öngördüğü cevaba eşit olabilir veya olmayabilir.

**Ayar Noktası ve Ayar Değeri:** Denetim sisteminde sabit bir kumanda değerinin ayarlandığı nokta ve ayarlanan değer. Örneğin sıcaklık denetim sisteminde arzu edilen sıcaklığın ayarlandığı nokta bir ayar noktası, ayarlanan sıcaklık ise ayar değeridir.

**Kumanda Girişi veya Arzu Edilen Giriş; v(t):** Sisteme uygulanan sevk edici giriş olup sistemin çıkışından bağımsızdır.

**Başvuru Giriş Elemanı:** Başvuru giriş değerini saptayan birimdir. Başvuru giriş elemanı sistem çıkışının arzu edilen birimleri cinsinden kalibre (ayar) edilir.

**Başvuru Girişi; r(t):** Denetlenen sistemin belirlenen bir eylemi kumanda etmek üzere denetim sistemine uyguladığı giriş sinyalidir. Başvuru giriş elemanının sağladığı bu sinyal sistem tarafından doğrudan doğruya kullanılabilir biçimde bir kumanda olarak ifade edilir. Denetim sistemi için gerçek sinyal giriş olup çoğunlukla ideal sistemin çıkış davranışını temsil eder.

**Karşılaştırıcı veya Hata Seçici:** Başvuru giriş sinyali ile geri besleme sinyalini karşılaştırıp mukayese eden ve bu iki sinyal arasındaki farka eşit bir hata sinyali üreten

elemandır. Karşılaştırıcı aynı boyut ve birimlerdeki sinyalleri karşılaştırarak onların cebirsel toplamını alır.

**Hata ve Sapma; e(t):** Çıkışın herhangi bir anda, arzu edilen değere göre farkına **hata** denir. Hata sinyali, başvuru girişi ile geri besleme sinyali arasındaki farka eşittir. Karşılaştırma elemanı, çıkışı arzu edilen değerle karşılaştırarak hata değişimlerini belirler. Hata sinyali, sistemin çıkışından arzu edilen değeri sağlamak üzere denetim organını hareket ettirir. Denetim organı bu değişimlerini giriş olarak alır ve kendi yapısına da bağlı olarak son denetim organı (motor eleman) için uygun bir denetim sinyali üretir. Denetlenen değişkenlerin belirli değerler etrafında değişimleri **sapma** olarak ifade edilir.

**Denetim Organı (Kontrol Edici):** Denetlenen sisteme uygulanacak uygun bir denetim sinyali sağlayan elemandır.

**Son denetim Elemanı veya Motor Eleman:** Denetim organından alınan sinyale göre belli bir fiziksel yapıda güç sağlayan elemandır. Bu eleman denetlenen sistemde meydana gelen hatayı veya sapmayı düzeltmek için gerekli hareketi sağlayan bir elemandır.

**Denetim Sinyali; m(t):** Denetim elemanları grubunun denetlenen sisteme uyguladıkları nicelik veya koşuldur. Bu sinyal, denetim organı çıkışında denetim sinyali olarak son denetim elemanına gönderilir. Burada yeteri kadar kuvvetlendirilerek denetlenen sistemin denetlenen değişkenini değiştirecek şekilde bir düzeltme işlemi meydana getirir. Fakat esas denetim, denetim organı tarafında üretilir.

**Denetlenen Sistem:** Özel bir niceliğin denetlendiği tesisat, süreç veya bir makinedir.

**Çıkış veya Denetlenen Değişken; c(t):** Denetlenen sistemin niceliği veya koşuludur. Bir niceliğin, sistemin bozucu girişlerinden etkilenmeksizin kumanda girişini izleyecek şekilde önceden tanımlanan bir değerde sabit tutulması gerekir.

**Bozucu Girişler; d(t):** Sistemin denetlenen çıkışı üzerinde arzu edilmeyen yönde etki yapan girişlerdir. Eğer bozucu etkiler sistem içinden meydana geliyorsa **iç bozucular**, sistem dışından geliyorsa **dış bozucular** adını alır ve her ikisi de sistem için bir giriştir. Bozucu giriş, denetim sistemi döngüsünde her hangi bir noktadan etki edebilir.

**Geri Besleme Sinyali; b(t):** Denetlenen değişkenin bir fonksiyonu olup, başvuru girişi ile karşılaştırılarak hata sinyalinin elde edilmesini sağlar.

**Geri Besleme Elemanı:** Denetlenen çıkış sinyali ile geri besleme sinyali arasında işlevsel bağlantı kuran elemandır. Geri besleme elemanları özellikle denetlenen değişken ile başvuru giriş sinyalinin farklı fiziksel yapıda olduğu durumlarda bir dönüştürücüden (transducer) ibarettir. Geri besleme elemanı bir ölçü elemanı biçimindedir.

**Geri Besleme Yolu:** Denetlenen çıkış sinyalinden geri besleme sinyaline kadar uzanan iletim yoludur.

**İleri Besleme Elemanları:** Arzu edilen çıkışı sağlamak üzere hata sinyaline tepki gösteren birimdir.

**İleri Besleme Yolu:** Hata sinyalinden denetlenen çıkış sinyaline kadar uzanan iletim yoludur.

**Açık Döngü Denetim Sistemi:** Denetim faaliyetinin denetim sistemi çıkışından bağımsız olduğu sistemdir. Açık döngü denetim sisteminde arzu edilen çıkışın denetlenen çıkış değişkeni ile hiçbir karşılaştırması yoktur (Şekil 2.2).

**Kapalı Döngü Denetim Sistemi:** Denetim faaliyetinin denetim sistemi çıkışına bağımlı olduğu sistemdir (Şekil 2.3).

**Geri Beslemeli Denetim:** Denetlenen çıkış değerlerinin arzu edilen giriş değerleri ile karşılaştırıldığı kapalı döngü bir denetim sistemidir. Sistem çıkışı arzu edilen çıkış değeri ile karşılaştırılıp oluşan hata sinyalleri tespit edilir. Sistem çıkışı, arzu edilen çıkış değerini sağlayabilecek şekilde değişkenler üzerine etkili olan bir denetim sinyali uygulanır. Geri besleme etkisi negatif geri besleme ve pozitif geri besleme olmak üzere ikiye ayrılır. Negatif geri besleme çıkışın girişe ters yönde etkidiği, pozitif geri besleme de çıkışın girişe aynı yönde etkidiği sistem olarak tanımlanır.

**Düzenleyici Denetim:** Başvuru girişinin uzun zaman aralıkları içerisinde belli bir çalışma koşulu için, değişmez veya sabit tutulduğu geri beslemeli denetim sistemidir. Düzenleyici denetimde sisteme etki eden bozucu girişlere rağmen sistem çıkışının arzu edilen değerde tutulması esastır. Bu tür denetimde sabit bir sinyal (ayar değeri) girişi ile sabit kalıcı durum çıkışı elde edilir.

**Servomekanizma veya Servo:** Çıkışı mekaniksel konum, hız veya ivme olan geri beslemeli denetim sistemidir.

**Süreç Denetim Sistemi:** Çıkışı sıcaklık, basınç, akış, seviye ve pH gibi değişkenler olan düzenleyici denetim türündeki geri beslemeli denetim sistemidir.

**Uzaktan Kumanda:** Bir sistemde donanımı, aygıtları ve işlemleri belli bir uzaklıktan çalıştırma imkânı sağlayan denetim türüdür. Bu tür denetimde gösterge, kaydetme ve denetim aygıtları denetlenen sistemden birkaç yüz metre mesafeye yerleştirilebilir. Böylece bir sistemde kumanda odasından denetlenen işleme veya sisteme ve sistemden kumanda odasına sinyal aktarımı ya pnömatik ya da elektriksel biçimde gerçekleşmelidir.

**Duyarga (Sensör/Algılayıcı):** Ölçülen ve dolayısıyla denetlenen niceliğin değerini algılayan veya seçen bir aygıttır. Duyarga eleman veya algılama organı ölçme sistemi zincirlerinin ilk halkası olup duyarga ifadesi bazen sezici veya dönüştürücü yerine de kullanılmaktadır.

**Dönüştürücü (Transducer):** Genel olarak herhangi bir enerji biçimini diğer bir enerji biçimine dönüştüren aygıt olarak tanımlanır. Örneğin; mekaniksel enerjiyi elektriksel enerjiye dönüştüren potansiyometreler, ışık enerjisini elektriksel enerjiye dönüştüren fotoseller, v.s. Özellikle; ölçme sistemi açısından bir enerji dönüştürme aygıtı olarak çalışan dönüştürücü, uyarıyı fiziksel ortamdan alır ve bu uyarıyı bir ölçme sistemi girişine göre daha uyun olan bir sinyale dönüştürür.

**Robot:** İnsan müdahalesi gerektirmeden otomatik olarak çalışmak üzere tasarlanmış programlı bir aygıttır. Endüstriyel bir robot bir insanın bel, omuz, dirsek ve bilek hareketlerine denk serbestlik dereceleri içeren bir kol ve elden ibaret olup, ulaşma mesafesi içerisindeki herhangi bir noktaya uzanabilir ve iş parçalarını veya aletleri kolayca yakalayabilir. Robot, endüstride özellikle malzeme aktarma işlevlerinin yerine getirilmesinde çok kullanılır. Robotlar endüstriyel uygulamalarda büyük bir potansiyel arz etmekle beraber günümüzde daha çok, makinelerin yüklenmesi ve boşaltılması, parçaların konumlandırılması, aktarılması, kaynak, boyama (otomobil endüstrilerinde) işlemlerinde yaygın olarak kullanılmaktadırlar.

### 2.2.3 Denetim Sistemleri ve Türleri

Denetim sistemlerinin türü ne olursa olsun mutlaka denetleyen ve denetlenen olmak üzere iki temel unsuru vardır. Bu anlamda üç temel denetim sistemi vardır[7,8]:

i. Doğal denetim sistemi,
ii. Endüstride kullanılan insan yapısı denetim sistemi,
iii. Hem doğal ve hem de insan yapısı unsurlar içeren karma denetim sistemi.

Denetim sistemleri denetim etkisi açısından ise iki ana sınıfa ayrılır[7,8]. Bunlar;

i. Açık döngü denetim sistemi,
ii. Kapalı döngü denetim sistemidir.

(i) Açık döngü denetim sisteminde; denetim eylemi sistem çıkışından bağımsızdır. Açık döngü sistemde çıkışın ölçülmesi ve geri beslemesi söz konusu değildir. Dolayısıyla sistemin girişi çıkış bilgisinden haberdar olmaz. Açık döngü denetim sistemi giriş-çıkış bağlantıları önceden belli olan ve iç veya dış bozuculara maruz kalmayan sistemlerde kullanılır. Çıkış ve girişin bir karşılaştırması yapılmadığı için sistemin çalışma doğruluğu yapılan kalibrasyon (ayarlama) derecesine bağlıdır. Açık döngü denetim sistemleri ya zaman esasına göre ya da sıralama esasına göre çalışır. Örneğin, otomatik çamaşır makinesinde olduğu gibi, sistem girişi bir program şeklinde verilir ve sistem program sırasını izler[7,8].
(ii) Kapalı döngü denetim sisteminde; denetim etkisi sistem çıkışına bağlıdır. Sistemin çıkışı ölçülüp geri besleme değerleri elde edildikten sonra arzu edilen giriş değeri ile karşılaştırılır. Böylece sistemin girişi çıkıştan haberdar olur. Sistem çıkışından elde edilen geri besleme değerlerinin girişe uygulandığı bu tür kapalı döngü denetim sistemine aynı zamanda **geri beslemeli sistem** de denir. Açık döngü sistem ile kapalı döngü sistemi birbirinden ayıran en önemli unsur geri besleme etkisidir. Geri besleme etkisi de **negatif geri besleme** ve **pozitif geri besleme** olmak üzere ikiye ayrılabilir[7,8].

Negatif geri beslemede, çıkıştaki gelişmeler girişe ters yönde etki eder. Böyle bir sistemde çıkış, arzu edilen değere göre bir artış gösterecek olursa denetim etkisi azaltılarak çıkışın istenen değere geri dönmesi sağlanır. Aksi bir durumda, eğer çıkış arzu edilen değere göre bir azalma (negatif değişme) gösterirse denetim etkisi artırılarak çıkışın istenen değere yükselmesi sağlanır. Negatif geri beslemede daima giriş ile çıkışın bir farkı alınır ve bu fark negatif veya pozitif değerli olabilir. Denetim organına bir hata girişi olarak iletilen bu değer, yukarıda da açıklandığı gibi, çıkışın istenen değere getirilmesini ve bu değerde sabit tutulmasını sağlar. Negatif geri besleme endüstriyel sistemlerin en önemli özelliğidir ve daima hatayı en küçük değerde tutmaya veya sıfır yapmaya çalışır[7,8].

Pozitif geri beslemede, çıkış girişe aynı yönde etki eder. Buna göre çıkışta herhangi bir artış meydana gelecek olursa bu giriş ile toplanarak hata sinyalinde bir artış

ve dolayısıyla da denetim sinyalinde bir artış meydana getirir. Bu, sistemde çıkışı daha da artıracak yönde bir etki yaratır. Sonuçta artış sistemin fiziksel sınırlamalarına kadar devam eder ve sistem denetlenebilirliğini kaybeder. Pozitif geri besleme, iç döngüler hariç, bir kapalı döngü denetim sisteminde kullanılmaz[7,8].

Denetim sistemleri uygulama alanları ve çalışma biçimlerine göre;

-Düzenleyici denetim,

-İzleyici denetim,

-Servomekanizma denetim gibi isimler de alırlar. Endüstriyel alanda kullanılan bu sistemlerin tümü negatif geri beslemelidir.

Düzenleyici denetim, sabit bir ayar değeri esasına göre çalışan ve sistem çıkışını, tüm bozucu girişlerin varlığına rağmen sabit değerde tutmaya çalışan denetim sistemidir. Sıcaklık, seviye, debi, basınç v.b. değişkenlerin denetiminde kullanılan bu çalışma biçimi yaygın olarak süreç endüstrilerinde kullanılır[6].

İzleyici denetimde giriş değişken olup, sistem çıkışı bu girişi izlemeye çalışır. Burada sistem çıkışının hem başvuru girişi ve hem de bozucu girişten doğan değişmeleri izlemesi ve arzu edilen değere getirilmesi esastır. Bu tür denetim daha çok imalat endüstrisinde kullanılır. Örneğin; takım tezgâhlarında kesici bıçağın bir şablonu izleyerek istenen parça biçimini işlemesi bu türden çalışma olur. İzleyici denetim daha çok konum denetim sistemi şeklindedir[6].

Servomekanizma, mekaniksel çıkışlı güç yükseltilmesi gerekli sistemlerde kullanılır. Yerine göre izleyici türde, yerine göre de düzenleyici türde çalışabilir. Örneğin, büyük güç yükseltmesi gereken konum servomekanizması izleyici türde, buna karşılık bir hız denetim servomekanizması düzenleyici türde denetim çalışması gösterir. Endüstriyel alanda ise örneğin, sıcaklık yanında, basınç, debi, seviye gibi diğer değişkenlerin denetimi de gerekebilir. Böyle bir durumda tek girişli – tek çıkışlı bir denetim sistemi yeterli olmaz. Modern endüstriyel sistemler daha çok çok girişli – çok çıkışlı sistemlerdir ve hatta bu giriş ve çıkışlar arasında karşılıklı etkileşimler de mevcut olabilir. Bu tür denetim sistemlerinin temel yapısı yine de geri beslemeli biçimde olmakla beraber sistem içinde birden fazla döngü yer alır. Çeşitli giriş ve çıkışlar arasında en uygun denetim sinyalini sağlamak için daha çok bilgisayarlardan ve mikroişlemcilerden yararlanılır[6].

### 2.2.4 Denetim Sistemlerinin Yapısı ve Kullanım Alanları

Endüstriyel anlamda bir denetim sistemi;

i. denetlenen sistem veya süreç,
ii. denetim elemanları donanımı olmak üzere iki ana kısımdan ibarettir.

Denetim organları donanımı ise;

i. ölçme sistemi,
ii. karşılaştırma elemanı,
iii. denetim organı,
iv. sürücü veya kumanda elemanından meydana gelir.

Denetlenen sistem ve denetim elemanları donanımı birbirlerine karşılıklı olarak bir takım hatlarla bağlanmıştır. Bunlar;

i. süreç ürünlerinin aktığı enerji ile ilişkili olanı, süreç hattı,
ii. ölçme ve denetim sinyallerinin aktığı hat, bilgi veya sinyal hattı,
iii. denetim elemanları ve sistem çalıştıran cihazlar için gerekli gücün aktığı, güç hattı isimlerini alırlar.

Genellikle karşılaştırma elemanını da içine alan denetim organı; sistem çıkışının denetim amaçlarına uygunluğunun saptanması ve sistemi denetim amaçları doğrultusunda yönetmek için gerekli denetim değişimlerini belirlemek gibi işlevleri yürütür. Denetim organının temel görevi karşılaştırıcıdan kendisine hata veya sapma olarak iletilen giriş bilgisine göre denetim sinyali şeklinde uygun karar üretmektir. Denetim organı fiziksel yapısı günümüzde daha çok elektronik ve pnömatik olmaktadır. Elektronik olarak benzeşik (analog) devrelerden meydana gelebildiği gibi sayısal (dijital) devrelerden de meydana gelebilir. Sayısal devrelerden (mikroişlemci) meydana gelmiş denetim organı aynı zamanda gerekli hesaplamaları yapabilme yeterliliğine sahiptir[6].

Son denetim organı veya motor eleman adlarını da alan kumanda elemanı, denetlenen sistemi doğrudan etkileyen sürücü bir elemandır. Son denetim elemanı denetim organından aldığı denetim sinyaline bağlı olarak dış enerji kaynağından aldığı güçle bir hareket üretir. Bu hareket denetlenen değişkenin enerji akışını modüle eder. Kullanılan enerjinin biçimine bağlı olarak motor elemanı, elektriksel, pnömatik veya hidrolik yapıda olabilir. Uygulamada genellikle motor elemanın yapısına bağlı olarak denetim elemanları da elektriksel, pnömatik veya hidrolik denetim elemanı isimlerini

alırlar. Çok küçük hacimlerde büyük güçler sağlayan hidrolik sistemlerin özel kullanım alanları mevcuttur. Elektrik enerjisinin her yerde kolaylıkla sağlanabilir olması ve elektronik teknolojisindeki gelişmeler günümüzde elektriksel sistemleri çok yaygın olarak kullanılır hale sokmuştur. Pnömatik sistemler, özellikle yangın tehlikesi olan yerlerde kullanılmaktadır. Ayrıca bu sistemlerin bileşimi olan karma sistemler de yaygın olarak kullanılmaktadır. Bunlardan çoğunlukla denetim sistemi döngüsünün ölçme sistemi ve karşılaştırma elemanları elektrik-elektronik olarak çalışmakta, motor elemanı ise elektriksel pnömatik veya hidrolik yapıdadır[6].

### 2.2.5 Otomatik Denetim Sistemi Uygulamaları

Doğada, çevremizde, günlük işlerimizde ve endüstriyel alanda kullandığımız sınırsız sayıda denetim sistemi örneklerine rastlamak mümkündür. Doğadaki örneklerini, ekolojik denetim sistemi, kan basıncı denetim sistemi, solunum denetim sistemi, adrenalin denetim sistemi v.s., çevremizdeki örneklerini, ekonomi denetim sistemi, sosyoloji denetim sistemi v.s., günlük işlerimizdeki örneklerini, ütü, buzdolabı, banyo sıcak su kazanı gibi yerlerde kullanılan sıcaklık denetim sistemleri ve endüstriyel alandaki örneklerini ise, süreç ve imalat endüstrilerinde kullanılan sıcaklık, basınç, debi, pH, konsantrasyon, konum, hız v.b. denetim sistemleri olarak sayabiliriz[7,8].

Doğada bulunan ve endüstride kullanılan denetim sistemlerinin çok büyük bir çoğunluğu kapalı döngü geri beslemeli denetim sistemleri teşkil eder. Açık döngü denetim sistemleri uygulamalarının sayısı ise oldukça sınırlıdır. Evlerimizde kullandığımız tam otomatik çamaşır makinesi, trafik akışını düzenleyen trafik lambaları, imalat endüstrisinde kullanılan sıralama denetim sistemleri v.b. açık döngü denetim sistemi örnekleri olarak ele alınabilir. Tam otomatik çamaşır makinesi kendisine program olarak sunulan önceden saptanmış belli bir işlem sırasına göre çalışır. Amacı belli bir temizlikte yıkanmış çamaşır elde etmektir. Kendisine belli bir program şeklinde verilen ayar değerine, yıkama, durulama, yumuşatma, sıkma v.b. işlemler sırasıyla yerine getirir ve sonuçta çıkış olarak belli temizlikte yıkanmış çamaşır elde edilir. Her ne kadar yıkama işlemi sırasında kullanılan suyun ısıtılması termostatik bir düzenekle geri beslemeli olarak yerine getirilmiş olsa da, sistemin esas giriş olan çamaşırların arzu edilen temizlik derecesi ile sonuçta gerçekleşen temizlik derecesinin işlem sırasında bir karşılaştırması yoktur. Elde edilen temizlik derecesi arzu ettiğimiz temizlik derecesine

eşit olabilir veya olmayabilir. Dolayısıyla sistem denetim eylemi açısından açık döngü denetim tipindedir. Trafik ışıkları yoluyla yapılan trafik denetim sistemi de açık döngü şeklinde çalışır. Ortalama trafik yoğunluğuna göre belli bir merkezden zaman esasına göre aylanan trafik ışıklarının sırasıyla yanıp sönmesi trafiğin sıkışmadan akmasını sağlar. Fakat yeşil ışığın yanma süresi her zaman bekleyen trafiğin tamamının geçmesini sağlamayabilir. Çünkü burada arzu edilen trafik akışının (girişin) gerçekleşen trafik akışı (çıkış) ile bir karşılaştırması yoktur[7,8].

# BÖLÜM 3

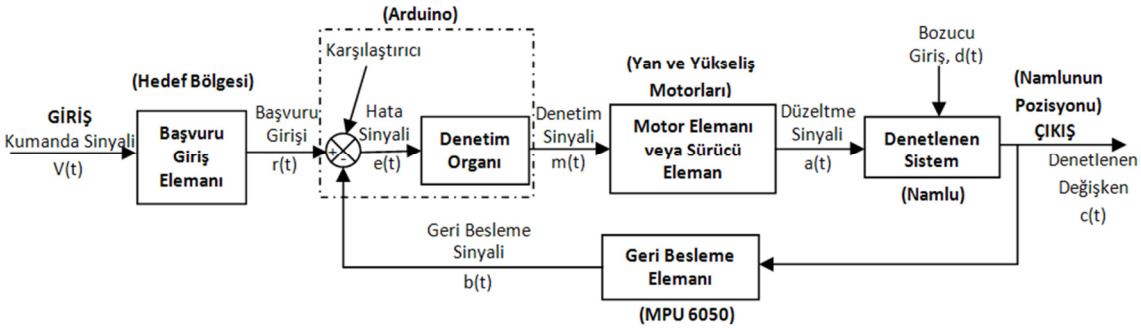
# MATERYAL VE TEORİ

## 3.1 Tank Namlusu Stabilizasyon Sisteminin Tasarımı

Tank namlusu stabilizasyon sistemi aslında bir kapalı döngü otomatik kontrol sistemidir. Nişancı tank topu namlusunu hedef bölgesine getirdikten sonra tank gövdesinin yön değişiklikleri ve bozuk/engebeli arazi şartları nedeni ile namluda hedef bölgesinden sapmalar meydana gelmektedir. Otomatik kontrol sistemi kullanılarak bu sapmaların giderilmesi sağlanır. Otomatik kontrol sistemi sensörlerden gelen tank gövdesi konumunu algılar ve mikroişlemcisi ile yapmış olduğu hesaplamalar sonucunda sapmaları gidermek üzere çıkış sinyalleri üretir. Bu sinyaller namluya yön veren sistemlere iletilir ve sapmaların tersi yönünde hareketle tank topu namlusunun hedef bölgesinde kalması sağlanır. Burada bilinmesi gereken husus otomatik kontrol sisteminin referans olarak tankın gövdesini almasıdır. Yani stabilizasyon sistemi bir hedef takip sistemi değil sadece namlunun hedef bölgesinde kalıp, arazi şartlarından en az etkilenerek nişancının nişan almasına ve vuruş yüzdesi yüksek atışlar yapabilmesine yardımcı olan bir sistemdir[4,5,9].

Yapılan çalışmada oluşturulmuş olan kapalı döngü otomatik kontrol sisteminde konum algılayıcı olarak MPU-6050 sensörü, işlemci olarak Arduino Uno, motor kontrol kartı ve eyleyici olarak doğru akım (DC) motorları kullanılmıştır. MPU-6050 sensörü bünyesinde barındırdığı ivmeölçer (accelerometer) ve jiroskop sayesinde tank topu namlusu için tanımlanan referans noktasına göre x, y ve z eksenlerinde yapmış olduğu açısal değişiklikleri algılar. Elde edilen veriler $I^2C$ iletişim protokolü kullanılarak karşılaştırıcı ve denetim organı olarak kullandığımız Arduino Uno mikroişlemci kartına iletilir. ATmega328 işlemci kullanan mikroişlemci kartı kendisine ulaşan verileri daha önceden tanımlanan yazılım algoritmasına uygun olarak değerlendirir ve tank topu

namlusu için tanımlanan referans noktasına namluyu tekrar yöneltmek için eyleyicilere motor kontrol kartı aracılığı ile gerekli sinyalleri göndermeye başlar. Eyleyici olarak kullanılan doğru akım motorları mikroişlemci kartından gelen sinyallerle tank topu namlusunu tanımlanan referans noktasında tutacak şekilde harekete başlarlar ve namlu tanımlanan referans noktasına ulaştığında mikroişlemci tarafından gönderilen sinyaller kesilir. Burada otomatik kontrol sistemi tarafından uygulanan işlem kısaca MPU-6050 sensörü ile tespit edilen açısal değişikliklerin aksi yönünde ve aynı miktarda doğru akım motorlarına hareket vermektir. Tüm bu otomatik kontrol sisteminin blok diyagramı Şekil 3.1'de gösterilmektedir.




**Şekil 3.1** Tank Namlusu Stabilizasyon Sistemi Blok Diyagramı

## 3.2 Tank Namlusu Stabilizasyon Sisteminin Tasarımında Kullanılan Malzemeler

### 3.2.1 Arduino Uno Mikroişlemci Kartı

Arduino Uno ATmega328 işlemci kullanan mikroişlemci kartıdır. 14 adet dijital giriş/çıkış pini bulunur, bunlardan 6′sı PWM çıkışı olarak kullanılabilir. Ayrıca 6 analog giriş pinine sahiptir. 16 MHz kristal osilatörü, bir USB bağlantısı, bir güç girişi, bir ICSP başlığı ve reset düğmesi vardır. Arduino Uno, USB bağlantısı üzerinden veya harici güç kaynağı (2.1mm'lik AC -DC adaptörü veya pil gücü) ile güç alabilir. Güç kaynağı otomatik olarak seçilir. Bu kart mikroişlemciyi desteklemek için gerekli her şeyi içermektedir[10-12].

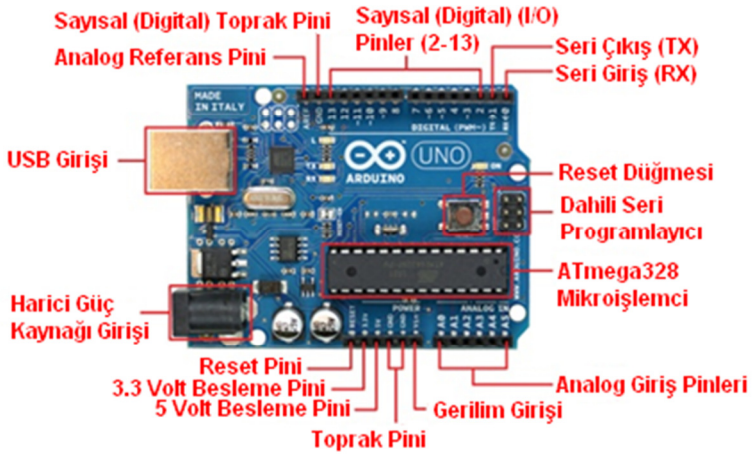


**Şekil 3.2** Arduino Uno Mikroişlemci Kartı (Ön ve Arka Yüzü)

ATmega328, dijital 0 (RX) pini ve 1 (TX) pini üzerinden UART TTL (5V) seri iletişimi desteklemektedir. Kart üzerindeki ATmega16U2 kanalları ile seri iletişim bilgisayarda sanal COM port olarak görünen USB bağlantı noktası üzerinden yapılmaktadır. ATmega16U2 standart USB COM sürücülerini kullanır ayrıca sürücü gerektirmez. ATmega328 aynı zamanda $I^2C$ (TWI) ve SPI iletişimi destekler. Arduino Uno üreticinin sağladığı "Arduino" isimli ücretsiz yazılım ile kolayca programlanabilmektedir. Arduino yazılımı $I^2C$ kanalının kullanımını kolaylaştıran bir "Wire" kütüphanesi içermektedir[10,11].

**Donanımsal Özellikler:**

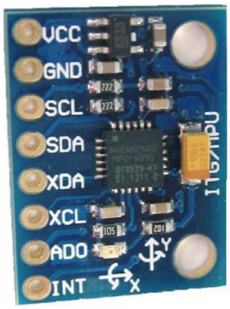
Mikro denetleyici : ATmega328
Çalışma gerilimi : 5V
Tavsiye edilen giriş gerilimi : 7-12V
Giriş gerilimi için alt ve üst sınırlar: 6-20V
Dijital giriş/çıkış pinleri : 14 pin (6'sı PWM)
Analog giriş pinleri : 6 pin
Giriş/çıkış pini başına akım : 40mA
Giriş/çıkış pini başına akım (3.3V): 50mA
Flash : 32KB (ATmega328) (0.5KB bootloader için kullanılır)
SRAM : 2KB (ATmega328)
EEPROM : 1KB (ATmega328)
Hızı : 16MHz




**Şekil 3.3** Arduino Uno Mikroişlemci Kartı

### 3.2.2 MPU-6050 Sensörü

MPU-6050 sensörü bir çip içerisinde bir adet ivmeölçer (accelerometer) ve bir adet jiroskoptan oluşmaktadır. Aynı çip üzerinde 3 eksenli jiroskop ve 3 eksenli ivmeölçer sensörü birleştirerek yerleşik Dijital Hareket İşlemci™ (DMP™) ile birlikte her hangi karmaşık 6-eksenli hareket füzyon algoritmaları işleyebilmektedir[13].

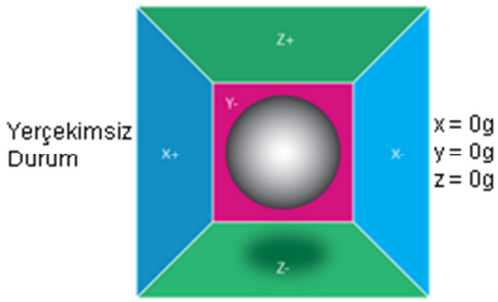



**Şekil 3.4** MPU-6050 Sensörü

MPU-6050 sensörü içerisinde bulunan ivmeölçer, kendisi üzerlerine düşen yerçekimi kuvveti (statik) ve aniden hızlanma veya durma (dinamik) ivmesini ölçmektedir. Sensörden aldığımız değer yer çekimi (g) veya $m/s^2$ türünden ifade edilebilir. Eğer uzayda değilseniz veya herhangi bir çekim alanının kapsamında değilseniz yeryüzünde sensör üzerine 1g lik bir yerçekimi kuvveti etki etmektedir. Yerçekimi kuvveti yaklaşık olarak $9.8m/s^2$ dir ve dünyada bulunduğunuz noktaya göre değişiklik göstermektedir. Yer çekimi etkisi sürekli olduğu için ve sensör sürekli olarak bu etki altında kaldığından eğimölçer veya hareket algılayıcı olarak kullanılabilmektedir. İvmeölçerin bir, iki ve üç eksende ölçüm yapabilen türevleri vardır. MPU-6050 sensörü üç eksende ölçüm yapabilmektedir[14].

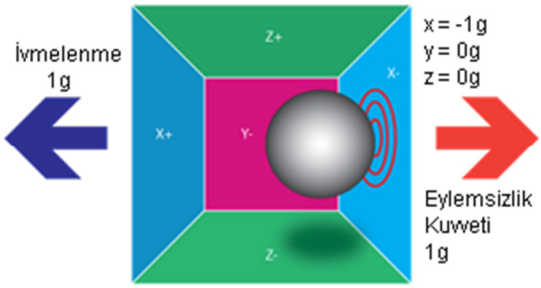
İvmeölçer (accelerometer)'in çalışma prensibi şu şekildedir;

Uzayda herhangi bir çekim etkisine maruz kalmayan ve dolayısıyla ağırlıkları 0 olan bir kutu ve küre ele alınıp, Şekil 3.5'te olduğu gibi küre kutunun ortasına yerleştirildiğinde herhangi bir çekim etkisi olmadığı için küre hiç bir yüzeye temas etmeden hareketsiz bir şekilde duracaktır. Kürenin kutu içerisindeki hareketini görebilmek için kutuya +Y yönünden bakılmalıdır[15].



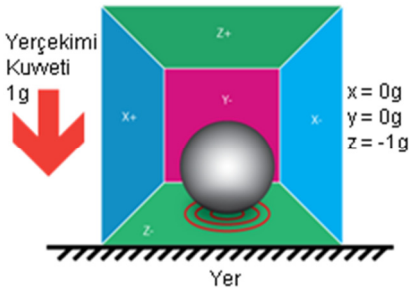

**Şekil 3.5** İvmeölçerin Yerçekimsiz Ortam Modellemesi

Örnekteki kutu Şekil 3.6'da olduğu gibi +X yönünde 1g kuvveti ile hareket ettirildiğinde küre kutunun –X yüzeyine eylemsizlikten dolayı 1g'lik bir kuvvet uygulayacaktır.




**Şekil 3.6** İvmeölçerin Yerçekimsiz Ortamda Hareket Modellemesi

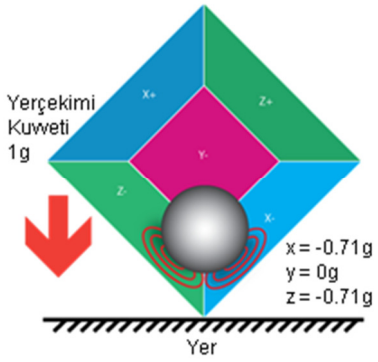
Örnekteki kutu ve küre dünyada Şekil 3.7'de olduğu gibi iç içe konulup yere bırakıldığında dünyanın 1g lik yer çekimi kuvvetinden dolayı küre kutunun –Z yüzeyine 1g kuvvet uygular.




**Şekil 3.7** İvmeölçerin Yerçekimi Ortamında Modellemesi

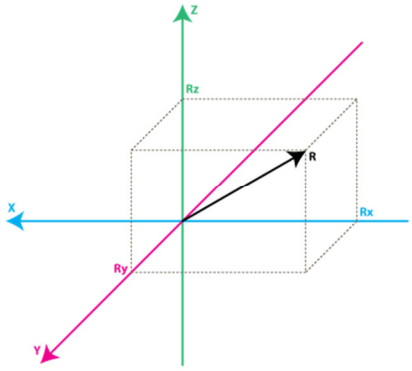
İvmeölçer yukarıda anlatılan örneğe benzer şekilde çalışmaktadır. Yüzeyleri basınca veya manyetik alana tepki verecek şekilde üretilmiştir ve ivmeölçer bu tepkiyi ölçerek bir değer üretmektedir. Sensörün yeryüzü ile yaptığı açının değişmesiyle

sensörün eksenlerine uygulanan kuvvet de değişecektir ve yeni değerler okunarak yeryüzü ile oluşan yeni açı trigonometri yardımıyla hesaplanabilir. Örneğin Şekil 3.8'de olduğu gibi kutunun 45 derece sağa doğru çevrilmesi durumunda kürenin –X ve –Z yüzeylerine kök içinde ½ lik bir kuvvet uygulanır ve bu değer 0,707g ye eşittir.




**Şekil 3.8** İvmeölçerin Yerçekimi Ortamında $45^0$ Sağa Doğru Çevrilmesi

Yukarıda örneklendirilen kutu modeli aşağıda koordinat sistemi üzerinde örneklendirilmiştir. Bu örnekte R vektörü ivmeölçer üzerine düşen kuvvet vektörüdür. Bu kuvvet yerçekimi veya sensörün hareketi sonucu yerçekimi kuvveti ile eylemsizlik kuvvetinin bileşkesi olabilir. Şekil 3.9'da olduğu gibi R vektörünün 3 bileşeni vardır ve bunlar R=[Rx,Ry,Rz] olarak ifade edilir. Pisagor teoremi ( $R^2 = Rx^2 + Ry^2 + Rz^2$ ) ile R vektörü bu bileşenlerden hesaplanabilir.




**Şekil 3.9** R Vektörünün Koordinat Sistemi Üzerinde Gösterimi

Rx, Ry ve Rz bileşenleri bilindiğinde trigonometrik fonksiyonlar yardımı ile R vektörünün X, Y ve Z eksenleri ile yaptığı açılar hesaplanıp sensörün yeryüzüne göre konumu tespit edilebilir. İvmeölçerin, R vektörünün bileşenlerini g kuvveti türünden vermesi için bir dizi işlem yapılmalıdır.

İvmeölçerden ADC yardımı ile elde edilen değerler ham değerlerdir (binary değerleri) ve her bir eksen için -32768 ile +32768 arasında bir değer olabilir. Yani AdcRx, AdcRy. AdcRz değerleri -32768 ile +32768 arasında değişir. Ancak elde edilen bu ham değerler ile işlem yapabilmek mümkün değildir. İşlem yapabilmek için g değeri elde edilmelidir. g değerine ulaşabilmek için ise ADC den okunan ham değerler önce voltaj türünden ifade edilmelidir. Ham değerleri volt türünden ifade edebilmek için LSB başına düşen gerilim miktarı (VDD 3.3 Volt) ile çarpılması gerekmektedir. Yani ham değerlerin her birinin ayrı ayrı besleme gerilimi ile çarpılıp $2^{\text{ADC çözünürlüğü}}$ ile bölünmesi gerekmektedir[15,16].

$VoltsRx = AdcRx * Vdd / 2^{16}$

$VoltsRy = AdcRy * Vdd / 2^{16}$

$VoltsRz = AdcRz * Vdd / 2^{16}$

İvmeölçer sensörünün kullanma kılavuzundan (datasheet) istifade ederek elde edilen voltaj değerlerinin g kuvveti türünden karşılığı bulunmalıdır.

İvmeölçerin 0 g de vermiş olduğu bir gerilim değeri vardır ve Zero-G değeri olarak adlandırılmaktadır. Bu değer sensörün kullanma kılavuzunda belirtilir ve genelde Vdd/2'dir. Yukarıda belirtilen voltaj değerlerinden bu değer çıkartılıp sensörün hassasiyetine bölünürse hangi eksene ne kadar kuvvet uygulandığı bulunur. Böylece tam denklem aşağıdaki gibi olur ve eksenlere uygulanan kuvvetler g türünden ifade edilir.

$Rx = (AdcRx * Vdd / 2^{16} - VzeroG) / Hassasiyet$

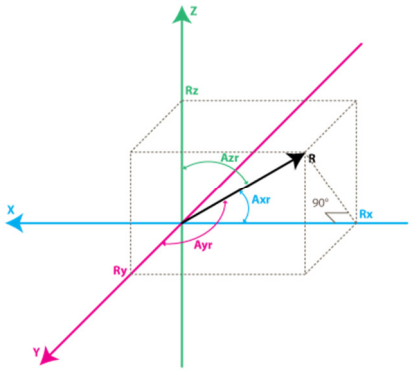
$Ry = (AdcRy * Vdd / 2^{16} - VzeroG) / Hassasiyet$

$Rz = (AdcRz * Vdd / 2^{16} - VzeroG) / Hassasiyet$

Böylece elde edilen değerler trigonometrik fonksiyonlar ile açıya çevrilir.

Şekil 3.10'da gösterilen R vektörünün X ve Y eksenleri ile yaptığı açılar şu şekilde hesaplanır. Şekilde görülen Axr açısının cosinüsü Rx/R değerini vermektedir. Rx ve R değerleri bilindiğine göre arccosinüs ile Axr açısı elde edilir. Benzer şekilde Ayr açısı ve Azr açısı hesaplanabilir. R vektörü de yukarıdaki birinci eşitlikten hesaplanıp aşağıdaki arccos fonksiyonlarına değerleri koyduğunda istenilen açılar elde edilir[15,16].

$$cos(Axr) = Rx / R \ , Axr = arccos(Rx/R)$$
$$cos(Ayr) = Ry / R \ , Ayr = arccos(\mathrm{Ry}/R)$$
$$cos(Azr) = Rz / R \ , Azr = arccos(Rz/R)$$



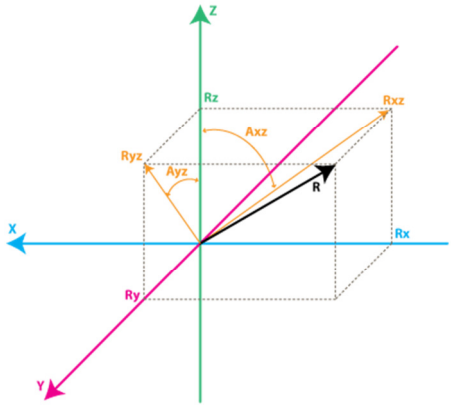

**Şekil 3.10** R Vektörünün Koordinat Sistemi Üzerinde Açısal Gösterimi

MPU-6050 sensörü içerisinde bir de jiroskop bulunmaktadır. Jiroskop basitçe bir diskin kendi ekseni etrafında hızla döndürülmesi sonucu ortaya çıkar. (Şekil 3.11)



**Şekil 3.11** Örnek Bir Jiroskop Modeli

Jiroskop bir nesnenin bir eksen etrafında ne kadar hızla döndüğünü (açısal hızını) tespit etmek için kullanılır. Tespit edilen hız dakikadaki dönüş sayısı (RPM) ya da saniyedeki dönüş derecesi (°/sn) olarak tanımlanır. İvmeölçerde olduğu gibi jiroskop bir, iki veya üç eksende ölçüm yapabilmektedir. Jiroskoplar saniyedeki dönüş hızını ölçebilme kapasitesine göre değişiklik gösterebilirler. Kapasitelerinin üzerindeki dönüş hızları sensör çıkışlarına anlamsız sinyaller gönderebilmektedir.




**Şekil 3.12** R Vektörünün Koordinat Sistemi Üzerindeki İz Düşümleri

Şekil 3.12'deki grafikte iki eksenli (X,Y) jiroskop kullanılmıştır. Buradaki R vektörünün XZ uzayındaki izdüşümü Rxz, YZ uzayındaki izdüşümü ise Ryz vektörüdür. Rxz ve Ryz vektörleri Pisagor teoremi kullanılarak hesaplanır. ($Rxz^2=Rx^2+Rz^2$ *ve* $Ryz^2=Ry^2+Rz^2$)

Vektörlerin Z ekseni ile yapmış olduğu açılar ise Axz ve Ayz dir. Grafik Y ekseni etrafında döndürüldüğünde Axy açısı, X ekseni etrafında döndürüldüğünde ise Ayz açısı değişecektir. Jiroskop dönüş hızını ölçtüğüne göre, dönüş hızı zaman ile çarpılarak dönüş açısını elde edilir. t0 anındaki açı Axz0 ve t1 anındaki açı Axz1 ise ($Axz1 - Axz0$) = $RateAxz * (t1 - t0)$ olur. Sensörden alınan değerleri dönüş hızına çevirmek için aşağıdaki formüller kullanılır.

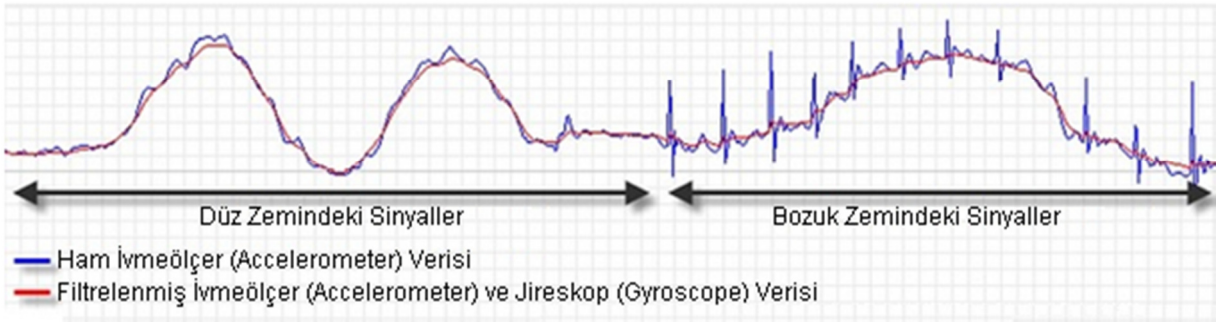
$RateAxz = (AdcGyroXZ * Vref / 2^{16} - VzeroRate) / Hassasiyet$

$RateAyz = (AdcGyroYZ * Vref / 2^{16} - VzeroRate) / Hassasiyet$

Sensörlerin hareketsiz durumda sabit olarak verdiği bir gerilim vardır ve VzeroRate olarak adlandırılır. Bu gerilim ikili değerden elde edilen gerilimden çıkartılmalıdır. Elde edilen değerler açısal hız değerleridir. Elde edilen bu değerler de iki ölçüm arasında geçen süre ile çarpılarak dönüş açısı bulunur.

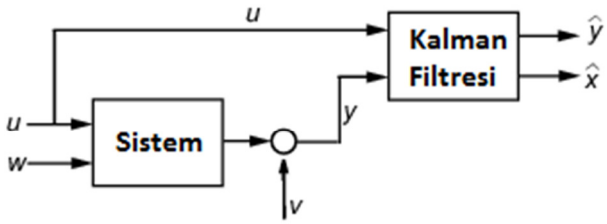
MPU-6050 sensöründe bulunan jiroskop ve ivmeölçerin ayrı ayrı göndermiş olduğu ham değerler çok güvenli bilgiler değildir. Çünkü jiroskop ve ivmeölçer eğimsel sapma (bias drift) adı verilen bir sapma yapar ve bundan dolayı hassas açı ölçümünde tek başlarına kullanılamazlar. Ayrıca ivmeölçer kuvvete karşı çok duyarlıdır ve en ufak titreşimlerde bile çok yüksek gürültüler oluşturmaktadır. Ancak jiroskop bu kuvvetlerden etkilenmemektedir.

Bu yüzden bu iki sensör birleştirilip yönelim, hız, pozisyon gibi bilgiler tek bir üniteden alınmalıdır. Böylece jiroskop ve ivmeölçer çıkışlarını filtreleyerek daha doğru bir ölçüm yapılmasını sağlanır. Bu üniteye IMU (Inertial Measurement Unit) denir. Serbestlik derecesi DOF (Degrees of Freedom) olarak ifade edilir. MPU-6050 sensörü 3 eksen jiroskop ve 3 eksen ivmeölçeri olduğu için 6 serbestlik derecesi IMU elde edilir.




**Şekil 3.13** Jiroskop ve İvmeölçer (Accelerometer) Çıkış Sinyalleri

Filtreleme işlemi için çeşitli algoritmalar bulunmaktadır. Bunlar Kalman ve Tamamlayıcı (Complementary) Filtrelerdir. En yaygın olarak kullanılan algoritmalardan bir tanesi Kalman Filtresidir. Sistemin bir önceki çıkışları ile yeni ölçümlerinden elde edilen çıkışları karşılaştırarak bir sonraki çıkışları tahmin edecek şekilde çalışmaktadır. Bu durum Şekil 3.13'te şematik olarak ifade edilmiştir.




**Şekil 3.14** Kalman Filtresi Akış Diyagramı

Şekil 3.14'te bulunan w'ye işlem gürültüsü denir ve sistem modelinde yukarıda sayılan nedenlerden dolayı meydana gelen idealden sapmaları temsil eder. Ayrıca v'ye ölçüm gürültüsü denir ve ölçümde meydana gelen sapmaları temsil eder. Kalman filtresinde bu gürültülerin sisteme aşağıdaki gibi etki ettiği varsayılır.

$$\dot{x} = Ax + Bu + Gw$$

$$y = Cx + Du + Hw + v$$

Kalman filtresinin sistemin gerçek çıkışı için yaptığı tahmin $\hat{y}$ ile, sistemin durumu için yaptığı tahmin $\hat{x}$ ile gösterilir.

Her ne kadar Kalman Filtresi çok iyi de olsa kullanımını zorlaştıran iki önemli problem vardır. Bunlar; anlaşılması güç karmaşık bir yapısı olması ve 8 bitlik işlemcilere uygulanması imkânsız olmasa da çok zor olmasıdır. Bu nedenle yapılan çalışmada anlaşılması ve uygulaması çok daha kolay Tamamlayıcı Filtre kullanılmıştır.

Yukarıda bahsedildiği gibi ivmeölçer kuvvete karşı çok duyarlıdır ve en ufak titreşimlerde bile çok yüksek gürültüler oluşturmaktadır. Ancak jiroskop bu kuvvetlerden etkilenmemektedir. Tamamlayıcı Filtre bu iki sensörden gelen verileri birleştirilip yönelim, hız, pozisyon gibi bilgileri tek bir açısal değer olarak vermektedir. Bunun için şu formül kullanılır;

$$açı = 0.98 \text{ x } (açı + jiroskop\ verisi \text{ x } dt) + 0.02 \text{ x } (ivmeölçer\ verisi)$$

Böylece jiroskop ve ivmeölçerin çıkışlarını filtreleyerek daha doğru bir ölçüm yapılmasını sağlar. Burada jiroskop verisi zaman ve mevcut açı değeri ile bütünleştirilmektedir. Daha sonra bu sonuç alçak geçirgen ivmeölçer verisi ile birleştirilmektedir. Tamamlayıcı Filtre ayarı kat sayılar (0.98 ve 0.02) değiştirilerek yapılabilir. Tamamlayıcı Filtrenin kullanıldığı sonsuz döngü sistemlerde sensörlerden gelen x, y ve z açısal değerleri her döngüde yeni jiroskop verileri ile güncellenir. Tamamlayıcı Filtre ivmeölçer tarafından okunan kuvvet büyüklüğünün makul değerde olup olmadığını kontrol eder. Eğer bu değer çok küçük veya çok büyükse bunun bozucu bir giriş olduğu değerlendirilir ve hesaplamalara dâhil edilmez. Daha sonra x, y, z açıları mevcut değerlerinin %98'i ve ivmeölçer ile hesaplanan açıların %2'si ile toplanarak güncellenir. Bu süreç ölçümde sapma olmamasını ve kısa zaman aralığında daha doğru bir ölçüm yapılmasını sağlar.

MPU-6050 sensörünün her kanalında 16bit'lik analog değerin dijitale dönüşümünü sağlayan donanım mevcuttur. Bu sayede X, Y ve Z eksenlerinde aynı zamanda ölçüm yapmaya olanak sağlar ve hem jiroskopun hem de ivmeölçerin ham değerleri kolaylıkla okunabilir. MPU-6050 sensörü bir de 1024 bayt'lık FIFO tampon içermektedir. MPU-6050 sensörü, değerleri FIFO tamponun içerisine yerleştirilmesi için programlanabilir ve tampon mikroişlemci tarafından okunabilir. FIFO tampon kesme sinyali ile birlikte kullanılır. Eğer sensör içerisindeki FIFO tampon bölgesinde veri varsa bu kesme sinyali sayesinde mikroişlemci sensör içerisindeki FIFO tampon bölgesinde bekleyen FIFO verisi olduğunu bilir.

**Donanımsal Özellikler:**

| | |
|---|---|
| Besleme gerilimi | : 2.4V–3.46V |
| Lojik gerilimi | : 1.8V±5% veya besleme gerilimi(VDD) |
| Jiroskop işletme akımı | : 3.6mA |
| Jiroskop + ivmeölçer işletme akımı | : 3.8mA |

3 eksen açısal hız sensörü(Jiroskop) : 131 LSBs/dps hassasiyete varan ve ±250, ±500, ±1000, ve ±2000dps tam ölçek aralıklı

3 eksen ivmeölçer : Programlanabilir tam ölçek aralıklı ±2g, ±4g, ±8g ve ±16g

İvmeölçer düşük güç modu işletme akımı: 10uA 1Hz'de, 20uA 5Hz'de, 70uA 20Hz'de, 140uA 40Hz'de

Bekleme modu besleme akımı : 8µA

Bağlantı ara yüzü :400kHz hızlı I²C seri sunucu

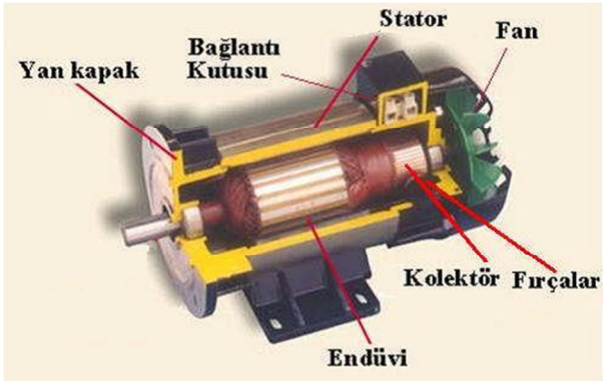
Darbe toleransı :10,000g

Dijital çıkış 9 eksen döndürme matrisi, Euler açısı ve ham data formatında MotionFusion verisi

### 3.2.3 Doğru Akım (DC) Motoru

Elektrik enerjisini, mekanik enerjiye çeviren elektrik makinesine doğru akım (DC) motoru denir. Herhangi bir iletkene doğru akım uygulandığında iletken, N ve S kutuplarından oluşan sabit bir manyetik alan meydana getirir. Meydana gelen bu sabit manyetik alan içindeki iletken cisimlere bir kuvvet uygular. Bu kuvvet iletken cisimleri ve farklı kutuplu manyetik alanlar birbirini çeker, aynı kutuplu manyetik alanlar ise iter. N kutbundan S kutbuna doğru meydana gelen bu kuvvete de manyetik akı denilmektedir[17].

Doğru akım motorları endüktör (stator), endüvi (rotor), kollektör, fırça, yataklar ve kapaklardan oluşmaktadır[18]. (Şekil 3-15)




**Şekil 3-15** Doğru Akım (DC) Motoru

Endüktör (stator), doğru akım motorlarının dış tarafında kalan, duran kısımdır. Bu kısım manyetik alanın oluşmasını sağlar ve doğru akımla beslenir. Doğru akım

motorlarında ana kutuplar ve bunarlın arasına yerleştirilmiş yardımcı kutuplar vardır. Endüktör (stator) kutupları mıknatıs veya elektromıknatıs olabilir.

Endüvi (rotor), doğru akım motorlarının dönen kısmı olup gerilim indüklenen bölümdür. Endüvi (rotor) milinden mekanik enerji alınır.

Kollektör sayesinde endüviye gerilimin iletilmesi sağlanır. Endüvide üretilen alternatif gerilim, kollektör tarafından doğrultulur ve kollektör üzerine temas eden fırçalar yardımıyla dış devreye iletilir. Kollektör dilimleri yüksek ısıya dayanıklı sert bakırdan yapılır.

Fırça, karbonlardan imal edilen, dış devredeki akımı endüviye ileten yardımcı elemandır. Bazen iletkenliği arttırmak için karbon bakır alaşımlı yapılırlar. Fırça, kollektör ile elektriksel bağlantıyı sağlar. Fırçanın, kollektör üzerinde durması ve bu durumun işletme sırasında devamlılığının sağlanması fırça tutucuları ile gerçekleştirilir.

Doğru akım motorlarında kendi kendine yağlanan metal yataklar veya bilyeli yataklar kullanılır. Bu yataklar, gürültüsüz az bir kayıpla motorun hareketli kısmının bir eksen etrafında dönmesini sağlar. Kapaklar ise, makinenin dış yüzeyindeki parçadır.

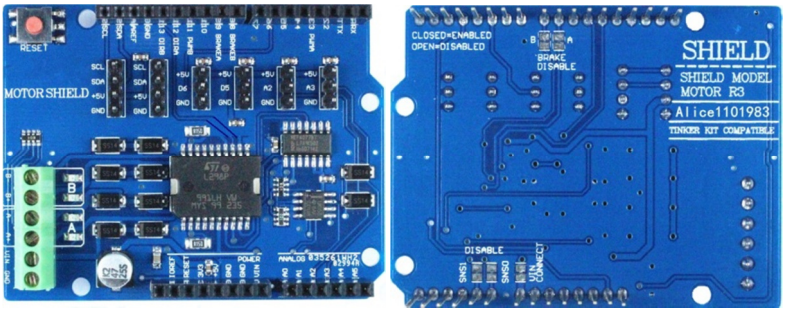
Doğru akım motorlarına bir güç kaynağı bağlandığında motorun dönüş yönü akımın yönüne bağlıdır. Akımın yönü terslendiğinde doğru akım motorunun dönüş yönü de terslenmiş olur. Doğru akım motorlarının hızı rpm (rotations per minute - bir dakikada tamamlanan devir sayısı) ile ölçülür. Motorun hızı uygulanan gerilim ve yüke bağlıdır.

Doğru akım motorlarının çeşitleri şunlardır; Fırçasız Doğru Akım Motoru, Şönt Motor, Seri Motor, Kompunt Motor (Ters Kompunt, Eklemeli Kompunt), Fırçalı Doğru Akım Motoru.

Doğru akım motorlarının daha verimli kullanılabilmesi için motor kontrol kartı gerekmektedir. Kontrol kartının özellikleri kullanılan doğru akım motoruna uygun olmalıdır.

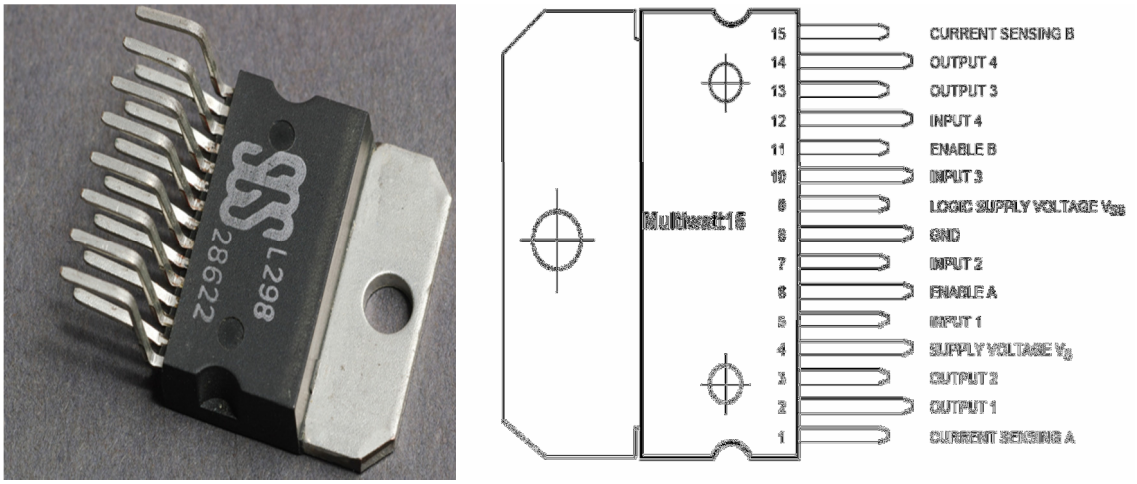
### 3.2.4 Motor Kontrol Kartı (L298 destekli)

"Arduino Motor Shield" motor kontrol kartı, DC ve step motorlar gibi endüktif yükleri çift yönlü sürmek için tasarlanmış ve bünyesinde L298 motor sürücü entegresi bulunduran bir karttır. L298 entegresi üzerinde 2 adet H köprüsü mevcuttur. H köprüsü




**Şekil 3.16** Motor Kontrol Kartı (L298 destekli)

DC motoru iki yönde de sürmeye yarar. L298 entegresi ile A ve B isimli iki adet DC motorun hızları ve yönlerini birbirinden bağımsız olarak kontrol edilebileceği gibi bağlantılar birleştirilerek bir adet iki kutuplu step motor da kontrol edilebilir. L298 entegresinde IN1, IN2, OUT1, OUT2, ENA, SENSA A motoru için, IN3, IN4, OUT3, OUT4, ENB, SENSB B motoru için olup toplam 15 adet bacak bulunmaktadır[20,21].




**Şekil 3.17** L298 Motor Sürücü Entegresi ve Bacak Bağlantıları

**IN 1, IN2 (5,7):** Bu bacaklar A motoru için olan girişlerdir. +5 volt ile çalışır. Eğer IN1'e 5V, IN2'ye 0V verildiğinde motor ileri dönerse, tam tersini verildiğinde motor geri döner. Her iki bacağa da aynı değer verilirse (0V-0V veya 5V-5V) motor dönmez.

**IN 3, IN 4 (10,12):** Bu bacaklar B motoru için olan girişlerdir. A motoruyla aynı şekilde çalışır.

**OUT 1, OUT2 (2,3):** A motoru için çıkış bacaklarıdır. Bu çıkışlar motorun her iki ucuna bağlanır.

**OUT 3, OUT 4 (13,14):** B motoru için çıkış bacaklarıdır. Bu çıkışlar da motorun her iki ucuna bağlanır.

**ENA, ENB (6,11):** A ve B motorlarını etkinleştirmek için bu bacaklara +5 volt gerilim uygulanmalıdır.

**SENSA, SENSB (1,15):** A ve B motorlarının çalışması için bu bacaklar toprağa bağlanmalıdır.

**VS (4):** Kontrol edilmek istenen motora uygun voltaj değeri verilmelidir ve bu voltaj değeri 46 V değerini aşmamalıdır.

**VSS (9):** L298' in çalışması için +5 volta gerilim uygulanmalıdır.

**GND (8):** L298' in çalışması için toprağa bağlanmalıdır.

Arduino Uno üzerinde bulunan pinlerden dört tanesi A motoru için ve dört tanesi B motoru için olmak üzere toplamda 8 adet pin motor kontrol kartı tarafından kullanır. Yapılacak çalışmada fren ve akım algılamaya ihtiyaç duyulmaz ve ayrıca uygulama için daha fazla pine ihtiyaç duyulur ise motor kontrol kartının arkasındaki sıralı jumper'lar kesilerek bu özellikler devre dışı bırakılabilir[19].

Çalışmada motor kontrol kartı tarafından kullanılan pinler aşağıdaki tabloda gösterilmiştir;

| Fonksiyon | A motoru pinleri | B motoru pinleri |
|---|---|---|
| Doğrultu | D12 | D13 |
| PWM | D3 | D11 |
| Fren | D9 | D8 |
| Akım algılama | - | - |

Motor kontrol kartı, altında bulunan pinler vasıtası ile Arduino Uno mikroişlemci kartının üzerine monte edilir ve harici bir güç kaynağı ile beslenmelidir. Çünkü kartın üzerine yerleştirilen L 298 entegresi; birisi mantık diğeri motor besleme sürücüsü olmak üzere iki ayrı güç bağlantısına sahiptir. İhtiyaç duyulan motor akımı genellikle maksimum USB akımını geçmektedir. Harici (USB olmayan) güç bir AC-DC adaptöründen veya bataryadan gelebilir. Adaptör, Arduino Uno mikroişlemci kartının 2.1mm'lik harici güç girişine bağlanabilir veya kutuplarına dikkat etmek koşulu ile motor kontrol kartı üzerinde bulunan $V_{in}$ ve GND vida terminallerine bağlanabilir[19].

Motor kontrol kartının monte edildiği Arduino Uno mikroişlemci kartında oluşabilecek olası hasarları engellemek için 7-12V arası gerilim sağlayan harici bir güç kaynağı kullanılmalıdır. Eğer kullanılan motor 9V'tan daha fazla gerilime ihtiyaç duyuyorsa motor kontrol kartı ile Arduino Uno mikroişlemci kartının güç hatları

birbirinden ayrılmalıdır. Güç hatları ayrılması motor kontrol kartının arkasında bulunan "$V_{in}$ Connect" bağlantısı kesilerek gerçekleştirilir[19].

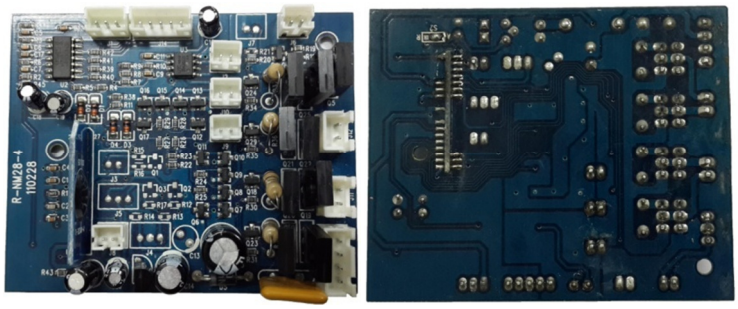
Vida terminal bloğundaki $V_{in}$ motor kontrol kartına bağlanan motorun giriş gerilimidir. Bu girişe bağlanan harici güç kaynağı aynı zamanda Arduino Uno mikroişlemci kartına da güç sağlar. "$V_{in}$ Connect" bağlantısı kesildiğinde bu hat motora özgü bir güç hattı haline gelir[19].

**Donanımsal Özellikler:**

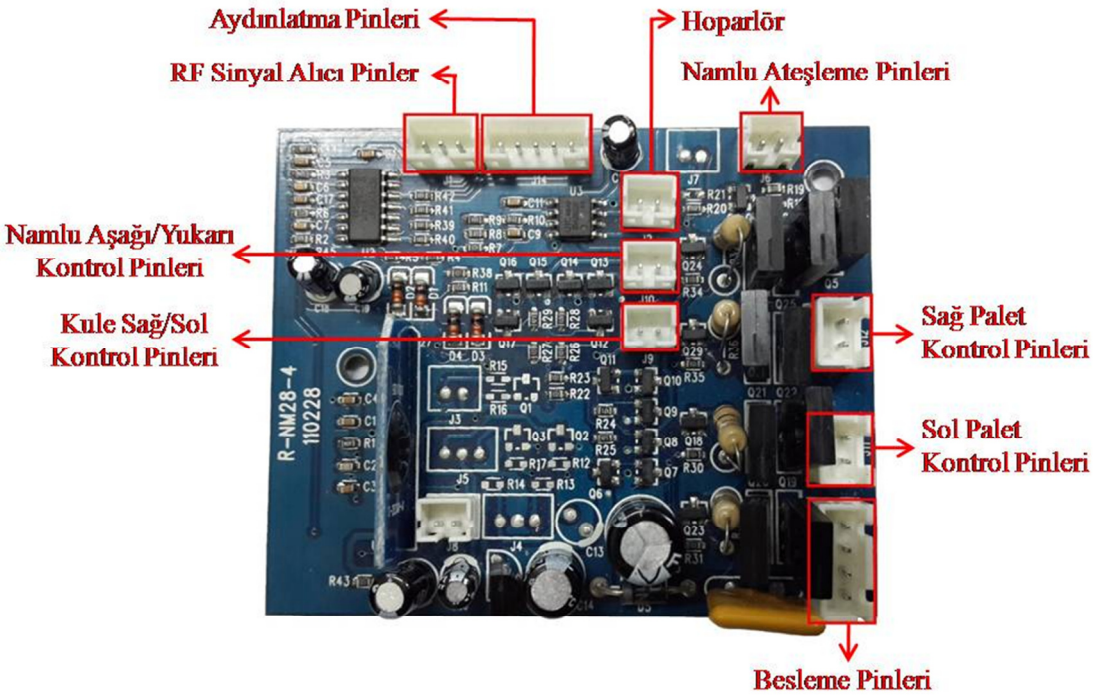
Çalışma gerilimi : 5V - 12V
Motor kontrolcüsü : L 298, 2 adet DC motor veya 1 adet step motor sürer
Azami akım : Her bir motor için 2A veya maksimum 4A (harici güç kaynağı ile)
Akım algılama : 1,65 V/A
Uzunluk : 2.7 inç (6.85 cm)
Genişlik : 2.1 inç (5.33 cm)
Kendiliğinden durma ve fren fonksiyonu
Motorları ve güç kaynaklarını bağlamak için vida terminali

### 3.2.5 Model Tank

Çalışmada kullanılan tank gerçeğinin 1/24 oranında küçültülmüş, uzaktan kumanda (R/F Control) ile kontrol edilen bir model tanktır. Model tankın içerisinde üzerinde mikroişlemcisi bulunan, sistemin çalışmasını ve kontrolünü sağlayan bir elektronik kontrol devresi mevcuttur[22, 23]. (Şekil 3-18, 3-19)



**Şekil 3-18** Model Tank Elektronik Kontrol Devresi (Ön ve Arka Yüz)




**Şekil 3-19** Model Tank Elektronik Kontrol Devresi (Ön Yüz)

**Model Tankın Özellikleri :**



**Şekil 3-20** Model Tank

Marka/Model :

VSTank / ABRAMS M1A2

Hız : 1.33 km/s

Uzunluk : 415 mm

Genişlik : 152 mm

Yükseklik : 142 mm

Ağırlık : 1300 gr

Tırmanış : 40 Derece

Batarya : Tank 8 adet / kumanda 6 adet kalem pil

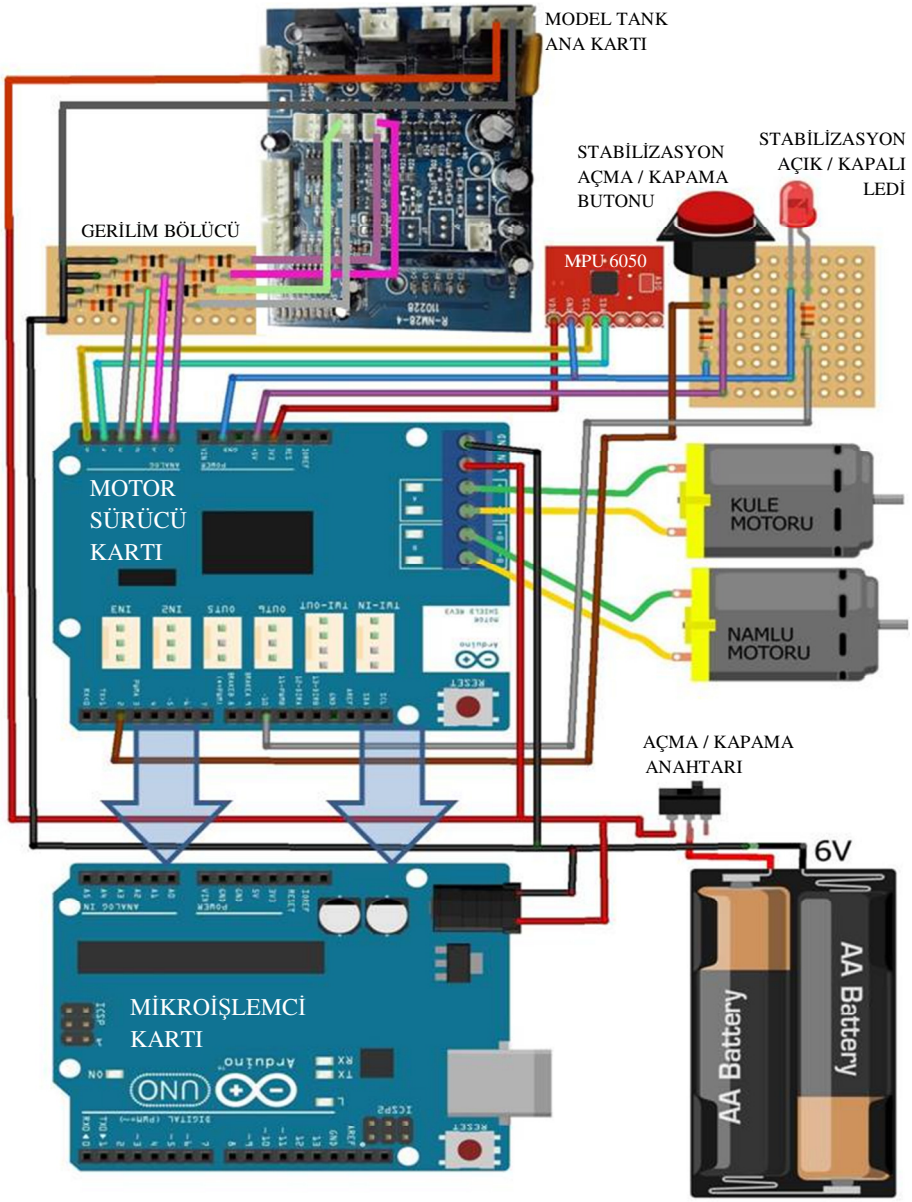
Kontrol : Üç farklı hız seçeneği ile ilerleme, iki farklı hız seçeneği ile geri gitme, iki farklı hız seçeneği ile sağa ve sola dönüş kabiliyeti. Birden fazla açılı dönüş özelliği.

Süspansiyon Özelliği : Her tekerlekte ayrı ergonomik süspansiyon

Kule ve Tank Topu Özelliği : $330^0$ dönebilen tank kulesi, yukarı ve aşağı yönde tank topu hareketi ve 25m'ye kadar boncuk fırlatabilme

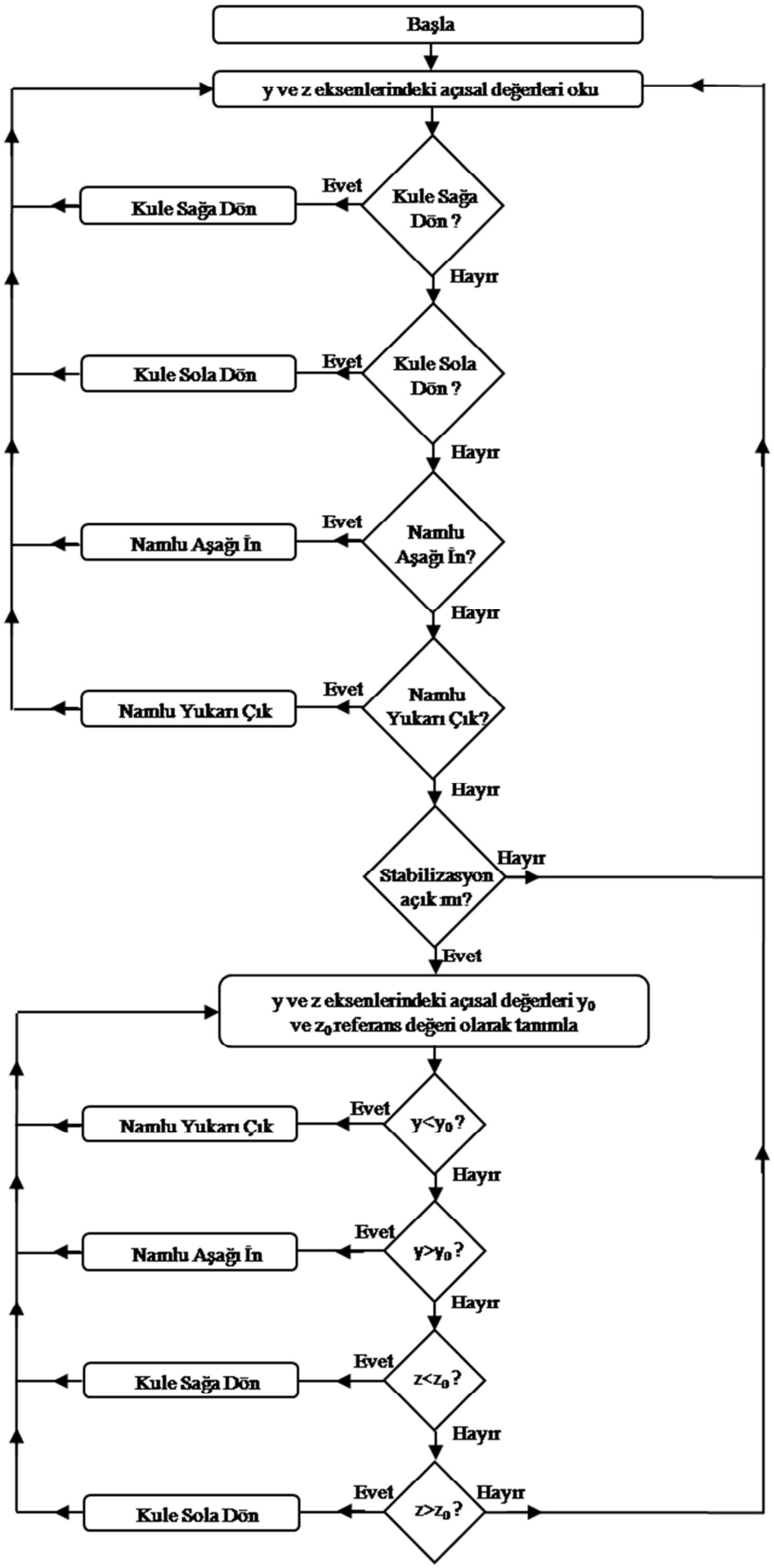
### 3.3 Tank Namlusu Stabilizasyon Sisteminin Modellenmesi

Yapılan çalışmada kullanılan Arduino Uno mikroişlemci, konum algılayıcı MPU-6050 sensörü, eyleyici doğru akım motorları ve motor kontrol kartı aracılığı ile oluşturulmuş olan kapalı döngü otomatik kontrol sisteminin elektronik bağlantıları Şekil 3.21'de olduğu gibidir.




**Şekil 3.21** Tank Namlusu Stabilizasyon Sistemi Elektronik Modeli

Arduino Uno mikroişlemci kartı üreticinin sağladığı "Arduino" isimli ücretsiz yazılım ile programlanmıştır. Söz konusu yazılım Wiring yazılım dilidir. Yapılan çalışmada kullanılan yazılım algoritmasına ait akış diyagramı Şekil 3.22'de olduğu gibidir.




**Şekil 3.22** Yazılım Algoritmasına Ait Akış Diyagramı

Oluşturulan model yazılım algoritmasına ait akış diyagramına uygun olarak MPU-6050 sensöründen gelen veriler ile tank gövdesinin x, y ve z eksenlerinde yapmış olduğu açısal değişiklikleri algılar. Elde edilen veriler $I^2C$ iletişim protokolü kullanılarak karşılaştırıcı ve denetim organı olarak kullanılan Arduino Uno mikroişlemci kartına iletilir. Mikroişlemci tarafından stabilizasyon sisteminin devrede olup olmadığını kontrol edilir ve sistem tank nişancısı tarafından devreye sokulmuş ise y ve z eksenlerindeki açısal değerleri $y_0$ ve $z_0$ referans değeri olarak tanımlanır. MPU-6050 sensöründen sürekli olarak gelen veriler sayesinde tank topu namlusu için tanımlanan referans noktasına göre y ve z eksenlerinde yapılan açısal değişiklikler algılanır. ATmega328 işlemcisi kullanan Arduino Uno mikroişlemci kartı kendisine ulaşan verileri daha önceden tanımlanan yazılım algoritmasına uygun olarak değerlendirir ve tank topu namlusu için tanımlanan referans noktasına namluyu tekrar yöneltmek için doğru akım motorlarına gerekli sinyali motor kontrol kartı aracılığıyla gönderir. Eyleyici olarak kullanılan doğru akım motorları Arduino Uno mikroişlemci kartından gelen sinyallerle tank topu namlusunu tanımlanan referans noktasında tutacak şekilde harekete başlarlar. Arduino Uno mikroişlemci kartı MPU-6050 sensöründen sürekli olarak gelen verilerle namlunun konumunu referans noktasıyla karşılaştırır ve namlu referans noktası geldiğinde eyleyici motorlara gönderdiği sinyali sonlandırır. Burada otomatik kontrol sistemi tarafından uygulanan işlem kısaca MPU-6050 sensörü ile tespit edilen açısal değişikliklerin aksi yönünde ve aynı miktarda doğru akım motorlarına hareket vermektir. Tüm bu otomatik kontrol sisteminin yazılım tasarımı EK-A'da gösterilmiş ve açıklanmıştır.

# BÖLÜM 4

# SONUÇLAR VE TARTIŞMA

Bu çalışma ile değişik/bozuk yol ve arazi şartlarında hareket halinde olan bir silah sistemini kullanan nişancıların vuruş yüzdesi yüksek olan atışlar yapabilmeleri için tank namlusu stabilizasyon sisteminin Arduino Uno ile uygulanması ve deneysel düzeneğinin hazırlanması, bu düzenek ile tank topu namlusunun sürekli hedef bölgesinde ve sabit bir şekilde durması amaçlanmıştır. Söz konusu deneysel düzenek model tank üzerinde hazırlanmış model tank $15^0$ sola döndüğünde namlu $15^0$ sağa dönerek referans noktasında kalmıştır. Aynı test model tank $15^0$ sağa döndüğünde namlu $15^0$ sola dönerek aynı başarıya ulaşmıştır. Model tanktaki namlu hareketi aşağı ve yukarı yönde de test edilmiştir. Model tank $5^0$'lik bir meyilden aşağı inerken namlu $5^0$'lik yukarı hareketle referans noktasında kalmayı başarmıştır.

Çalışmada kullanılan algoritmanın bazı zamanlarda (çok yüksek hızlı hareketlerde) kararsız davrandığı tespit edilmiştir. Bu kapsamda filtreleme programı ve yazılım algoritması geliştirilebilir. Ayrıca MPU-6050 sensöründen elde edilen ham verilerin mikroişlemci üzerinde hesaplanması mikroişlemcinin işlem yükünü artırmaktadır. Elde edilen ham verilerin mikroişlemci üzerinde değil MPU-6050 sensöründe yerleşik olan Dijital Hareket İşlemci™ (DMP™) ile hesaplanması sağlanabilir.

Çalışmada kullanılan model tankın kulesinin hareketi $330^0$ ile sınırlı olduğu için sisteme bunun dışında hareket verilememektedir. Oluşturulan sistemin daha kapsamlı ve etkin olarak kullanılabilmesi maksadı ile $360^0$ dönebilen kuleye sahip model tankta kullanılabilir.

# EKLER

## **EK-A:** Tank Namlusu Stabilizasyon Sisteminin Yazılım Tasarımında Yer Alan Kodlar

```
#include <Wire.h>                       // I2C iletişimi için "Wire" kütüphanesi eklenir.

int stabled=10;                         // 10 numaralı pin stabled isimli değişken olarak tanımlanır.
volatile int stab = LOW;                // "stab" isimli değişken tanımlanır ve ilk durumunu "0"
                                        olarak atanır.
volatile int stabledstat = LOW;         // "stabledstat" isimli değişken tanımlanır ve ilk durumunu
                                        "0" olarak atanır.
long LastDebounceTime=0;                // Son Debounce zamanı tanımlanır.
long DebounceTime=300;                  // Debounce zamanı 300 ms olarak ayarlanır.
float targetPosKule, targetPosNam;      // Hedef açı ve şimdiki açı değişkenleri tanımlanır.

const int                               // DC motor sabit pini tanımlanır.

PWM_A   = 3,                            // Kule motoru için darbe genişlik modülasyonu pin 3'e
                                        tanımlanır.
DIR_A   = 12,                           // Kule motoru için motor yönü kontrolü pin 12'ye tanımlanır
BRAKE_A = 9,                            // Kule motoru için motor freni pin 9a' tanımlanır.
R_A     = A0,                           // Kumandadan gelen kule motoru sağa dön sinyani A0 pinine
                                        atanır.
L_A     = A1,                           // Kumandadan gelen kule motoru sola dön sinyani A1 pinine
                                        atanır.

PWM_B   = 11,                           // Namlu motoru için darbe genişlik modülasyonu pin11'e
                                        tanımlanır.
DIR_B   = 13,                           // Namlu motoru için motor yönü kontrolü pin 13'e
                                        tanımlanır.
BRAKE_B = 8,                            // Namlu motoru için motor freni pin 8'e tanımlanır.
U_B     = A2,                           // Kumandadan gelen namlu motoru yukarı dön sinyali A2
                                        pinine atanır.
D_B     = A3;                           // Kumandadan gelen namlu motoru aşağı dön sinyali A3
                                        pinine atanır.
```

//MPU6050'nin register adresleri birer etiket ismi olarak tanımlanarak bu registerlar adresler yerine etiketleri ile kullanılırlar.

```
#define MPU6050_ACCEL_XOUT_H     0x3B      // Oku
#define MPU6050_ACCEL_XOUT_L     0x3C      // Oku
#define MPU6050_ACCEL_YOUT_H     0x3D      // Oku
#define MPU6050_ACCEL_YOUT_L     0x3E      // Oku
#define MPU6050_ACCEL_ZOUT_H     0x3F      // Oku
#define MPU6050_ACCEL_ZOUT_L     0x40      // Oku
#define MPU6050_PWR_MGMT_1       0x6B      // Oku/Yaz
#define MPU6050_PWR_MGMT_2       0x6C      // Oku/Yaz
#define MPU6050_WHO_AM_I          0x75     // Oku
#define MPU6050_I2C_ADDRESS       0x68     // Oku/Yaz
```

// MPU6050den gelen baytların sırası derleyici ve AVR çipi bayt sıralamasıyla eşleşmemektedir. Arduino üzerindeki AVR çip, düşük adreste düşük bayt bilgisi bulundurmaktadır. Fakat MPU-6050 faklı sıralamaya sahip yüksek bayt düşük adrestedir, o nedenle düzeltilmesi gerekmektedir.

```
typedef union accel_t_gyro_union
{
 struct
 {
   uint8_t x_accel_h;                    // İvmeölçer X ekseni yüksek bayt
   uint8_t x_accel_l;                    // İvmeölçer X ekseni düşük bayt
   uint8_t y_accel_h;                    // İvmeölçer Y ekseni yüksek bayt
   uint8_t y_accel_l;                    // İvmeölçer Y ekseni düşük bayt
   uint8_t z_accel_h;                    // İvmeölçer Z ekseni yüksek bayt
   uint8_t z_accel_l;                    // İvmeölçer Z ekseni düşük bayt

   uint8_t t_h;                          // Sıcaklık yüksek bayt
   uint8_t t_l;                          // Sıcaklık düşük bayt

   uint8_t x_gyro_h;                     // Jiroskop X ekseni yüksek bayt
   uint8_t x_gyro_l;                     // Jiroskop X ekseni düşük bayt
   uint8_t y_gyro_h;                     // Jiroskop Y ekseni yüksek bayt
   uint8_t y_gyro_l;                     // Jiroskop Y ekseni düşük bayt
   uint8_t z_gyro_h;                     // Jiroskop Z ekseni yüksek bayt
   uint8_t z_gyro_l;                     // Jiroskop Z ekseni düşük bayt
 } reg;
```

```
struct
  {
    int x_accel;                                    // Yeni değişkenler tanımlanır.
    int y_accel;
    int z_accel;
    int temperature;
    int x_gyro;
    int y_gyro;
    int z_gyro;
  } value;
};

// Sensörün tüm dönüş açılarının kaydedilmesi için aşağıda tanımlanan küresel (global) değişkenler fonksiyonları kullanılır.
unsigned long last_read_time;                       // Okunan son zaman.
float         last_x_angle;                         // Okunan son filtrelenmiş açılar.
float         last_y_angle;
float         last_z_angle;
float         last_gyro_x_angle;                    // Jiroskop açıları (kayma karşılaştırılması için)
float         last_gyro_y_angle;
float         last_gyro_z_angle;
float         angles[2]={0,0};

//Son okunan açı verilerinin değişkenlere atama fonksiyonu;

void set_last_read_angle_data(unsigned long time, float x, float y, float z, float x_gyro, float y_gyro, float z_gyro) {
  last_read_time = time;
  last_x_angle = x;
  last_y_angle = y;
  last_z_angle = z;
  last_gyro_x_angle = x_gyro;
  last_gyro_y_angle = y_gyro;
  last_gyro_z_angle = z_gyro;
}
```

```
// Geçen zaman tanımlanır.
inline unsigned long get_last_time() {return last_read_time;}
inline float get_last_x_angle() {return last_x_angle;}
inline float get_last_y_angle() {return last_y_angle;}
inline float get_last_z_angle() {return last_z_angle;}
inline float get_last_gyro_x_angle() {return last_gyro_x_angle;}
inline float get_last_gyro_y_angle() {return last_gyro_y_angle;}
inline float get_last_gyro_z_angle() {return last_gyro_z_angle;}

// Kalibrasyon yapabilmek için küresel (global) değişkenler tanımlanır.
float   base_x_accel;
float   base_y_accel;
float   base_z_accel;
float   base_x_gyro;
float   base_y_gyro;
float   base_z_gyro;

// Ham değerler okunur. 14 bayt tek seferde okunur. (3 eksen ivmeölçer = 6 bayt, sıcaklık 2 bayt, 3 eksen
jiroskop = 6 bayt). MPU-6050 varsayılan ayarlarına göre filtrelenmemiş veri gönderir ve kararlı veriler
değildirler, hata değeri oluştururlar.

int read_gyro_accel_vals(uint8_t* accel_t_gyro_ptr) {    // Jiroskop ve ivmeölçer ham değerleri okunur.
accel_t_gyro_union* accel_t_gyro = (accel_t_gyro_union *) accel_t_gyro_ptr;
int error = MPU6050_read (MPU6050_ACCEL_XOUT_H, (uint8_t *) accel_t_gyro,
sizeof(*accel_t_gyro));

// Tüm yüksek ve düşük baytlar değiştirilir. Daha sonra register değerleri değiştirilir, artık "x_accel_l"
isimli yapıda düşük bayt yoktur.
uint8_t swap;
#define SWAP(x,y) swap = x; x = y; y = swap                    // Değiştirme fonksiyonu tanımlanır.
SWAP ((*accel_t_gyro).reg.x_accel_h, (*accel_t_gyro).reg.x_accel_l);    // İvmeölçer X ekseni düşük ve
                                                                        yüksek baytlar değiştirilir.
SWAP ((*accel_t_gyro).reg.y_accel_h, (*accel_t_gyro).reg.y_accel_l);    // İvmeölçer Y ekseni düşük ve
                                                                        yüksek baytlar değiştirilir.
SWAP ((*accel_t_gyro).reg.z_accel_h, (*accel_t_gyro).reg.z_accel_l);    // İvmeölçer Z ekseni düşük ve
                                                                        yüksek baytlar değiştirilir.
SWAP ((*accel_t_gyro).reg.t_h, (*accel_t_gyro).reg.t_l);                // Sıcaklık düşük ve yüksek
                                                                        baytlar değiştirilir.
SWAP ((*accel_t_gyro).reg.x_gyro_h, (*accel_t_gyro).reg.x_gyro_l);      // Jiroskop X ekseni düşük ve
                                                                        yüksek baytlar değiştirilir.
```

```
SWAP ((*accel_t_gyro).reg.y_gyro_h, (*accel_t_gyro).reg.y_gyro_l);      // Jiroskop Y ekseni düşük ve
                                                                        yüksek baytlar değiştirilir.
SWAP ((*accel_t_gyro).reg.z_gyro_h, (*accel_t_gyro).reg.z_gyro_l);      // Jiroskop Z ekseni düşük ve
                                                                        yüksek baytlar değiştirilir.
return error;
}

// Kalibrasyon fonksiyonu tanımlanır. Kalibrasyon yapılırken sensör yatay pozisyonda hareketsiz
durmalıdır.
void calibrate_sensors() {
  int                 num_readings = 10;
  float               x_accel = 0;
  float               y_accel = 0;
  float               z_accel = 0;
  float               x_gyro = 0;
  float               y_gyro = 0;
  float               z_gyro = 0;
  accel_t_gyro_union    accel_t_gyro;

// IMU sıfır konumunu belirlemezse sistem uzay boşluğunda gibi davranır.
read_gyro_accel_vals((uint8_t *) &accel_t_gyro);     // IMU'dan okunan değerlerin ilk setini yok say.

// Sensörden ham değerler okunur ve ortalamasını alınır. Bunun için ilk açılışta bir miktar veri okunur ve
ortalamasını alıp sıfır kabul edilir.
for (int i = 0; i < num_readings; i++) {                    // 10 tane değeri topla.
    read_gyro_accel_vals((uint8_t *) &accel_t_gyro);
    x_accel += accel_t_gyro.value.x_accel;
    y_accel += accel_t_gyro.value.y_accel;
    z_accel += accel_t_gyro.value.z_accel;
    x_gyro += accel_t_gyro.value.x_gyro;
    y_gyro += accel_t_gyro.value.y_gyro;
    z_gyro += accel_t_gyro.value.z_gyro;
    delay(100);
  }
  x_accel /= num_readings;                                  // Ortalama al.
  y_accel /= num_readings;
  z_accel /= num_readings;
  x_gyro /= num_readings;
  y_gyro /= num_readings;
  z_gyro /= num_readings;
```

```
// Ham kalibre değerleri küresel (global) değişkenlere kaydedilmelidir.
  base_x_accel = x_accel;
  base_y_accel = y_accel;
  base_z_accel = z_accel;
  base_x_gyro = x_gyro;
  base_y_gyro = y_gyro;
  base_z_gyro = z_gyro;
  }
void setup()
{
  int error;
  uint8_t c;

// Motor Sürücüsüde A ve B isimli motorlar için tanımlamalar yapılmalıdır.
  pinMode(BRAKE_A, OUTPUT);              // A motorunun fren pini çıkış olarak tanımlanır.
  pinMode(DIR_A, OUTPUT);                // A motorunun yön pini çıkış olarak tanımlanır.
  pinMode(L_A, INPUT);                   // A motorunun sola dön sinyali giriş olarak
                                         tanımlanır.
  pinMode(R_A, INPUT);                   // A motorunun sağa dön sinyali giriş olarak
                                         tanımlanır.
  digitalWrite(BRAKE_A, LOW);            //A motorunun freni serbest bırakılır.

  pinMode(BRAKE_B, OUTPUT);              // B motorunun fren pini çıkış olarak tanımlanır.
  pinMode(DIR_B, OUTPUT);                // B motorunun yön pini çıkış olarak tanımlanır.
  pinMode(U_B, INPUT);                   // B motorunun yukarı dön sinyali giriş olarak
                                         tanımlanır.
  pinMode(D_B, INPUT);                   // B motorunun aşağı dön sinyali giriş olarak
                                         tanımlanır.
  digitalWrite(BRAKE_B, LOW);            // B motorunun freni serbest bırakılır.

  attachInterrupt(0, stabilization, RISING);  //Stabilizayon için yükselen kenar "0" pinine
                                              "stabilization" adında kesme fonksiyonu tanımlanır.
  pinMode(stabled,OUTPUT);               //stabled değişkenini(10. pini) çıkış olarak tanımlanır
  Serial.begin(19200);                   //Seri port 19200bps olarak tanımlanır ve başlatılır.
  Wire.begin();                          // I2C haberleşmesini tanımlanır ve başlatılır.

  error = MPU6050_read (MPU6050_WHO_AM_I, &c, 1);       // MPU-6050 tespit edilemezse hata ver.
  error = MPU6050_read (MPU6050_PWR_MGMT_2, &c, 1);  // MPU-6050 uyku modunda ise hata ver.
  MPU6050_write_reg (MPU6050_PWR_MGMT_1, 0);          // MPU-6050'yi uyku modundan çıkar.
```

```
//Kalibrasyonu başlatılıp, ilk değerler atanmalıdır.
 delay(2000);                                  // Jiroskopun kalibre edilmesi için 2000 mili sn. bekletilir.
                                               (Model tankın altındaki ana akım düğmesini açıp elimizi
                                               çekmek içi zaman tanınmalı.)
 calibrate_sensors();
 set_last_read_angle_data(millis(), 0, 0, 0, 0, 0, 0);          // ilk açı değerleri 0 atanmalıdır.
}

// loop'da kullanmak üzere jiroskoptan açıları okumak için fonksiyon oluşturulur.

void sensPos(){                                                 //sensPos isimli fonksiyon oluşturulur.
 int error;                                                     // error değişkeni tanımlanır.
 double dT;                                                     // Zaman değişkeni tanımlanır.
 accel_t_gyro_union accel_t_gyro;

// Ham değerler okunur.
 error = read_gyro_accel_vals((uint8_t*) &accel_t_gyro);
 unsigned long t_now = millis();                                // Dönüş hesaplamalarının okunmasındaki
                                                                zaman elde edilir.

// Jiroskop değerleri alınır ve derece/saniye formatına dönüştürülür.
 float FS_SEL = 131;                            // MPU-6050 datasheetine göre ham değerler sırasıyla 131'e
                                                bölünerek derece/sn değeri elde edilir.
 float gyro_x = (accel_t_gyro.value.x_gyro - base_x_gyro)/FS_SEL;
 float gyro_y = (accel_t_gyro.value.y_gyro - base_y_gyro)/FS_SEL;
 float gyro_z = (accel_t_gyro.value.z_gyro - base_z_gyro)/FS_SEL;

// İvmeölçer değerleri alınır.
 float accel_x = accel_t_gyro.value.x_accel;
 float accel_y = accel_t_gyro.value.y_accel;
 float accel_z = accel_t_gyro.value.z_accel;

// İvmeölçerden alınan değerler ile açılar hesaplanır.
 float RADIANS_TO_DEGREES = 180/3.14159;                // Radyan dereceye çevrilir.

float accel_angle_y = atan(-1*accel_x/sqrt(pow(accel_y,2) +
pow(accel_z,2)))*RADIANS_TO_DEGREES;
float accel_angle_x = atan(accel_y/sqrt(pow(accel_x,2) + pow(accel_z,2)))*RADIANS_TO_DEGREES;
float accel_angle_z = 0;
```

```
// Filtrelenmiş jiroskop açıları hesaplanır.
float dt =(t_now - get_last_time())/1000.0;                    // Zamandaki değişimi hesaplanır.
float gyro_angle_x = gyro_x*dt + get_last_x_angle();           // Tamamlayıcı filtre formülü
float gyro_angle_y = gyro_y*dt + get_last_y_angle();           // açı = omega x delta t + son açı
float gyro_angle_z = gyro_z*dt + get_last_z_angle();

// Jiroskopun kayma açıları hesaplanmalıdır. Burada filtrelenmiş veri kullanılarak açıdaki kayma
hesaplanır.
float unfiltered_gyro_angle_x = gyro_x*dt + get_last_gyro_x_angle();
float unfiltered_gyro_angle_y = gyro_y*dt + get_last_gyro_y_angle();
float unfiltered_gyro_angle_z = gyro_z*dt + get_last_gyro_z_angle();

// Tamamlayıcı Filtre algoritması uygulanır. Alfa örnekleme hızına bağlıdır ve kestirim (0.98) yapılır.
float alpha = 0.98;
float angle_x = alpha*gyro_angle_x + (1.0 - alpha)*accel_angle_x;
float angle_y = alpha*gyro_angle_y + (1.0 - alpha)*accel_angle_y;
float angle_z = gyro_angle_z;                                  // İvmeölçer "Z" açısı vermez. Bu yüzden
                                                               jiroskoptan alınan "Z" açısına eşitlenir.

// Hafızadaki veriler elde edilen son verilerle güncellenmelidir.
set_last_read_angle_data(t_now, angle_x, angle_y, angle_z, unfiltered_gyro_angle_x,
unfiltered_gyro_angle_y, unfiltered_gyro_angle_z);
angles[1] = angle_y;
angles[2] = angle_z;
}

// PID Kontrol Algoritması tanımlanır.
float K = 1.4;                                                 // error kat sayısı
int   Kp = 3;                                                  // Oransal kat sayı (Hedef noktasına ulaşma
                                                               zamanı ayarlanır.)
int   Ki = 1;                                                  // İntegral katsayısı (Hedef noktası
                                                               etrafındaki salınım ayarlanır.)
int   Kd =6;                                                   // Türevsel katsayı (Hedef noktasına oturma
                                                               zamanını ayarlanır.)
int last_err = 0;                                              // Hatanın son hali için değişken atanır.
int integrated_err = 0;                                        // Hata toplamları için değişken atanır.
int pTerm = 0, iTerm = 0, dTerm = 0;                           // PID için tek tek değişken atanır.
int GUARD_GAIN=1;                                              // Hata toplamı sınırı 1 olarak tanımlanır.
```

```
// PWM değeri hesaplama fonksiyonu tanımlanır.
int updatePid(int targetPosition, int currentPosition)   {
  int err = targetPosition - currentPosition;                    // Hata= kule pozisyonu - şimdiki pozisyon
  if (abs(err)>=3){                                              // Eğer hata mutlak 3 ise
  pTerm = Kp * err;                                              // Hatayı Kp ile çarp.
  integrated_err += err;                                         // Hataları topla.
  iTerm = Ki * constrain(integrated_err, -GUARD_GAIN, GUARD_GAIN); // Hataların sınırları
                                                                 belirlenir. (-1 ile +1 arası)(constrain
                                                                 sınırlama yapan fonksiyon)
  dTerm = Kd * (err - last_err);                                 // Hatadaki değişim Kd ile çarpılır. (Hatanın
                                                                 azalıp azalmadığı kontrol edilir yani
                                                                 motorlar doğru yönemi dönüyor kontrol
                                                                 edilir.)
  last_err = err;                                                // Hata güncellenir.
  return -constrain(K*(pTerm + iTerm + dTerm), -255, 255);} // Heseplanan PWM sonucuna uygun sinyal
                                                                 üretilir ve bu sonuç ±255 ile sınırlandır.
}

// Stabilizasyo sisteminin çalışması veya çalışmaması için "stabilization" adında kesme fonksiyonu
tanımlanmalıdır.
void stabilization()
{
  if ((millis()-LastDebounceTime)>DebounceTime){   // Bu koşul ile düğmenin debounce etkisi
                                                   filtrelenir.(millis komutu o anki zamanı saate
                                                   bakarak oluşturur.)
                                                   // Düğmye ikinci kez basılma 300 ms'den önce
                                                   yapılırsa sistem tarafından işleme alınmaz.
                                                   (300=debounce time yukarıda tanımlandı.)
    stab = !stab;                                  // Düğmeye basılınca mevcut çalışma durumunun
                                                   tersini uygulanır.
    LastDebounceTime=millis();                     // En son düğmeye basılan zaman tespit edilir.

    if (stab == HIGH)                              // Stabilizasyon aktif ise sensörün yönelimi
                                                   referans alınır.
    {
     digitalWrite (stabled,HIGH);                  // Led yakılır.
    }
     else if (stab == LOW)                         // Stabilizasyon aktif değilse
    {
     digitalWrite (stabled,LOW);                   // Led söndürülür.
```

```
   }
  }
}

// Kule ve namlu motorlarının hareket fonksiyonları tanımlanmalıdır.
void kuleMotYon(int tork){                  // Kule motorunu döndürmek için fonksiyon tanımlanır.
   if (tork >= 0){                          // Kule motoru için sağa dönüş koşulu belirlenir.(tork = PWM
                                            değeri)
   digitalWrite(DIR_A, LOW);                // Kule motoru için sağa dönüş yönü belirlenir.
  } else if ( tork < 0){                    // Kule motoru için sola dönüş koşulu belirlenir.(tork = PWM
                                            değeri)
   digitalWrite(DIR_A, HIGH); }             // Kule motoru için sola dönüş yönü belirlenir.
   tork=abs(tork);                          // PWM değeri pozitif olmalıdır. Bunun için mutlağı alınır.
   if (tork>=10){                           // Çok küçük PWM değerleri motora hareket vermeye
                                            yetmeyebilir o yüzden bu değer 10'un üzerinde olduğunda
                                            uygulanmalıdır.
   tork=map(tork, 0,255,150,255);           // PWM motoru döndürebilecek seviyeye yeniden eşlenir.
   analogWrite(PWM_A, tork);}               // Motoru döndür komutu gönderilir.
   else{
   analogWrite(PWM_A, 0);}
 }

 void namMotYon(int tork){                  // Namlu motorunu döndürmek için fonksiyon tanımlanır.
   if (tork >= 0){                          // Namlu motoru için aşağı dönüş koşulu belirlenir.
                                            (tork = PWM değeri)
   digitalWrite(DIR_B, HIGH);               // Namlu motoru için aşağı dönüş yönü belirlenir.
  } else if ( tork < 0){                    // Namlu motoru için yukarı dönüş koşulu belirlenir.
                                            (tork = PWM değeri)
   digitalWrite(DIR_B, LOW); }              // Namlu motoru için yukarı dönüş yönü belirlenir
   tork=abs(tork);                          // PWM değeri pozitif olmalıdır. Bunun için mutlağı alınır.
   if (tork>=10){                           // Çok küçük PWM değerleri motora hareket vermeye
                                            yetmeyebilir o yüzden bu değer 10'un üzerinde olduğunda
                                            uygulanmalıdır.
   tork=map(tork, 0,255,200,255);           // PWM motoru döndürebilecek seviyeye yeniden eşlenir.
   analogWrite(PWM_B, tork);}               // Motoru döndür komutu gönderilir.
   else{
   analogWrite(PWM_B, 0);}
 }
```

```
// Kumandadan kontrol için aşağıdaki fonksiyonlar tanımlanır.
void kuleMotSol(){                  // Kule motorunu sola döndürmek için fonksiyon tanımlanır.
 digitalWrite(DIR_A, HIGH);
 analogWrite(PWM_A, 255);
}
void kuleMotSag(){                  // Kule motorunu sağa döndürmek için fonksiyon tanımlanır.
 digitalWrite(DIR_A, LOW);
 analogWrite(PWM_A, 255);
}
void namMotYkr(){                   // Namlu motorunu yukarı döndürmek için fonksiyon tanımlanır.
 digitalWrite(DIR_B, LOW);
 analogWrite(PWM_B, 255);
}
void namMotAsg(){                   // Namlu motorunu aşağı döndürmek için fonksiyon tanımlanır.
 digitalWrite(DIR_B, HIGH);
 analogWrite(PWM_B, 255);
}
void kuleMotDur()  {                // Kule motorunu durdurmak için fonksiyon tanımlanır.
 analogWrite(PWM_A, 0);
}
void namMotDur()  {                 // Namlu motorunu durdurmak için fonksiyon tanımlanır.
 analogWrite(PWM_B, 0);
}

// Ana program fonksiyonu oluşturulur.
void loop(){
  sensPos();                        // Yukarıda tanımlanmış olan jiroskoptan açıları okuma fonksiyonu
                                    çağırılır. (Her loop'ta yeniden açıları okumak üzere)

// Ana programda kule motoru kontrolü için fonksiyon oluşturulur.
 if (analogRead(R_A) >= 500){       // Kumandadan gelen R_A sinyali 500'den büyük ise kule sağa
                                    döndürülür.
   while(analogRead(R_A) >= 500)
          {kuleMotSag(); }          //while kodu ile sinyal geldiği müddetçe kule motoru sağa döndürülür.
   kuleMotDur();                    // Sinyal kesildiğinde kule motoru durdurulur.
   sensPos();                       // Kulenin yeni açısı okunur.
   targetPosKule=angles[2];}        // Tespit edilen yeni açı yeni z0 değeri olarak atanır.
  else if (analogRead(L_A) >=500){  // Kumandadan gelen L_A sinyali 500'den büyük ise kule sola
                                    döndürülür.
```

```
  while(analogRead(L_A) >= 500)
        {kuleMotSol();}          // while kodu ile sinyal geldiği müddetçe kule motoru sola döndürülür.
  kuleMotDur();                  // Sinyal kesildiğinde kule motoru durdurulur.
  sensPos();                     // Kulenin yeni açısı okunur.
  targetPosKule=angles[2];}      // Tespit edilen yeni açı yeni z0 değeri olarak atanır.
  else if (stab==HIGH){          // Kulenin stabilizasyonda kalması sağlanır.
  kuleMotYon(updatePid(targetPosKule, angles[2]));          // PID fonksiyonu çalıştırılır. Çıkan PWM
                                 sonucu kule motorunun hareketi için girdi olarak alındıktan sonra kule
                                 motoru fonksiyonu çalıştırılır.
 }

// Ana programda namlu kontrolü için fonksiyon oluşturulur.
 if (analogRead(U_B) >= 400){    // Kumandadan gelen U_B sinyali 400'den büyük ise namlu aşağı
                                 indirilir.
  while(analogRead(U_B) >= 400)
        {namMotYkr();}           // while kodu ile sinyal geldiği müddetçe namlu aşağı indirilir.
  namMotDur();                   // Sinyal kesildiğinde namlu motoru durdurulur.
  sensPos();                     // Namlunun yeni açısı okunur.
  targetPosNam=angles[1];}       // Tespit edilen yeni açı yeni y0 değeri olarak atanır.
 else if (analogRead(D_B) >= 400){     // Kumandadan gelen D_B sinyali 400'den büyük ise namlu
                                 yukarı kaldırılır.
  while(analogRead(D_B) >= 400)
        {namMotAsg();}           // while kodu ile sinyal geldiği müddetçe namlu yukarı kaldırılır.
  namMotDur();                   // Sinyal kesildiğinde namlu motoru durdurulur.
  sensPos();                     // Namlunun yeni açısı okunur.
  targetPosNam=angles[1];}       // Tespit edilen yeni açı yeni y0 değeri olarak atanır.
 else if (stab==HIGH){           // Namlunun stabilizasyonda kalması sağlanır.
   namMotYon(updatePid(targetPosNam, angles[1]));          // PID fonksiyonu çalıştırılır. Çıkan PWM
                                 sonucu namlu motorunun hareketi için girdi olarak alındıktan sonra
                                 namlu motoru fonksiyonu çalıştırılır.
  }
 }
```

// "MPU6050_read" fonksiyonu $I^2C$ iletişim protokolünden çoklu bayt okumak için kullanılan genel bir fonksiyondur. Sadece veri okumak için kullanılabilir. Tek bayt okumak için fonksiyon barındırmaz.

```
int MPU6050_read(int start, uint8_t *buffer, int size)     // Arduino ile MPU-6050 sensörü arasında
```

iletişim kurmak için gerekli olan $I^2C$ protokolü tanımlanır.

```
{
 int i, n, error;
```

```
 Wire.beginTransmission(MPU6050_I2C_ADDRESS);
 n = Wire.write(start);
 if (n != 1)
  return (-10);
 n = Wire.endTransmission(false);                    // I2C veri yolunu beklet.
 if (n != 0)
  return (n);
Wire.requestFrom(MPU6050_I2C_ADDRESS, size, true);   // Veri okunduktan sonra I2C veri yolu
                                                     boşaltılır.
 i = 0;
 while(Wire.available() && i<size)
 {
  buffer[i++]=Wire.read();
 }
 if ( i != size)
  return (-11);
  return (0);                                        // Hata yok bilgisi kontrol edilir ve geri
                                                     besleme sağlanır.
}

//"MPU6050_write" fonksiyonu I2C cihazına çoklu bayt yazmak için kullanılan genel bir fonksiyondur.
// Sadece tek register yazılacaksa "MPU_6050_write_reg()" isimli fonksiyon kullanılmalıdır.
// Parametreleri:
// start: Başlangıç adresini belirtir (register için tanımlama kullanılır).
// pData: Veri yazmak için kullanılan işaretleyicidir.
// size: Yazılacak bayt sayısıdır.
// Tek registera veri yazılacaksa veriye bir işaretleyici kullanılmalıdır ve "size" tek bir bayt olmalıdır.

int MPU6050_write(int start, const uint8_t *pData, int size)   // Çoklu bayt gönderebilmek için gerekli
                                                               olan fonksiyondur.
{
 int n, error;
 Wire.beginTransmission(MPU6050_I2C_ADDRESS);
 n = Wire.write(start);                                        // Yazımın yapılacağı başlangıç adresi.
 if (n != 1)
  return (-20);
 n = Wire.write(pData, size);                                  // Veri baytları adrese yazılır.
 if (n != size)
  return (-21);
```

```
  error = Wire.endTransmission(true);                    // I2C veri yolu boşaltılır.
  if (error != 0)
    return (error);
    return (0);                                          // Hata yok bilgisi kontrol edilir ve geri besleme
                                                         sağlanır.
}
```

// "MPU6050_write_reg" fonksiyonu tek register yazabilmek için kullanılan ilave bir fonksiyondur. "MPU_6050_write()" fonksiyonuna ek olarak çalışan bir pakettir, sadece tek register yazmayı kolaylaştırmak için pratik bir fonksiyondur. Örnek: MPU-6050 sensörünün uyku modundan çıkartılması için sıfır yazılması.

```
int MPU6050_write_reg(int reg, uint8_t data)
{
  int error;
  error = MPU6050_write(reg, &data, 1);
  return (error);
}
```

# KAYNAKLAR


1. Teoman ENGİN, ***Türk Ordusunda Zırhlı Birlikler*****,** Yüksek Lisans Tezi, Kırıkkale, 2007
2. Ufuk SUADİYE, Sıtkı ÖZTÜRK, İsmet KANDİLLİ, ***Tank Namlu Stabilizasyon Uygulaması***, Fırat Üniversitesi Elektrik-Elektronik ve Bilgisayar Sempozyumu, Elazığ, 2011
3. http://www.aselsan.com.tr/content.aspx?mid=119&oid=639, Erişim Tarihi: 23.09.2013
4. Binroth, W., Cornell, G. A., & Presley, R. W., ***Closed-Loop Optimization Program for the M60A1 Tank Gun Stabilization System***, Bendix Research Labs, Southfield Mi, 1975
5. Pastrick, H. L., ***An Analysis and Simulation of the M60AIE2 Tank Main Gun's Elevation Control System***, Army Missile Research Development And Engineering Lab, Redstone Arsenal Al, 1969
6. http://hilmi.trakya.edu.tr/ders_notlari/Otomatik_kontrol/Otomatik_Kontrol_2.pdf, Erişim Tarihi: 19.11.2013
7. Benjamin C.KUO, ***Otomtik Kontol***, Literatür Yayıncılık, İstanbul, 1999
8. Prof.Dr. İbrahim Deniz AKÇALI, ***Otomatik Kontrol***, Karahan Kitabevi, Adana, 2011
9. Türker KARAYUMAK, ***Modelıng And Stabılızatıon Control Of A Maın Battle Tank***, ODTÜ Ankara, 2011
10. http://arduino.cc/en/Main/ArduinoBoardUno, Erişim Tarihi: 05.09.2013
11. http://arduino.cc/en/Guide/Windows, Erişim Tarihi: 10.09.2013
12. http://arduino.cc/en/Reference/HomePage, Erişim Tarihi: 11.09.2013
13. ***MPU-6000/MPU-6050 Product Specification***, Document Number: PS-MPU-6000A-00, Revision: 3.1, U.S.A., 2011
14. http://playground.arduino.cc/Main/MPU-6050 Erişim Tarihi: 05.09.2013
15. http://www.starlino.com/imu_guide.html, Erişim Tarihi: 11.10.2013
16. http://www.starlino.com/dcm_tutorial.html, Erişim Tarihi: 11.10.2013
17. T.C. Millî Eğitim Bakanlığı, ***Elektrik Elektronik Teknolojisi Elektrikli Ev Aletlerinde D.C. Motorlar***, Ankara, 2007

18. http://www.elektrikrehberiniz.com/elektrik-motorlari/dc-motor-nedir-454/, Erişim Tarihi: 05.01.2014
19. http://arduino.cc/en/Main/ArduinoMotorShieldR3, Erişim Tarihi: 15.01.2014
20. ***L298 Dual Full-Bridge Driver Product Specification***, STMicroelectronics, Italy, 2000
21. İsmail ARSLAN, ***L298 Motor Sürücü Entegresinin Kullanımı,*** Hunrobotx – Makaleler, Hacettepe Üniversitesi Robot Topluluğu, Ankara, 2009
22. http://www.vstank.com/vspro/product_M1A2_desert/product_M1A2_desert.html, Erişim Tarihi: 12.02.2014
23. http://www.technomodel.com/urun/VSTANK12A/US-M1A2-ABRAMS-NATO-AIR-SOFT, Erişim Tarihi: 18.02.2014

# ÖZGEÇMİŞ

1984 yılında Bulgaristan'da doğdu. İlköğrenimini Bilecik'te tamamladı. 1999 yılında girdiği Işıklar Askeri Lisesi'nden 2003 yılında mezun oldu. Aynı yıl girdiği Kara Harp Okulu'ndan 2007 yılında mezun oldu. 2010 yılı Eylül Ayı'nda, Trakya Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Ana Bilim Dalı Yüksek Lisans Programına başladı. 2007 yılından itibaren Türk Silahlı Kuvvetlerinde Tankçı Subayı olarak görev yapmaktadır.