

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAVACILIK BİLİMİ VE TEKNOLOJİLERİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

REDÜKTÖRLÜ DC MOTOR İLE YÖNETİLEN 6 EKSEN
HAREKETLİ UÇUŞ SİMÜLATÖRÜNÜN TASARIMI,
ANALİZİ VE UYGULAMASI

BURAK HAMDİ TUFAN

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAVACILIK BİLİMİ VE TEKNOLOJİLERİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

REDÜKTÖRLÜ DC MOTOR İLE YÖNETİLEN 6 EKSEN
HAREKETLİ UÇUŞ SİMÜLATÖRÜNÜN TASARIMI,
ANALİZİ VE UYGULAMASI

BURAK HAMDİ TUFAN

Doç. Dr. Satılmış ÜRGÜN
Danışman, Kocaeli Üniv.

.....

Prof. Dr. Faruk ARAS

Jüri Üyesi, Kocaeli Üniv.

.....

Dr. Öğr. Üyesi Serhat Aydoğan

Jüri Üyesi, Kapadokya Üniv.

.....

Tezin Savunulduğu Tarih: 14.01.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez, Kocaeli üniversitesi Fen Bilimleri Enstitüsü Havacılık Bilimi ve teknolojileri Programı’nda yüksek lisans tezi olarak hazırlanmıştır.

Bu çalışmanın gerçekleştirilmesinde göstermiş olduğu sabır ile birlikte her aşamasında vermiş olduğu destek ve katkılarından ötürü danışman hocam Doç. Dr. Satılmış ÜRGÜN’e sonsuz teşekkür ve minnetlerimi sunarım.

Bu tez çalışmasını FYL-2020-2054 proje numarası ile desteklerinden dolayı Kocaeli Üniversitesi | BAP Koordinasyon Birimine ve Fen Bilimleri Enstitüsüne teşekkür ederim.

Son olarak 2008-2012 yılları arasında Bolu Dörtdivan Anadolu Öğretmen Lisesinde, lise eğitimi gördüğüm yıllarda, bilişim teknolojileri ve yazılım alanlarında beni yetiştiren, benden desteğini hiçbir şekilde esirgemeyen ve bugüne kadar yaptığım bütün çalışmalarda dolaylı olarak katkısı bulunan Sayın Faruk YILDIRIM hocama sonsuz teşekkürü borç bilirim.

Tezimi; beni yetiştiren, her zaman bana destek olan; maddi ve manevi hiç bir fedakârlıktan kaçınmayan, aileme ithaf ediyorum.

Şubat - 2021

Burak Hamdi TUFAN

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
SİMGELER VE KISALTMALAR DİZİNİ	vi
ÖZET.....	viii
ABSTRACT.....	ix
GİRİŞ	1
1. HAREKETLİ UÇUŞ PLATFORMLARI	5
1.1. Uçuş Simülatörleri Tarihçesi.....	5
1.2. Simülatörlerin Kullanım Alanları.....	7
1.3. Simülatörlerin Avantajları.....	8
1.3.1. Maliyet	9
1.3.2. Güvenlik gereksinimleri	9
1.3.3. Çevresel etkiler.....	9
1.4. Simülatörlerin Elemanları	10
1.4.1. Kokpit.....	10
1.4.2. Görsel sistem	10
1.4.3. Aletler.....	11
1.4.4. Hareket sistemi	11
1.4.5. Ses sistemi	11
1.5. Günümüzde Uçuş Simülatörleri	11
2. HAREKETLİ UÇUŞ PLATFORMU ANALİZLERİ	13
2.1. Matematiksel Eşitlikler	13
2.2. Dönüş Eşitlikleri	13
2.2.1. Newton-Euler matrisleri	14
2.2.2. Tait-Bryan matrisleri	14
2.2.3. İki eksenli hareketin lineer matris gösterimi	16
2.2.4. Üç eksenli hareketin lineer matris gösterimi	18
2.3. Matematiksel Analizler	20
2.3.1. Kinematik analiz	20
2.3.2. İleri kinematik analizi	21
2.3.3. Ters kinematik analizi	21
2.3.4. İleri kinematik ve ters kinematik arasındaki farklar	21
3. HAREKETLİ PLATFORMLARIN KİNEMATİK ANALİZ İLE MODELLENMESİ	23
3.1. Stewart Platformu Modellemesi	23
3.2. Kullanılan Yardımcı Programlar	34
3.2.1. Matlab	34
3.2.2. Solidworks	36
4. ELEKTRONİK KONTROL DEVRESİ TASARIMI	38
4.1. PWM Sinyali Nedir	38
4.2. Mikroişlemci Seçimi	39
4.3. PID Uygulaması	39

4.4. PWM Sinyali Oluřturma	45
4.5. STM32F429 PWM Oluřturma	46
4.5.1. STM32F429 PID ile PWM sinyali oluřturma	48
4.6. DC Motor Nedir	50
4.7. DC Motor Kontrolü ve H-Bridge Devresi	50
4.8. PID Kontrol ile DC Motor kontrolü	52
5. HAREKETLİ UÇUŐ PLATFORMUN TASARIMI VE ÜRETİMİ	56
5.1. Malzemelerin Üretilmesi	56
5.2. Elektronik Kontrol Birimi Montajı	58
6. TASARLANAN 6DOF PLATFORMUN GERÇEKLENMESİ	63
7. SONUÇLAR VE ÖNERİLER	81
KAYNAKLAR	83
EKLER	85
KİŐİSEL YAYIN VE ESERLER	89
ÖZGEÇMİŐ	90

ŞEKİLLER DİZİNİ

Şekil 1.1.	The ruggles orientator (1917).....	5
Şekil 1.2.	Link trainer (1927-1929).....	6
Şekil 1.3.	Cessna 172 simülatör kokpiti	10
Şekil 1.4.	Boeing 777 uçuş simülatörü kokpiti	12
Şekil 2.1.	Küresel koordinat gösterimi	13
Şekil 2.2.	Bir cismin 3 ekseninde döndürülmesi	15
Şekil 2.3.	Bir noktanın 2 ekseninde gösterimi.....	16
Şekil 2.4.	Hareket ettirilen bir noktanın 2 ekseninde gösterimi	17
Şekil 2.5.	Bir noktanın 3 ekseninde gösterimi.....	18
Şekil 3.1.	6DOF Hareketin gösterimi.....	24
Şekil 3.2.	Bir manipülatörün hareket vektörlerinin gösterimi	25
Şekil 3.3.	Her bir servo motorun x-y eksenindeki dönüş hareketi	26
Şekil 3.4.	Motor çiftinin hareket eksenleri	27
Şekil 3.5.	Bir noktanın 3 ekseninde kartezyen olarak gösterimi	28
Şekil 3.6.	Hareket platformunun yandan görünümü	32
Şekil 3.7.	Hareket platformunun tepeden görünümü.....	33
Şekil 3.8.	Simülatör açısal değerleri MATLAB kod bloğu.....	34
Şekil 3.9.	Açıların radyan tipine dönüştürülmesi MATLAB kod bloğu	35
Şekil 3.10.	Simülatör bağlantılarının tanımlanması MATLAB kod bloğu	35
Şekil 3.11.	Dönüş matrisinin MATLAB programına tanımlanması	35
Şekil 3.12.	Dönüş matrisi ile bağlantı noktalarının MATLAB ile hesaplanması	36
Şekil 3.13.	Base kısımdan hareketli platforma olan vektörlerin hesaplanması	36
Şekil 3.14.	Platformun üst tablasının Solidworks çizimi ve üretilmiş hali	36
Şekil 4.1.	Örnek bir PWM sinyali gösterimi	38
Şekil 4.2.	Duty Cycle gösterimi	38
Şekil 4.3.	PID uygulaması diyagramı.....	40
Şekil 4.4.	Örnek bir PI uygulaması diagramı	41
Şekil 4.5.	Simulink uygulaması örnek PID diagramı	42
Şekil 4.6.	Simulink PID uygulaması çıktısı.....	43
Şekil 4.7.	Simulink PI uygulaması çıktı ekranı	43
Şekil 4.8.	Simulink P uygulaması çıktı ekranı	44
Şekil 4.9.	Simulink PID uygulaması Sinüs dalgası takibi	45
Şekil 4.10.	TIMER konfigürasyonu kod bloğu	47
Şekil 4.11.	Sensör verisi okuduktan sonra ölçeklendirip PWM çıktısı olarak atama	47
Şekil 4.12.	TIMER PWM çıktı örneği.....	48
Şekil 4.13.	Kesme içerisinde PID hesaplama fonksiyonları.....	49
Şekil 4.14.	H-Bridge diyagramı.....	51
Şekil 4.15.	Örnek IGBT H-Bridge devresi şeması	52
Şekil 4.16.	TIMER ve PWM başlatma kod bloğu.....	53

Şekil 5.17. TIMER ve PWM başlatma metodundan sonra oluşan PWM sinyali	54
Şekil 5.18. MATLAB Simulink PID şeması	54
Şekil 4.19. PID hesaplama fonksiyonu kod bloğu	55
Şekil 5.1. Solidworks ortamında hazırlanan Hareketli platform Modeli.....	56
Şekil 5.2. DC motor montajı.....	57
Şekil 5.3. Üretilen base kıs	57
Şekil 5.4. TheSim Interface yazılımı ekran görüntüsü	59
Şekil 5.5. Visual display programı	60
Şekil 5.6. TheSim Interface yazılımı ve Microişlemci hata ayıklama ekranı	61
Şekil 5.7. Elektronik kontrol sistem çalışma şeması	61
Şekil 6.1. Hareketli platformun ilk test aşaması.....	63
Şekil 6.2. Üretilen platformun yaw eksenli hareketi	64
Şekil 6.3. Platform yaw hareketi önden görünüş.....	65
Şekil 6.4. Yaw ve öne kayma hareketi	66
Şekil 6.5. Platformun sağa kayma hareketi	67
Şekil 6.6. Platformun yaw ve pitch hareketi.....	68
Şekil 6.7. Platformun 3 eksenli hareketi	69
Şekil 6.8. Platform burun yukarı hareketi	70
Şekil 6.9. Platformun burun aşağı ve sağa dönüş hareketi	71
Şekil 6.10. Hareketli platform pitch-up hareketi	72
Şekil 6.11. Hareketli platformun roll hareketi	73
Şekil 6.12. Platform burun aşağı ve roll hareketi	74
Şekil 6.13. Platformun roll ve yaw hareketi	75
Şekil 6.14. Platformun yatış, dalış ve sapma hareketlerini birlikte yapması.....	76
Şekil 6.15. Platform yatış açısı değerlerinin limit testi	77
Şekil 6.16. Motor limit değeri testi.....	78
Şekil 6.17. Güç kaynakları bağlantısı.....	79
Şekil 6.18. Alt sistem kablolu örneği.....	80

SİMGELER VE KISALTMALAR DİZİNİ

ϕ	: (x-y) ile düzlemleri üzerinde yapılan açı
φ	: (y-z) ile düzlemleri üzerinde yapılan açı
α	: Roll açısı (Yatış Açısı)
β	: Pitch Açısı (Dalış Açısı)
θ	: Yaw Açısı (Sapma Açısı)
A_i	: Servo kolu ile rodun birleştirildiği noktalar
a_i	: Servo kolu ile rod birleşim noktaları vektörleri
B_i	: Servoların Base kısma bağlandığı noktalar
b_i	: Servoların base kısım vektörleri
d_x	: Bir noktanın x eksenine olan uzaklığı
d_y	: Bir noktanın y eksenine olan uzaklığı
d_z	: Bir noktanın z eksenine olan uzaklığı
h_0	: Başlangıç durum yüksekliği
K_p	: Proportional Katsayısı (Oranlama Katsayısı)
K_i	: İntegral katsayısı
K_d	: Derivative Katsayısı (Türev Katsayısı)
l_i	: B_i ile P_i noktaları arasındaki vektör
P	: Proportional (Oransal)
I	: Integral
D	: Derivative (Türevsel)
P_i	: Platformun Rod ile bağlandığı noktalar
p_i	: Platform-Rod arası bağlantı vektörleri
R	: Üç eksen etrafında dönüş
r	: Küre merkezine olan uzaklığı
R_x	: x eksen etrafındaki dönüş matrisi
R_y	: y eksen etrafındaki dönüş matrisi
R_z	: z eksen etrafındaki dönüş matrisi
r_x	: Bir noktanın x kartezyen eksenine olan uzaklığı
r_y	: Bir noktanın y kartezyen eksenine olan uzaklığı
r_z	: Bir noktanın z kartezyen eksenine olan uzaklığı
S	: Rod uzunluğu
T	: Translation Vektörü
X	: X eksenine olan açısal değer
Y	: Y eksenine olan açısal değer
Z	: Z eksenine olan açısal değer
x'	: x eksen etrafındaki dönüşten sonra x eksenine olan uzaklık
y'	: y eksen etrafındaki dönüşten sonra y eksenine olan uzaklık
z'	: z eksen etrafındaki dönüşten sonra z eksenine olan uzaklık

Kısaltmalar

Cos	:	Cosinus
DC	:	Direct Current (Doğru Akım)
DMA	:	Direct Memory Access (Direk Bellek Erişimi)
HAL	:	Hardware Access Layer (Donanım Erişim Katmanı)
KHz	:	Kilohertz
MHz	:	Megahertz
NVIC	:	Nested Vector Interrupt Controller (İç içe Vektör Kesme Denetleyicisi)
RVDT	:	Rotary Variable Displacement Sensor (Dairesel Değişken Hareket Sensörü)
Sin	:	Sinus
VDC	:	Direct Current Voltage (Doğru Akım Gerilimi)



REDÜKTÖRLÜ DC MOTOR İLE YÖNETİLEN 6 EKSEN HAREKETLİ UÇUŞ SİMÜLATÖRÜNÜN TASARIMI, ANALİZİ VE UYGULAMASI

ÖZET

Yüksek güvenli hava taşımacılığında pilotların karşılaşabileceği hava olaylarına hazırlıklı olmaları gerekmektedir. Bu amaçla geliştirilen uçuş simülatörleri ticari hava taşımacılığının ciddi derecede ilerlemesinin en büyük sebeplerinden birisi olup, ticari havayolu taşımacılığı şirketlerinin pilotaj eğitim maliyetlerini ciddi oranda azaltmıştır. Aynı zamanda uçak üretim firmalarının test operasyonlarını kolaylaştırıp, üretim maliyetlerinin de düşmesini sağlamıştır. Geliştirilen uçuş simülatörleri pilotların uçuş operasyonu esnasında karşılaşabileceği rüzgar kesmesi, şiddetli kuyruk rüzgarı, türbülans, havada tutunamama, tek motora kalma gibi etkileri oluşturabilmesi gerekmektedir. Görsel eğitimin yanı sıra pilotluk eğitiminin hareketli sistemler ile desteklenmesi insani duyularında uçuşa hazır olması konusunda daha olumlu sonuçlar verecektir.

Bu çalışma 6 eksen hareketli bir Stewart Platformu tasarımı çalışmasıdır. Bu çalışmada döner-dişli tip motorlar kullanılmış olup hareketli platform bu motorun uçlarına monte edilen tek eklemlili rotalarla sağlanmaktadır. Kinematik analizler yapılarak oluşturulan matematiksel model ile birlikte yapıyı ani hareketlenmelerden korumak ve bununla birlikte en yumuşak ve en hızlı şekilde hedef açılara götürebilmek adına PID kapalı çevrim kontrol mekanizması eklenmiştir. Bilgisayar ortamında geliştirilen bu sistemin, hayata geçirmek adına uygun malzemeler seçilmiş, modellendiği şekilde üretilmiş ve elektronik kontrol sistemi ve geliştirilen ara yüz yazılımı ile denemeler yapıldıktan sonra montaj tamamlanmıştır.

Bu tez çalışması, Kocaeli Üniversitesi | BAP Koordinasyon Birimi tarafından FYL-2020-2054 proje numarası ile desteklenmektedir.

Anahtar Kelimeler: 6DOF Hareket, Asenkron Sistem Ara Yüzü, Stewart Platformu, Ters Kinematik Hesaplamaları, Uçuş Simülatörü.

DESIGN, ANALYSIS AND APPLICATION OF 6 AXIS MOTION FLIGHT SIMULATOR MANAGED BY REDUCER DC MOTOR

ABSTRACT

Pilots must be prepared for the weather events they may encounter in high security air transport. Flight simulators developed for this purpose and they are one of the biggest reason for the serious progress of commercial air transportation. Flight simulators has significantly reduced the pilotage training costs of commercial air transportation companies. At the same time, it facilitated the test operations of aircraft manufacturing companies and reduced production costs. The developed flight simulators have to be able to create effects such as wind shear, strong tail wind, turbulence, inability to hold in the air, engine failure that pilots may encounter during flight operation. Supporting pilot training with mobile systems in addition to visual training will give more positive results in human senses to be ready for flight.

This study is a 6-axis Stewart Platform design study. In this study, rotary-gear type motors are used and the moving platform is provided with single-joint rods mounted on the ends of this motor. Along with the mathematical model created by performing kinematic analysis, a PID closed-loop control mechanism has been added to protect the structure from sudden movements and to bring it to the target angles in the smoothest and fastest way. In order to implement this system developed in computer environment, materials were selected, produced as modeled, and after trials were made with the electronic control system and the developed interface software, the assembly was completed.

This thesis work is supported by the Kocaeli University | BAP Coordination Unit with the project number FYL-2020-2054.

Keywords: 6DOF Motion, Asynchronous System Interface, Stewart Platform, Inverse Kinematic Calculations, Flight Simulator.

GİRİŞ

Havacılık simülatörleri gerçekte var olan bir uçağın veya uçak sisteminin operasyonel eğitimi için geliştirilen daha ucuz ve daha güvenli araçlardır. Simülatörlerin amacı sistemin gerçek operasyonda maruz kaldığı tepkileri üreterek operatörüne bilinç kazandırması ve gerçek sistemi yönetirken gerçekleşen durumlarda nasıl tepki vereceğini bilmesini sağlamaktır. Bu algoritmaları geliştirmek oldukça zordur. Çünkü sabit bir işlem olmayıp aksine pilotun hareketleri ve ara yüz yazılımından gelen geri beslemelere göre manipülatörlerin hareketi sağlanmaktadır. Bütün bu hareketler, eğitim alan pilot adaylarının algılama sistemlerine uçuyor hissi verebilmelidir [1]. Bu yüzden oluşturulan görüntünün hareket sistemleri ile birlikte eş zamanlı olarak çalışması gerekmektedir. Ayrıca bunların yanında ses desteği de algıyı güçlendirecektir. Pilotların sert hava koşullarına karşı hazırlıklı olmaları gerekmektedir. Bu yüzden pilot eğitiminde kullanılan simülatörlerin bu tarz koşullarda gerçekteki bir uçağın verebileceği tepkileri verebilmesi gerekmektedir. Bu sayede pilotlar bu tarz durumların oluşabilme ihtimaline karşı eğitilmiş olacaklardır. Örnek olarak iniş ve kalkış esnasında gerçekleşen rüzgâr kesmesi gibi hava olayları, uçuş güvenliği için büyük tehlike oluşturmaktadır. Uçak rüzgar kesmesi, havada tutunamama, türbülans vb. gibi durumlara maruz kaldığında dengesizleşecek ve uçuşu tehlikeye girecektir.

Pilot eğitimi için gerekli özellikleri karşılaması gereken bir simülatörün, ani hareketler yapabilecek kapasitede olması gerekmektedir. Bu yüzden fiziksel dayanımı yüksek malzemeler seçilerek, yeterli hızlarda hareket edebilen manipülatörler kullanılmalıdır. Bu tarz simülatörler genellikle 6 adet doğrusal veya döner tip manipülatör ile hareket ettirilmektedir. Bu tip manipülatörler bir tarafı sabit zemine, diğer tarafı ise hareket ettirilecek olan platforma bağlanmaktadır. Hareket için gerekli kinematik hesaplamalar bu sistemin boyutlarına ve parçaların bağlantı konumlarına göre yapılmaktadır.

Bu çalışmada döner tipli manipülatörler kullanılarak dayanıklı, verimli ve dinamik bir simülatör geliştirilecektir. Araştırmada kullanılacak olan manipülatör döner tipli olduğu için desteğinin her iki ucu hareketli zeminlere bağlanmış olacak ve doğrusal tipli manipülatörün aksine Stewart platformu tasarımında kullanılan ters kinematik dönüşüm fonksiyonundaki zemin bağlantı noktalarının da değerleri sürekli değişken bir hal alacaktır. Ayrıca manipülatörlerin ve genel yapının limitleri de modele aktararak uygun ve etkili bir hareket algoritması geliştirilecektir. Fiziksel dayanımı arttırmak ve keskin hareketlerden yapının zarar görmesini engellemek amacıyla motor kontrollerinde PID denklemlerinden yararlanılacaktır [2].

Geliştirilen her dinamik sistem analiz edilmelidir [3]. Bir sistemi inşa etmeden önce analiz etmek ve eksiklikleri üretim aşamasından önce düzeltmek, üretimin maliyetini azaltarak güvenilirliğini arttıracaktır. Karmaşık dinamik sistemler genellikle birkaç tane yardımcı sistemlerden oluşur. Bu çalışmanın tasarımında kullanılan yardımcı sistemler ayrı ayrı hareket, konum algılama ve hesaplama, ara yüz ve haberleşme sistemlerinden oluşmaktadır. Merkezi işlem birimi, haberleşme sistemi tarafından ara yüz yazılımı aracılığı ile MS Flight Simulator uygulaması üzerinden anlık uçuş parametrelerini alarak platformun hedef konumunu hesaplayacak. Ayrıca bu hesaplama göre motorların olması gereken açıları konum algılama sistemi ile algılayıp, gerekli açılara ulaşması için hareket sistemine gerekli parametreler gönderilecektir.

Bu tez çalışmasında bir hareketli uçuş simülatörü tasarlanacaktır. Hareketli platformun hareket sistemi 6-DOF Stewart Platformu üzerine kurulmuştur [4]. Uçuş simülatörü için gereken hareketli bir platform, 6 ekseninde hareket edebilmelidir. Bu eksenler yatış, dalış ve sapma hareketleri ile birlikte; X, Y ve Z eksenleri üzerinde kayma hareketleridir. Gerçek bir uçak, uçuş esnasında kendi dengesini bu altı eksen üzerinde sağlar. Bu tezin konusu olan prototip uçuş simülatör tasarımı, pilot eğitiminde kullanılabilmek amacıyla geliştirilecektir. Bu sayede platformun hareketleri ile gerçek uçağın hareketleri simüle edilerek, pilot adayının gerçek uçuşa karşılaşabileceği durumlar için refleks oluşturulup hazır olmaları sağlanmış olunacaktır.

Bu çalışma ile birlikte gerçek bir 6 eksenli uçuş simülatör platformu üretimi üzerinde çalışılmıştır. Üretilen platformun matematiksel modellemesi ile matematiksel olarak modellenen tasarımın birebir aynı olmasına dikkat edilmiştir[5]. Bu tasarım üzerinde çalışılırken MATLAB ve SOLIDWORKS programlarından yardım alınmıştır. Ayrıca simülatördeki manipülatörlerin kontrolü için ARM Cortex-M4 tabanlı STM32 işlemcileri kullanıldı. ARM mimarisine sahip işlemci üzerinde çalışacak olan gömülü sistem yazılımı düşük seviyeli C dili ve HAL kütüphaneleri kullanılarak geliştirilmiştir. Geliştirme ortamı olarak uVision Keil programı seçilmiştir.

Görüntü oluşturmak için Microsoft tarafından geliştirilen Flight Simulator X uygulaması kullanılmaktadır. Bu uygulama ile gömülü sistem arasında QT Framework ve C++ programlama dili kullanılarak bir ara yüz yazılımı geliştirilmiştir. Bu ara yüz yazılımı iki yönlü çalışmaktadır. Gömülü sistem aracılığı ile gelen kullanıcı komutlarını MS Flight Simulator X programına aktarma göreviyle beraber, MS Flight Simulator X programındaki anlık uçuş ve diğer uçak parametrelerini sistemdeki diğer birimlere aktarmaktır. Gerekli kinematik hesaplamaların yapılabilmesi ve manipülatörlere uygulanabilmesi için MS Flight Simulator X uygulamasından anlık olarak uçuş parametrelerini sistem belleği üzerinden okuyarak hesaplamaların yapılacağı merkezi işlem birimine aktarılmıştır. Manipülatörlerin, alınan bu değerler ile yapılacak olan kinematik hesaplamaları sonucu oluşan değerlere doğru ve efektif bir şekilde ulaşması hareket sistemini yöneten işlemcinin sorumluluğundadır.

Hareket kontrol sisteminin, platformun hareketlerini sürekli olarak doğru bir biçimde oluşturabilmesi için motorların anlık açısal değerlerini, ilgili sensörlerden sürekli olarak alması gerekmektedir. Bu esnada ara yüz yazılımından gelen hedef yatış değerleri ile sürekli kinematik hesaplamalar yapılmaktadır ve bu hesaplamalar sonucu oluşan açısal değerler ile motorların anlık açısal değerleri sürekli karşılaştırılmaktadır. Aradaki farklar giderilerek motorlar hedef değerlerde tutulmaktadır. Bir manipülatör için gelen hedef değer ile o anki değeri arasındaki fark büyük ise motorun hedefe ulaşması için yapacağı hareket çok büyük bir tepki kuvveti oluşturacak ve bir süre sonra sisteme mekaniksel açıdan olumsuz etkileri olacaktır. Ayrıca en önemlisi ani hareket ve durmalar sonucunda hedef değerden sapmalar meydana gelebilir. Bunu engellemek için çok ani hareketlerden

kaçınılmalıdır. Bu yüzden motor kontrol sisteminde PID hesaplamaları kullanılmıştır [6]. PID hesaplamaları ile motorun kontrolü esnasında hız ayarlaması için PWM sinyali oluşturulmuştur. PWM sinyalinin değeri motorun hızını belirleyeceğinden, PWM sinyali motor konumu ve hedef değer arasındaki farka göre ayarlanmaktadır. PID fonksiyonu ile en uygun görev süresi değeri ve bununla beraber motor dönüş hızı ayarlanmıştır.

Bundan sonraki kısımda havacılık simülatörleri ile ilgili bilgi verilip tarihsel gelişiminden bahsedilecektir. Üçüncü bölümde tasarlanan hareketli platformda kullanılan matematiksel analizler anlatılacaktır. Dördüncü bölümde ise tasarlanan hareketli platformun matematiksel modellemesi anlatılacaktır. Beşinci bölümde ise tasarımın kontrolünü sağlayacak olan elektronik kontrol sistemi tasarımından bahsedilecektir. Altıncı bölümde hareketli uçuş simülatörü üretimi anlatılacaktır. Yedinci ve son bölümde ise çalışmanın sonuçlarından bahsedilerek, sonraki çalışmalar için öneriler paylaşılacaktır.

1. HAREKETLİ UÇUŞ PLATFORMLARI

1.1. Uçuş Simülatörleri Tarihçesi

Simülatörler var olan veya yapılmakta olan bir sistemin uyarlamasıdır. Simülatörler, gerçek sistemler ile aynı davranışları gerçekleştirebilmelidirler. Geliştirilen bütün sistemler karmaşık ve birbirinden farklı matematiksel denklemler barındırırlar. Uçuş simülatörleri muhakkak ki simülatör alanında en gelişmiş simülatörlerden tiplerinden biridir. Uçakların çok karmaşık olması bunun en büyük sebebidir [7].

1900'lü yılların başında uçuş simülatörleri: rüzgâr yardımıyla veya bir eğitmen tarafından manuel olarak hareket ettiriliyordu. Bu sayede eğitimi alan kişi üç eksen etrafındaki etkileri hissederek alıştırma yapıyordu. 1920'lerde The Ruggles Orientator geliştirilmiştir ve bu sayede 3 eksendeki harekete ek olarak dikey eksendeki hareketler elektrikli motorlar ile sağlamıştır [8]. Sonrasında üretilen en başarılı simülatör 1927-1929 yılında üretilen Link Trainer cihazıdır ve 1950'lere kadar kullanılmıştır. Pnömatik olarak, kokpitteki lövyeye ile hareket ettirilmiştir. Simüle edilen etkiler deneme-yanılma yoluyla kontrol yüzeylerinden bağımsız olarak ayarlanmıştır [9].



Şekil 1.1. The ruggles orientator (1917)

Bu sistem başarılı olsa da uçağın hareket ve tepkileri düzgün bir şekilde simüle edilememiştir. 1930'lerden sonra ikinci dünya savaşının da başlamasıyla beraber simülatörün hareketi ile birlikte görüşsüz bomba atışları içinde simülasyon denemelerine başlanmıştır. Bu sayede görüntü oluşturma çalışmaları da başlamıştır. İlk yıllardaki kokpit içi gösterge sistemlerinin simülasyonları pnömatik sistemlerden ve sensörlerden gelen verilerin analog bilgisayarların matematiksel eşitlikler çözmesiyle oluşturulmuştur.



Şekil 1.2. Link trainer (1927-1929)

İlk tam fonksiyonlu uçak simülatörü Boeing 377 uçağı üzerine geliştirilmiştir. O zamana kadar geliştirilen uçuş simülatörlerinin bir hareket sistemi yoktu. Bu sebeple hareketlerin hissedilmesi konusunda eksikliğe sebep oluşturmıştır. Uçakların karmaşıklığının artması ve daha fazla test verisinin oluşması ile birlikte gelen daha karmaşık hesaplama gerekliliklerinden sonra uçuş simülatörlerinde de analog bilgisayarlardan dijital bilgisayarlara geçilmiştir [10].

Fakat uçuş verileri, analiz yapabilmek için veri toplama konusunda yetersiz kalmıştır. Bununla beraber simülatör hareketleri ve geliştirilmiş görüntüleme sistemleri ile birlikte uçuş hissiyatı daha da güçlendirilmiştir. 1959 yılında NASA tarafından ilk kez hareketlerin araştırılması için bir simülatör geliştirilmiştir ve bu

sayede hareketin önemi konusunda ciddi veriler toplanmıştır. Bu veriler ile merkezkaç kuvvetinin etkisi fark edilmiştir. Bu tarz etkenlerden dolayı, titreşim ve stroke limitleri gibi etkenlerde göz önünde bulundurularak, bir tarafı sabit zemine tutturulup diğer tarafı hareket platformuna tutturulan simülatör modeline geçilmiştir. Ayrıca alternatif tasarımlara sahip simülatörler üretilmiştir. İlk 6-DOF uçuş simülatörü 1964 yılında NASA tarafından geliştirilmiştir [11].

70'li yılların sonunda uçuş simülatör hareketlerini sınıflandırmak için bir girişim olmuştur. Çalışma limitleri, tepki süresi, pürüzsüz hareket ve gürültü değerleri çok önemli bir konuma gelmiştir. Bu değerler ile simülatörün kapasitesi JAR ve FAA kurallarına göre belirlenebilir hale gelmiştir. 15 yıl boyunca bu kurallar değişmemiş ve simülatör yetenekleri bu değerlere göre belirlenmiştir. Simülatör geliştirme çalışmalarında bu etkenler göz önünde bulundurulmuştur. Ayrıca tarafsız testlerin ve doğru manevra ile hareket ölçütlerinin olmayışı yeni standartların getirilmesini zorlaştırmıştır. Bununla birlikte havayolu şirketlerinin uçuş eğitimleri için hazırda beklediği bir pazar talebi ortaya çıkmıştır.

80'ler ve 90'lar görsel ve hesaplama işlemleri için hızlı bir gelişmeye şahit olmuştur. Hareket sistemleri ilerlemiştir ve artık uçuş simülatörleri pazarında görüntü sistemleri iyi olana talep artmıştır. Eskiden yan sistemler daha önemliyken şimdilerde ise görüntü sistemleri daha önemli bir hale gelmiştir. Ayrıca hesaplamaların hızlanması ve sistemlerin güçlenmesi ile daha karmaşık ve daha hızlı tepkiler oluşturabilen cihazlar geliştirilmiştir. Bu sayede gerçeğe daha yakın uçuş simülatörleri geliştirilebilmiştir.

1.2. Simülatörlerin Kullanım Alanları

Simülatörlerin pilotaj eğitimleri uzay eğitimleri gibi birçok kullanım alanları bulunmaktadır. Aşağıda kullanım alanlarının bazılarından bahsedilmektedir.

Mühendislik için geliştirilen simülatörler ile geliştirilen bazı araçların simülasyonları yapılmaktadır. Bu simülasyonlar sayesinde geliştirilen araçların karakteristikleri test edilmektedir ve mühendisliğe yön verilmektedir [12]. Uçuş simülatörlerinin geliştirilmesiyle uçak üretiminin maliyetinin azaltılmasıyla geliştirme adımları hızlandığı için uçak sertifikasyonları daha hızlı alınabilmektedir. Birçok simülatör

yere sabit durumdadır ve hareketin önemsiz olduğu durumlarda hareketsizdir. Bununla birlikte, hareket sistemindeki hareketlerin tahmini ve iyileştirilmesi çok önemlidir, bu sayede pilotlar geliştirilen uçağın hareketlerini anlayabilir ve bunun sonucunda uçağın uçuş esnasındaki salınımlarını önleyebilmektedir.

Eğitim için geliştirilen uçuş simülatörleri ise pilotların hem prosedürel olarak hem de uçuş tecrübeleri konusunda yetiştirilmeleri içindir [13]. Pilotların uçağı manuel olarak kullanabilmeleri geliştirilmektedir. Askeri uygulamalarda prosedürel eğitimler sabit simülatörlerde gerçekleştirilmektedir. Uçuş tecrübesi için kullanılan simülatörler ise gerçek uçuş tepkilerini verebilmektedirler. Çok büyük manevralar gerektiğinde ise gerçek uçaklarda eğitim yapılmaktadır. Ticari havacılık firmalarında ise parasal durumlar göz önünde bulundurulduğunda prosedürel ve uçuş tecrübesi için hareketli simülatörler kullanılmaktadır. Havacılık otoriteleri tarafından onaylı simülatörlerde yapılan uçuşlar, pilotların gerçekleştirmesi gereken uçuş saati olarak kabul edilmektedir. Gerekli uçuş saatleri EASA için 1500 saat FAA için ise 2500 saattir.

Araştırma için geliştirilen simülatörler, araç ve pilotların etkileşimlerini ve insan algılarını anlayabilmek için geliştirilmiştir [14]. Bu tip simülatörler için performans önemlidir ve sensör girdileri bu tip simülatörlerde manuel kontrol için gereklidir. Ayrıca kaza araştırmalarında da bu simülatörler önemli bir yer edinmiştir. Simülatöre türbülans veya ani rüzgar gibi girdiler uygulanır ve sonuç olarak kaza anında olabilecek durumlar simülatör üzerinde gerçekleşmiş olur. Bu tip simülatörlerde hareket mekanizmasının kaza anında meydana gelen etkileri aldığı sensörel bilgilere göre gerçekleyebilmesi önemlidir.

1.3. Simülatörlerin Avantajları

Simülatörler bir sistemde neler olup bittiğini ve sistemin nasıl işlediğini anlamanın en etkili yoludur. Gerçek bir sistemde hata meydana geldiğinde bu hatayı çözmek için alternatif yöntemler geliştirilir. Simülatörler üzerinde uygulanan bu alternatif yöntemler, ilgili testleri geçtikten sonra gerçek sisteme uygulanır. Bu sayede maliyetin azaltılmasıyla birlikte oluşabilecek can ve mal kaybı en aza indirilmiş olur.

İnsanların duyuları iç kulakta bulunan vestibular sistem tarafından kontrol edilir. Simülatörlerin bir amacı da oluşturulan sistemlerin gerçekliklerini olduğu gibi buraya aktarmaktır. Kullanılan sistem ve matematiksel modeller ne kadar gerçekçi olursa, pilotun vestibular sistemine o kadar gerçeğe yakın etkide bulunmaktadır.

Simülatörler bugüne kadar birçok alanda kullanılmaya başlanmıştır. Kullanılan bu alanlarda bazı konularda avantajlar sağlamıştır. Aşağıda simülatörlerin bazı avantajlarından bahsedilmiştir.

1.3.1. Maliyet

Uçuş simülatörleri ile ticari havayolu firmalarının maliyetleri ciddi derecede azalma göstermiştir [15]. Çünkü havayolu firmaları pilotlarının ulaşması gereken uçuş saatlerini simülatörler ile karşılayabilmektedirler. Ayrıca kokpit prosedür eğitimleri ve pilot yetiştirme pratik eğitimlerini bu simülatörlerle gerçekleştirmektedirler. Aynı zamanda uçak üretim şirketleri de geliştirmelerini ve denemelerini simülatörler üzerinden yaparak, üretime devam etmektedirler. Bu sayede denemeler için sürekli uçuşlar yapmak yerine simülatörler ile uçağın karakteristiğini yansıtarak gerekli verileri toplamaktadırlar.

1.3.2. Güvenlik gereksinimleri

Yeni geliştirilen bir sistemin, parçanın veya metodolojinin ilk denemesinin gerçek uçuşta yapılması tehlike oluşturabilmektedir. Bu yüzden bu tarz denemelerin ilk olarak simülatör üzerinde yapılması daha güvenilir olacaktır. Pilotlara uçuş esnasında gerçekleşebilecek her türlü durumlar simülatörler üzerinde gösterilir. Tecrübesiz pilotlar simülatörler aracılığıyla gerekli eğitimleri güvenli bir şekilde almış olurlar. Ayrıca birçok tasarım ve eleman hataları simülatörler aracılığıyla fark edilir. Bu sayede gerçek uçakların yapabileceği kazalar engellenmiş olur ve uçuş güvenliği bu simülatörler aracılığı ile artırılmış olur.

1.3.3. Çevresel etkiler

Pilotların eğitimlerinin uçaklar yerine simülatörlerde yapılması yakıt tüketimini azaltmaktadır. Bu sayede tonlarca yakıttan tasarruf edilmektedir.

1.4. Simülatörlerin Elemanları

Geliştirilen uçuş simülatörleri pilot eğitimlerinde ve benzer işlemlerde kullanılacağı için, operatör hava aracını kullanırken gerekli uçuş görüntüsü ve hava olaylarıyla karşılaşmalıdır. Bu sebeple simülatör sisteminin uçuş eğitimi için olan tüm gereklilikleri karşılayabilmesi gerekmektedir. Aşağıda bir uçuş simülatörünün sahip olması gereken temel elemanlardan bahsedilmiştir.

1.4.1. Kokpit

Birincisi ve önemlisi pilotların kokpit prosedür ve uçuş eğitimi almaları için bir kokpite ihtiyaçları vardır. Dolayısıyla uçuş simülatörlerinin en önemli gereksinimi, simülasyonun yapıldığı uçak tipinin birebir aynısı bir kokpite ihtiyacı bulunmaktadır.



Şekil 1.3. Cessna 172 simülatör kokpiti

1.4.2. Görsel sistem

Geliştirilen bir uçak sisteminin daha etkili olabilmesi eğitim alan pilota uçuş hissiyatını vermesi gerekmektedir. Bu sebeple bir görüntü atım sistemi ile uçuş simülatörünün uçuş esnasında görüntüyü oluşturup uygun bir görüntüleme yöntemi ile uçuş görüntüsünü bir ekran aracılığı ile pilotların uçuş hareketlerini görmesi sağlanmalıdır.

1.4.3. Aletler

Uçuş ile ilgili aletlerin simülatördeki olması gereken yerlere konumlandırılması gerekmektedir ve uçuş esnasında olması gereken uyarıları da anlık uçuş durumuna göre vermesi gerekmektedir.

1.4.4. Hareket sistemi

Simülatör gerçek uçağın hareketi esnasındaki vereceği tepkilerin birebir aynılarını vermelidir. Bu sebeple uçağın manevra gereksinimlerini karşılayabilecek bir hareket sistemine ihtiyaç duymaktadır. Ayrıca bu hareketlerin gerçek etkileri gösterebilir şekilde gerçekleşmesi gerekmektedir. Dolayısıyla hareketlerin görsel olarak ekrana yansıtılan görüntüler ile eş zamanlı olarak yapılması gerekmektedir.

Geliştirilen simülatörlerin ilgili uçak tipine göre kontrol sistemleri geliştirilmelidir. Mesela Airbus A330 tipindeki uçaklarda joystick kullanılırken Boeing B777 tipi uçaklarında ise lövyeler kullanılmaktadır. Dolayısıyla kontrol sistemleri de farklı olacaktır. Uçakların kontrol yüzeylerinin etkileri kontrol sistemlerinin hareketlerine ve dış rüzgâr etkilerine göre olacağından hareketler de bu etkilere göre değişecektir. Bu yüzden sistemin hareketleri de bu gereklilikleri karşılayacak şekilde ayarlanmalıdır.

1.4.5. Ses sistemi

Uçak simülatörünün, uçağın ve dış etkenlerin seslerini oluşturması gerçekliği arttıracaktır. Uçağın gaz kollarının seviyesine bağlı motor çalışma sesleri, kontrol yüzeylerinin hareketlerinin sesleri, dış çevredeki rüzgâr sesleri, motor stall ve surge gibi sesleri gibi seslerin pilotlara duyurulmalıdır. Bu sayede pilotların algısı güçlenecek ve daha etkin bir uçuş eğitimi yapılmış olacaktır.

1.5. Günümüzde Uçuş Simülatörleri

Bu bölümde günümüzde üretilen ve aktif olarak kullanılan modern uçak simülatörlerinden bahsedilecektir.

Günümüzde havacılığın gelişmesine paralel olarak pilot ve pilot adaylarının eğitim ihtiyacı da artmaktadır. 1920’li yıllarda mekaniksel ve pnömatik olarak kontrol

edilebilen uçuş simülatörleri, savaş pilotlarının eğitimlerinde kullanılmıştır. Günümüzde ise savaş pilotu eğitiminin yanında ticari pilot eğitim talebi de artmaktadır. Bundan dolayı ticari uçaklar için de simülatör talebi ciddi oranda artmıştır. Özellikle ticari havayolu firmaları, bünyelerindeki pilotların uçuş lisanslarını yenileyebilmeleri için tamamlamaları gereken uçuş saatlerini tamamlayabilmeleri için uçuş simülatörü pazarının çok büyük bir kısmını oluşturmaktadır.



Şekil 1.4. Boeing 777 uçuş simülatörü kokpiti

Şekil 1.4’te bir Boeing 777 uçuş simülatörü kokpiti görülmektedir. Görüldüğü üzere simülatörün kokpit kısmı gerçek bir Boeing 777 uçağının kokpiti ile aynı olacak şekilde tasarlanmıştır. Aynı zamanda kokpitin hemen arkasına eğitmen için alan tasarlanmıştır. Bu alanda eğitmen, pilotlar için gerçek uçuşta karşılaşılabilecekleri durumları oluşturup, pilotların bu tarz durumlarda verecekleri tepkileri gözlemleyebilmektedir.

2. HAREKETLİ UÇUŞ PLATFORMU ANALİZLERİ

2.1. Matematiksel Eşitlikler

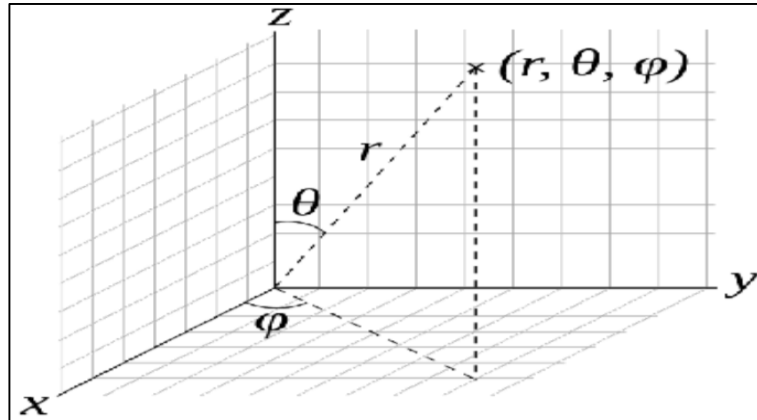
Bu bölümde, üretilecek olan uçuş simülatörünün hareket sistemi ile ilgili matematiksel hesaplamalar ve bu hesaplamaların ispatları yapılacaktır.

Matematiksel modelleme ile geliştirilmek istenen tasarım matematiksel ifadeler ve denklemlere aktarılır. Bu şekilde tasarım makinelerin de anlayabileceği evrensel bir şekilde ifade edilmiş olunur. Bu sayede gerekli analizler ve geliştirmeler bilgisayar ortamına aktarılarak daha hızlı ve daha efektif bir şekilde gerçekleştirilebilir. Sonuç olarak üretim hızlanır, maliyet düşer. Çalışma daha hızlı ve anlaşılır bir şekilde devam edebilir.

2.2. Dönüş Eşitlikleri

Bu bölümde dönüş hareketinin nasıl yapıldığı ve nasıl ifade edildiği anlatılacaktır. Daha sonra uçuş simülatörlerinin hareket mekanizmalarının geliştirilmesinde kullanılan matematiksel teoremlerden bahsedilecektir.

Pozisyon, bir nesnenin uzay boşluğunda bulunduğu konumu ifade eden terimdir. Bir nesnenin pozisyonu (x, y, z) koordinat noktaları ile Kartezyen bir şekilde gösterilir. Bu şekildeki gösterim 3 boyutlu gösterimdir.



Şekil 2.1. Küresel koordinat gösterimi

Küresel Koordinat, bir cismin uzaydaki konumunu bir kürenin çeperi üzerindeki konumu olarak düşünülerek gösterilir [17]. Şekil 2.1’de cismin konumunun küre merkezine olan uzaklığı (r) ve ϕ (x-y) ile θ (y-z) düzlemleri üzerinde yaptığı açılar kullanılarak gösterilir. Örnek olarak (r, θ, ϕ) şeklinde gösterilir.

İçsel Dönüş, bir dönüş hareketinin referans aldığı başlangıç noktasının bir önceki dönüş hareketinin bitiş noktası olduğu konumdur. Bir sonraki dönüş hareketinin eksenleri bir önceki dönüş hareketinin eksenlerine bağlıdır. Bahsi geçen dönüş hareketinin hareket eksenleri (z, x', z') şeklinde kullanılır.

Dışsal Dönüş, dönüş hareketleri sabit bir eksene bağlı kalarak yapılır. Bağlı kalınan eksenler genellikle dünyanın eksenleri olmaktadır. Yani içsel hareketteki gibi bir önceki hareketin hedef noktaya bağlı değildir. Bu hareket sisteminde dönüşler x, y, z eksenlerinde yapılır.

2.2.1. Newton-Euler matrisleri

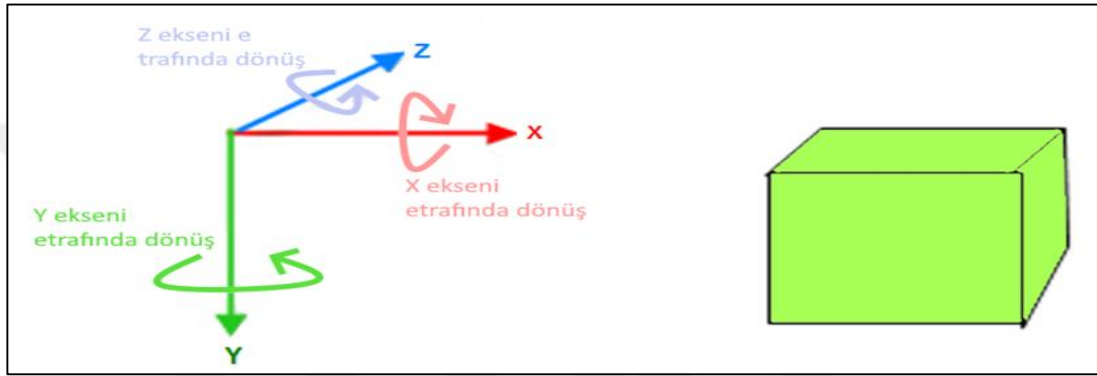
Euler Rotasyon teoremine göre dönüş hareketi 3 parametre ile açıklanabilir. Bir sayı dizisinin oluşturduğu vektörler dizisi Euler açıları olarak adlandırılır [18]. Dönüş Hareketi x, y, z gibi bir eksen dizisi ele alındığında bu dönüş hareketi ilk olarak z eksenini etrafında β kadar döner. Bu dönüşten sonra nesnenin eksenleri (x', y', z) olarak değişir. Yeni dönüş bu eksenler üzerinden yapılır. Bu sefer dönüş x' eksenini etrafında θ açısı kadar olur ve nesnenin eksenleri (x', y'', z') olarak değişir. Bu dönüşten sonra son olarak z' eksenini etrafında ϕ açısı kadar dönüş yapılır. Bu dönüş sonucunda nesnenin yeni eksenleri (x'', y'', z') olur. Bu üç dönüşten sonra cisim üç eksen etrafında dönüş yapmış olur. Eksenler ve dönüşleri incelendiğinde y eksenini etrafında direk olarak bir dönüş yapılmamıştır. Fakat cismin y eksenini değiştirmiştir.

2.2.2. Tait-Bryan matrisleri

Tait-Bryan rotasyonu üç eksen üzerinde yapılır. Euler rotasyonunda x ve z eksenlerinde yapılmaktaydı. Tait-Bryan rotasyonu (z, y, x) eksenlerini kullanırken Euler rotasyonu (z, x, z) eksenlerini kullanır. Farklı ardışık eksen serilerini kullansalar da ikisi de 3 boyutlu dönüş hareketi sağlar. Mühendislik uygulamalarında

genellikle Tait-Bryan rotasyonu kullanılır [19]. Şekil 2.2’de Tait-Bryan rotasyonunun içsel olarak nasıl yapıldığını açıklayan matematiksel model verilmiştir.

Şekil 2.2’deki cisim üç eksende α , β ve θ açıları ile döndürülecektir. Bu işlem Euler rotasyon matrisleri ile ardışık bir şekilde yapılacaktır. Öncelikle döndürme işlemini bir noktanın bir çember üzerinde hareket etmesi olarak düşünülmelidir. Bu dönüş hareketinde nesnenin merkez noktasının hareket ettiği düşünülerek döndürme işlemi yapılacaktır.



Şekil 2.2. Bir cismin 3 ekseninde döndürülmesi

İlk olarak cisim X eksenini etrafında α açısı kadar döndüğünde cismin merkezinin X eksenine olan uzaklığı değişmeyecektir. Oluşan yeni koordinat değerleri 2.1, 2.2 ve 2.3 numaralı denklemlerdeki gibi hesaplanır. Oluşacak olan yeni eksenler (x' , y' , z') olarak gösterilir.

$$x' = x \quad (2.1)$$

$$y' = y \cdot \cos(\alpha) - z \cdot \sin(\alpha) \quad (2.2)$$

$$z' = y \cdot \sin(\alpha) + z \cdot \cos(\alpha) \quad (2.3)$$

Daha sonra cisim oluşan bu son konumu başlangıç konumu olarak kabul edip y' eksenini etrafında β açısı kadar döndürülmüştür. Oluşan bu son y' eksenini etrafında dönüş yaptığı için noktanın bu eksene olan uzaklığı değişmeyecektir ve aynı kalacaktır. Yeni eksenler (x'' , y'' , z'') şeklinde gösterilir.

$$x'' = x' \cdot \cos(\beta) + z' \cdot \sin(\beta) \quad (2.4)$$

$$y'' = y' \quad (2.5)$$

$$z'' = -x' \cdot \sin(\beta) + y' \cdot \cos(\beta) \quad (2.6)$$

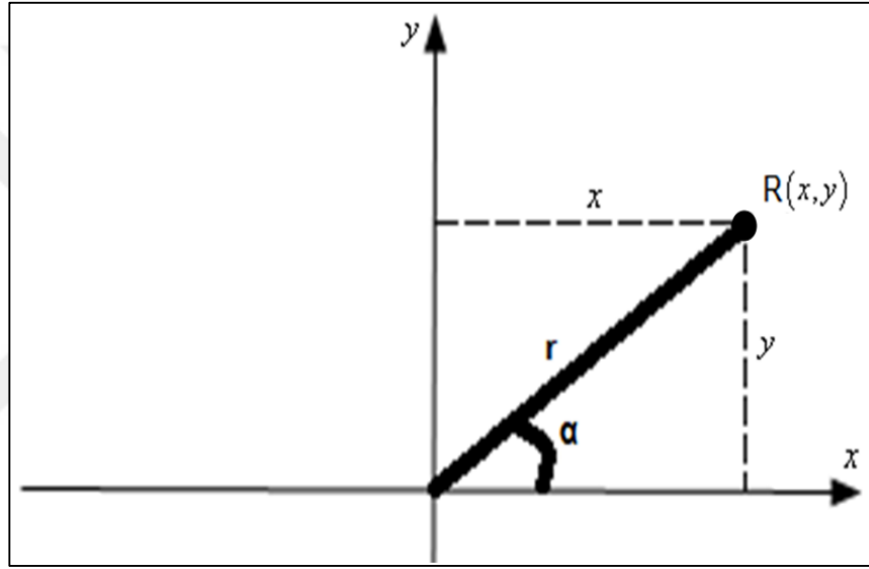
Son olarak oluşan bu eksenleri başlangıç konumu olarak alarak z'' eksenini etrafında θ açısı kadar döndürülsün. Oluşan yeni koordinatlar (x''', y''', z''') şeklinde gösterilir ve cisim z'' etrafında döndürüldüğü için cismin bu eksene olan uzaklığı değişmez.

$$x''' = x'' \cdot \cos(\theta) - y'' \cdot \sin(\theta) \quad (2.7)$$

$$y''' = x'' \cdot \sin(\theta) + y'' \cdot \cos(\theta) \quad (2.8)$$

$$z''' = z'' \quad (2.9)$$

2.2.3. İki eksenli hareketin lineer matris gösterimi



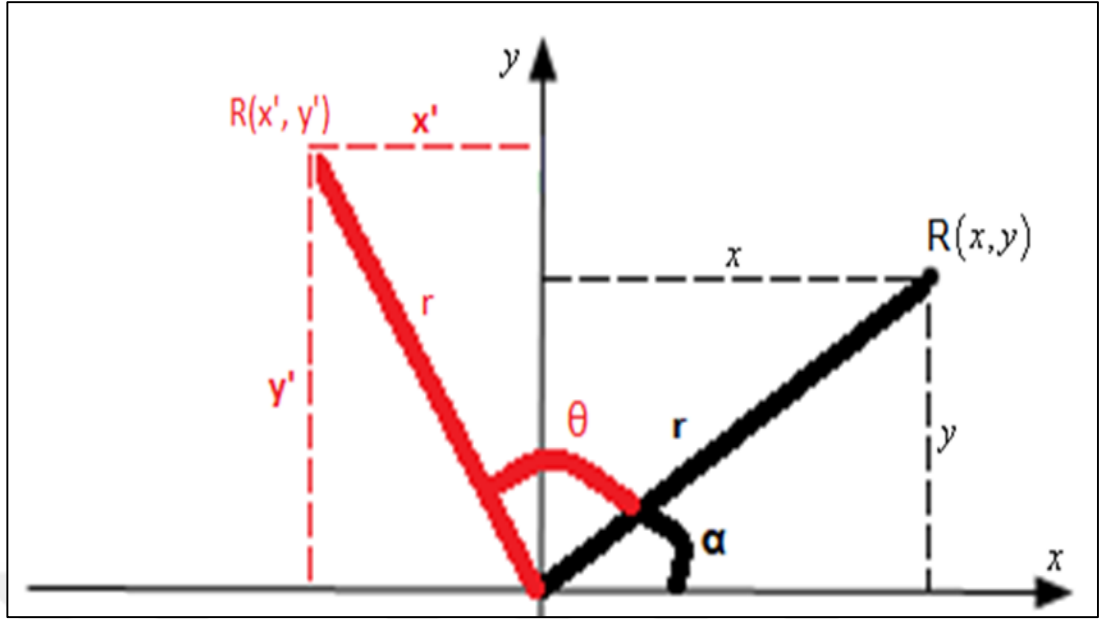
Şekil 2.3. Bir noktanın 2 eksenli gösterimi

Şekil 2.3'te iki boyutlu bir eksenli seçilen $R(x, y)$ noktası merkeze r kadar uzaklıkta ve α kadar açı yapacak şekilde seçilmiştir. Seçilen bu nokta üzerinde matematiksel işlemlerin daha kolay yapılabilmesi açısından matris şeklinde gösterilir.

$$R(x, y) = R \begin{bmatrix} x \\ y \end{bmatrix} = R \begin{bmatrix} r \cdot \cos(\alpha) \\ r \cdot \sin(\alpha) \end{bmatrix} \quad (2.10)$$

Temel olarak bir noktanın konumu Şekil 2.4'deki gibi gösterilir. Eğer bu nokta merkez etrafında θ açısı kadar döndürülmek istenirse yeni oluşan noktalar denklem 2.11'deki trigonometrik dönüşüm eşitlikleri ile ifade edilecektir.

$$R(x', y') = R \begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} r \cdot \cos(\alpha + \theta) \\ r \cdot \sin(\alpha + \theta) \end{bmatrix} \quad (2.11)$$



Şekil 2.4. Hareket ettirilen bir noktanın 2 eksende gösterimi

Bu trigonometrik denklemlere 2.12 ve 2.13 numaralı denklemlerdeki trigonometrik dönüşümler uygulandığında 2.14 ve 2.15 numaralı denklemler elde edilir:

$$\cos(\alpha + \theta) = \cos \alpha \cos \theta - \sin \alpha \sin \theta \quad (2.12)$$

$$\sin(\alpha + \theta) = \sin \alpha \cos \theta + \cos \alpha \sin \theta \quad (2.13)$$

Oluşan yeni koordinatlar denklem 2.14 ve 2.15'teki gibi olacaktır.

$$R \begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} r \cdot \cos(\alpha + \theta) \\ r \cdot \sin(\alpha + \theta) \end{bmatrix} = R \begin{bmatrix} r \cdot (\cos(\alpha) \cdot \cos(\theta) - \sin(\alpha) \cdot \sin(\theta)) \\ r \cdot (\cos(\alpha) \cdot \sin(\theta) + \sin(\alpha) \cdot \cos(\theta)) \end{bmatrix} \quad (2.14)$$

$$R \begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} r \cdot \cos(\alpha + \theta) \\ r \cdot \sin(\alpha + \theta) \end{bmatrix} = R \begin{bmatrix} r \cdot \cos(\alpha) \cdot \cos(\theta) - r \cdot \sin(\alpha) \cdot \sin(\theta) \\ r \cdot \cos(\alpha) \cdot \sin(\theta) + r \cdot \sin(\alpha) \cdot \cos(\theta) \end{bmatrix} \quad (2.15)$$

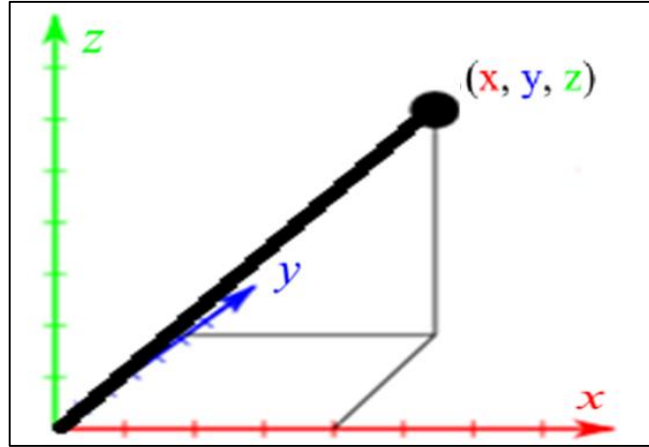
2.14 ve 2.15 numaralı denklemler üzerinde $x = r \cdot \cos(\alpha)$ ve $y = r \cdot \sin(\alpha)$ değişiklikleri yapıldığında, aşağıdaki hale gelecektir.

$$R \begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{bmatrix} \quad (2.16)$$

ve matris çarpanlarına ayrıldığında 2.17 numaralı denklemdeki sonuç elde edilecektir.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.17)$$

2.2.4. Üç eksenli hareketin lineer matris gösterimi



Şekil 2.5. Bir noktanın 3 eksenle gösterimi

Üç eksenli harekette, iki eksenli harekete ek olarak Z eksenli bulunmaktadır. Şekil-2.5'te görünen bir noktanın hareket matrisini tanımlanacaktır. Seçilen noktanın x, y ve z eksenlerine sırasıyla uzaklığı r_x , r_y ve r_z dir. Eksenlere yapılan açılar ise X, Y ve Z olarak ifade edilmektedir. Bu noktanın sırasıyla x, y ve z eksenleri etrafında dönüşlerinin trigonometrik olarak matrisleri oluşturulacaktır. İlk olarak x eksen etrafında α açısı kadar döndürülsün. Bu dönüşün ardından oluşan yeni eksenler denklem 2.18'deki gibi olacaktır.

$$R = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r \cdot \cos(X) \\ r \cdot \cos(Y + \alpha) \\ r \cdot \sin(Z + \alpha) \end{bmatrix} \quad (2.18)$$

Bu denklemde ilgili nokta X eksen etrafında α açısı kadar döndürüldüğü için X eksenine olan uzaklığı değişmemektedir. Dolayısıyla r_x değerini $r \cdot \cos(X)$ şeklinde olduğu gibi kalıyor. Ama Y ve Z eksenlerinde açısal olarak değişiklikler olduğu için noktanın o eksenlere olan uzaklığı açıya bağlı olarak değişecektir ve buna bağlı olarak $r_y = r \cdot \cos(Y + \alpha)$ ve $r_z = r \cdot \sin(Z + \alpha)$ şeklinde yeni değerler oluşur ve oluşacak olan yeni noktanın eksenlere olan uzaklık değerleri aşağıdaki gibi olur. Aşağıdaki denklemlerde x, y ve z olarak belirtilen değerler r_x , r_y ve r_z olarak kullanılan noktanın ilgili eksene olan uzaklığını belirtmektedir.

Yapılan döndürme işlemi sonucunda noktanın geldiği yeni koordinatın x, y ve z eksenlerine yaptığı açılar sırasıyla X, Y ve Z şeklinde alınır.

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_x \\ r \cdot \cos(Y + \alpha) \\ r \cdot \sin(Z + \alpha) \end{bmatrix} = \begin{bmatrix} x + y \cdot 0 + z \cdot 0 \\ x \cdot 0 + y \cdot \cos(\alpha) - z \cdot \sin(\alpha) \\ x \cdot 0 + y \cdot \sin(\alpha) + z \cdot \cos(\alpha) \end{bmatrix} \quad (2.19)$$

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.20)$$

2.19 ve 2.20 numaralı denklemler ile x eksenini etrafındaki dönüş hesaplanır.

Y eksenini etrafındaki dönüş hesaplanacak olursa, yukarıda x eksenini için yapılan işlemin aynısı burada y eksenini için yapılacaktır. Dolayısıyla y eksenini etrafındaki β açısı kadar dönüş hareketi denklem 2.21 ve 2.22'deki gibi olacaktır.

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r \cdot \cos(X + \beta) \\ r_y \\ r \cdot \sin(Z + \beta) \end{bmatrix} = \begin{bmatrix} x \cdot \cos(\beta) + y \cdot 0 - z \cdot \sin(\beta) \\ x \cdot 0 + y \cdot 1 + z \cdot 0 \\ x \cdot \sin(\beta) + y \cdot 0 + z \cdot \cos(\beta) \end{bmatrix} \quad (2.21)$$

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.22)$$

Z eksenini etrafındaki θ açısı kadar dönüş denklem 2.23 ve 2.24 gibi hesaplanır.

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r \cdot \cos(Z + \theta) \\ r \cdot \sin(Z + \theta) \\ r_z \end{bmatrix} = \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) + z \cdot 0 \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) - z \cdot 0 \\ x \cdot 0 + y \cdot 0 + z \cdot 1 \end{bmatrix} \quad (2.23)$$

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.24)$$

Standart olarak Tait-Bryan dönüşüm matrisleri için 6 farklı dönüş sıralaması vardır. Bunlar x-y-z, x-z-y, y-z-x, y-x-z, z-x-y, z-y-x şeklindedir. Daha öncede bahsedildiği üzere Newton-Euler dönüşüm matrislerinde hesaplamalar z-y-z sırasıyla yapılır. Dolayısıyla bu çalışmada kullanılacak olan dönüş matrisleri de Tait-Bryan denklemlerinde olduğu gibi z-y-x eksenleri etrafında hesaplanacaktır. Matris çarpma işlemlerinde çarpanların sırası değiştiğinde sonuçta değiştiği için yukarıdaki sıralamaya uygun olarak çarpma işlemi yapılacaktır. Yapılan matematiksel işlemden sonra hareketin konumu denklem 2.25 ve 2.26 ile hesaplanır.

$$R = R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \quad (2.25)$$

$$R = \begin{bmatrix} \cos(\beta) \cos(\theta) & \sin(\alpha) \sin(\beta) \cos(\theta) - \cos(\alpha) \sin(\theta) & \cos(\alpha) \sin(\beta) \cos(\theta) + \sin(\alpha) \sin(\theta) \\ \cos(\beta) \sin(\theta) & \sin(\alpha) \sin(\beta) \sin(\theta) + \cos(\alpha) \cos(\theta) & \cos(\alpha) \sin(\beta) \sin(\theta) - \sin(\alpha) \cos(\theta) \\ -\sin(\beta) & \sin(\alpha) \cos(\beta) & \cos(\alpha) \cos(\beta) \end{bmatrix} \quad (2.26)$$

2.3. Matematiksel Analizler

Bu bölümde tasarlanacak olan 6-DOF uçuş simülatörünün matematiksel olarak hareket analizi yapılacaktır. Bu bölümde ilk olarak kinematik analiz incelenip sonrasında gerekli kinematik hesaplamalar yapılacaktır.

2.3.1. Kinematik analiz

Kinematik, kuvvet etkisinde hareket eden bir cismin, hareketinin özelliklerini ve sonuçlarını inceleyen bilim dalıdır. Kinematik bir cisim üzerindeki kuvveti incelemeyi sadece cismin hareketi ile ilgilenir. Kinematik sayesinde bir hareket cebirsel olarak ifade edilebilir ve bu sayede hareketin belli bir zamanda konumu, ivmesi, hızı gibi kuvvetten bağımsız değerleri bulunabilir. Kinematik sayesinde zamana bağlı üretilen grafikler ile değerler bulunabilir. Buna örnek olarak ivmeli bir hareketin hız-zaman grafiğinde herhangi bir noktanın eğimi(türevi) alındığında o anki hızı bulunur. Eğer bu grafiğin belli bir andaki integrali alınırsa hareketlinin aldığı yol bulunur. Bütün bu açıklamalar kinematiğe örnektir [20].

6-DOF özelliği olan bir hareket platformu için altı adet birbirinden bağımsız ve farklı noktalara bağlanmış manipülatör bulunur. Bu manipülatörler merkezi işlem biriminden aldığı komutlara göre belli mesafelerde uzar veya kısalır. Bu hareketi altısı birden paralel bir şekilde yaptığında platformun hareketli kısmı merkezi işlem biriminin ayarladığı açısal değerlere gelmiş olur. Kinematik hesaplamasının önemi ise bu manipülatörlerin uzunluğunun zamana bağlı olarak değişimini sensörler aracılığı ile hesaplayarak gerekli komutun gönderilmesini devam ettirir veya durdurur. Merkezi işlem birimi, bunu kendisine daha önceden tanımlanmış olan kinematik hesaplamaları yaparak hesaplar ve hareketi belirlenen değerlere ulaşmaya kadar devam ettirir. Bu bölümde 6-DOF hareket için kullanılan ileri ve ters kinematik hesaplamaları ve analizleri incelenecektir.

2.3.2. İleri kinematik analizi

Eklem değişkenlerinin belirlenmesiyle hareketli eklem sisteminin işlem yapacak olan uç noktasının geleceği konumu sistemin merkezine göre belirler.

Bir hareket sistemi birbirine seri olarak bağlanan hareketli manipülatörlerden oluşur. Bu iki manipülatör arasındaki hareket ve konum ilişkisi tek matris ile açıklanır. Bu işlemde seri bağlı olan manipülatörlerin hareket matrisleri dizilim sırasına göre sırasıyla çarpıldığında sonuç olarak hareket sisteminin ana çerçeve kısmı ile en uç noktası arasındaki ilişkili olan matris açıklanmış olur. Açıklanan bu matris içeriğinde hareketli kısmın ana çerçeveye göre geldiği son konum bulunmuş olur. İleri kinematik hesaplaması için manipülatörlerin uzunlukları ve eklemlerin açılarını bilmek gerekmektedir.

2.3.3. Ters kinematik analizi

Çok bağlantılı bir hareket sisteminde hareket ettirmek istenilen kısmın hedef olan konuma gelebilmesi için manipülatörlerin gelmesi gereken uzunlukları hesaplar.

Robotların ileri yön kinematiğinde eklem açıları ve robotun fiziksel değişkenleri bulunur. Bulunan bu değişkenler dönüşüm matrisinde yerlerine konulup ana çerçeveden araç çerçevesine doğru robotun yönelimi ve konumu hesaplanır. Robotlar için ters kinematik çözümü ise, araç çerçevesinin ana çerçeveye göre yönelimi ve konumu verildiğinde, robotun bu yönelim ve konuma ulaşabilmesi için gerekli olan açı setlerinin hesaplanması şeklinde tanımlanır. Yani ileri kinematik hesaplamalarının tersine olarak yapılmasıdır.

2.3.4. İleri kinematik ve ters kinematik arasındaki farklar

İleri ve ters kinematik hesaplamalarının eklemlere sahip olan hareketli bir sistemin hareketi sonucunda varacağı son noktaya nasıl ulaşacağını hesaplar. İki işlemde birbirlerinin sağlaması şeklinde kullanılabilir. Çünkü birisi verilen hedef koordinattan olması gereken açı ve uzunlukları bulurken diğeri verilen uzunluk ve açı değerlerinde hedef noktayı bulmaktadır. İleri kinematik işlemlerinde hedef olan koordinatlara ulaşmak için eklemlerin açılarını ve manipülatörlerin uzunluklarının bilinmesi gerekmektedir. Ters kinematikte ise hareketli olan kısmın hedef olan

koordinatlarına ulaşabilmek için gerekli olan manipölatör uzunlukları ve eklem açıları bulunur.

İnsansı robotik tasarımlarda hareket sistemindeki eklem sayısı arttıkça ters kinematik işlemleri karmaşılaşacaktır. Fakat ileri kinematik işlemlerinde eklem açıları ve manipölatör uzunlukları bilinirken bu eklemlerin matrisleri sıralı bir şekilde çarpıldığında hedef daha kolay bir şekilde bulunur. Bu yüzden fazla eklem sayısı olan sistemlerde ileri kinematik kullanmak işlem hızı açısından daha avantajlı olacaktır.

Bu tezin konusunda bir uçuş simülatörü tasarlanacaktır. Bu uçuş simülatöründe kullanılacak olan manipölatörler sadece sabit bir zemin ve hareketli bir platforma bağlanacaktır. Projemizde kullanılacak olan hareket sisteminde eklem sayısı insansı hareket kabiliyetindeki robotik tasarımlara göre daha az eklem bulunduğu için ters kinematik hesaplamaları yeterli olacaktır. Bununla beraber uçuş görüntüsünü oluşturmak için kullanılacak olan Microsoft Flight Simulator X programından okunacak olan değerler direk olarak hareketli platformun hedef olarak alacağı değerlerdir. Bu değerler görüntü yazılımındaki uçağın roll, pitch ve yaw değerleri olacaktır. Bu değerlerle birlikte uçuş hissiyatını sağlayabilmek adına platformun x, y ve z eksenleri üzerinde de hareket ettirilmesi gerekmektedir. Dolayısıyla 6 eksen üzerinde hareket sağlayabilmek için ters kinematik işlemlerini kullanmak bu proje için daha avantajlı olacaktır.

3. HAREKETLİ PLATFORMLARIN KİNEMATİK ANALİZ İLE MODELLENMESİ

Bu bölümde hareketli bir simülatörünün hareketleri, bilgisayar ortamında oluşturulacak ve üretim için hazır hale getirilecektir. Bu işlemler için bazı yardımcı programlar kullanılacak. Bu programlar ile gerekli matematiksel model oluşturulacak ve oluşturulan bu matematiksel model fiziksel modele aktarılacaktır.

3.1. Stewart Platformu Modellemesi

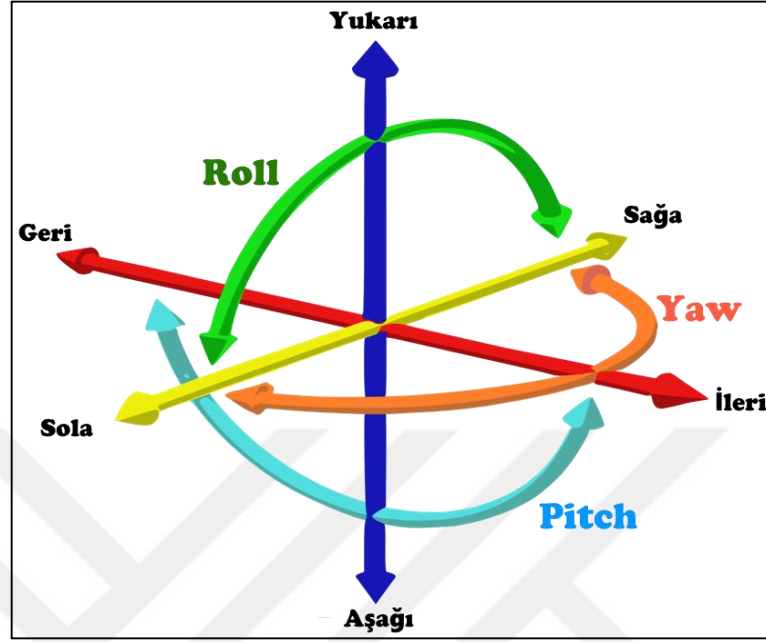
Stewart platformu biri sabit biri hareketli olmak üzere iki yapının birbirine paralel 6 tane değişebilen uzunluklara sahip manipülatörlerin bağlanmasıyla oluşur. Tasarımda sabit kısım Base, hareketli kısım ise Platform olarak adlandırılacaktır.

Base kısım yere göre paralel olan ve birbirine dik olan x , y ve z eksenlerini kullanmaktadır. Base kısmın üzerindeki, manipülatörlerin bağlantı noktaları bu eksenler ile ifade edilerek görelî hareket matrisinde bu şekilde kullanılacaktır.

Tasarımdaki hareketli kısım ise base kısma bağımlı olarak hareket etmektedir. Bu yüzden hareket eksenleri birbirine dik olan x' , y' ve z' eksenleri ile ifade edilmektedir. Stewart platformlarının hareketli kısmı platform kısmı olduğu için, 6 özgürlük eksenindeki hareketlerde platform kısmına aittir. Platform kısmın kabiliyeti dahilinde olan bu hareketleri base kısma bağlı olduğu için base kısma göreceli olarak yapmaktadır.

6 özgürlük eksenindeki hareketlerin üç tanesi base yapıya göre kayma ekseninde üç tanesi de base eksenlerine göre dönüş eksenindedir. Dönüş eksenleri roll, pitch ve yaw; kayma eksenleri ise aşağı-yukarı, ileri-geri ve sağa soladır. Dönüş hareketleri eksenlerin etrafında gerçekleşirken, kayma hareketleri eksenlerin üzerinde gerçekleşir. Yani platform x eksenî etrafında dönüş yaparken yine x eksenî üzerinde bu sefer ileri geri yaparak kayma hareketi yapmaktadır. Aynı şekilde platform y eksenî etrafında pitch hareketi ve z eksenî etrafında yaw hareketi yapabilmektedir. Aynı zamanda y eksenî üzerinde sağa-sola hareket ederken, z eksenî üzerinde yukarı-

aşağı kayma hareketi yapabilmektedir. Bu hareket kabiliyetlerine sahip olduğu için 6 DOF olarak kabul edilmektedir.



Şekil 3.1. 6DOF Hareketin gösterimi

Platformun hareket kabiliyetindeki eksenlere karar verildikten sonra kullanılacak olan matematiksel yapının oluşturulması gerekmektedir. Bu çalışmada tasarlanan Stewart platformu base kısma sabitlenmiş, döner tip manipülatörlerden oluştuğu için üç tane bağlantı noktası içermektedir. Yapının bu özelliğinden dolayı ileri kinematik hesaplamaları uygulanıp platformun durumları için gereken motor açıları bulunacaktır.

Dolayısıyla her base ve platform bağlantılarını her bir bağlantı noktası için denklem 3.1 ve 3.2'deki gibi ifade edilecektir.

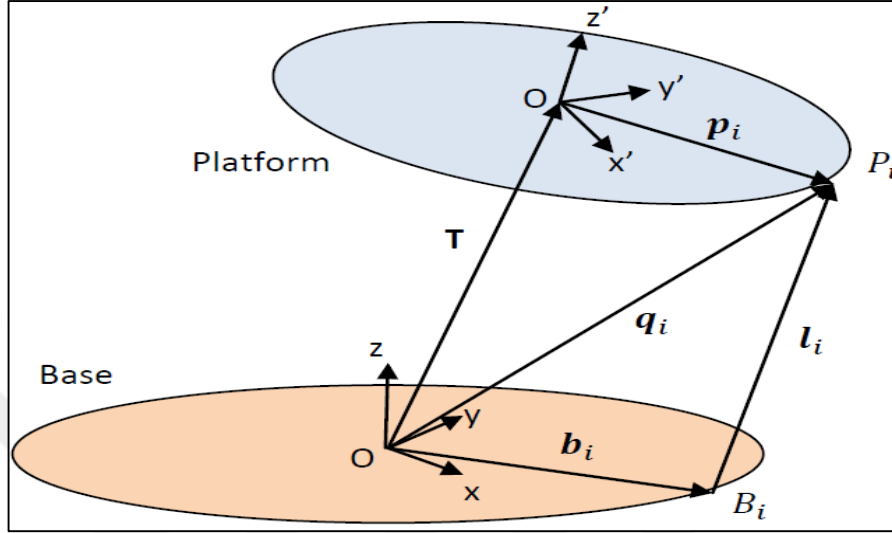
$$P_i = \{P_{xi}, P_{yi}, P_{zi}, 1\} \quad (3.1)$$

$$B_i = \{B_{xi}, B_{yi}, B_{zi}, 1\} \quad (3.2)$$

Stewart platformunun merkez noktası hareketli platformun orta noktası olarak kabul edildiği için $P_{zi} = 0$ olarak kabul edilmektedir ve B_{zi} değeri negatif değer alacaktır.

Hareketli platformun dönüş hareketini yaptığı esnada sabit kısma göre bir Translation (T) vektörü oluşmaktadır. Kinematik hesaplamalarına bu vektörde

eklendiğinde platformun hareket matrisi aşağıdaki gibi olmaktadır. Base kısma bağlı olan her bir manipülatörün platforma olan uzaklığının bulunmasında bu T vektörü kullanılmaktadır. Şekil 3.2’de her bir manipülatör için her iki platformunda eksenel ve bu eksenlere bağlı olarak hareket vektörlerinin durumları gösterilmiştir.



Şekil 3.2. Bir manipülatörün hareket vektörlerinin gösterimi

Şekil 3.2’ye göre 3.3, 3.4 ve 3.5 numaralı denklemler oluşmaktadır. Bu işlemlerin sonucunda manipülatörden platform bağlantı noktasına olan mesafe hesaplanmaktadır.

$$q_i = b_i + l_i \quad (3.3)$$

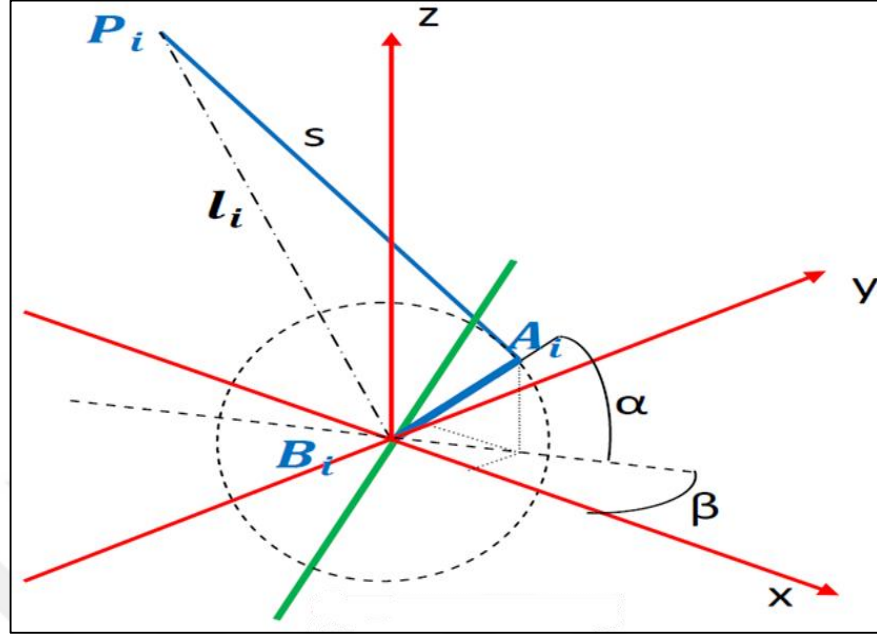
$$q_i = T + R_{PB} \cdot p_i \quad (3.4)$$

$$l_i = T + R_{PB} \cdot p_i - b_i \quad (3.5)$$

Bu işlemlerde b_i vektörü her bütün actuatörler için base kısmın merkez noktasından manipülatör hareketinin başlangıç noktasına (B_i) olan vektörü temsil etmektedir. Aynı şekilde p_i vektörü de hareketli platformun merkez noktasından her bir bağlantı rodunun konumuna (P_i) olan vektördür.

Bu vektörel işlemler doğrusal manipülatörler kullanıldığında geçerli olacaktır. Bizim tasarımıımızda sabitlenmiş döner manipülatörler kullanılacaktır. Dolayısıyla bu hesaplamalara ek olarak birde döner motorun uç kısmına bağlı olan kol uzunluğu ile birlikte l_i değeri hesaplanmalıdır [21].

Şekil 3.3'te döner manipülatörün l_i vektörüne olan etkisini görülmektedir.



Şekil 3.3. Her bir servo motorun x-y eksenindeki dönüş hareketi

- P_i : Platformun Rod ile bağlandığı noktalar ve vektörleri: p_i
- B_i : Servoların Base kısma bağlandığı noktalar ve vektörleri: b_i
- A_i : Servo kolu ile rodun birleştirildiği noktalar ve vektörleri: a_i
- S : Rod uzunluğu
- l_i : B_i ile P_i noktaları arasındaki vektörler

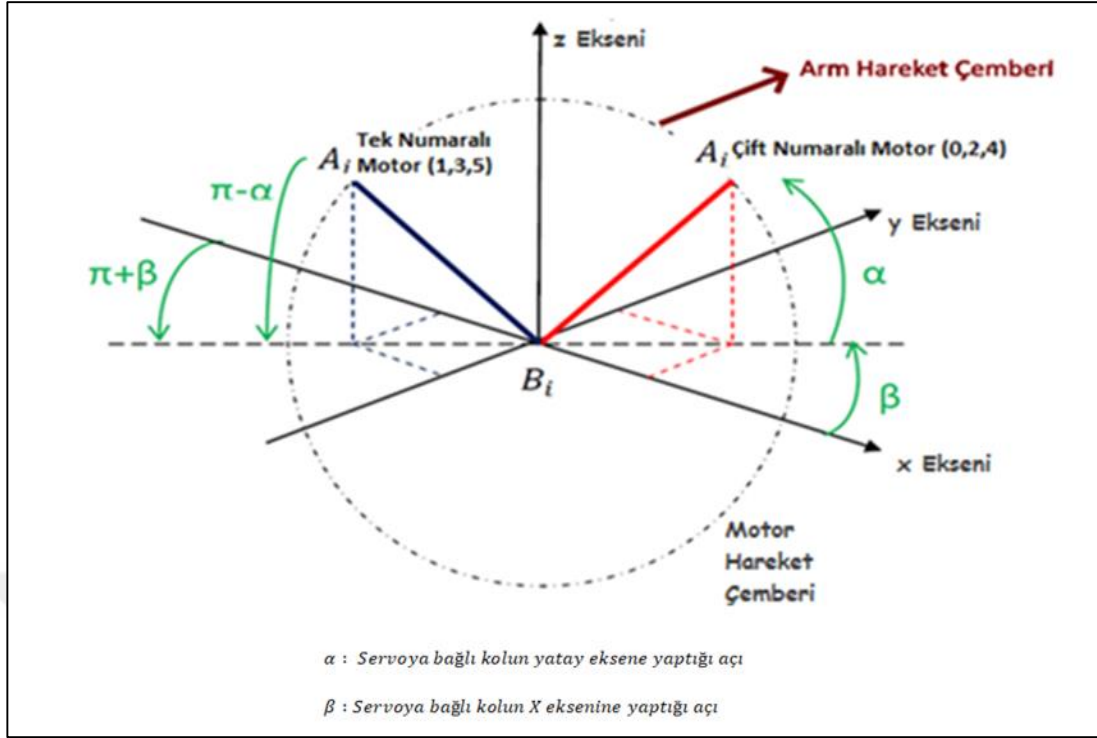
Her bir manipülatör için yukarıda belirtilen konum vektörleri ise denklem 3.6, 3.7 ve 3.8'deki gibi ifade edilir.

$$p_i = [x_{pi} \ y_{pi} \ z_{pi}]^T \quad (3.6)$$

$$b_i = [x_{bi} \ y_{bi} \ z_{bi}]^T \quad (3.7)$$

$$a_i = [x_{ai} \ y_{ai} \ z_{ai}]^T \quad (3.8)$$

Bu tasarımda motorlar çiftler olarak ele alınmaktadır. Motorların dönüş yönleri birbirlerine ters yönde döneceklerdir. Dolayısıyla motorların dönüş yönlerini her iki yöne dönüşlerini belirtmek adına numaralandırırken tek ve çift numaralı motorlar olarak isimlendirilecektir. Ayrıca motorların dönüşleri kendi eksenlerine göre y ekseninde ve x ekseninde boyunca olmaktadır.



Şekil 3.4. Motor çiftinin hareket eksenleri

Yukarıdaki bilgilere göre platforma bağlı olan kolların ve rodların konum vektörlerini ekstenel olarak hesaplanacaktır. Motorların dönüş yönleri birbirlerine zıt olduğu için çift sayılı motorların dönüşlerini α açısı kadar hesaplarken ters yönde dönen tek sayılı motorlar için $\pi - \alpha$ kadar döndüğü varsayılacaktır. Aynı zamanda base ile bağlantı yapılan konum base kısım merkezinden x_{bi} kadar ötelenmiş olması hesaba katılmalıdır.

Dolayısıyla Çift sayılı motorların kol ve rod birleşim noktaları 3.9, 3.10 ve 3.11'deki gibi olur

$$a_{xi} = a \cdot \cos\alpha \cdot \cos\beta + b_{xi} \quad (3.9)$$

$$a_{yi} = a \cdot \cos\alpha \cdot \sin\beta + b_{yi} \quad (3.10)$$

$$a_{zi} = a \cdot \sin\alpha + b_{zi} \quad (3.11)$$

Tek sayılı motorlar için birleşim noktaları ise denklem 3.12, 3.13 ve 3.14 gibi olur:

$$x_{ai} = a \cdot \cos(\pi - \alpha) \cdot \cos(\pi + \beta) + x_{bi} \quad (3.12)$$

$$y_{ai} = a \cdot \cos(\pi - \alpha) \cdot \sin(\pi + \beta) + y_{bi} \quad (3.13)$$

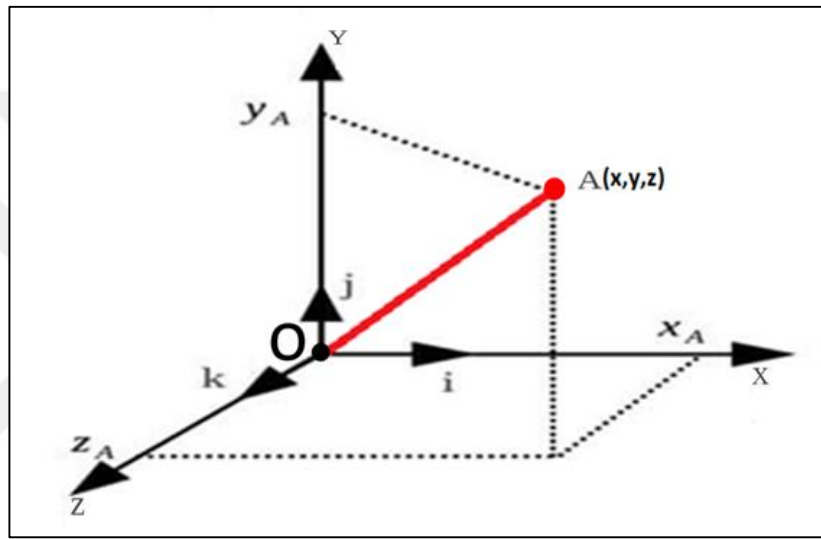
$$z_{ai} = a \cdot \sin(\pi - \alpha) + z_{bi} \quad (3.14)$$

Ayrıca aşağıdaki trigonometrik dönüşüm eşitlikleri tek sayılı motorlar için yazılan eşitlikte yerlerine yazıldığında çift sayılı motorların konum formülleri elde edilmiş olunacaktır.

$$\sin(\pi - \alpha) = \sin\alpha \text{ ve } \cos(\pi - \alpha) = -\cos\alpha \quad (3.15)$$

$$\sin(\pi + \beta) = -\sin\beta \text{ ve } \cos(\pi + \beta) = -\cos\beta \quad (3.16)$$

Pisagor teoremine göre üç boyutlu bir uzaydaki noktanın konumu şekil 3.5'deki gibi ifade edilir [22].



Şekil 3.5. Bir noktanın 3 eksenli kartezyen olarak gösterimi

$$(A)^2 = (A_x - O_x)^2 + (A_y - O_y)^2 + (A_z - O_z)^2 \quad (3.17)$$

Dolayısıyla bu tasarımdaki konum hesaplamaları aşağıdaki gibi olacak.

$$(a_i)^2 = (a_{xi} - b_{xi})^2 + (a_{yi} - b_{yi})^2 + (a_{zi} - b_{zi})^2 \quad (3.18)$$

$$a_i^2 = (a_{xi}^2 + b_{xi}^2 - 2 \cdot a_{xi} \cdot b_{xi}) + (a_{yi}^2 + b_{yi}^2 - 2 \cdot a_{yi} \cdot b_{yi}) + (a_{zi}^2 + b_{zi}^2 - 2 \cdot a_{zi} \cdot b_{zi}) \quad (3.19)$$

$$a_i^2 = (a_{xi}^2 + a_{yi}^2 + a_{zi}^2) + (b_{xi}^2 + b_{yi}^2 + b_{zi}^2) - 2 \cdot (a_{xi} \cdot b_{xi} + a_{yi} \cdot b_{yi} + a_{zi} \cdot b_{zi}) \quad (3.20)$$

$$(l_i)^2 = (p_{xi} - b_{xi})^2 + (p_{yi} - b_{yi})^2 + (p_{zi} - b_{zi})^2 \quad (3.21)$$

$$l_i^2 = (p_{xi}^2 + b_{xi}^2 - 2 \cdot p_{xi} \cdot b_{xi}) + (p_{yi}^2 + b_{yi}^2 - 2 \cdot p_{yi} \cdot b_{yi}) + (p_{zi}^2 + b_{zi}^2 - 2 \cdot p_{zi} \cdot b_{zi}) \quad (3.22)$$

$$l_i^2 = (p_{xi}^2 + p_{yi}^2 + p_{zi}^2) + (b_{xi}^2 + b_{yi}^2 + b_{zi}^2) - 2.(p_{xi} \cdot b_{xi} + p_{yi} \cdot b_{yi} + p_{zi} \cdot b_{zi}) \quad (3.23)$$

$$(s_i)^2 = (p_{xi} - a_{xi})^2 + (p_{yi} - a_{yi})^2 + (p_{zi} - a_{zi})^2 \quad (3.24)$$

$$s_i^2 = (p_{xi}^2 + a_{xi}^2 - 2.p_{xi} \cdot a_{xi}) + (p_{yi}^2 + a_{yi}^2 - 2.p_{yi} \cdot a_{yi}) + (p_{zi}^2 + a_{zi}^2 - 2.p_{zi} \cdot a_{zi}) \quad (3.25)$$

$$s_i^2 = (p_{xi}^2 + p_{yi}^2 + p_{zi}^2) + (a_{xi}^2 + a_{yi}^2 + a_{zi}^2) - 2.(p_{xi} \cdot a_{xi} + p_{yi} \cdot a_{yi} + p_{zi} \cdot a_{zi}) \quad (3.26)$$

$(l_i)^2$ ve $(a_i)^2$ denklemlerini sadeleştirmek için $(s_i)^2$ denkleminde yerlerine yazılırsa:

$$s_i^2 = l_i^2 - [(b_{xi}^2 + b_{yi}^2 + b_{zi}^2) - 2.(p_{xi} \cdot b_{xi} + p_{yi} \cdot b_{yi} + p_{zi} \cdot b_{zi})] + a_i^2 - [(b_{xi}^2 + b_{yi}^2 + b_{zi}^2) - 2.(a_{xi} \cdot b_{xi} + a_{yi} \cdot b_{yi} + a_{zi} \cdot b_{zi})] - 2.(p_{xi} \cdot a_{xi} + p_{yi} \cdot a_{yi} + p_{zi} \cdot a_{zi}) \quad (3.27)$$

$$s_i^2 - l_i^2 - a_i^2 = -(b_{xi}^2 + b_{yi}^2 + b_{zi}^2) + 2.(p_{xi} \cdot b_{xi} + p_{yi} \cdot b_{yi} + p_{zi} \cdot b_{zi}) - (b_{xi}^2 + b_{yi}^2 + b_{zi}^2) + 2.(a_{xi} \cdot b_{xi} + a_{yi} \cdot b_{yi} + a_{zi} \cdot b_{zi}) - 2.(p_{xi} \cdot a_{xi} + p_{yi} \cdot a_{yi} + p_{zi} \cdot a_{zi}) \quad (3.28)$$

$$l_i^2 + a_i^2 - s_i^2 = 2.(b_{xi}^2 + b_{yi}^2 + b_{zi}^2) + 2.a_{xi} \cdot (p_{xi} - b_{xi}) + 2.a_{yi} \cdot (p_{yi} - b_{yi}) + 2.a_{zi} \cdot (p_{zi} - b_{zi}) - 2.(p_{xi} \cdot b_{xi} + p_{yi} \cdot b_{yi} + p_{zi} \cdot b_{zi}) \quad (3.29)$$

Yukarıdaki denklemde a_{xi} , a_{yi} ve a_{zi} yerlerine yazılırsa aşağıdaki denklem elde edilecektir.

$$l_i^2 + a_i^2 - s_i^2 = 2.(b_{xi}^2 + b_{yi}^2 + b_{zi}^2) + 2.(a \cdot \cos\alpha \cdot \cos\beta + b_{xi}) \cdot (p_{xi} - b_{xi}) + 2.(a \cdot \cos\alpha \cdot \sin\beta + b_{yi}) \cdot (p_{yi} - b_{yi}) + 2.(a \cdot \sin\alpha + b_{zi}) \cdot (p_{zi} - b_{zi}) - 2.(p_{xi} \cdot b_{xi} + p_{yi} \cdot b_{yi} + p_{zi} \cdot b_{zi}) \quad (3.30)$$

Denklemdaki parantezleri dağıtırsak aşağıdaki denklemler elde edilir.

$$\begin{aligned}
&= 2.b_{xi}^2 + 2.b_{yi}^2 + 2.b_{zi}^2 + (2.a.\cos\alpha.\cos\beta + 2.b_{xi}).(p_{xi} - b_{xi}) \\
&\quad + (2.a.\cos\alpha.\sin\beta + 2.b_{yi}).(p_{yi} - b_{yi}) \\
&\quad + (2.a.\sin\alpha + 2.b_{zi}).(p_{zi} - b_{zi}) \\
&\quad - 2.(p_{xi}.b_{xi} + p_{yi}.b_{yi} + p_{zi}.b_{zi})
\end{aligned} \tag{3.31}$$

$$\begin{aligned}
&= 2.b_{xi}^2 + 2.b_{yi}^2 + 2.b_{zi}^2 + 2.a.\cos\alpha.\cos\beta.p_{xi} - 2.a.\cos\alpha.\cos\beta.b_{xi} \\
&\quad + 2.b_{xi}.p_{xi} - 2.b_{xi}^2 + 2.a.\cos\alpha.\sin\beta.p_{yi} \\
&\quad - 2.a.\cos\alpha.\sin\beta.b_{yi} + 2.b_{yi}.p_{yi} - 2.b_{yi}^2 + 2.a.\sin\alpha.p_{zi} \\
&\quad - 2.a.\sin\alpha.b_{zi} + 2.b_{zi}.p_{zi} - 2.b_{zi}^2 - 2.p_{xi}.b_{xi} \\
&\quad - 2.p_{yi}.b_{yi} - 2.p_{zi}.b_{zi}
\end{aligned} \tag{3.32}$$

$$\begin{aligned}
&= 2.a.\cos\alpha.\cos\beta.p_{xi} - 2.a.\cos\alpha.\cos\beta.b_{xi} + 2.a.\cos\alpha.\sin\beta.p_{yi} \\
&\quad - 2.a.\cos\alpha.\sin\beta.b_{yi} + 2.a.\sin\alpha.p_{zi} - 2.a.\sin\alpha.b_{zi}
\end{aligned} \tag{3.33}$$

$$\begin{aligned}
&= 2.a.\cos\alpha.\cos\beta.(p_{xi} - b_{xi}) + 2.a.\cos\alpha.\sin\beta.(p_{yi} - b_{yi}) \\
&\quad + 2.a.\sin\alpha.(p_{zi} - b_{zi})
\end{aligned} \tag{3.34}$$

$$\begin{aligned}
&= 2.a.\sin\alpha.(p_{zi} - b_{zi}) + 2.a.\cos\alpha.[\cos\beta.(p_{xi} - b_{xi}) \\
&\quad + \sin\beta.(p_{yi} - b_{yi})]
\end{aligned} \tag{3.35}$$

Yukarıdaki denklem aşağıdaki denklem formatına gelmiş durumda [23]:

$$L = m.\sin\alpha + n.\cos\alpha = \sqrt{m^2 + n^2}.\sin(\alpha + p) \tag{3.36}$$

$$\tan p = \frac{n}{m} \tag{3.37}$$

$$p = -\text{Arctan}\left(-\frac{n}{m}\right) \tag{3.37}$$

$$L = \sqrt{m^2 + n^2}.\sin\left(\alpha + \text{Arctan}\left(\frac{n}{m}\right)\right) \tag{3.38}$$

$$\frac{L}{\sqrt{m^2 + n^2}} = \sin\left(\alpha + \text{Arctan}\left(\frac{n}{m}\right)\right) \tag{3.39}$$

$$\alpha = \text{Arcsin}\left(\frac{L}{\sqrt{m^2 + n^2}}\right) - \text{Arctan}\left(\frac{n}{m}\right) \tag{3.40}$$

Yukarıdaki denkleme göre ilgili değerler aşağıdadır.

$$L = l_i^2 + a_i^2 - s_i^2 \tag{3.41}$$

$$m = 2.a.(p_{zi} - b_{zi}) \tag{3.42}$$

$$n = 2.a.[\cos\beta.(p_{xi} - b_{xi}) + \sin\beta.(p_{yi} - b_{yi})] \tag{3.43}$$

Yukarıdaki hesaplamalar ile belirli açılarla konumlandırılmış olan servoların ve bu servolara bağlanan kol ve rod uzunlukları kullanılarak, servolardan hareketli platformun bağlantı noktalarına olan uzaklıklar hesaplanabilir. Fakat bu işlemden önce hareketli platformun bazı değişmezleri belirlenmelidir. Bu değişmez değerler, platformun başlangıç durumu olacak ve hesaplamalar bu değerler baz alınarak yapılacaktır.

Hareket platformunun başlangıç durumu base ve hareketli platform kısmı birbirlerine paralel durumda olacak ve motorların kol kısımları aynı çizgi üzerinde şekilde konumlandırılacaktır. Bunun için motorların dönüş dereceleri 90 derece olarak belirlenecektir. Bu durumdayken platformda herhangi bir dönüş hareketi yapılmamış olacaktır. Bu durumda base ve platform kısımlarının merkezleri arasındaki değer başlangıç yükseklik değeri h_0 olarak adlandırılacaktır. Bu yükseklik değeri aynı zamanda T vektöründeki z eksenini üzerindeki konumunu belirtmiş olacaktır.

Dolayısıyla başlangıçta aşağıdaki durum elde edilir:

$$q_0 = T + R_{PB} \cdot p_i \quad (3.44)$$

$$q_0 = \begin{bmatrix} 0 \\ 0 \\ h_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (3.45)$$

$$q_0 = \begin{bmatrix} p_x \\ p_y \\ h_0 + p_z \end{bmatrix} \quad (3.46)$$

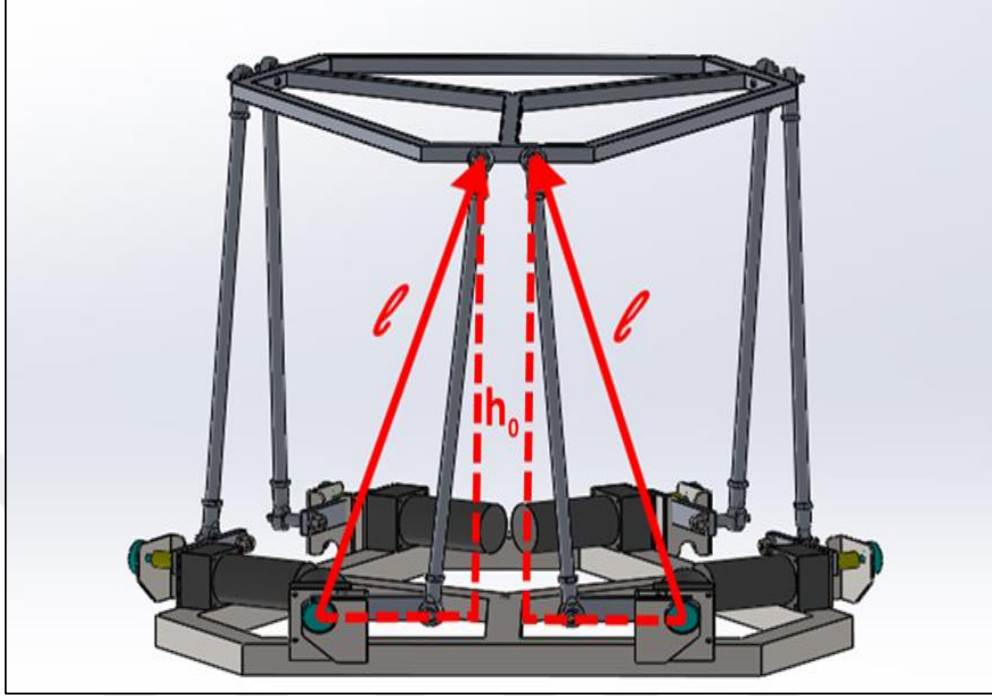
$$(l_i)^2 = (p_{xi} - b_{xi})^2 + (p_{yi} - b_{yi})^2 + (h_0 + p_{zi} - b_{zi})^2 \quad (3.47)$$

Eğer başlangıç pozisyonu olarak ilgili manipülatör 90 derece açıdayken, platforma bağlı olan rod tam olarak kola 90 derece olduğu durumda motor-platform vektör uzunluğu için 3.48 numaralı eşitlik elde edilir.

$$l_i^2 = s_i^2 + a_i^2 \quad (3.48)$$

Fakat bu tasarımda başlangıç durumu aşağıdaki görseldeki gibi olacaktır ve bu durumda yukarıdaki eşitliği olduğu gibi tasarım üzerinde uygulanamayacaktır. Platform hareketi ile kol-rod arasındaki açı sürekli değişken ve diğer motorların durumlarına göre değişken olacaktır. Dolayısıyla diğer motorları da hesaba katarak yapılan bir hesaplama mikroişlemci üzerinde çok fazla işlem yüküne sebep olacaktır.

İşlemi basitleştirmek adına bağlantı noktaları ve başlangıç nokta açıları ile şekil 3.6'daki görselde kesik çizgiler ile gösterilen uzunluklar hesaplanacaktır.

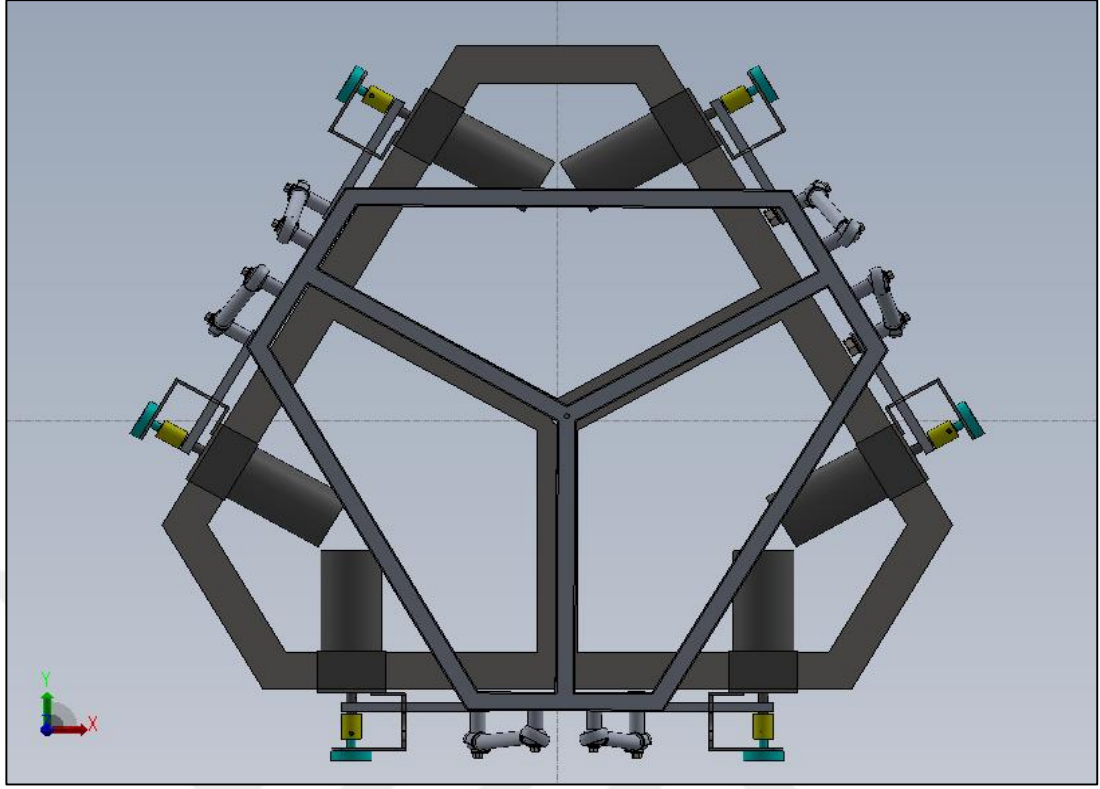


Şekil 3.6. Hareket platformunun yandan görünümü

Tasarımın bilgisayar destekli çizimine tepeden bakıldığında rod ve kol parçalarının aynı hizada olduğu görülmektedir. Dolayısıyla başlangıç yüksekli değerini hesaplariken Y eksenini yok sayarak 2 boyutlu hesaplama yapılabilecektir. Aşağıdaki eşitlik herhangi bir motor üzerinden başlangıç yükseklik değerinin hesaplanmasını sağlayacaktır.

$$h_0 = \sqrt{(p_{xi} - b_{xi})^2 - l_0^2} \quad (4.49)$$

Ayrıca tasarımda kullanılan motorların istenilen dönüş ve kayma hareketleri için gelmesi gereken açının hesaplanması için yukarıdaki eşitlikler kullanılabilir. Bu sayede mikroişlemci tarafından yapılan hesaplamalar sonucunda motorlar hareket ettirilecek ve sensörlerden alınan konum verileriyle konuma gelip gelmediği kontrol edilecek. Bu hesaplama için ihtiyaç olan kinematik hesaplamalardan alınan her bir leg vektörü gerekmektedir. Ayrıca base kısmın yerde olduğu göz önünde bulundurulmalı. Bu nedenle $b_z = 0$ olacaktır.



Şekil 3.7. Hareket platformunun tepeden görünümü

$$l_i = T + R_{PB} \cdot p_i - b_i \quad (3.50)$$

$$l_0 = \begin{bmatrix} 0 \\ 0 \\ h_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (3.51)$$

$$l_0 = \begin{bmatrix} p_x \\ p_y \\ h_0 + p_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ 0 \end{bmatrix} \quad (3.52)$$

Yukarıdaki denklem sadeleştirildiğinde aşağıdaki denklem elde edilir. Yukarıdaki işlemler vektörel olarak yapıldığı için leg vektörünün uzunluğunu bulabilmek için Pisagor teoremi uygulandığında aşağıdaki sonuç elde edilecektir.

$$(l_i)^2 = (p_{xi} - b_{xi})^2 + (p_{yi} - b_{yi})^2 + (h_0 + p_{zi})^2 \quad (3.53)$$

Yukarıdaki eşitlikler aracılığıyla hareketli platformun hedef değerlerine ulaşabilmesi için gereken değerler hesaplanarak motorların bulunması gereken açılar hesaplanabilecektir.

3.2. Kullanılan Yardımcı Programlar

3.2.1. Matlab

Matlab matematiksel hesaplamaların ve modellemelerin bilgisayar ortamında hızlı bir şekilde gerçekleştirilebildiği bir programdır. Matlab ile tasarlanan model matematiksel olarak ifade edilip, geliştirilip uygulanabilir hale getirilmesi sağlanacaktır.

Matlab ortamında çalışmanın fiziksel değişkenleri oluşturulacaktır. MATLAB modeline simülatörün bağlantı noktaları, bağlantıların rod uzunlukları gibi değerler tanımlanacaktır. Girilen bu değerler ile ters kinematik hesaplamaları yapılacaktır. Bölüm 3 te anlatıldığı üzere hesaplamalar karmaşıklığı azaltması açısından matrisler üzerinden yapılmaktadır. Matlab programına tanımlanan değişkenler, program içerisindeki işlemlerde matrisler olarak çağrılmaktadır ve matematiksel işlemler matrisler olarak yapılmaktadır. Gerekli olan sabit ve değişken değerler MATLAB programına matris olarak girilecek ve gerekli hesaplamaların yapılması sağlanacaktır.

Ters kinematik hesaplamalarının yapılabilmesi için ilk olarak gerekli değişken değerleri ilgili değişkenlere atanmalıdır.

İlk olarak şekil 3.8'deki kod bloğunda görüldüğü üzere uçak simülatörünün yapacağı yatış açıları belirlenmiştir. Program içerisindeki hesaplamalar bu değerler temel alınarak yapılacaktır.

```
roll_deg = 0;  
pitch_deg = 0;  
yaw_deg = 0;  
x_ax = 0;  
y_ax = 0;  
z_ax = 0;  
hz = 70;
```

Şekil 3.8. Simülatör açısal değerleri MATLAB kod bloğu

Ardından şekil 3.9’da görüldüğü üzere, derece cinsinden girilen açısal değerlerin radyan cinsine dönüştürülmesi gerekmektedir. Bilgisayar sistemlerinde yapılan açısal ve trigonometrik hesaplamalar radyan cinsi açılar üzerinde yapılmaktadır.

```
roll = roll_deg *pi/180;
pitch = pitch_deg *pi/180;
yaw = yaw_deg *pi/180;
```

Şekil 3.9. Açıların radyan tipine dönüştürülmesi MATLAB kod bloğu

Sonrasında gerekli olan işlem ise bağlantı noktalarının tanımlanmasıdır. Burada önemli olan nokta her iki düzleminde yere paralel bir şekilde ve yerde olarak kabul edilmesidir. Bu yüzden şekil üç boyutlu düşünüldüğünde x ve y koordinat bilgileri girilmiştir. Aynı zamanda z koordinat bilgisi sıfır olarak girilmiştir.

```
base_joint1 = [33; 16; 0; 1];
base_joint2 = [37; -3; 0; 1];
base_joint3 = [21; -35; 0; 1];
base_joint4 = [-21; -35; 0; 1];
base_joint5 = [-39; -2; 0; 1];
base_joint6 = [-16; 34; 0; 1];

platform_joint1=[29; 20; 0; 1];
platform_joint2=[32; 15; 0; 1];
platform_joint3=[3; 35; 0; 1];
platform_joint4=[-3; 35; 0; 1];
platform_joint5=[-31; 15; 0; 1];
platform_joint6=[-29; 20; 0; 1];
```

Şekil 3.10. Simülatör bağlantılarının tanımlanması MATLAB kod bloğu

Ardından üçüncü bölümde tanımlanan dönüş matrisi şekil 3.10’daki gibi tanımlanmıştır. Bu matris içerisindeki hedef yatış değerleri, radyan cinsine dönüştürülmüş değerlerdir.

```
%sabit kısımdan hareketli kısma doğru dönüş matrisi
T=[cos(pitch)*cos(yaw), -cos(pitch)*sin(yaw), sin(pitch), x_ax;
sin(roll)*sin(pitch)*cos(yaw)+cos(roll)*sin(yaw), -sin(roll)*sin(pitch)*sin(yaw)+cos(roll)*cos(yaw), -sin(roll)*cos(pitch), y_ax;
-cos(roll)*sin(pitch)*cos(yaw)+sin(roll)*sin(yaw), cos(roll)*sin(pitch)*sin(yaw)+sin(roll)*cos(yaw), cos(roll)*cos(pitch), z_ax-hz;
0, 0, 0, 1];
```

Şekil 3.11. Dönüş matrisinin MATLAB programına tanımlanması

Dönüş matrislerine göre yapılan hesaplamalar ile hareketli platformun bağlantı noktalarının konumları şekil 3.11’deki kod bloğu ile hesaplanmıştır.

```

platform1 = T * platform_joint1;
platform2 = T * platform_joint2;
platform3 = T * platform_joint3;
platform4 = T * platform_joint4;
platform5 = T * platform_joint5;
platform6 = T * platform_joint6;

```

Şekil 3.12. Dönüş matrisi ile bağlantı noktalarının MATLAB ile hesaplanması

Son olarak yukarıda belirtilen matrisler kullanılarak, sabit ve hareketli kısım arasındaki vektörler Pisagor bağıntısı kullanılarak hesaplanmaktadır.

```

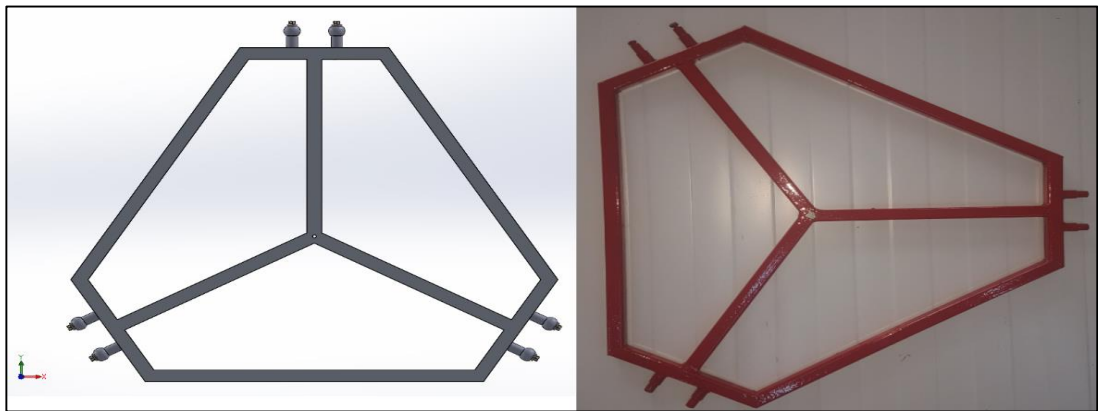
vec1=sqrt((abs(base1(1)-platform1(1)))^2+(abs(base1(2)-platform1(2)))^2+(abs(base1(3)-platform1(3)))^2);
vec2=sqrt((abs(base2(1)-platform2(1)))^2+(abs(base2(2)-platform2(2)))^2+(abs(base2(3)-platform2(3)))^2);
vec3=sqrt((abs(base3(1)-platform3(1)))^2+(abs(base3(2)-platform3(2)))^2+(abs(base3(3)-platform3(3)))^2);
vec4=sqrt((abs(base4(1)-platform4(1)))^2+(abs(base4(2)-platform4(2)))^2+(abs(base4(3)-platform4(3)))^2);
vec5=sqrt((abs(base5(1)-platform5(1)))^2+(abs(base5(2)-platform5(2)))^2+(abs(base5(3)-platform5(3)))^2);
vec6=sqrt((abs(base6(1)-platform6(1)))^2+(abs(base6(2)-platform6(2)))^2+(abs(base6(3)-platform6(3)))^2);

```

Şekil 3.13. Base kısımdan hareketli platforma olan vektörlerin hesaplanması

3.2.2. Solidworks

Solidworks bir bilgisayar destekli çizim programıdır. İki ve üç boyutlu katı parçaların çizim modellerinin oluşturulmasında ve bu modellerin birleştirilip ana yapının oluşturulmasında yardımcı olan bir programdır. Oluşturulan bağlantı noktalarının hareketlerini test ederek sistemin hareket limitlerini belirleyerek doğruluklarını test etmeye olanak tanımaktadır. Bu sayede Solidworks programı çok farklı alanlarda etkin bir şekilde kullanılmaktadır.



Şekil 3.14. Platformun üst tablasının Solidworks çizimi ve üretilmiş hali

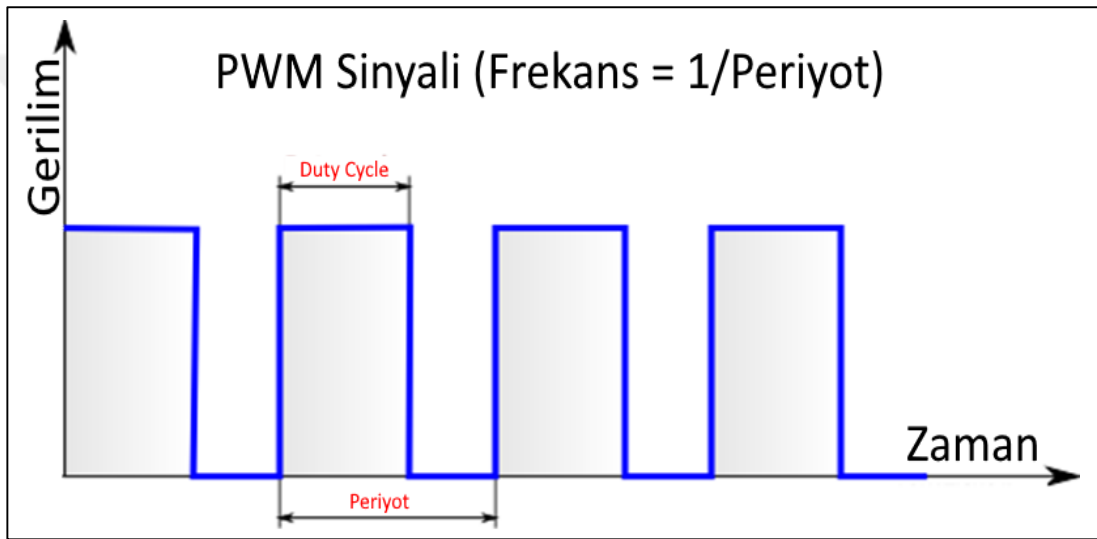
Bu alıřmada solidworks programı kullanılarak uuř simlatrnn btn paraları ayrı ayrı oluřturulmuřtur. Oluřturulan bu paralar yine solidworks ierisinde birleřtirilmiřtir. Ardından hareket kabiliyeti kontrol edilerek sistemin birbiri ile uyum ierisinde alıřtıėından emin olunmuřtur ve alıřma limitleri belirlenmiřtir. Ardından paralar řekil 3.14'deki gibi tek tek retilmiřtir.



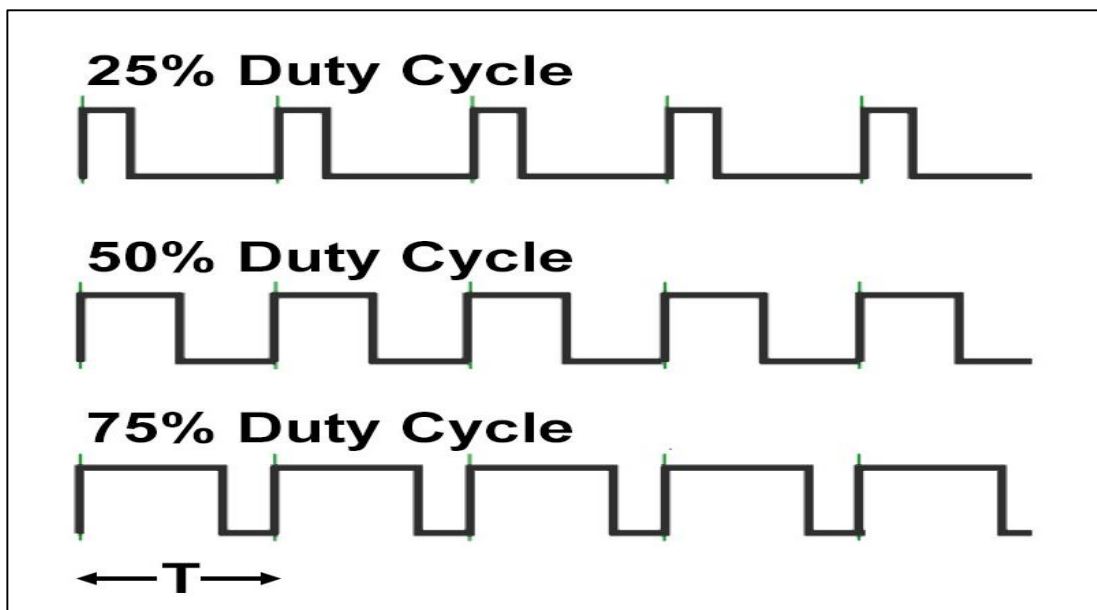
4. ELEKTRONİK KONTROL DEVRESİ TASARIMI

4.1. PWM Sinyali Nedir

PWM, Pulse Width Modulation teriminin kısaltmasıdır. PWM bir kare dalganın HIGH-LOW durumlarının değişme zamanlarını ayarlamaktır. Dolayısıyla dalganın bir zaman periyodu içerisindeki HIGH kalma zamanını değiştirme işlemidir [24].



Şekil 4.1. Örnek bir PWM sinyali gösterimi



Şekil 4.2. Duty Cycle gösterimi

Periyot kullanılan işlemcinin ayarlanan birim zamanıdır. Duty cycle ise periyot zamanında sinyalin toplam HIGH kalma süresidir. Duty cycle ise bir elektrik-pulse grafiğinde bir cycle içerisindeki HIGH kısmın bütün periyota oranıdır. Duty cycle değeri sinyalin ortalama çalışma süresini belirler ve yüzdelik olarak gösterilir. Şekil-4.2’de görüldüğü üzere T ile periyod zamanı gösterilmiştir ve bir periyod içerisindeki HIGH kalma oranı yüzdelik olarak Duty cycle ile belirtilmiştir.

4.2. Mikroişlemci Seçimi

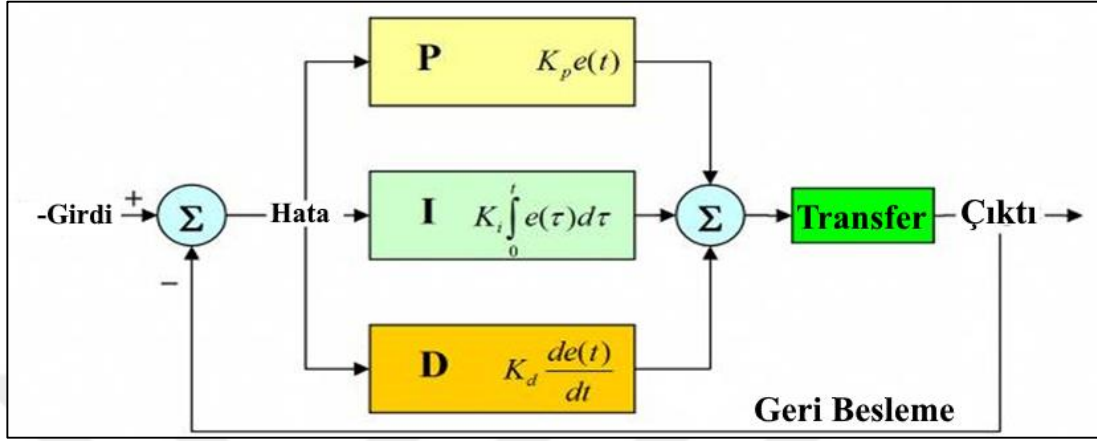
Bu uygulamanın matematiksel modelinde yoğun matematiksel hesaplama bulunduğundan dolayı çalışma frekansı yüksek bir işlemci seçilmiştir. Ayrıca işlemlerin hassasiyeti açısından virgüllü sayılarla da işlemler yapılabilmesi önemlidir. Bu çalışmada ARM mimarisine sahip STM32F429 işlemcisini kullanılacaktır. İşlemci aracılığı ile girdi olarak girilen platformun gelmesi gereken açılış değerleri için manipulatorlerin pozisyonları hesaplanacak oluşturulan hedef değerlere gelmeleri sağlanacaktır. Buradaki girdi değerleri, bilgisayar tarafından çalıştırılan MS Flight Simulator programından alınan anlık yatış değerleri bir ara yüz aracılığı ile asenkron olarak seri port üzerinden aktarılacaktır. Aktarılan bu değerler mikroişlemci tarafından işlendikten sonra motor yön ve PWM sinyali çıktısı olarak motorlara uygulanacaktır.

4.3. PID Uygulaması

PID geri beslemeli bir kontrol mekanizmasıdır. Bir denetleyici yardımıyla oluşturulan sinyalin istenilen sinyale ulaşılmasını sağlar. Şekil 4.3’te örnek bir PID çevriminin çalışması gösterilmiştir.

Şekil 4.3’ten anlaşılacağı üzere bir PID kontrol işleminde üretilen bir sinyal vardır ve bu sinyal üzerinde Proportional (oransal), Integral (alansal) ve Derivative (türevsel) işlemleri yapılır. Yapılan bu işlemler ile yeni bir sinyal değeri oluşur. Oluşan bu sinyal değeri istenilen değer ile yeniden karşılaştırılır ve ana girişe geri besleme sinyali olarak eklenir ve burada daima geri besleme değeri ile gönderilen değer karşılaştırılır. Geri besleme olarak gelen değer sürekli çıktı olan değere yaklaştırılmaya çalışılır ve hata sıfıra çekilir [25].

PID işlemlerinde ilk olarak hata (Error) değeri hesaplanmalıdır. Hata değeri üretilen sinyalin istenen sinyale olan uzaklığıdır. PID işlemlerinin amacı hata değerini sıfıra çekmektir.



Şekil 4.3. PID uygulaması diyagramı

$$\text{Desired Value} = \text{İstenilen Değer} \quad (4.1)$$

$$\text{Current Value} = \text{Şu anki değer} \quad (4.2)$$

$$\text{Error} = \text{Desired Value} - \text{Current Value} \quad (4.3)$$

Proportional işlemi yapılırken adından da anlaşılacağı üzere geri besleme değeri ile hesaplanan hata değerini bir katsayı ile çarparak elde edilir. Bu katsayı K_p değişkeni ile gösterilir. K_p değerinin seçilirken çok büyük seçilmemesine özen gösterilmelidir. Bu bir çarpma işlemi olduğundan katsayı olarak seçilen değer büyüdükçe sinyali daha fazla salınım yapması muhtemeldir. Bu durumda PID işlemi sonucunda üretilen sinyal Set sinyal değerinden uzaklaşabilir veya çok geç ulaşır. Proportional işleminin diğer kontrol işlemlerinden bağımsız bir şekilde tek başına kullanılması durumunda ise eğer sistem kararlıysa set sinyal değerine yaklaştırır. Çünkü sinyal sürekli sabit bir değer ile oranlanmış olacaktır. Fakat değişken sinyalli bir sistemde bu işlemi tek başına kullanmak sonuç değerlerinin yanlış olarak çıkmasına sebep olacaktır.

$$P = K_p \cdot \text{Error} \quad (4.4)$$

Proportional kontrolünde bazı saptmalar oluşacaktır. Oluşan bu saptmaları Integral işlemi düzeltilecektir. Aritmetik işlemlerde Integral, bir sayı grafiğinin altında kalan alanı hesaplar. PID hesaplamalarında ise proportion işleminde oluşacak olan saptmaları engellemek için kullanılır. Proportion işlemi tek başına uygulandığında

sabit bir katsayı ile çarpma işlemi yapılarak hatayı sıfıra çekmeye çalıştığı için sapmalar olacaktır. Oluşan bu hata sinyaline referans sinyali ile arasında kalan bölgeye Integral uygulanarak arada kalan bölgenin alanı bulunur ve buraya müdahale edilerek Proportion işleminden kalan sapma değeri sıfıra çekilmeye çalışılır. Hata değeri büyükse sistemin tepkisinde büyük olacaktır ve büyük tepkiler verecektir. Bunu yaparken sinyali hızlı bir şekilde azaltır veya arttırır. Integral değişkeni K_i ile ifade edilir.

dt = pid fonksiyonuna her girdiğindeki geçen zaman.

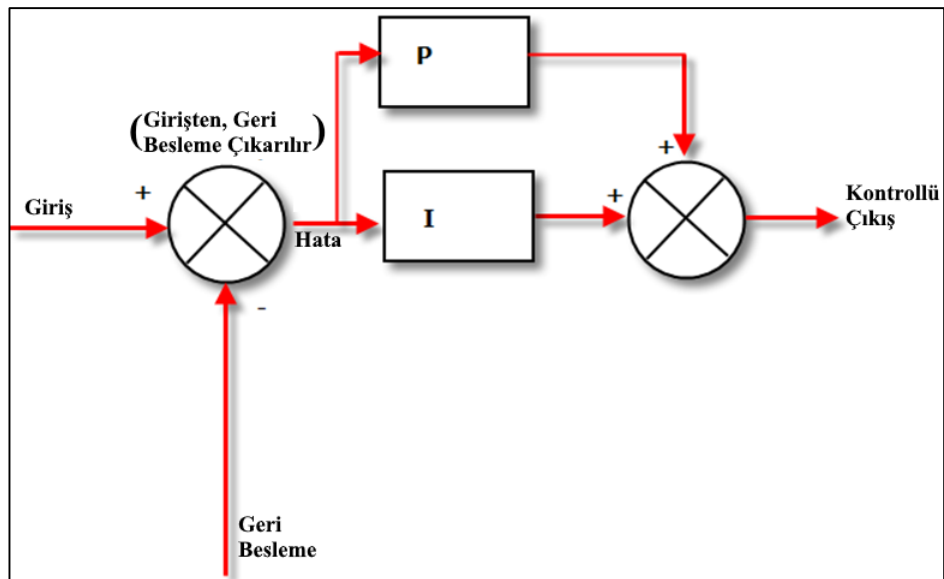
$$I = I_{prev} + (K_i \cdot \text{Error} \cdot dt) \quad (4.5)$$

Bu şekilde hata değerini sıfıra çekerek sistemin ürettiği sinyali set sinyal değerine eşitlemeye çalışır. Eğer sistemin tepki süresi konusunda problemi yoksa Proportion ve Integral işlemleri ile sistem set sinyali yakalar. Bu şekilde sadece bu ikisini kullanarak yapılan hata düzeltme işlemine PI kontrol denir.

Eğer sistemin tepki süresini kısa tutması isteniyorsa Derivative kontrolü kullanılır. Derivative kontrolü K_d ile ifade edilir.

$$\text{Error} = \text{PresentError} - \text{PrevError} \quad (4.6)$$

$$D = (K_d \cdot \text{Error})/dt \quad (4.7)$$



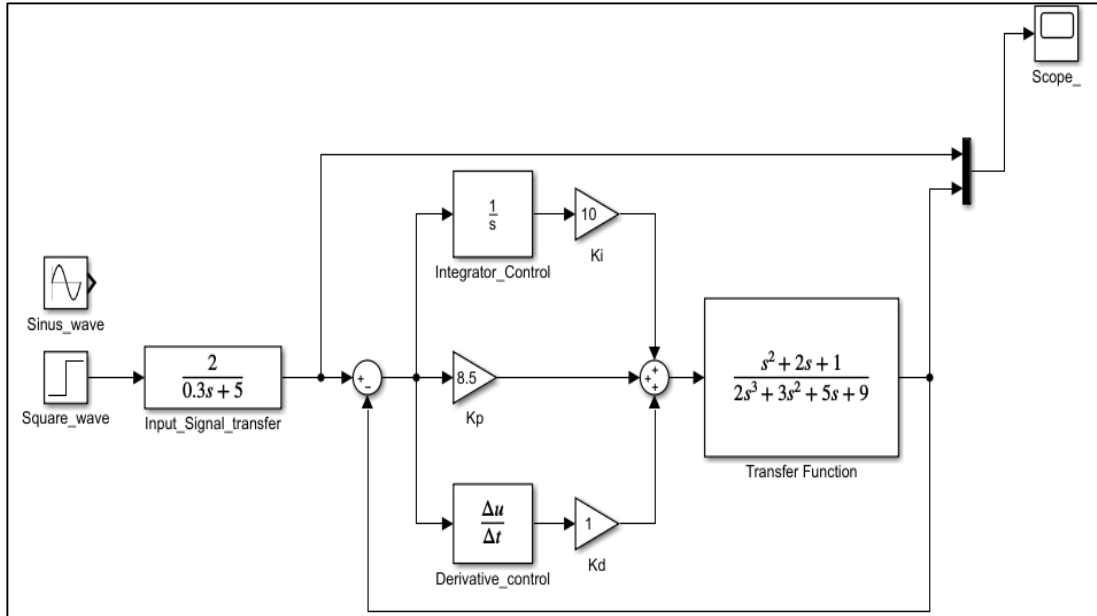
Şekil 4.4. Örnek bir PI uygulaması diagramı

Integral kontrolü bir sonraki durumu kontrol edemez. Çünkü grafiğin ilgili zaman bölgesine kadar gerekli kontrolü yapar ve sonrası için bir şey diyemez. Burada gerekli kontrolü Derivative(Türev) işlemi yapar. Türev işlemi bir grafiğin ilgili zamandaki eğimini vermektedir. Bu nedenle grafiğin nereye gittiğini görerek ileriye dönük yorumlama yapabilen kontrol Derivative kontrolüdür. Grafiğin türevi sıfır olduğunda grafik düz olduğu anlamına gelir ve hata değeri sıfıra gelmiş olur. Türev kontrolünün hızı Proportion ve Integral kontrolünün yanıt hızına göre daha hızlıdır. Bu işlem PI işleminin çıktısı ile çarpılarak sonuca ulaşılır ve PID hesaplaması yapılmış olur.

PID kontrolünün ortak çıktısını alabilmek için bu üç kontrolün çıktılarını toplamak gerekmektedir.

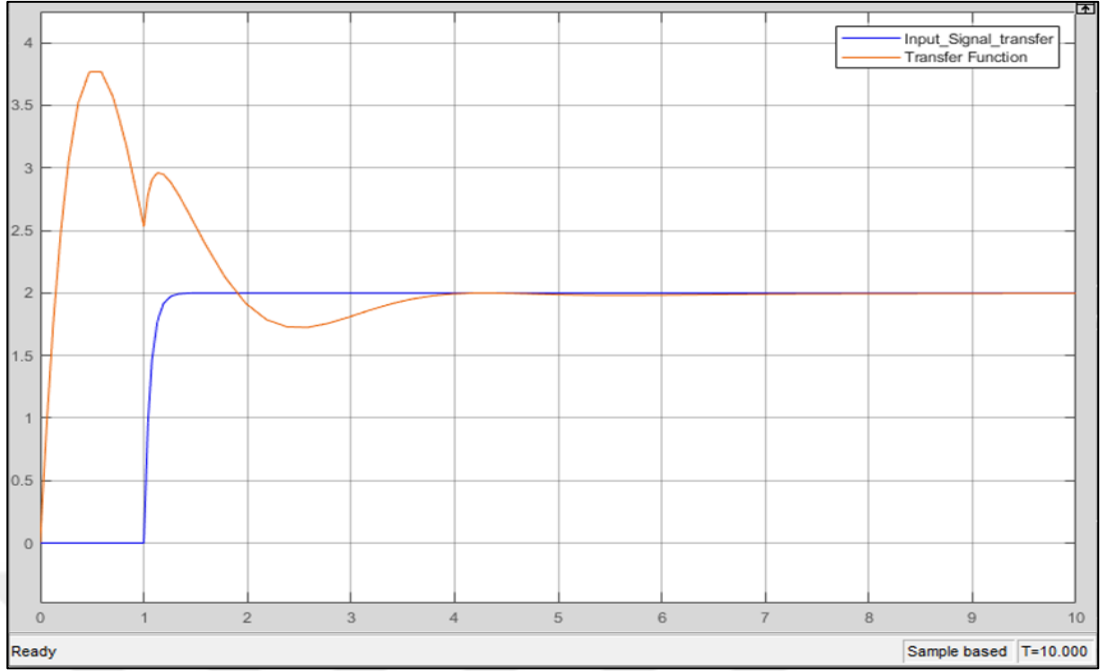
$$PID = P + I + D \quad (4.8)$$

Şekil 4.5'te MATLAB Simulink ile yapılmış temel bir PID uygulaması bulunmaktadır. Bu örnekte bir transfer fonksiyonu ve öncesinde ise transfer fonksiyonu çıkışından gelen geri besleme sinyali ile yapılan PID işlemi gösterilmiştir.



Şekil 4.5. Simulink uygulaması örnek PID diagramı

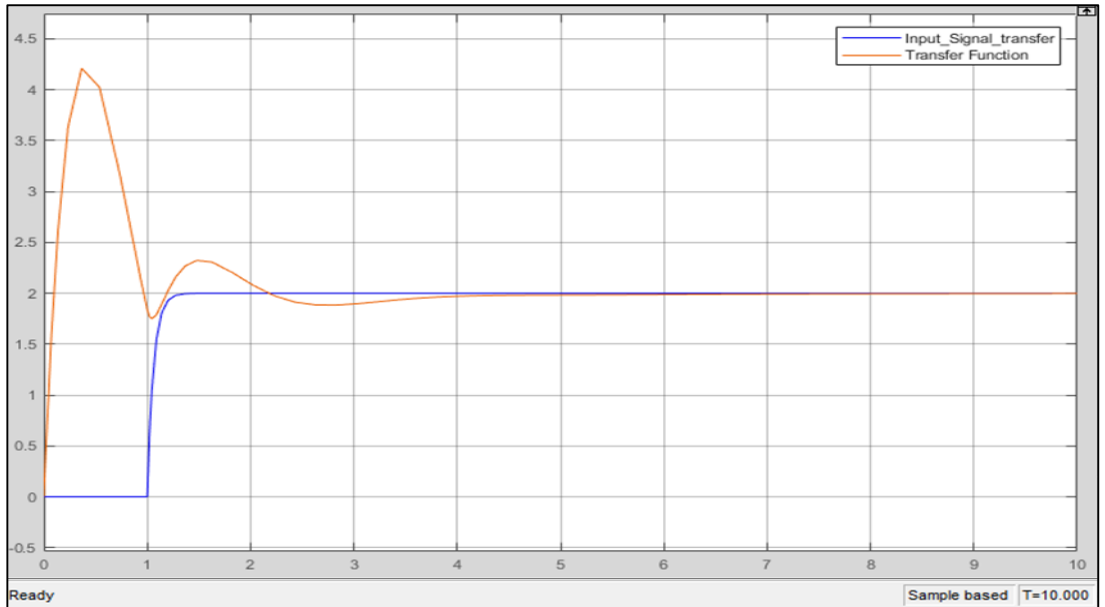
Şekil 4.5'deki Simulink uygulamasında gördüğümüz basit PID uygulamasının osiloskop ekranı şekil 4.6'daki gibi olacaktır.



Şekil 4.6. Simulink PID uygulaması çıktısı

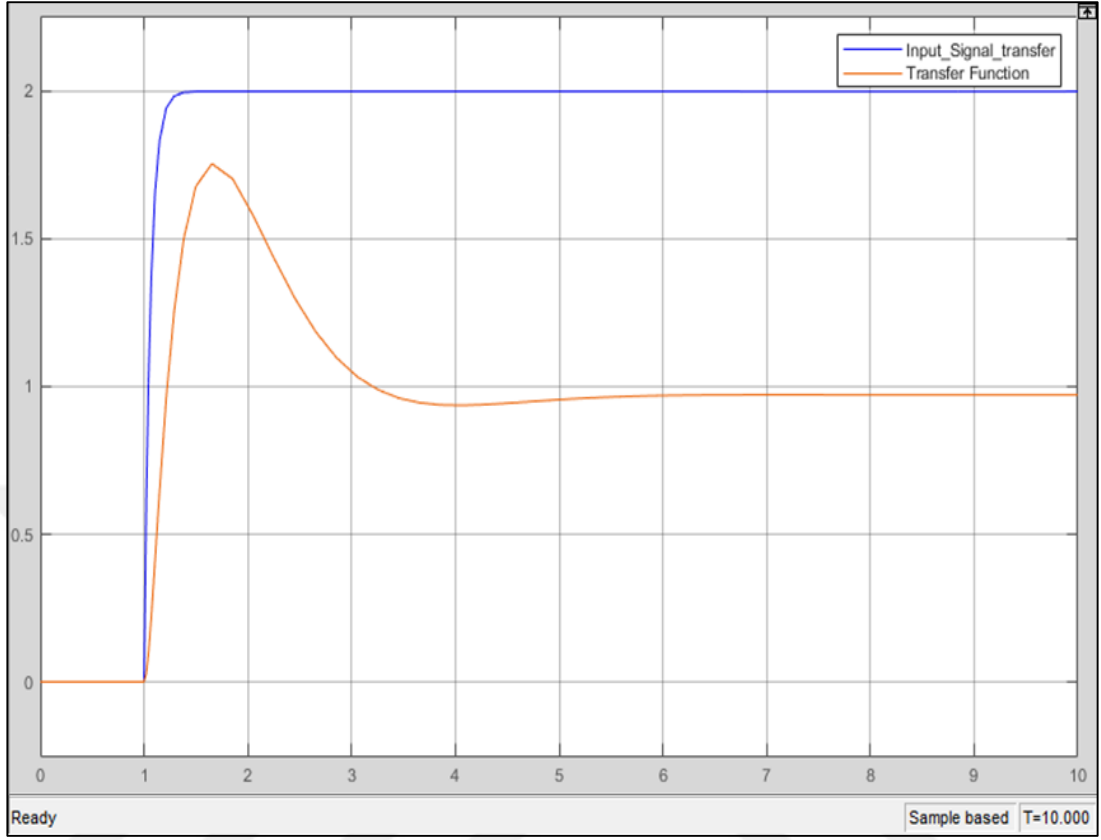
Şekil 4.6'daki grafikte görüldüğü üzere transfer fonksiyonundan çıkan sinyal yaklaşık olarak 4. saniyede hedef sinyale gelmiş durumdadır. Bu süreci karşılaştırmak adına transfer fonksiyonlarını değiştirmeden P ve PI uygulamalarının osiloskop çıktıları Şekil 4.7 ve Şekil 4.8'de verilmiştir.

PI uygulaması scope ekranı:



Şekil 4.7. Simulink PI uygulaması çıktı ekranı

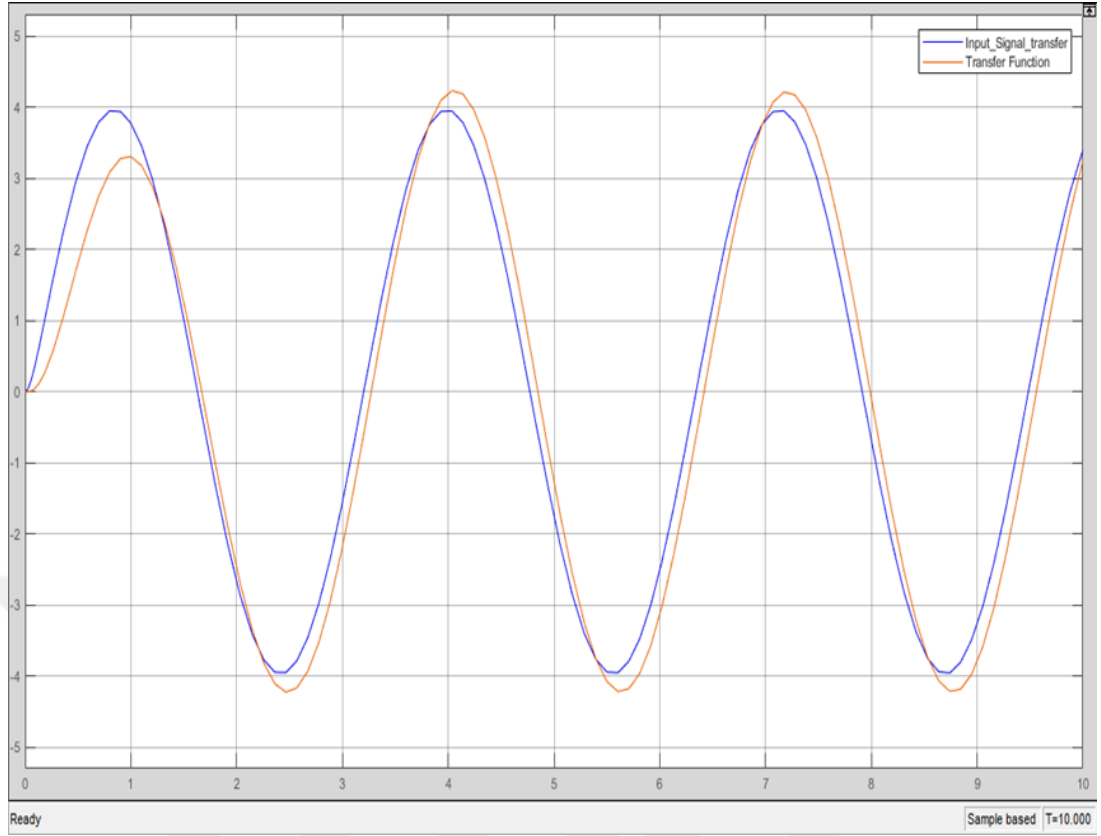
Sadece P uygulaması Scope ekranı:



Şekil 4.8. Simulink P uygulaması çıktı ekranı

Yukarıdaki üç grafik incelendiğinde PID ve PI kontrollerinin birbirine çok yakın sonuç verdiği görülmektedir. PID kontrolü yaklaşık 4. saniyede hedef sinyale ulaşırken PI kontrolü ise yaklaşık olarak 6. saniyede hedefe ulaşmıştır. Bu demek oluyor ki Proportional ve Integrator kontrolleri sinyali hedefe ulaştırmaya yeterlidir. Derivative işlemi ise PID tepki süresini kısaltır ama fazlada hesaplama ve elektronik devre üzerinde fazladan eleman anlamına gelir. Tepki süresinin çok önemli olmadığı yerlerde PI devresi kullanılabilir. Sadece P kontrolünü kullandığımızda oluşan grafikte ise transfer fonksiyonunun hedef sinyale hiç gelemediği görülmüştür. Yani tek başına P kontrolünü kullanmak sinyali yakalamada yeterli kalmayacaktır.

Yukarıdaki grafikler sabit bir değere ulaşmak için kullanılmıştır. Şimdi birde aynı PID değerleri ile devrenin girişine sinüs dalgası uygulandığında yeni oluşacak olan grafik şekil 4.9'daki gibi olacaktır.



Şekil 4.9. Simulink PID uygulaması Sinüs dalgası takibi

Şekil 4.9'daki sinüs dalgası için yapılan PID uygulamasında görüldüğü üzere sinüs dalgasını takip eden bir PID sinyali bulunmaktadır. Bu sinyal incelendiğinde sinüs dalgasının anlık durumuna göre tepki vererek biraz geriden de olsa PID sinyalinin hedef sinyali takip ettiği görülmüştür. Derivative hesaplamasının önemi bu grafikte ortaya çıkmaktadır. Sinyalin eğim değerine göre gittiği yön tahmin edilerek sinyalin daha hızlı yakalanabilmesi sağlanmıştır.

4.4. PWM Sinyali Oluşturma

Bu kısımda çalışmanın temel fonksiyonlarından biri olan PWM sinyalinin üzerinde durulacak ve motorlara, sürücüler aracılığıyla bu değerlerin nasıl aktarılacağından bahsedilecektir. Bu bölümde ARM Cortex-M4 tabanlı bir mikroişlemci olan STM32F4 mikro işlemcisini kullanarak bir PWM sinyali oluşturulacak. Hedef PWM sinyalinin sabit tutulması gereken değere bir konum algılama sensörü yardımıyla yaklaşılabilecektir.

4.5. STM32F429 PWM Oluřturma

Bu kısımda STM32F429 mikro denetleyicisi ile motor sürücülerinin hızlarını kontrol etmek amacıyla istenen değerdeki PWM işaretinin nasıl oluşturulacağı anlatılacaktır. PWM sinyali bir hedef değere göre oluşturulacaktır. Dolayısıyla ilk olarak bir konum sensörü aracılığı ile işlemcinin analog girişine konumu belirten bir gerilim uygulanacaktır ve PWM çıktısı ile yön bilgisi gelen bu değere göre değişecektir. Bu sayede motorlar sürekli olarak hedef açılara götürülecektir. PWM oluşturma esnasında dikkat edilmesi gereken işlemci frekansı ve oluşturulacak olan PWM değerlerinin özellikleridir. STM32F429 işlemcisi içerisinde PWM sinyali Timer donanımlarından oluşturulmaktadır. Timer aracılığıyla zamanlama ayarlanacak ve ayarlanan periyotlarda sinyal üretilecektir. Bu işlem için ilk olarak çalışma frekansları ele alınarak periyot, frekans ve duty cycle ayarlamaları yapılmalıdır.

İşlemci üzerindeki kullanacağımız Timer donanımı 42MHz clock bus frekansına sahiptir. Ayrıca kendi içinde barındırdığı PLL sayesinde bu frekansı ikiye katlayarak kullanmaktadır. Dolayısıyla donanımın beslediği bu bus 84MHz değerinde bir clock frekansına sahiptir. Aşağıdaki eşitlik ile Timer Frekansı hesaplanmaktadır.

$$\text{Timer Frequency} = \text{Bus Frequency} / (\text{Prescaler} + 1) \quad (4.9)$$

Bu çalışmada herhangi bir ön bölücü kullanılmayacaktır ve sonuç olarak bus frekansı hiç bölünmeden kullanılacaktır.

$$\text{Timer Frequency} = 84\text{MHz} / (0 + 1) = 84\text{MHz} \quad (4.10)$$

Timer üzerinden PWM oluştururken, periyot ve PWM frekansları birbirleri ile ilişkili değişkenlerdir. Aşağıdaki eşitlik ile bu ikisi arasındaki ilişkiyi görülmektedir.

$$\text{PWM Frequency} = \text{Timer Frequency} / (\text{Timer Period} + 1) \quad (4.11)$$

Buna bağlı olarak Timer periyot değerini aşağıdaki gibi hesaplanmaktadır.

$$\text{Timer Period} = \text{Timer Frequency} / (\text{PWM Frequency} - 1) \quad (4.12)$$

Bu aşamada motor sürücüsünün anlayabileceği PWM sinyal frekansı belirlenmelidir. Her iki cihazın senkronize bir şekilde çalışabilmesi için PWM frekans değerleri

senkronize olması gerekir. Farklı değerlerde haberleşme frekansları olursa istenen PWM değerlerinden sapmalar olacaktır. Bu durum manipulatörlerin hatalı çalışmasına sebebiyet verecektir.

Şekil 4.10'deki kod bloğu Timer konfigürasyonlarını tanımlamaktadır.

```
TIM_BaseStruct.TIM_Period = 8999; /* 10kHz PWM */
TIM_BaseStruct.TIM_ClockDivision = TIM_CKD_DIV1;
TIM_BaseStruct.TIM_Prescaler = 1;
TIM_BaseStruct.TIM_RepetitionCounter = 0;
/* Initialize TIM4 */
TIM_TimeBaseInit(TIM4, &TIM_BaseStruct);
/* Start count on TIM4 */
TIM_Cmd(TIM4, ENABLE);
```

Şekil 4.10. TIMER konfigürasyonu kod bloğu

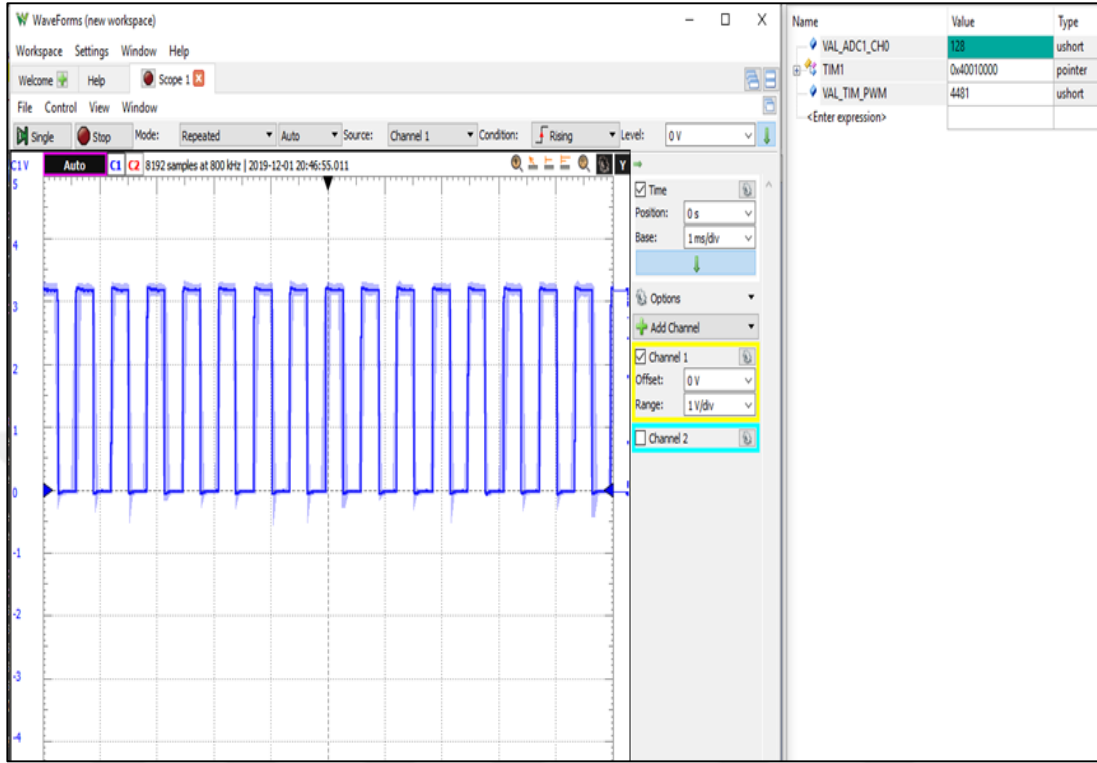
Timer yapılandırmalarını oluşturulduktan sonra PWM haberleşmesi üzerinde ayarlamalar yapılacaktır. PWM haberleşme sisteminde “Duty Cycle” kavramının bilinmesi gerekmektedir. Duty cycle terimi PWM sinyalinin bir periyodunun ne kadarlık süresinin HIGH durumunda kaldığını belirtmektedir [26]. Eğer hedeflenen PWM değeri yukarıdaki kod bloğundaki TIM_Period değerinden büyük olursa PWM çıktımız sürekli HIGH durumunda olacaktır. Dolayısıyla hedef PWM değeri 0 ile TIM_Period değeri arasında olmalıdır. Eğer daha geniş periyod isteniyorsa Timer donanımının ön bölücü değerinin artırılması gerekmektedir.

```
TM_TIMER_Init();
// Init PWM
TM_PWM_Init();
while(1==1)
{
    VAL_ADC1_CH0 = fnc_readADC();
    VAL_TIM_PWM = fnc_scale(VAL_ADC1_CH0,0,255,0,8999);
    TIM4->CCR1 = VAL_TIM_PWM;
    bekle_cycle(1000);
}
```

Şekil 4.11. Sensör verisi okuduktan sonra ölçeklendirip PWM çıktısı olarak atama

Şekil 4.11'teki kod bloğunda TIMER ve PWM yapılandırma fonksiyonları çağırıldıktan sonra konum sensörü üzerindeki gerilim alınarak 0 – 8999 arası ölçeklendirilir. Ölçeklenen bu değer direk olarak PWM çıkışına aktarılarak gerekli

PWM alınmış olur. Bu kodların çalışmasıyla birlikte PWM çıktısı şekil 4.12'deki gibi olacaktır.



Şekil 4.12. TIMER PWM çıktı örneği

4.5.1. STM32F429 PID ile PWM sinyali oluşturma

Bu kısımda bir önceki bölümdeki işlemin PID fonksiyonu kullanılarak istenen PWM değerine PWM kullanarak nasıl gelineceği anlatılacaktır. Bu kısımda PWM sinyali, hedef değeri PID ile işlenerek oluşturulacaktır.

Bu hesaplama yapılırken işlemci içerisinde normal program çalışmasını engellememesi için farklı bir donanım katmanı kullanılacaktır. Bir işlem katmanı hesaplamayı yaparken diğer katmanda bu hesaplama sonucu oluşan değeri PWM çıkışına aktaracaktır. Aksi takdirde bu iki işlemi aynı katmanda yapılırsa PWM sinyalinde gecikmeler meydana gelecektir. Assembly kodları incelendiğinde bu iki işlem neredeyse birbiri kadar işlem adımıyla yapılabilmektedir. Dolayısıyla hesaplama işlemi ve oluşan sonucu çıkışa aktarma işlemi iki kat hızlanmış olacaktır.

ARM mimarisine sahip STM32 serisi mikroişlemciler kendi içlerinde DMA donanımları bulundurmaktadır. DMA donanımları sayesinde işlemci normal işlemine

devam ederken analog girişe bağlı sensör verisi direk olarak bellekteki ilgili değişkene aktarılacak. Sensör değerini okuma işlemiyle eş zamanlı olarak iç içe çalışan kesmeler gerekli hesaplamayı yapabilmek adına ilgili fonksiyonu belli zaman aralıklarıyla tetikleyecek ve PWM çıkışı PID algoritmasıyla birlikte hesaplanacak. Hesaplamanın ardından işlemci, oluşan PWM değerini, PWM çıkışına aktaracak. Bu işlemler ile işlemcinin temel fonksiyonları çok daha kısa bir zaman içerisinde, ana program döngüsünü zaman kaybetmeden yapabilecektir. Aşağıdaki örnek kod bloğunda bu hesaplamanın nasıl yapıldığı gösterilmiştir. Bu hesaplama için gerekli olan hedef değer, anlık sensör değeri, önceki fark değeri gibi değerleri kullanarak anlık değeri sürekli hedefe değere yaklaştırmaya çalışacak.

```
void TIM2_IRQHandler()
{
    if(TIM_GetITStatus(TIM2, TIM_IT_Update) == SET) //check the TIMER is active
    {
        //always reset the interrupt
        TIM_ClearITPendingBit(TIM2, TIM_IT_Update);
        MOTOR_1_PID_PWM = fnc_calculatePID(MOTOR_1_SENSOR_VALUE, MOTOR_1_DESIRED_VALUE);
    }
}

uint16_t fnc_calculatePID(uint16_t _sensor_position, uint16_t _desired_position)
{
    error_value = _desired_position - sensor_position;
    integral_value = integral_value + (error_value * Kt);
    derivative_value = (error_value - previous_error_value) / Kt;
    output_value = (Kp * error_value) + (Ki * integral_value) + (Kd * derivative_value);
    previous_error_value = error_value;

    return output_value;
}
```

Şekil 4.13. Kesme içerisinde PID hesaplama fonksiyonları

Şekil 4.13'daki kod bloğunda görüldüğü üzere TIM2_IRQHandler fonksiyonu NVIC tarafından belli zaman aralıklarıyla tetiklenen fonksiyon ve bu fonksiyonun içerisinde PID hesaplama formülasyonları uygulanıp ilgili değişkenlere atanacaktır. PID fonksiyonunun içerisine gönderilen sensör ve istenen pozisyon değerleri referans alınarak gerekli PID hesaplamaları yapılmaktadır. Oluşan çıktı değeri, ilgili PWM değişkenine atanmaktadır. Bu değişken değerine göre işlemci çıkışında PWM sinyali oluşturulmaktadır.

4.6. DC Motor Nedir

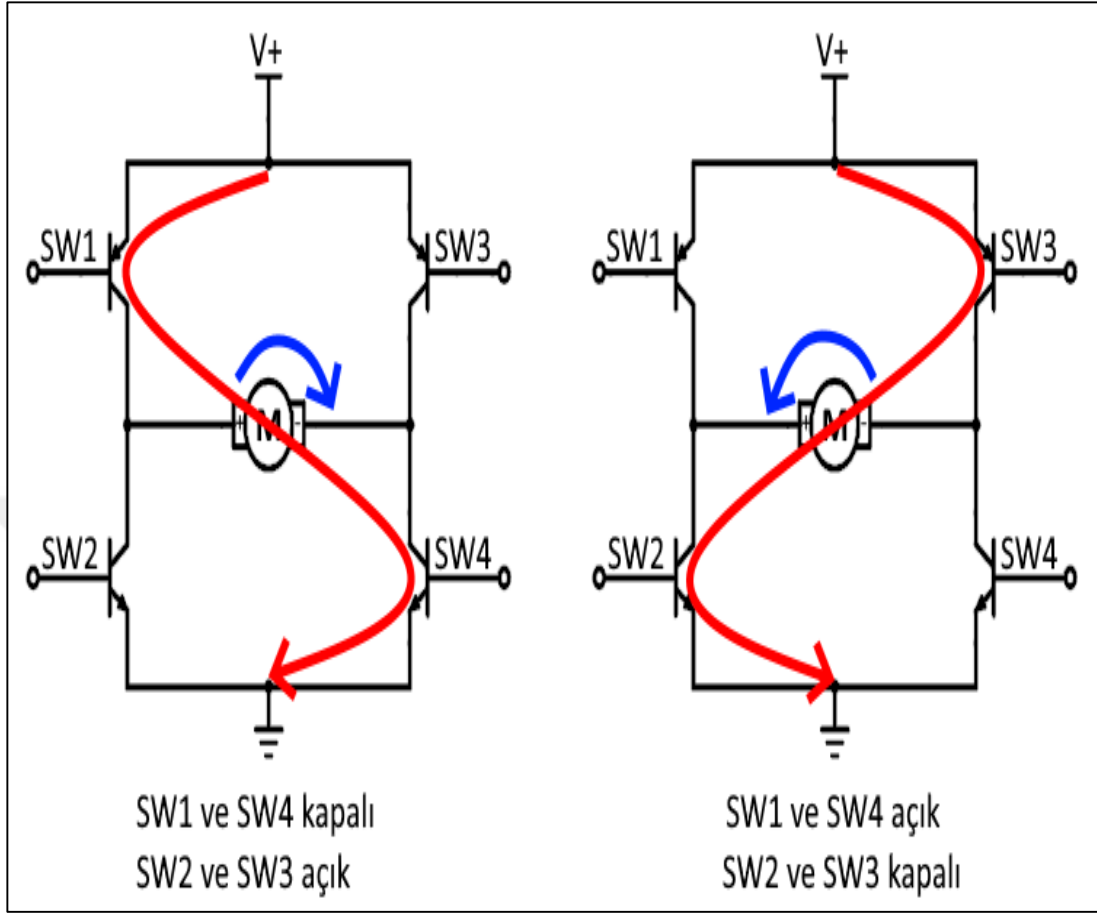
DC Motorlar, DC elektrik enerjisini harekete çeviren motorlardır. Bu motorlar birkaç farklı yöntem ile çalışmaktadırlar. Motorun içerisindeki sabit stator sarımlarına pozitif gerilim uygulanır ve hareketli rotor kısmına negatif gerilim uygulanır. Uygulanan bu gerilim sonucu bu iki kısım arasında bir çekim kuvveti oluşur ve motor hareket etmeye başlar.

Temel olarak fırçalı, fırçasız, step ve servo motorlar olmak üzere 4 temel çeşidi vardır. Fırçalı motorlar, rotor ve stator kısımları birbirlerine fırçalar veya iletken kömürler ile temas ederek üzerinde enerji geçmesi ile hareket ederler. Bu tip motorlarda, fırça ve kömürlerin ömürleri bittiğinde motorlar çalışamaz ve bu parçaların değiştirilmeleri gerekmektedir. Fırçasız motorlar, fırçalı motorları aksine elektrik enerjisini fırça veya kömür üzerinden iletmek yerine manyetik olarak temassız bir şekilde hareket etmektedirler. Fırçasız motorlar harici bir sürücüye ihtiyaç duyarlar. Bu sürücü motorun içerisindeki sabit mıknatısları yüksek enerji ile anahtarlayarak rotor üzerindeki iletkenlere manyetik kuvvet uygular ve hareketi oluşturur. Step motorlar, konum hassasiyeti gerektiren işlemlerde kullanılmaktadır. Hareketini girişlerine uygulanan enerjiyi anahtarlama mantığıyla rotora aktarır ve adım adım hareket eder. Hareketin hassas olması için çeyrek veya yarım adım şeklinde de hareket modu bulunmaktadır. Servo motorlar, belli bir derece açı yapmak için kullanılan motorlardır. İçerisindeki sensör ile konumunu net olarak belirtir. Step motorlardan farkı step motorlarda konum geri bildirimi bulunmaz. Servo motorlar konumlarını bildirerek hassas ölçümler ve işlemler yapılmasını sağlar.

4.7. DC Motor Kontrolü ve H-Bridge Devresi

DC motorlar motor hareketini sağlamak için pozitif ve negatif olmak üzere iki adet girişe sahiptir. Bu kutuplar enerjilendirildiğinde motor hareket yönü akımın akış yönüne bağlıdır. Motorun dönüş yönünü değiştirmek için akımın akış yönünü değiştirmek gerekmektedir. Bağlantıları terslediğimizde DC motorunda dönüş yönü değişecektir. Normal çalışma esnasında yön değiştirmek istendiğinde bağlantılar değiştirilemeyeceği için bu işlem H-Bridge devresi ile gerçekleştirilmektedir [27].

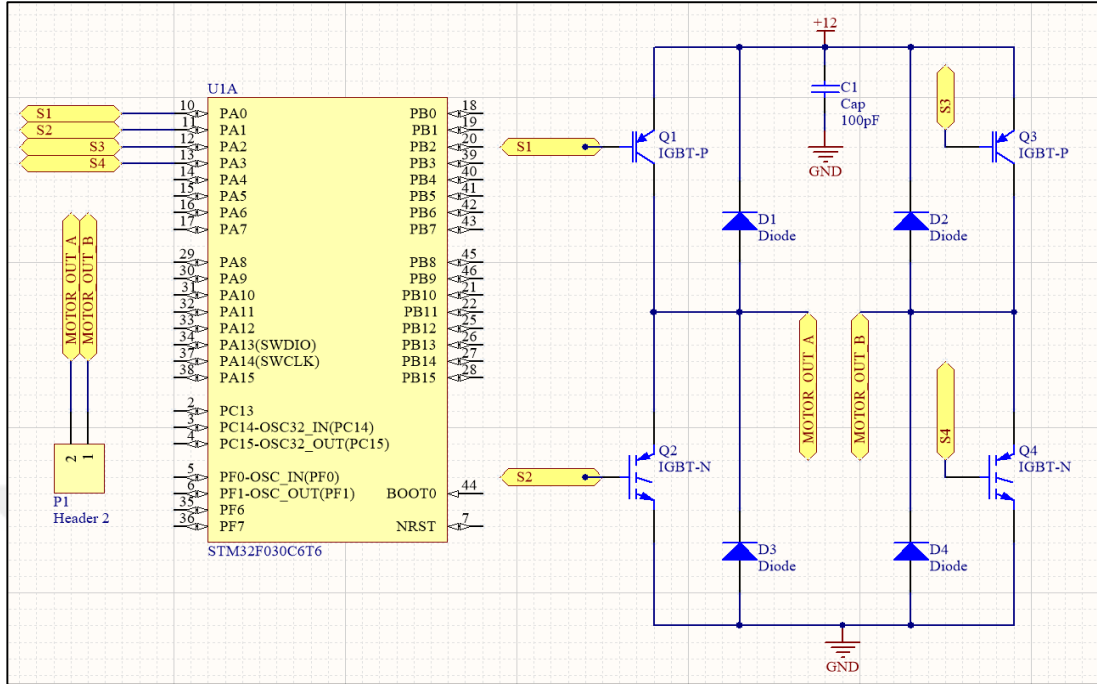
Şekil 4.14'te temel olarak bir H-Bridge devresi görülmektedir.



Şekil 4.14. H-Bridge diyagramı

Şekil 4.14'te görüldüğü üzere 4 adet anahtarlama elemanı ile birlikte gerilim girişi ve gerilim çıkışı bulunmaktadır. Motorun dönüş yönü değiştirilmek istendiğinde gerilimin yolu anahtarlama elemanları ile değiştirilerek motorunda dönüş yönü değiştirilmiş olur. Burada anahtarlama elemanları olarak genellikle transistör veya mosfet kullanılmaktadır. Bazen rölelerde anahtarlama elemanları olarak kullanılmaktadırlar. Fakat röleler manyetik olarak anahtarlama yaptığı için yarı iletken elemanlara göre çok daha yavaş çalışacaktır ve hızlı açılıp kapanma durumlarında manyetik kısmı hasarlanıp sürekli açık veya kapalı konumda kalma durumları oluşabilir. Bunun sonucu olarak hasarlanın rölenin pozisyonlarına göre devrede kısa devre olabilir veya istenen şekilde çalışmayabilir.

Şekil 4.15'te basit olarak bir H-Bridge köprüsü şematiği görülmektedir.



Şekil 4.15. Örnek IGBT H-Bridge devresi şeması

Şekil 4.15'te anahtarlama elemanları olarak yüksek güçlü IGBT kullanılmıştır. IGBT aracılığı ile elektronik olarak çok hızlı bir şekilde anahtarlama yapılabilmektedir. Eğer burada IGBT yerine röle kullanılsaydı anahtarlama için manyetik bir hareket gerekecekti ve hassas işlemler için uygun anahtarlama hızı elde edilemeyecekti.

Bu çalışmada motorların konumları kontrol edilecektir ve motor açılarının değerleri hareketin etkili olabilmesi için çok önemli olduğundan, elektronik olarak kontrol edilebilen ve üzerinden yüksek akım geçiren transistör tipi anahtarlama araçları kullanılacaktır. Ayrıca motorun taşıyacağı ağırlık yüksek olduğundan bu ağırlığı taşıyabilmek için çekeceği akımda yüksek olacaktır. Bu yüzden üzerinden sürekli olarak yüksek akım geçirebilen bir transistör tipi anahtarlama elemanı seçilmiştir.

4.8. PID Kontrol ile DC Motor kontrolü

PID ile motor sürme işleminde motor hızını işlemcinin PWM çıkışından alınan sinyalin duty cycle değeri ile sürücüye dönüşün hızı belirlenmektedir. Bu çalışmada kullanılacak olan sürücü en fazla 40KHz PWM sinyalini desteklemektedir. İşlemci PWM çıktısı bu değere göre ayarlanacaktır.

PWM değerini hesaplama işleminde ayrı bir donanım katmanı kullanılacak ve bu sayede hesaplamanın arka planda yapılması sağlanacaktır. Arka planda yapılan hesaplama işleminde işlem ana işlem parçacığında kullanılan cycle sayısını azaltarak birim zamanda yapılan hesaplanmış PWM çıkış sayısı arttırılmış olunacaktır. Bu sayede PWM sinyal çözünürlüğü artmış olacaktır.

Şekil 4.16'daki kod bloğunda seçilen motor sürücüsü için oluşturulan 20 kHz frekansa sahip PWM oluşturan bir timer donanım katman yapılandırması verilmiştir. Yapılandırmanın ardından donanım katmanı çıkış PWM Duty Cycle değeri %25 olarak ayarlanmıştır.

```
void FNC_TIMER_Start(void) {
    TIM_TimeBaseInitTypeDef TIM_BaseStruct;
    /* Enable clock for TIM4 */
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
    TIM_BaseStruct.TIM_Prescaler = 0;
    TIM_BaseStruct.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_BaseStruct.TIM_Period = 17999; /* 20kHz PWM->17999 */
    TIM_BaseStruct.TIM_ClockDivision = TIM_CKD_DIV1;
    TIM_BaseStruct.TIM_Prescaler = 1;
    TIM_BaseStruct.TIM_RepetitionCounter = 0;
    /* Initialize TIM4 */
    TIM_TimeBaseInit(TIM4, &TIM_BaseStruct);
    /* Start count on TIM4 */
    TIM_Cmd(TIM4, ENABLE);
}

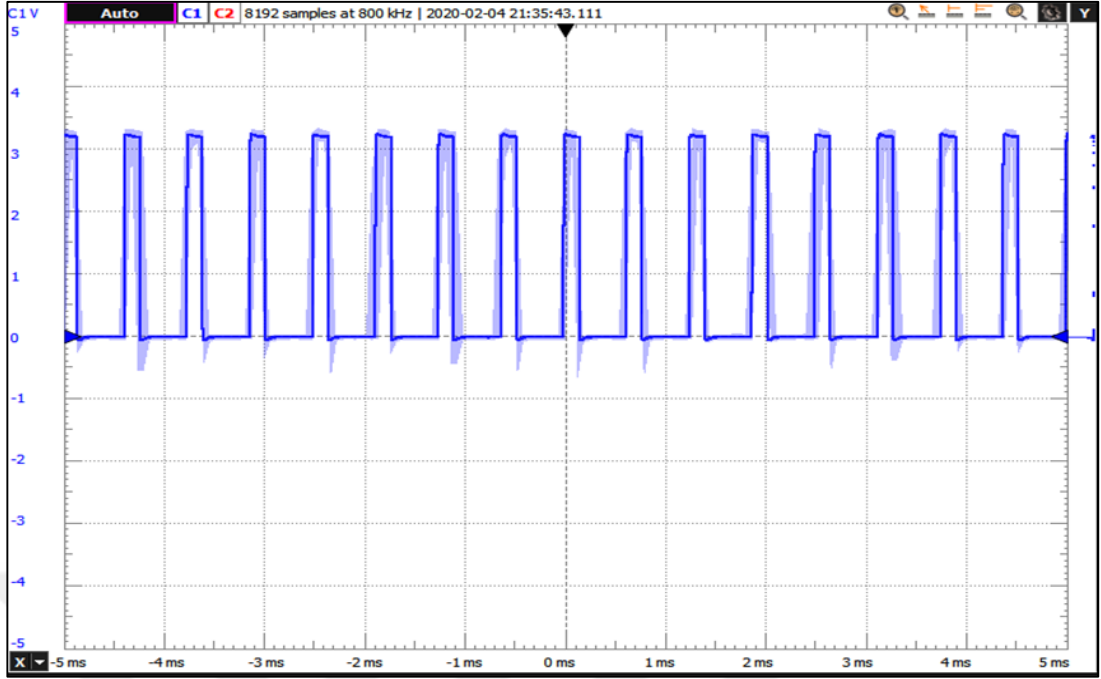
void FNC_PWM_Start(void) {
    TIM_OCInitTypeDef TIM_OCStruct;

    TIM_OCStruct.TIM_OCMode = TIM_OCMode_PWM2;
    TIM_OCStruct.TIM_OutputState = TIM_OutputState_Enable;
    TIM_OCStruct.TIM_OCPolarity = TIM_OCPolarity_Low;

    TIM_OCStruct.TIM_Pulse = 4499; /* 25% duty cycle */
    TIM_OC1Init(TIM4, &TIM_OCStruct);
    TIM_OC1PreloadConfig(TIM4, TIM_OCPreload_Enable);
}
```

Şekil 4.16. TIMER ve PWM başlatma kod bloğu

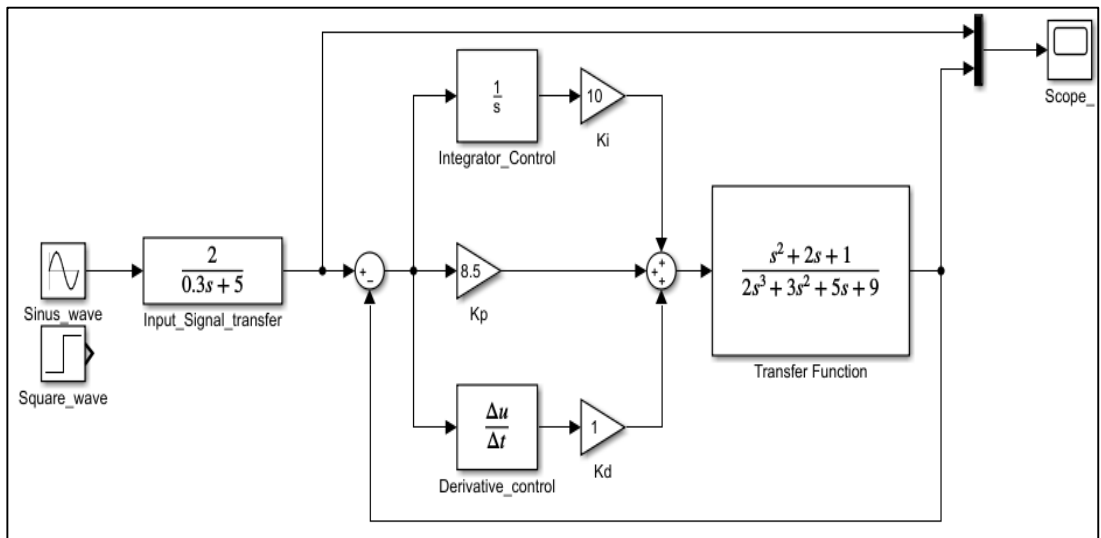
Yukarıdaki TIMER ve PWM başlatma fonksiyonlarının çalıştırılmasından sonra işlemcinin PWM çıkışından alınacak olan PWM sinyali şekil 4.17'deki gibi olacaktır.



Şekil 4.17. TIMER ve PWM başlatma metodundan sonra oluşan PWM sinyali

Görüldüğü üzere %25 olarak ayarlanan Duty Cycle değeri ile PWM çıkışından 1ms içinde yaklaşık olarak %25 kadar çıkış HIGH durumunda kalmaktadır. Bu sayede PWM sinyali %25 olarak ayarlanmış olmaktadır.

PWM sinyalini PID hesaplaması ile oluşturmak için ise motorun konumunu hesaplayan sensörden ve komut sinyal değerinin karşılaştırılması ile hesaplanacaktır. Şekil 4.18’de motorun dönüş hızını belirlemede kullanılacak olan PID fonksiyonunun Matlab şeması görüntüsü bulunmaktadır.



Şekil 4.18. MATLAB Simulink PID şeması

Şekil 4.19’da ise yukarıda verilen matlab görüntüsünün gömülü sistem üzerinde yazılmış kodları bulunmaktadır. Geliştirme STM32F4 seri işlemci ailesi için C dilinde ve uVision Keil programında yapılmaktadır.

```
#define Kd 1
#define Ki 0.1
#define Kp 10
#define Kt 100

uint16_t fnc_calculatePID(long _sensor_position, long _desired_position)
{
    //calculate the difference between sensor and command values
    error_value = _desired_position - sensor_position;
    //calculate the integral and derivateive values
    integral_value = (error_value * Kt) + integral_value ;
    derivative_value = (error_value - previous_error_value) / Kt;
    //calculate the PID value
    output_value = (Kp * error_value) + (Ki * integral_value) - (Kd + derivative_value);
    //keep the value to use again when needed
    previous_error_value = error_value;
    //decide the direction at loop, make positive the value
    return output_value < 0 ? (output_value * (-1)) : output_value ;
}
```

Şekil 4.19. PID hesaplama fonksiyonu kod bloğu

Şekil 4.19’daki kod bloğunda ilk olarak komut değeri ile sensörden alınan değer arasındaki fark alınarak hata değeri hesaplanmıştır. Hesaplanan bu hata değeri ile integral ve derivative değerleri belirlenen katsayılar ile hesaplanmıştır. Sonrasında çıkış değeri oluşturmak için ise hata değerine proportional değeri ile hata değeri arasında çarpma işlemi yapılarak oluşturuldu. Bu çıkış değeri daha sonra kullanılmak üzere bir değişkende tutuldu ve son olarak PID çıktısının pozitif olarak çıkması sağlandı. Çıkış değerinin pozitif olarak çıkarılmasının sebebi motorun dönüş yönü ise ana program döngüsü içerisinde belirlenecektir.

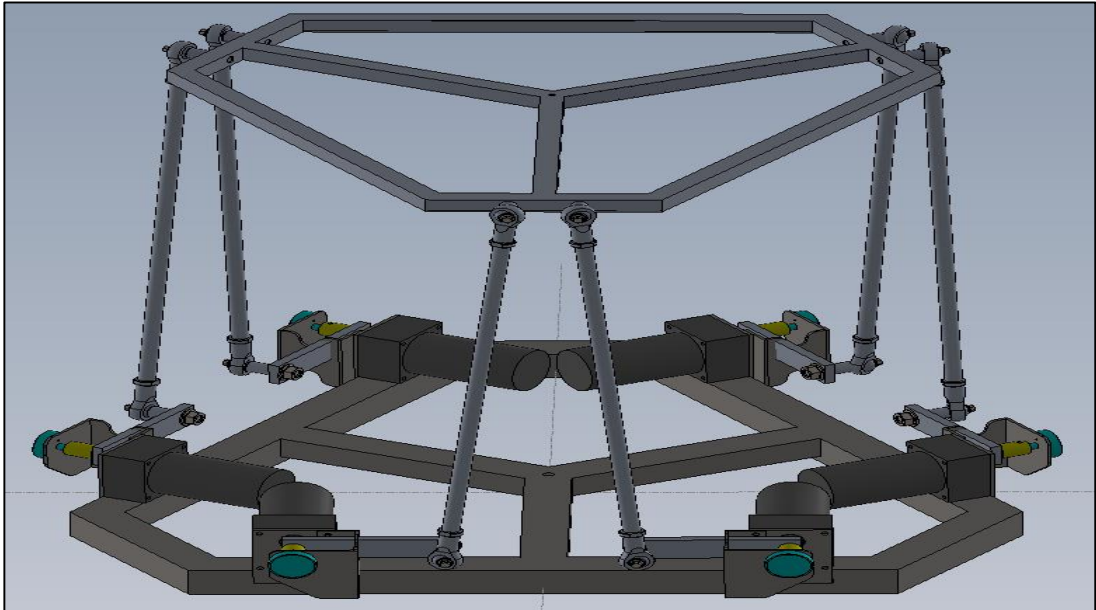
5. HAREKETLİ UÇUŞ PLATFORMUN TASARIMI VE ÜRETİMİ

5.1. Malzemelerin Üretilmesi

Matematiksel modellemesi yapılan malzemeler, modelleme esnasında belirlenen değerlere göre Solidwork programında üç boyutlu olarak parça parça tasarlanmıştır. Ardından tasarlanan bu malzemeler için yük taşıma durumlarına göre malzeme seçimleri yapılmıştır. Yük taşımayan malzemeler sert plastik malzemeden; platformu taşıyacak olan malzemeler ise esnek alüminyum malzemeden üretilmiştir. Seçilen esnek malzeme sayesinde platform maruz kalabileceği burulma kuvvetine karşı hasarlanmayacak ve yük taşımaya devam edecektir.

Solidworks ile tasarlanan parçalar, yine solidwork programı üzerinde monta edilerek hareketli platform tasarımının çalışma sonundaki halini göstermektedir. Aynı zamanda montaj tasarımında platform bağlantıları hareketli olarak tasarlandığı için platformun çalışma sonundaki hareketi buradan kontrol edilebilecektir.

Şekil 5.1, şekil 5.2 ve şekil 5.3'teki görsellerde Solidworks ortamında hazırlanan modeli ve bu modelden üretilen sabit kısım parçaları görülmektedir.



Şekil 5.1. Solidworks ortamında hazırlanan Hareketli platform Modeli



Şekil 5.2. DC motor montajı



Şekil 5.3. Üretilen base kısım

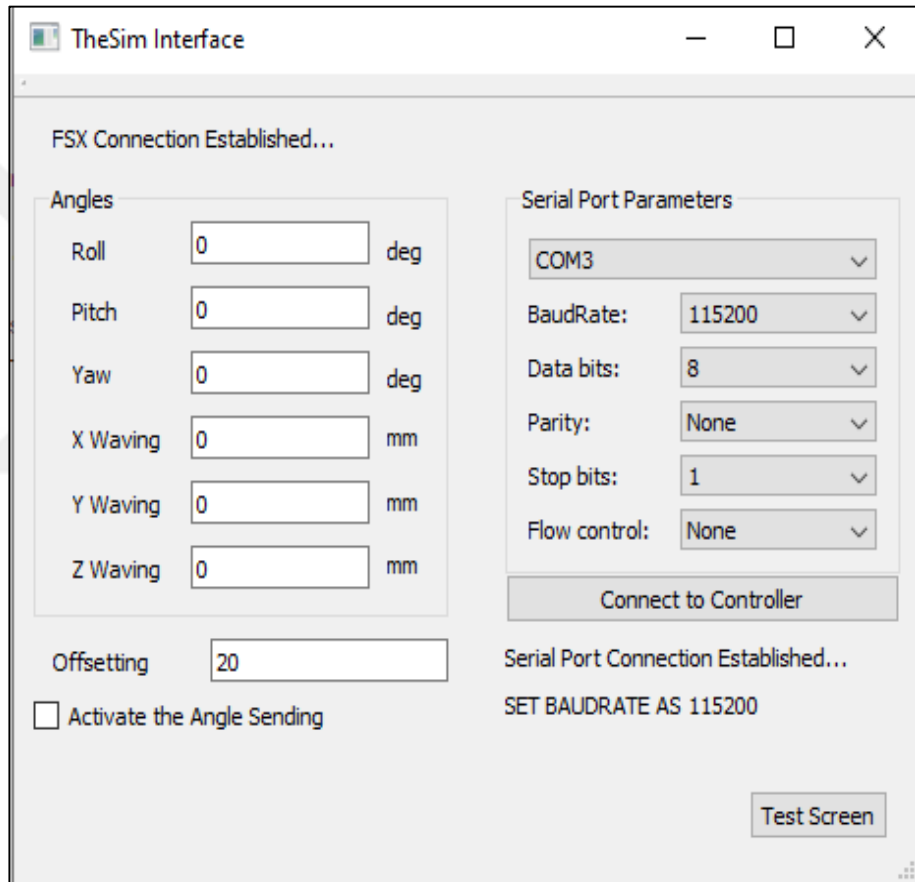
5.2. Elektronik Kontrol Birimi Montajı

Bu bölümde hareketli uçuş platformunun elektronik kontrol biriminin üretim aşamalarından bahsedilecektir. Üretim üç ana kısımdan oluşmaktadır. Bu aşamalardan ilki üretilmek istenen hareketli platformun matematiksel olarak modellenmesi ve gerekli matematiksel hesaplamaların istenen platform boyutlarına uygunluk durumundan emin olmaktır. Bu aşama üzerinde çalışma esnasında ilk olarak kullanılmak istenen dönüşüm teoremi belirlenmiş olup sonrasında ise bu teoremin matematiksel olarak hesaplamaları, motorların dönüşlerini kontrol edecek olan mikroişlemci için programa dökülmüştür. Bu programın tasarımı esnasında motorların konumları, başlangıç pozisyonları, motor kolu ve platform rodları ve platform-rod bağlantı konumlarının net olarak belirlenmesi ve hesaplamaların bu düzlemde yapılması gerekmektedir.

Hazırlanan programın test edilmesinin ardından alınan başarılı sonuçlardan sonra ikinci aşama olan hareket platformunun belirlenen boyutlara göre tasarlanması ve üretilmesi aşamasına devam edildi. Bu aşamada üç boyutlu çizime olanak tanıyan Solidworks yardımcı programından yardım alındı ve çizimler bu program aracılığı ile üç boyutlu olarak tasarlandı. Tasarlanan bu üç boyutlu çizimler kullanılacağı yerlerdeki güç taşıyabilme durumları göz önüne alınarak seçilen malzemeler ile üretilmiştir. Yük taşımayacak olan malzemeler 3D yazıcı yardımı ile sert plastik malzemeden; yük taşıyacak olan malzemeler ise yerine göre alüminyum alaşımlardan üretilmiştir.

Mekaniksel malzemelerin üretiminden sonra montaj işlemine kısmen başlanılmıştır. İlk olarak motorların ve motorlara bağlı motor kollarının ilgili konumlara bağlantılar yapılmıştır. Ardından motorların dönüş esnasındaki konum bilgisini alabilmek adına hareket sensörleri ile konum limit sensörlerinin montajı yapılmıştır. Buraya kadar yapılan montajda elektronik sistem sabit kısma bağlanıp test edilebilir duruma gelmiştir ve tasarlanan devrenin bağlantıları yapılmıştır. Bu esnada manuel olarak motorlar hareket ettirilerek hareket-konum sensörlerinin çalışma durumları kontrol edilmiştir. Ayrıca limit sensörlerinin de doğru bir şekilde çalıştığı kontrol edilerek motorların bireysel olarak çalışma limitleri değerlerini oluşturulduğundan emin olunmuştur.

Çalışmanın üçüncü aşaması olarak, hareket verilerinin otomatik olarak MS Flight Simulator uygulamasında simüle edilen uçuşun, uçuş verilerini anlık olarak elde edip hareket kontrol sistemine seri port üzerinden gönderen bir ara yüz uygulaması geliştirilmiştir. Bu ara yüz uygulaması C++ programlama dilinde QT Frameworkü kullanılarak geliştirilmiştir. Bu uygulama MS Flight Simulator X tarafından yürütülen simülasyondaki anlık uçuş verilerini alarak belirlenen öteleme değerleriyle öteledikten sonra seri port üzerinden belirtilen haberleşme hızında göndermektedir. Şekil 5.4'teki görselde program ara yüzü görülmektedir.



Şekil 5.4. TheSim Interface yazılımı ekran görüntüsü

TheSim Interface programı ile birlikte operatörün vestibular sistemine yapılacak olan etkinin daha da arttırılabilmesi adına bir gösterge yazılımı geliştirildi. Bu yazılımda uçakların temel hareket göstergeleri olan ve Basic-T olarak adlandırılan göstergeler ilk olarak tasarlanmıştır. Sonrasında diğer göstergelerin tasarımları tamamlanmıştır. Bu gösterge yazılımı sayesinde simüle edilen uçuş esnasında, simüle edilen uçağın anlık olarak yatış değerleri takip edilebilmektedir. Aynı zamanda simüle edilen

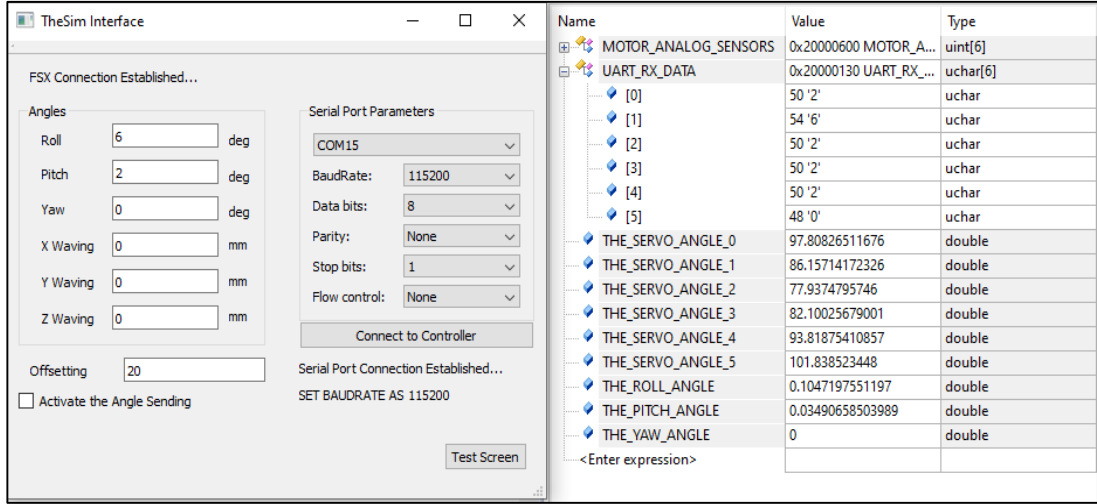
uçağın kalan yakıt, yakıt harcaması, yağ sıcaklığı gibi diğer parametreleri de uçak ile birebir aynı olacak şekilde tasarlanmıştır.



Şekil 5.5. Visual display programı

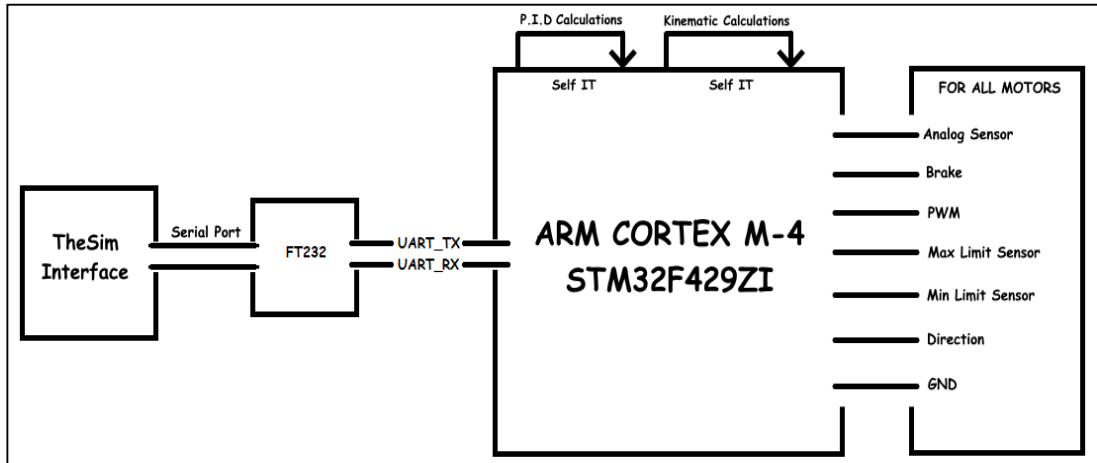
Şekil 5.5'te gösterge yazılımından bir görsel bulunmaktadır. Şekilde görüldüğü üzere gösterge yazılımında uçağın temel olarak hız, suni ufuk, dikey hız, yatış, pusula ve irtifa göstergeleri öncelikli olarak sol tarafa konumlandırılmıştır. Aynı zamanda diğer önemli parametrelerin gösterilebilmesi ve daha gerçekçi bir uçuş sağlanabilmesi adına uçağın diğer parametrelerini gösteren gösterge ekranları da oluşturulmuştur. Bu yazılım da, TheSim Interface yazılımı gibi Qt Framework kullanılarak C++ programlama dilinde geliştirilmiştir.

Mikro işlemci tarafında tasarlanan program ise seri port üzerinden gelen verileri sürekli bekleyerek gelmesi gereken boyutta veri geldikten sonra gerekli doğruluk kontrolünden geçirmektedir. Ardından hesaplamaları yapabilmek için değerleri radyana dönüştürüp radyanal değerler üzerinden gerekli hesaplamalar yapılmaktadır. Mikro işlemci tarafında geliştirilen program C dilinde geliştirilmiştir. C dilindeki standart kütüphaneler trigonometrik hesaplamaları radyanal cinsten yapmaktadır. Bu yüzden bu uygulamada da seri port üzerinden gelen yatış parametrelerini hesaplama fonksiyonuna göndermeden önce radyan karşılıklarına dönüştürmeliyiz. Şekil 5.6'teki görselde bu dönüşüme bir örnek görülmektedir.



Şekil 5.6. TheSim Interface yazılımı ve Mikroişlemci hata ayıklama ekranı

Üçüncü aşamada geliştirilen ara yüz programının mikroişlemci üzerinde yazılan program ile doğru haberleşmesi uzunca bir süre hatasız çalışabilmesi adına test edildikten sonra kontrol devresinin bağlantıları yapılmıştır. Bu bağlantılar aşağıdaki şemada görülmektedir. TheSim ara yüz programından seri port aracılığı ile alınan veriler, transfer çipi aracılığı ile ARM Cortex M-4 mimarisi ile geliştirilen STM32F429 işlemcisine gönderilmektedir. İşlemci üzerinde oluşturulan iki adet kendi içerisinde kesme kaynağı yardımıyla sürekli olarak kinematik hesaplamalar gerçekleştirilmektedir. Bu hesaplamalar sonucu oluşan değerler ve sensör sürekli olarak kontrol edilen sensör değerleri yardımıyla PID hesaplamaları sürekli olarak gerçekleştirilmektedir. Oluşan bu değerlere göre bütün motorların PWM ve yön çıkışları oluşturulup her bir motora ayrı ayrı gönderilmektedir. Bu sayede motorların hareketi ile hareketli platform istenen konuma gelmiş olmaktadır.



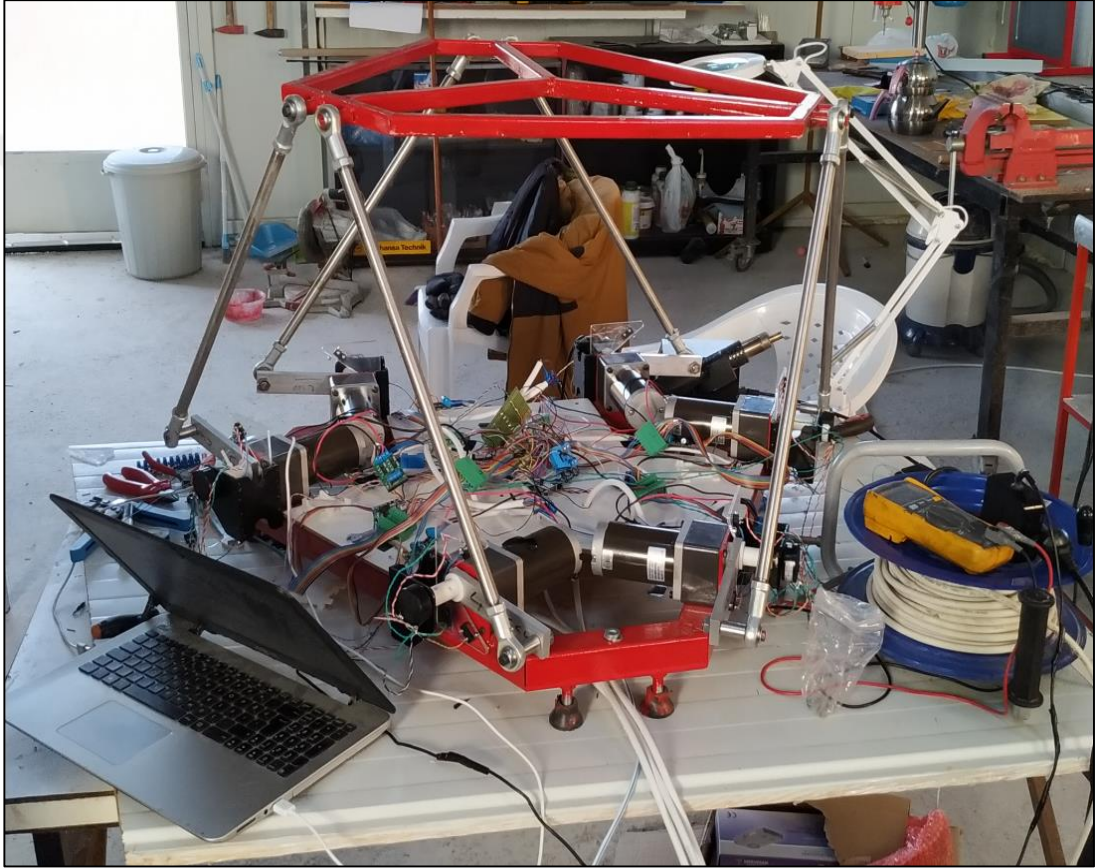
Şekil 5.7. Elektronik kontrol sistem çalışma şeması

Yapılan bu bağlantıların ve testlerin ardından ikinci aşamada başlanılan bağlantılara devam edilmiştir. Burada hareketli platformun bağlantıları tamamlanmıştır. Ardından hesaplanan başlangıç yükseklik değerleri kontrol edilmiştir. Üretim ve montaj tamamlanmıştır.



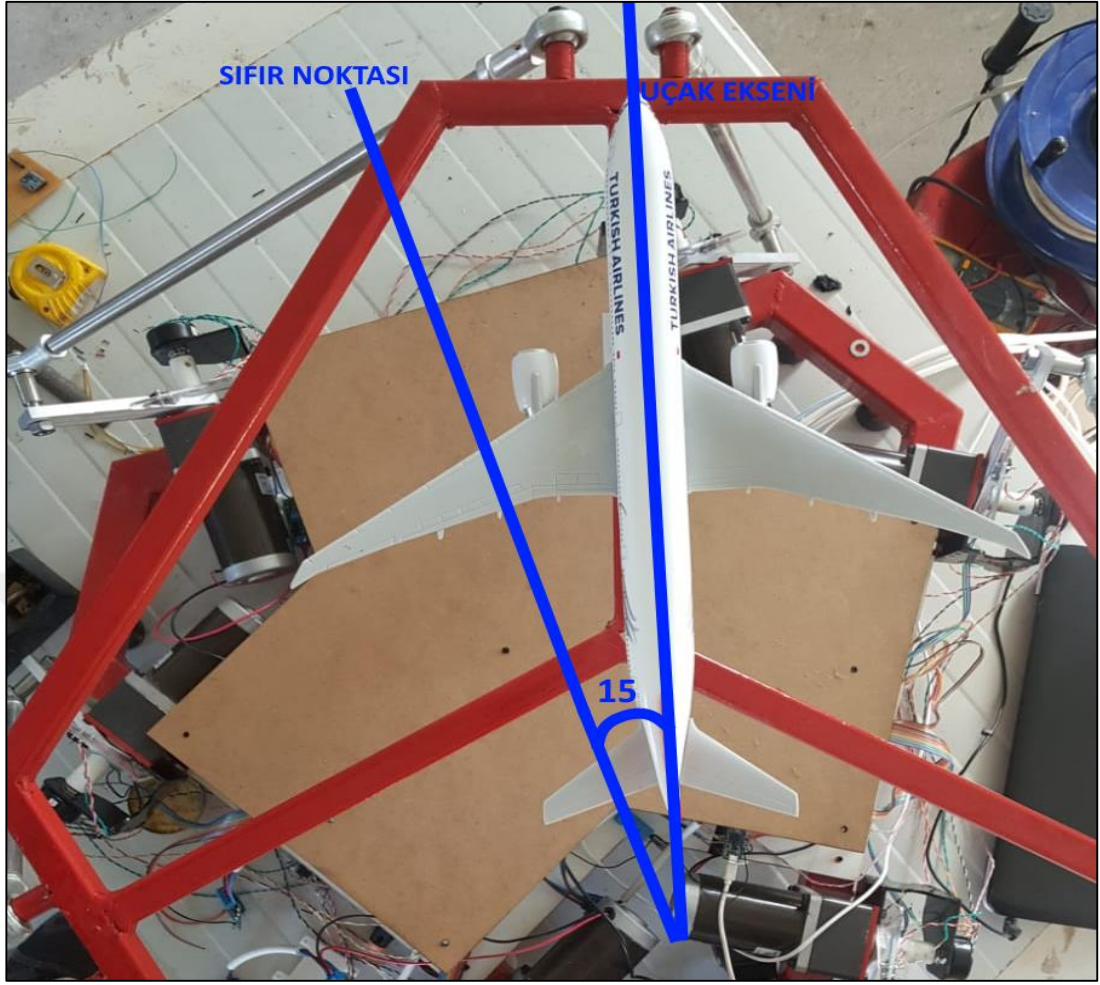
6. TASARLANAN 6DOF PLATFORMUN GERÇEKLENMESİ

Bu bölümde, gerçekleştirilen çalışmanın sonuçları hakkında bilgiler verilecektir. Çalışma ile ilgili görseller üzerinden hareket mekanizmasını ve hareketlerin istenen yatış değerlerine gelme durumlarından bahsedilecektir.



Şekil 6.1. Hareketli platformun ilk test aşaması

Şekil 6.1’de gerçekleşen tasarımın test aşaması görülmektedir. Test aşamasında tasarlanan elektronik sistemin bağlantıları kontrol edildikten sonra motorları sürme işleminin doğru bir şekilde yapıldığı kontrol edilmiştir. Geliştirilen test uygulaması ile hareket platformunun ilk olarak MS Flight Simulator uygulamasından bağımsız olarak uç noktalara gitmesi test edilmiştir. Bu testlerden başarıyla geçtikten sonra test modu devre dışı bırakılarak MS Flight Simulator programından alınan açısal değerler ile işlemler yapılmaya başlanmıştır.



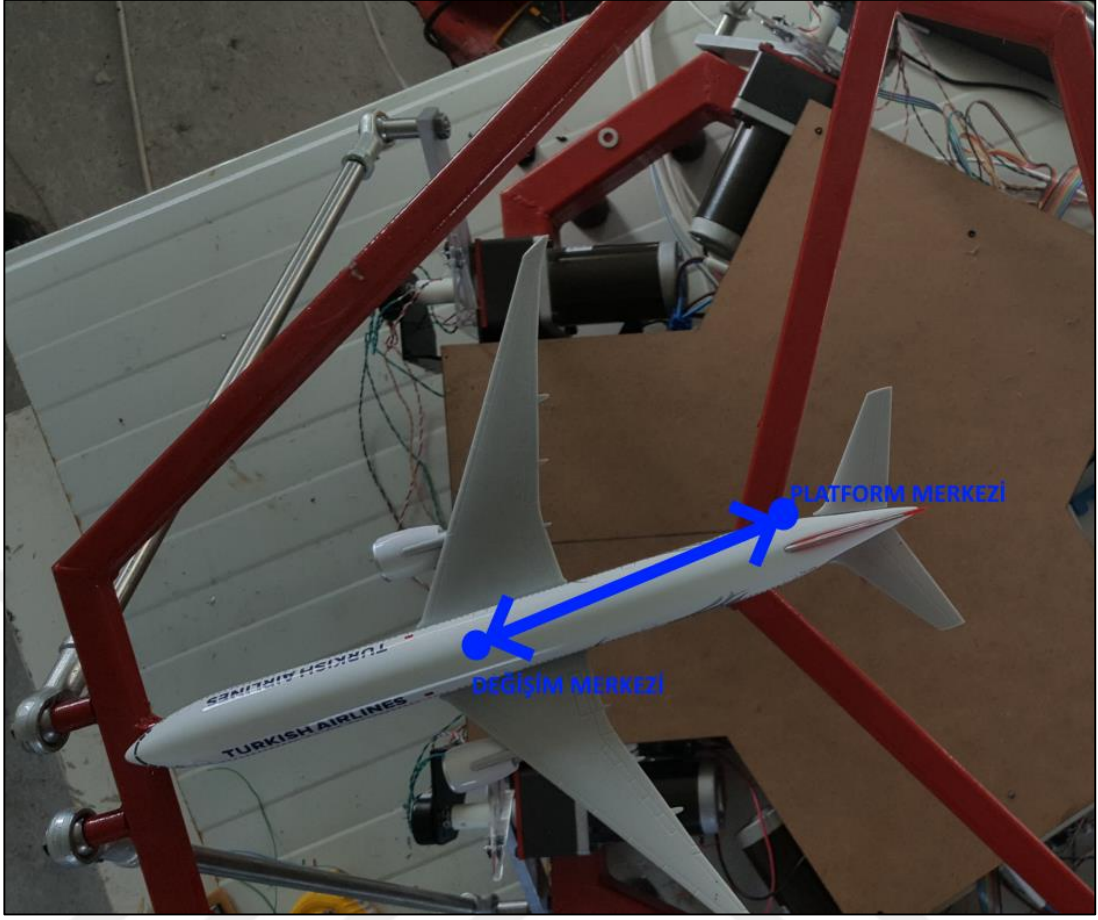
Şekil 6.2. Üretilen platformun yaw eksenini hareketi

Şekil 6.2’de görülen uçak sembolü, referans olarak alınan sıfır noktasından maksimum yaw hareketi yaptığında 15 derece olarak hareket etmektedir. MS Flight Simulator uygulamasında simüle edilen uçağın yaw hareketi olarak 15 derece yapması sağlanmış ve TheSim Interface programı aracılığı ile okunan bu değerler seri port üzerinden hedef yatış değerleri olarak elektronik kontrol kartına gönderilmiştir. Bu durumda 15 derece yaw açısı gönderilmiştir. Gönderilen bu hedef yatış değerleri kullanılarak elektronik kontrol kartı tarafından kinematik hesaplamalar yapılmış ve motorların dönüş açıları hesaplanmıştır. Elektronik kontrol kartı, motorlara bağlı olan sensörler ile motorların açısal konumlarını alarak dönüş yönlerine karar vermektedir. Ardından PID hesaplamalarını kullanarak motor hızı için PWM değerleri belirlenmiş ve bu hızlarla motorların istenen açılara dönmeleri sağlanmıştır. Sonuç olarak elektronik kontrol kartı, motorları hedef olarak hesaplanan açılara döndürmüştür.



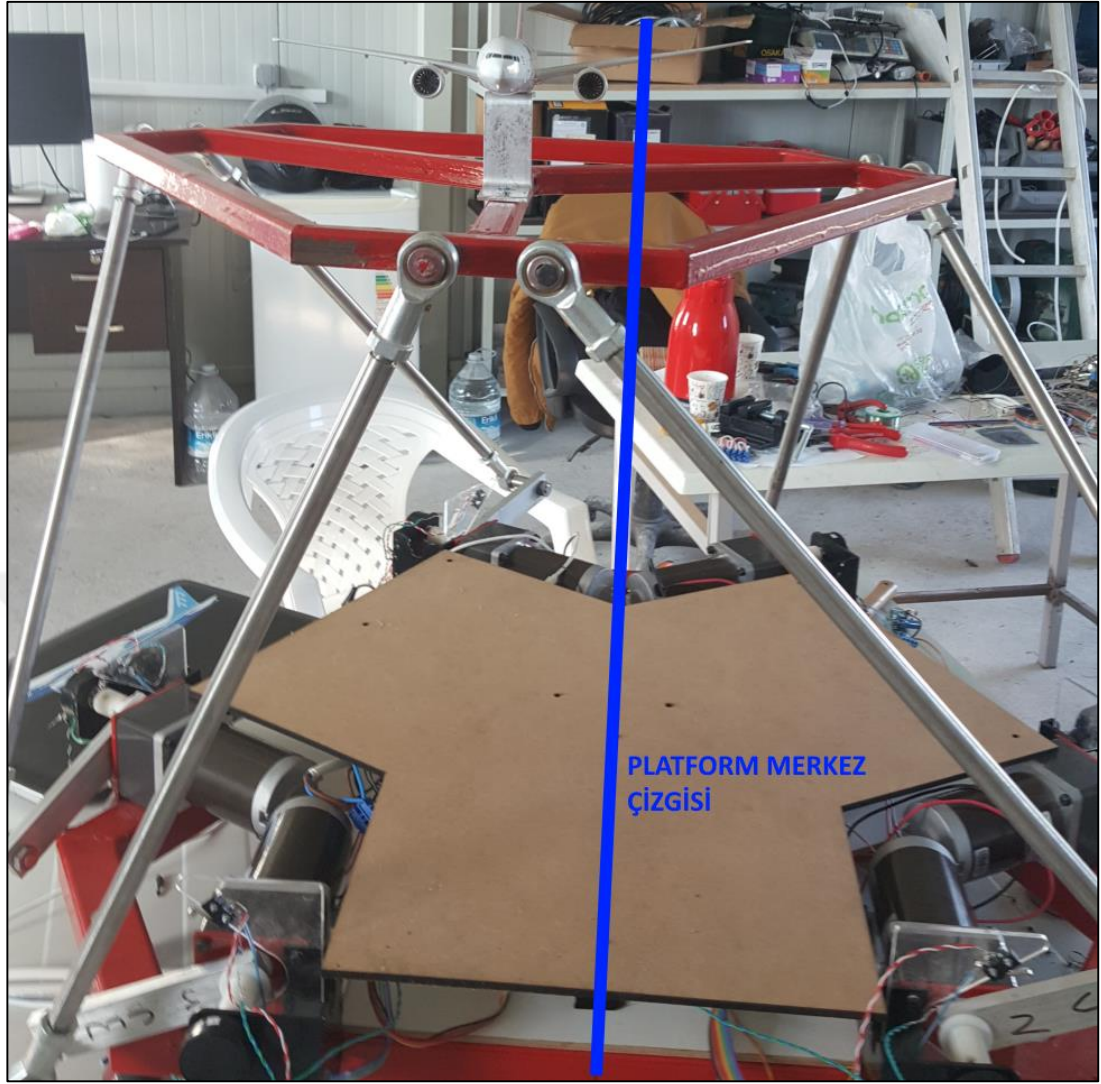
Şekil 6.3. Platform yaw hareketi önden görünüş

Şekil 6.3'te yaw hareketi yapan platformun önden görünüşü bulunmaktadır. Hareketin miktarını belirtmek için eklenen çizgi ile hareket merkezi arasında yaklaşık 15 derecelik açı bulunmaktadır. Hareketli platform motorları aracılığı ile istenen bu hedef açılara getirilmektedir. Burada yaw hareketi ile birlikte bir miktarda roll hareketi de yapılmıştır. Hareketler tekil olarak yapıldığında, hareketin dönüş açısına göre ilgili motorların dönüşleri hareket sadece o yöne hareket ettirecek şekilde olmaktadır. Şekil 6.3'deki harekette 2 ve 3 numaralı motorların hareketleri incelendiğinde, 2 numaralı motorun platformu kendi tarafına çekmeye çalışmaktadır. Aynı zamanda 3 numaralı motorunda platformu aynı tarafa doğru itme hareketi yaptığı görülmektedir. Diğer motor çiftleri de aynı hareketi, aynı yön için yaptığında platformun yaw hareketi tekil olarak sağlanmaktadır.



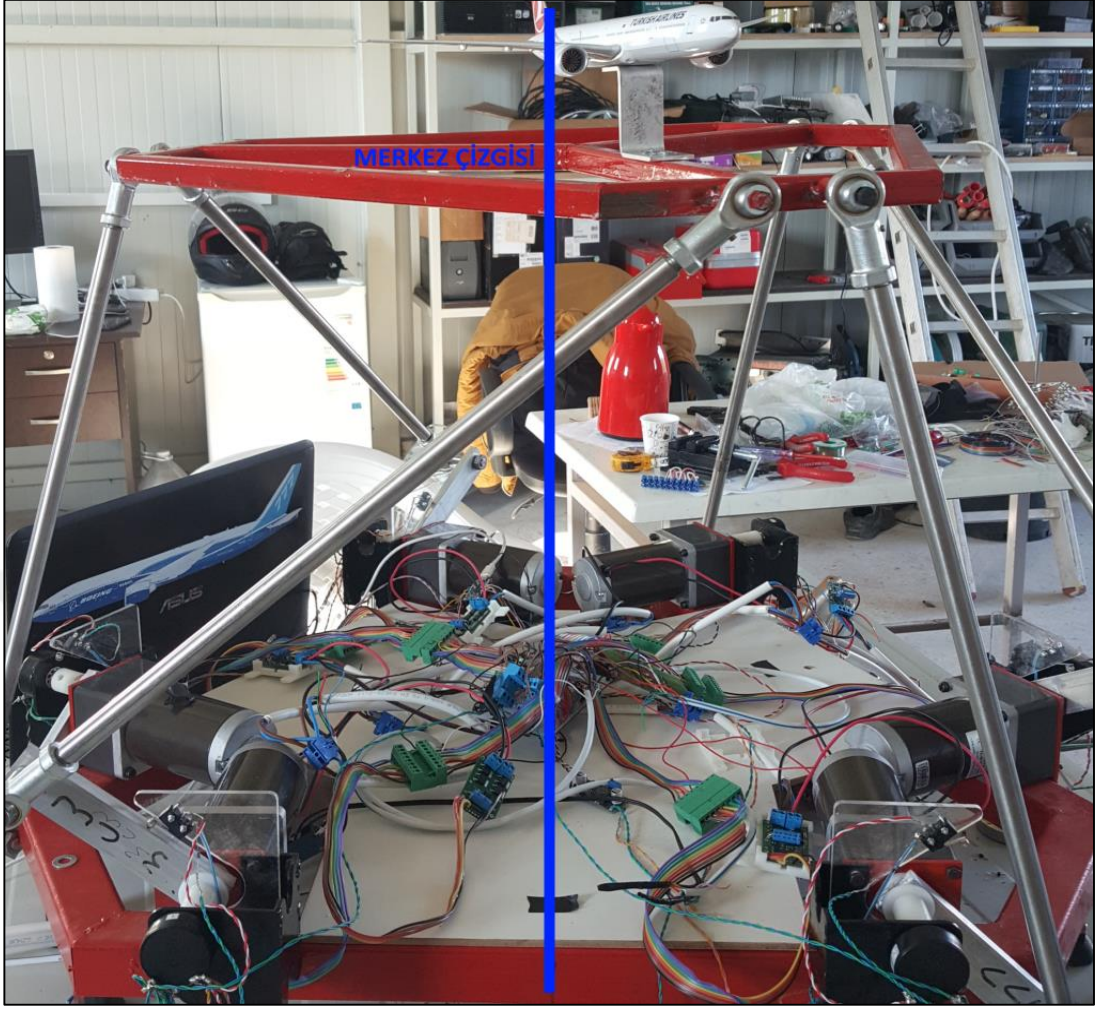
Şekil 6.4. Yaw ve öne kayma hareketi

Şekil 6.4'te görüldüğü üzere platform hem yaw hareketi hem de öne gitme hareketi yapmaktadır. Bu tarz iki hareketin birlikte yapıldığı durumlarda, her iki hareketin gerektirdiği motor açıları birbirlerine eklenerek yapıldığı görülmüştür. Yani yaw hareketindeki gibi motor çiftleri ilgili yönde platformu itme ve çekme hareketi yapmıştır. Bu hareket ile birlikte 1 ve 4 numaralı motorların birlikte çalışarak platformu, hareket simgesi olarak kullanılan uçak maketinin ön tarafına doğru ittiği görüldü. Bu iki hareketin karışımı olarak platformun, hem yaw hem de öne doğru gidiş hareketlerini birlikte yapabildiği görülmüştür. Kinematik hesaplamalarında görüldüğü üzere platformun 3 eksen etrafındaki dönüşünün birlikte hesaplanmıştır. Sonrasında hesaplama kısmında yapılan bu 3 eksen hesaplamalarına ek olarak platformun kayma hareketinin hesaplamaları da yapılmıştır. Mikroişlemci tarafından gerçekleştirilen bu hesaplamaların sonuçları birleştirilerek motorların hareketleri gerçekleştirilmiştir. Bu sayede platform her iki yöne doğru birlikte hareket ettirilmiştir.



Şekil 6.5. Platformun sağa kayma hareketi

Şekil 6.5'te görüldüğü üzere platform belli bir miktar sağ tarafa doğru kayma hareketi ile birlikte uçak simgesinin sol tarafına doğru bir miktar roll hareketi yapmıştır. Bu hareketin gerçekleştirilmesi esnasında 3,4 ve 5 numaralı motorların platformu kendilerine doğru çekme hareketi esnasında 0,1 ve 2 numaralı motorların platformu çekilen yöne doğru aynı oranda ittirdiği görüldü. Bu hareket ile birlikte verilmek istenen roll hareketi için 4 numaralı motorun çekiş açısı ile 1 numaralı motorun itiş açısının diğerlerine oranla daha az olduğu görüldü. Bu sayede hareketli platform kayma hareketi yaparken aynı anda 1 ve 4 numaralı motorların dönüş açıları miktarınca roll hareketi yapmıştır. Platform bu iki hareketi birlikte yapabilmesi için dönüş ve kayma eşitlikleri birlikte kullanılmıştır. Bu kullanım sayesinde 6DOF hareket sağlanmıştır.



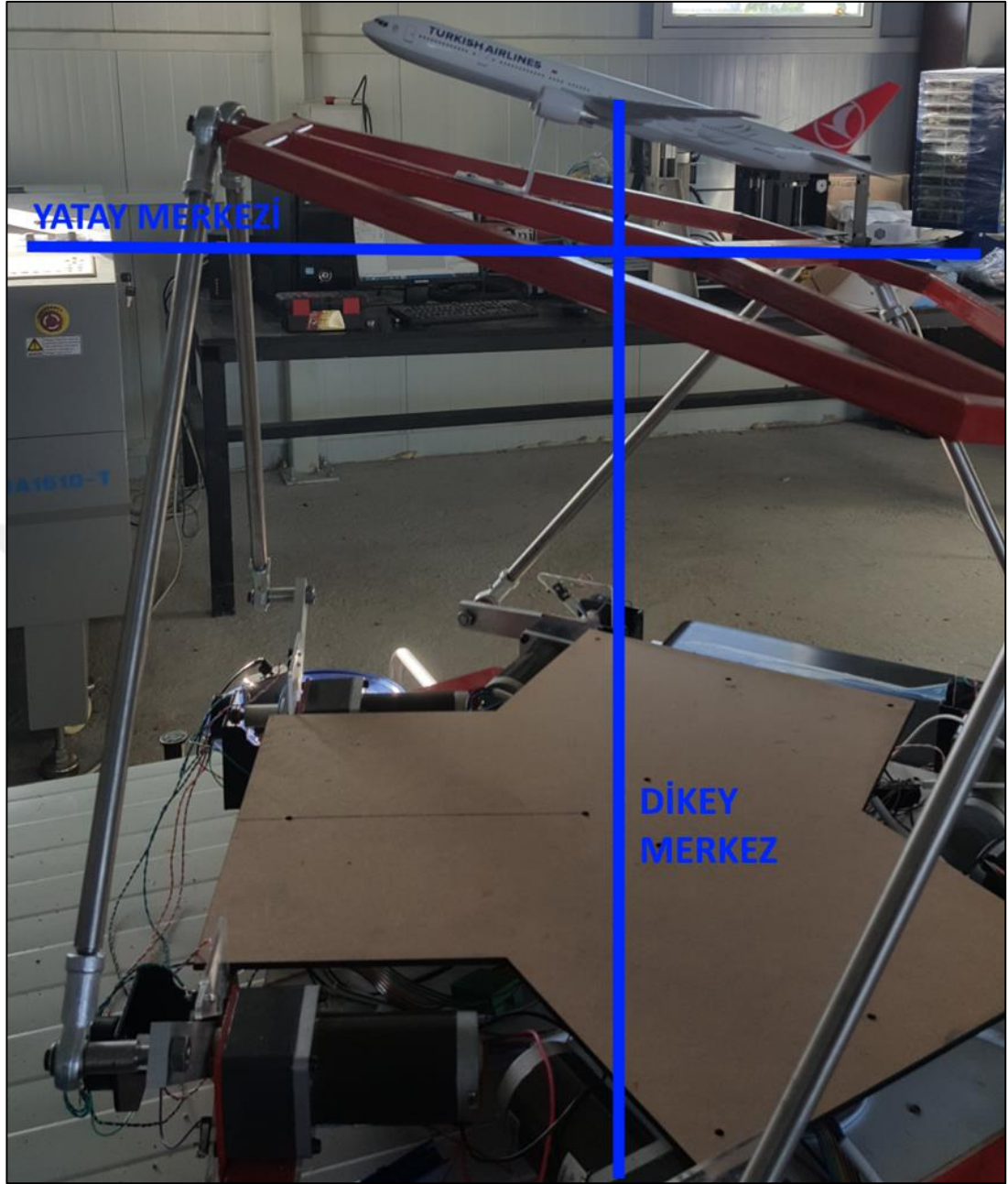
Şekil 6.6. Platformun yaw ve pitch hareketi

Şekil 6.6 üzerinde de görüldüğü üzere platform yaw hareketi yaparken aynı zamanda bir miktar burun yukarı hareketi yapmaktadır. Daha öncede bahsedildiği üzere yaw hareketinin tekil olarak gerçekleşmesi için motor çiftlerindeki motorların, birisi ilgili yöne çekme hareketi yaparken diğerinin de aynı yöne doğru itme hareketi yaptığı görülmektedir. Aynı zamanda motorların sensör bağlantıları ve uç limit algılama sensörleri incelendiğinde motorların son noktaya geldiğinde daha fazla dönmedikleri görülmüştür. Bu sayede motorların platforma veya kullanıcıya zarar verme durumlarının önüne geçilmiştir. Motorun hareket aktarım mekanizmasına bağlı olan kollar sayesinde, motorlardan bir tanesi limit değere geldiğinde bütün sistem çalışmayı durdurmaktadır. Şekil 6.6 üzerinde de yaw hareketinin sınırları denenmiş olup sonrasında, küçük bir roll hareketi ile sistemine tamamen durmayıp sadece limit değerlerin aşılması durumlarında sistemin durması ve limit içi olarak hesaplanan sonraki değerlerde çalışmasına devam ettiği görülmüştür.



Şekil 6.7. Platformun 3 eksenindeki hareketi

Şekil 6.7’de görüldüğü üzere platformun 3 eksen etrafında hareket etmesi sağlanmıştır. Görselde mavi renk ile gösterilen çizgiler ile platformun sıfır durumları belirtilmiştir. Uçak simgesi olarak kullanılan maketin yatış durumlarından anlaşılacağı üzere, aynı anda roll, pitch ve yaw hareketlerini birlikte yapabilmektedir. Bu 3 eksenindeki hareket 3DOF olarak adlandırılmaktadır. Eksenlerin üzerinde yapılan kayma hareketleri ile birlikte 6DOF olarak adlandırılan eksenlerde bu hareketin yapılması sağlanır. Bu çalışmanın asıl amacı olan 6DOF hareket, tasarlanan elektronik devre sayesinde kapalı çevrim olarak sağlanmıştır. Tasarlanan elektronik sistem ile hareketli platformun bütün parametreleri sürekli olarak kontrol edilerek, merkezi işlem birimi tarafından yapılan kinematik hesaplamalara uygun hareket ettiği gözlenmiştir.



Şekil 6.8. Platform burun yukarı hareketi

Şekil 6.8’de görüldüğü platformun yaptığı burun yukarı hareketi görülmektedir. 1 ve 4 numaralı motorların başlangıç durumuna göre herhangi bir dönüş hareketi yapmadığı görülmektedir. Bununla birlikte 2 ve 3 numaralı motorlar platformu bağlandığı noktalardan yukarı doğru kaldırırken, 0 ve 5 numaralı motorların platformu aşağı doğru indirdiği görülmektedir. Bu sayede bu iki motor çifti birlikte hareket ederek platforma burun yukarı hareketi yaptırılmıştır.



Şekil 6.9. Platformun burun aşağı ve sağa dönüş hareketi

Şekil 6.9’da görüldüğü üzere uçak simgesi bir miktar burun aşağı ve sağa yatış hareketi yapmaktadır. Buradaki hareket uç limitlere gelmeden ara değerlerde yapılan yatış hareketlerini göstermektedir. Sağa yatış hareketinin ana etmenleri, 0 ve 1 numaralı motorların platformu bağlandığı noktalardan kaldırmalarıdır. Bu hareketlere destek olarak 5 numaralı motorunda platformu bağlandığı noktadan aşağıya doğru çekecek şekilde hareket etmesi sağa yatışa destek olmuştur. Burun aşağı hareket için ise 1 ve 4 numaralı motorların platformun harekete göre arka konumdan kaldırarak, hareketin gerçekleşmesini sağlamışlardır. Elektronik devre tasarımında, motor sensörlerinden sürekli olarak veri alınmasına dikkat edilmiştir. Kapalı çevrim sistem kontrolündeki en önemli unsurun, gönderilen komuttan sonra motorun hedef açıya dönme durumunun sürekli olarak kontrol edilmesi olduğu anlaşılmıştır. Sensörlerden veri alma sıklığının artması ile birlikte dönüşün güvenilirliği ve etkinliği artırılmıştır. Buna bağlı olarak, kapalı sistem kontrol kalitesi de artırılmıştır.



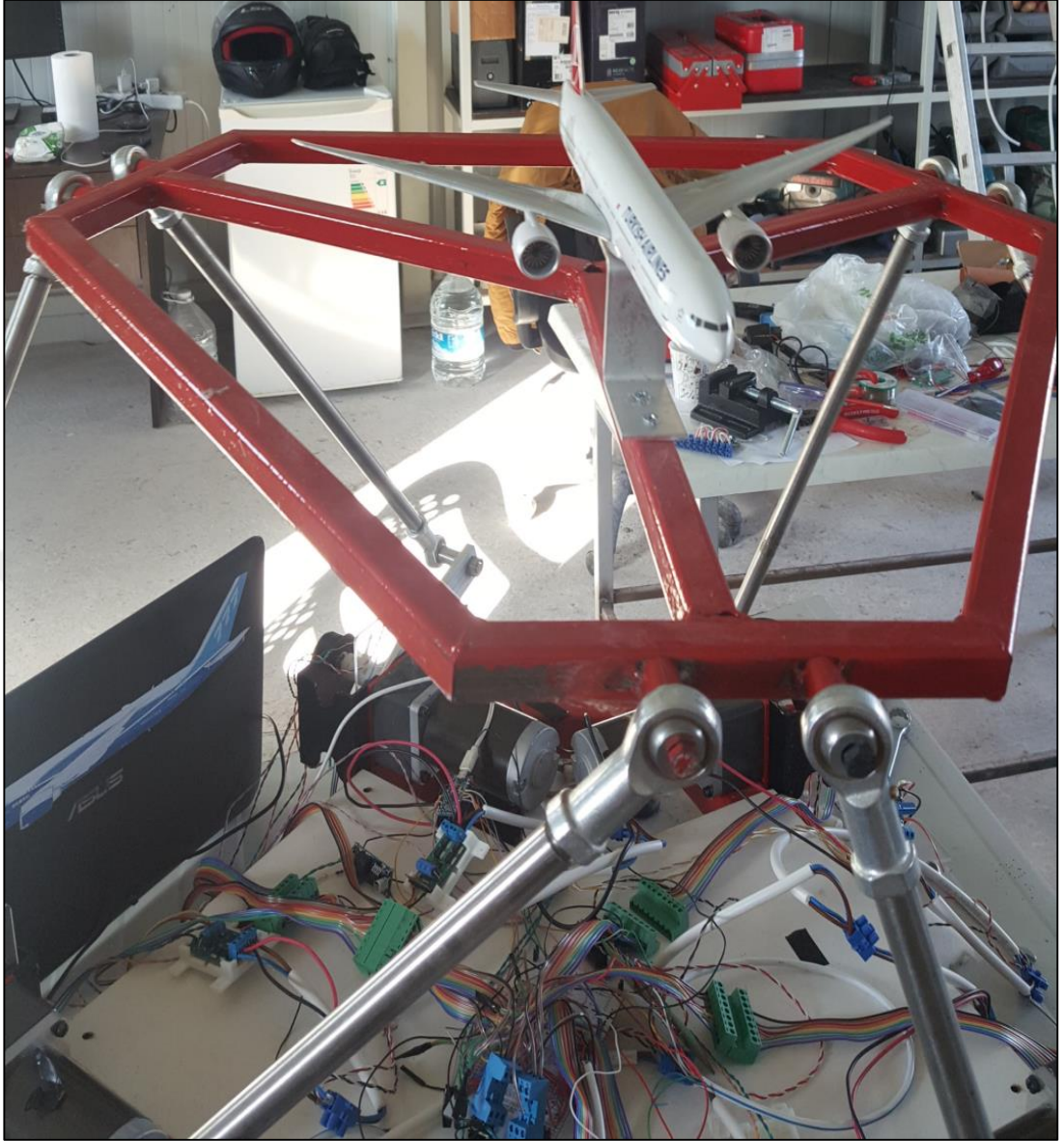
Şekil 6.10. Hareketli platform pitch-up hareketi

Şekil 6.10’da hareketli platformun burun yukarı hareketi yaptığı görülmektedir. MS Flight Simulator uygulamasında simüle edilen uçağın, belli bir miktar burun yukarı hareketini yapması sağlanmış ve bu çalışma kapsamında geliştirilen TheSim Interface programı tarafından seri port aracılığı ile gönderilen hedef yatış değerleri kullanılarak yapılan hesaplamalar sonucu platform burun yukarı hareketini yapmıştır. Platformun referans noktasından burun yukarı yapabilmesi için gereken hedef yatış değerleri TheSim Interface programından alındıktan sonra gerekli kinematik hesaplamalar elektronik kontrol kartı tarafından yapılmıştır. Hesaplanan bu açısız değerler ile motorların anlık açısız değerleri karşılaştırılarak, motorların hesaplanan değerlere getirilmesi sağlanmıştır. Bu sayede platformun burun yukarı hareketi yapması sağlanmıştır. Motor sürücülerin PWM girişlerine uygulanan sinyal ile motorların dönüş hızları belirlenmiştir. PWM değerleri PID yöntemi kullanılarak belirlenmiştir.



Şekil 6.11. Hareketli platformun roll hareketi

Şekil 6.11’de görüldüğü üzere hareketli platform belli bir miktar roll hareketi yapmıştır. MS Flight simülatör uygulaması üzerinde simüle edilen uçağın anlık durumu alınıp TheSim Interface programı aracılığı ile seri port üzerinden elektronik kontrol kartına gönderilmiştir. Burada yapılan kinematik hesaplamalardan sonra motorların dönüş yönleri belirlenmiştir. Ardından motorların hedefleri ile anlık konumları arasındaki farklara göre PID hesaplama yöntemi kullanılarak PWM değerleri belirlenmiştir. Belirlenen bu PWM değerleri motor sürücülerinin PWM girişlerine uygulanıp, belirlenen yön bilgisi dijital sinyal şeklinde sürücünün yön girişine uygulanmıştır. Motorların konumları sürekli olarak kontrol edildiği için PID ile hesaplanan PWM değeri de aradaki farka bağlı olarak sürekli değişecektir. PID hesaplamaları sayesinde motorların anlık açısı hedef değere yaklaştıkça yavaşlayacak olup hedef değere ulaşınca anlık olarak hedef değerden sapma durumlarının meydana gelmesi önlenmiş olacak ve ileri-geri titremesi engellenecektir.



Şekil 6.12. Platform burun aşağı ve roll hareketi

Şekil 6.12’de görüldüğü üzere hareketli platform hem roll hareketini hem de burun aşağı hareketini birlikte yapabilme kapasitesine sahiptir. Platformun 6 eksen olarak tasarlanmasının önemi burada oraya çıkmaktadır. Gerçek uçuşlarda uçaklar yatış, dalış ve sapma hareketlerini ve bu hareketlerin eksenleri üzerindeki kayma hareketlerini birlikte yapabilmektedir. Bu tasarımda da görüldüğü üzere platform her iki hareketi birlikte yapabilmektedir. Matematiksel modellemede kullanılan Tait-Bryan hareket matrisi ile bu özellik sağlanmış olup motor konumlarından platform bağlantı noktalarına olan vektörler hesaplanmıştır. Bu vektörleri sağlayabilmek için motorların gelmesi gereken dönüş açılarını hesaplayabilmek için ek olarak ilgili bölümlerde anlatılan trigonometrik teoremlerden faydalanılmıştır.



Şekil 6.13. Platformun roll ve yaw hareketi

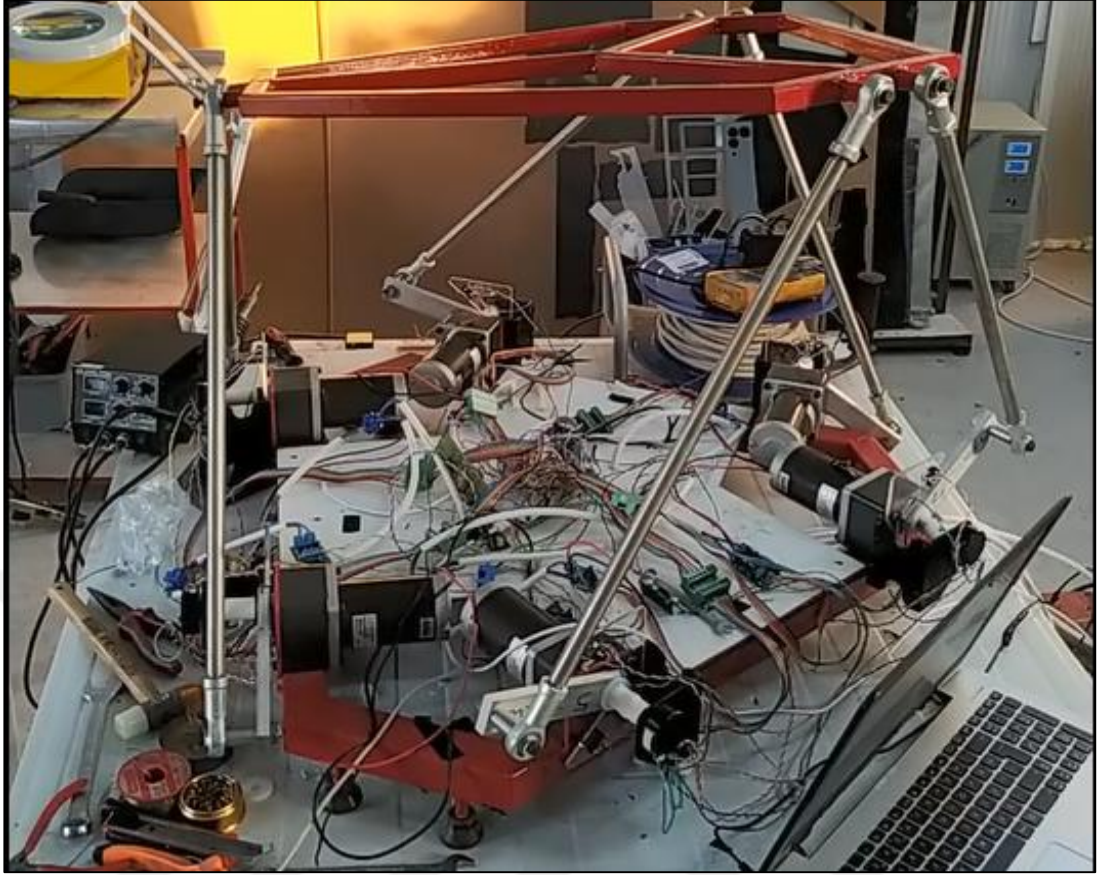
Şekil 6.13’de görüldüğü üzere hareketli platform hem yaw hem de roll hareketi birlikte yapabilmektedir. MS Flight simülör uygulamasında simüle edilen uçağın roll ve yaw hareketlerini birlikte yapması sağlanmıştır. TheSim Interface uygulaması MS Flight simülör programını sürekli olarak takip etmektedir. Bundan dolayı oluşan herhangi bir yatış değerini direk olarak seri port üzerinden elektronik kontrol kartına aktarmaktadır. Aynı zamanda program ara yüzünde açısal değerlerin gönderimi deaktif etmek için bulunan seçim butonu ile platformun hareketleri duraklatılabilmektedir. Bu durum programın sürekli olarak eş zamanlı çalışabilmesinin yanı sıra herhangi bir olumsuz durum olduğunda hareketli platformun duraklatılabilmesini sağlamaktadır. Açısal değerlerin gönderimi tekrar aktifleştirildiğinde elektronik kontrol kartı tekrardan çalışmaya başlayacak ve platformun hareket etmesini sağlayacaktır. TheSim Interface programı ileriye dönük bu tarz geliştirmelere açık bir şekilde geliştirilmiştir.



Şekil 6.14. Platformun yatış, dalış ve sapma hareketlerini birlikte yapması

Hareketli platformu kalibre edebilmek adına bazı testler yapılmıştır. Bu testler, bütün yatış açısı değerlerinin limit testi ve motor limit değeri testleridir. Bu testler aracılığı ile sistemin çalışma doğruluğu aralıklı olarak kontrol edilebilecektir.

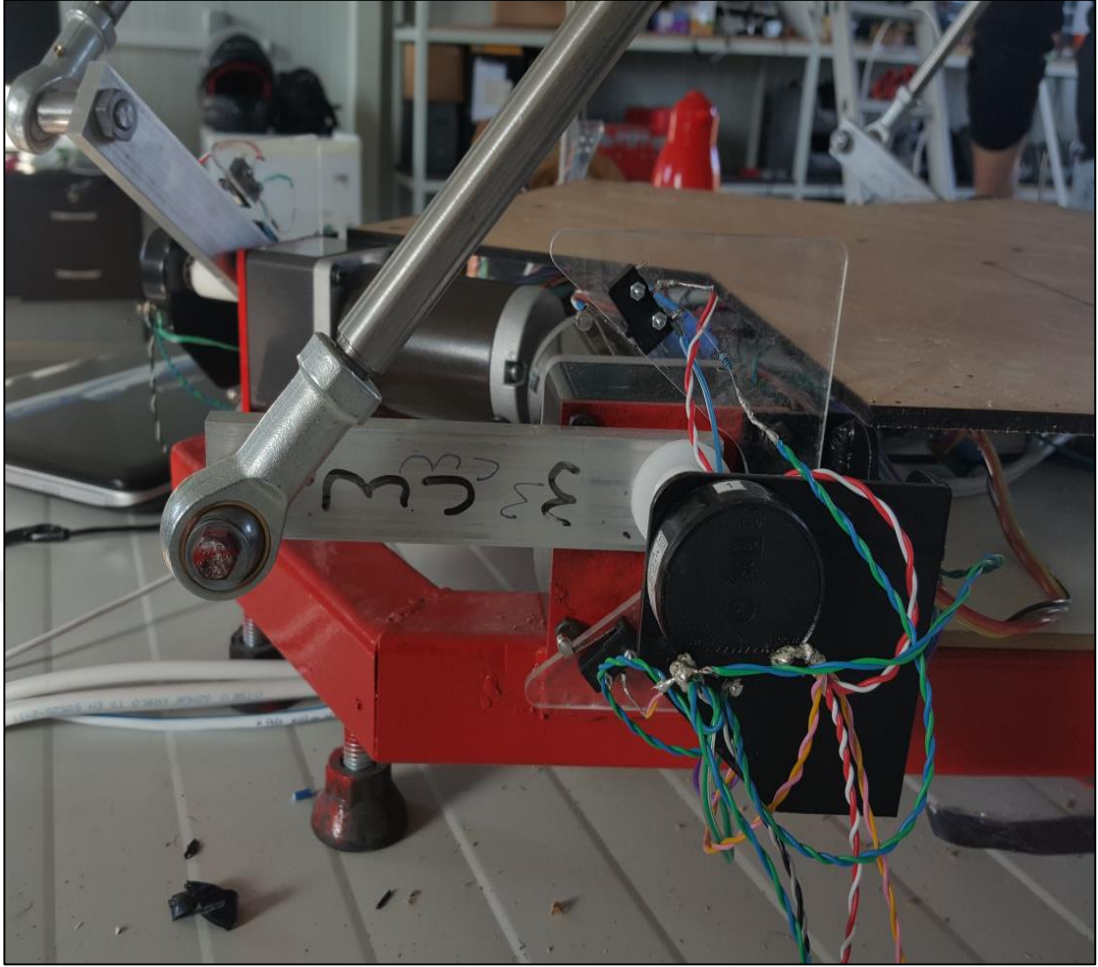
Yatış açılarının testi direk olarak elektronik kontrol kartı üzerinden yapılmaktadır. Bu test esnasında, kontrol kartının kendi özelliklerinden birisi olan iç kesme üretici kullanılmıştır. Bu kesme üretici zamanlanarak elektronik kontrol kartına seri port üzerinden yatış değerleri gönderiliyormuş gibi test simülasyonu yapmaktadır. Kesme üretici tarafından iç içe yazılan test döngüleri sayesinde, kontrol kartının seri port kontrolünü devre dışı bırakarak ana program döngüsüne girmesini sağlamaktadır. Ardından hedef yatış değerlerinin tutulduğu program değişkenlerini seri porttan gönderiliyormuş gibi değiştirerek, platformun hareketini sağlamaktadır. Bu değişkenleri sürekli olarak uç noktalara götürmektedir ve sistem hareketinin doğruluğundan emin olunmasını sağlamaktadır.



Şekil 6.15. Platform yatış açısı değerlerinin limit testi

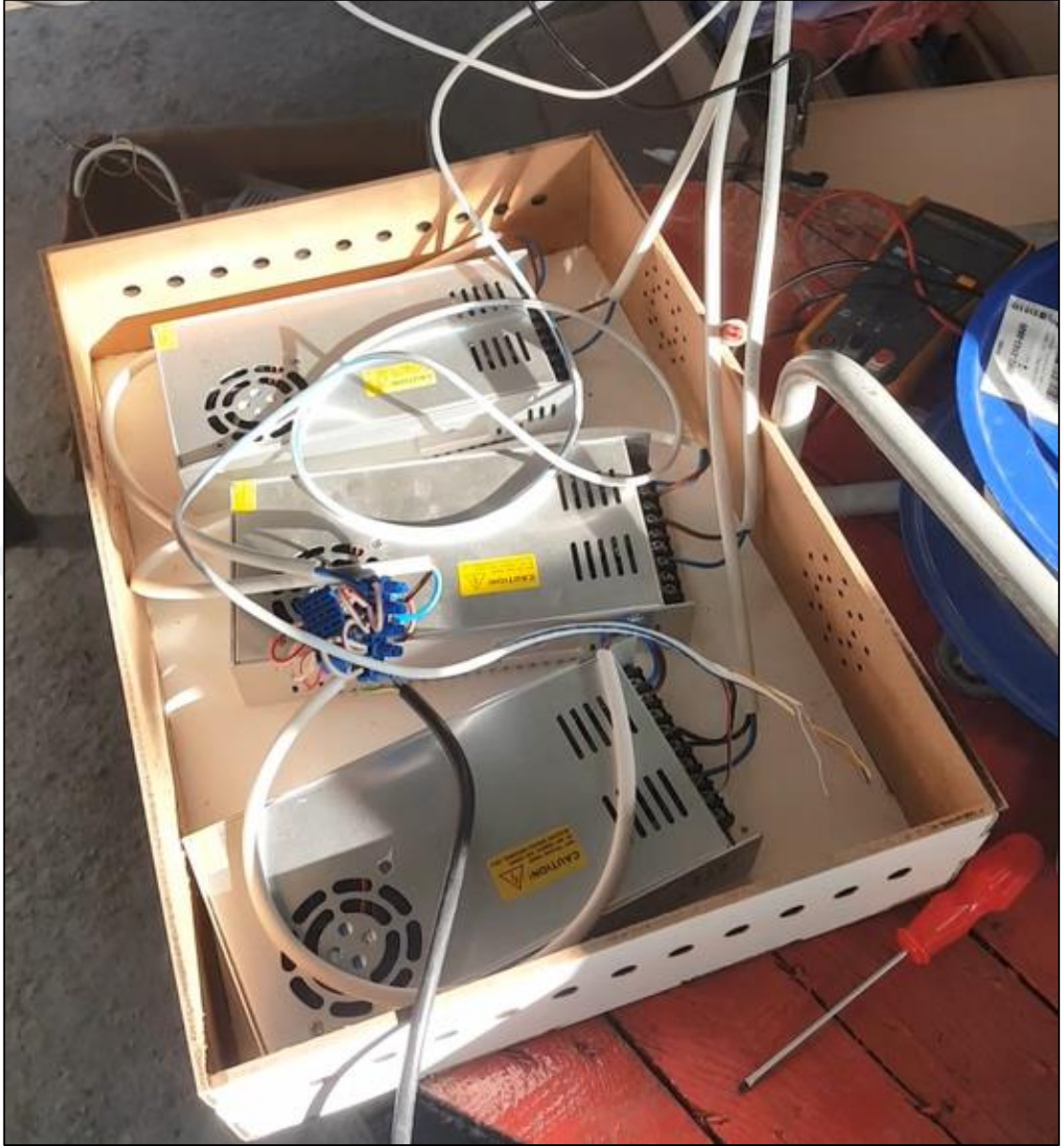
Motor limit değeri testi ile motorların limitleri test edilmektedir. Motorların hareket limitlerini aşarak sisteme zarar vermesi engellenmesi amacıyla, motor hareket limitlerine, limit sensörleri yerleştirilmiştir. Motorların açıları belirlenen bu uç noktalara geldiğinde, sistem çalışmayı durduracak şekilde çalıştırılmaktadır. Buradaki sensörün bilgisi elektronik kontrol kartına gönderilmektedir. Şekil-7.16'da bir motor ve bu motora ait limit sensörleri görülmektedir. Motorlar ile birlikte dönen kol bağlantısı, bu sensörlere geldiğinde karta giden bilgi sayesinde sistem limit bölgeye geldiğini anlayıp, motorların hareketini durdurmaktadır. Sonraki gelen komutların tekrar limit içi olduğunda hesaplanan yerlere gitmektedir.

Bazı durumlarda bağlantılara gelen küçük bir gürültüden kaynaklı sensörler aktifleşip, ilgili motor uç noktaya gelmemiş olsa bile aktifleştikten dolayı limit değere erişildi bilgisi gönderilebilmektedir. Bu olumsuz durumu engellemek adına elektronik olarak filtreleme işlemi yapılmaktadır. Bu sayede gelen gürültü sinyalinin filtrelenecek şekilde sistem topraklama hattına aktarılması sağlanmış ve sensörlerin gürültüye karşı korunması sağlanmıştır.



Şekil 6.16. Motor limit değeri testi

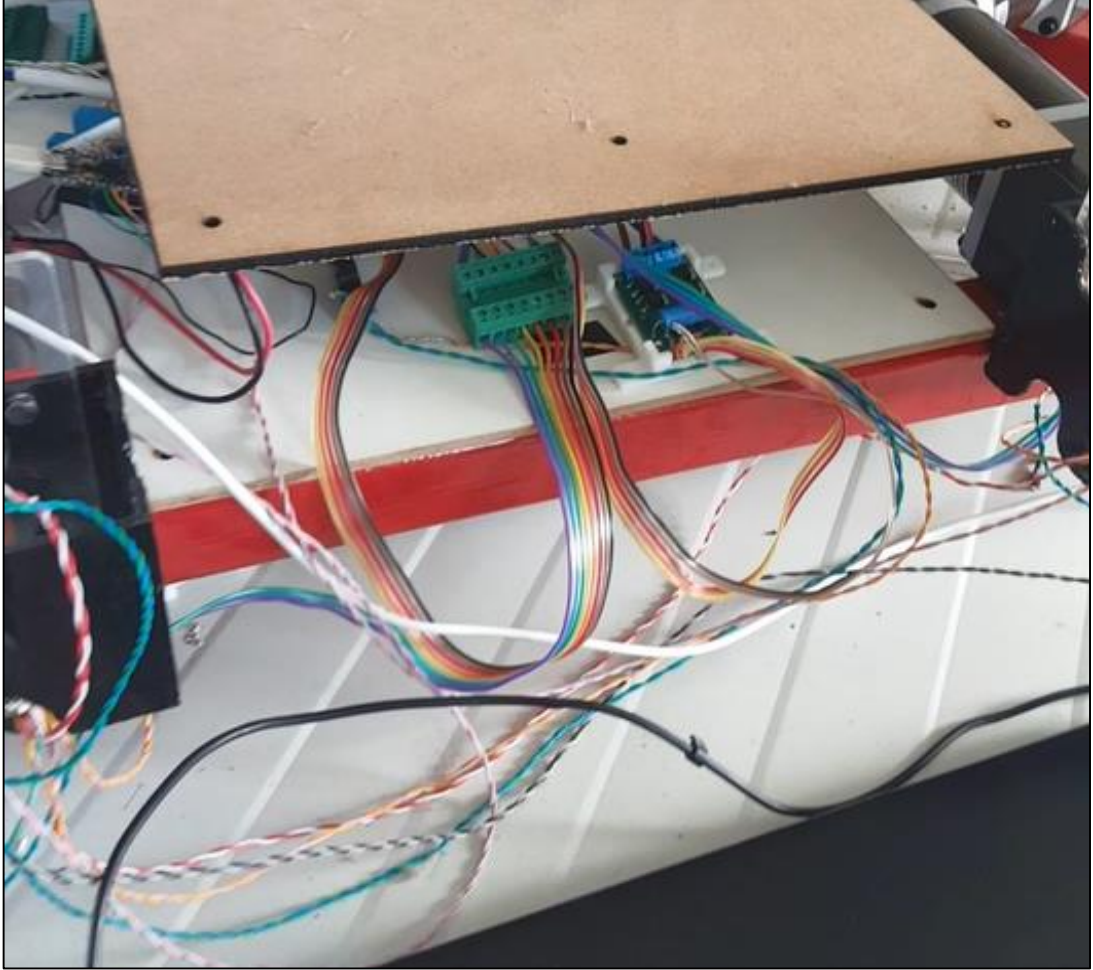
Sistemde kullanılan bütün elektronik elemanlar DC elektrik enerjisi ile beslenmektedir. Dolayısıyla doğrudan alınan şebeke elektriği sistemde güç kaynağı olarak kullanılamamaktadır. Sistemin ana beslemesi olarak kullanılan dönüştürücüler ile şebekeden alınan elektrik enerjisi 24VDC enerjiye dönüştürülmektedir. Motor sürücüleri direk olarak bu enerji hattından beslenmektedir. Dönüştürücülerden fazla akım çekme durumlarını engellemek adına her iki motora bir dönüştürücü olacak şekilde elektrik beslemeleri paylaştırılmıştır. Şekil 6.17’de sistemin elektrik enerjisi beslemesi için kullanılan güç kaynaklarının bağlantısı gösterilmektedir. Ayrıca elektronik kontrol sistemi bu dönüştürücüler aracılığı ile beslenmektedir. Elektronik kontrol sistemi beslemesi için gereken enerjilerde, dönüştürücü hattına bağlanan gerilim düşürücü elemanlar yardımıyla beslenmektedir. Elektronik kontrol kartı için gereken enerji 12VDC ile beraber diğer analog ve dijital sinyal gönderen sensörler aynı hat üzerinden beslenmektedir.



Şekil 6.17. Güç kaynakları bağlantısı

Sistem tasarımında aynı tip elemanın birden fazla yerde kullanımı bulunmaktadır. Örnek olarak, sistemde 6 adet motor, sürücüler, sensörler bulunmaktadır. Bununla beraber 6 tane motor yukarı limit sensörü ile birlikte 6 tane motor aşağı limit sensörü bulunmaktadır. Sistemdeki bağlantıların karmaşıklığını engellemek adına, her bir işlevi yerine getiren bağlantı için aynı etiketli kablolar kullanılmalıdır. Bu çalışmadaki kablolama sistemi oluşturulurken, renkli kablo seti kullanıldı. Aynı zamanda enerji taşıyan kablolar için ikili sarmal kablo yapısı kullanıldı. Şekil 6.18'deki görselde bir alt sisteme ait kablolama görülmektedir. Buradaki kablolar bir geçmeli klemens aracılığı ile elektronik kontrol sistemine bağlanmaktadır.

Bağlantının sistem tarafında ise kablolar sensörlere ve sürücülere dağılmaktadır. Bu bağlantı şeması bütün sistemlerde aynıdır. Bu tasarım sayesinde montaj hızlandı ve karmaşıklık azaltıldı. Aynı zamanda motorlar, elektronik kartlara gelen geçmeli klemensler sayesinde ayrılabilir. Bu şekilde tasarlanan modüler yapı sayesinde kontrol kartında veya motorlarda bir arıza olduğunda birbirlerinden kolayca ayrılıp üzerinde daha rahat işlem yapılabilmesine olanak tanımaktadır.



Şekil 6.18. Alt sistem kablolama örneği

Profesyonel olarak tasarlanan pek çok sistemde görülen standart kablolama prensibi sayesinde tasarımlar modüler olarak rahat bir şekilde birkaç koldan hızlı ve güvenilir bir şekilde devam edebilmektedir. Aynı zamanda alt sistemlerde bir arıza meydana geldiğinde onarım için rahat bir şekilde diğer sistemlerden izole ederek işlem yapmak mümkün hale gelmektedir. Bu tasarımda bu şekilde bir yöntemin kullanılmasının en büyük sebeplerinden biriside modüler yapıyı sağlayabilmektir.

7. SONUÇLAR VE ÖNERİLER

Hareket platformu genel fonksiyonlarını yerine getirerek beklendiği gibi çalışmaktadır. Bu sayede matematiksel modellemenin doğru bir şekilde oluşturulduğu ve doğru bir şekilde gerçek mekanizmaya dönüştürüldüğü görülmüştür. Programa aktarılan matematiksel modelin ve kinematik hesaplamaların doğru bir şekilde çalıştığı ve beklenen sonuçları oluşturduğu gözlemlenmiştir. Bu sonuçlar ile birlikte tez çalışmasının büyük ölçüde başarılı olduğu söylenebilmektedir. Aşağıda bu tez çalışması ile ilgili öneriler bulunmaktadır. Bu öneriler ile üretimi yapılan hareket platformu sonraki üretimlerde daha kararlı bir şekilde üretililebilecektir.

Motorların konumlarını algılamak için kullanılan sensörler direk olarak motorların kuvvet aktarım millerine sabitlenmiştir. Kısa devreye karşı gerekli önlemler alınarak direk olarak elektronik kontrol sistemine bağlanmıştır. Açısal değerleri analog olarak ilgili portlardan elektronik kontrol kartına analog olarak aktarılması sağlanmıştır. Burada kullanılan analog veri kaynağı motor sürücüsünden bağımsız olduğu için açısal değer okuma ve hesaplama konusunda eş zamanlı çalışamama durumları meydana gelmektedir. Bu yüzden sonraki çalışmalarda direk olarak, manipülatör olarak kullanılan DC motorların açısal değerleri için bir RVDT sensörü tasarlanarak daha hassas veriler okunmalıdır. Ayrıca bir motor sürücü sensörü tasarlanıp, içerisine entegre edilecek olan encoder ile konum algılama işlemi yapılmalıdır. Aynı zamanda bu RVDT sensöründe bu motor sürücüsüne bağlanarak, motor konum kontrolünü ana işlemciden alarak yardımcı sistemlere aktarmak sistem çalışmasını daha stabil hale getirilecektir.

Elektronik kontrol sistemindeki sistemler birlikte çalışmaktadır. Dolayısıyla bu sistemlerin senkron çalışması için elektronik ground noktalarının doğru bir şekilde belirlenmesi gerekmektedir. Elektronik kontrol sistemi içerisindeki yan sistemlerin çalışma esnasında elektronik olarak izole edilmesi sistemi daha güvenli hale getirilecektir. Elektronik olarak izolasyon işlemi için elektronik devrelerde

optocoupler kullanılması gerekmektedir. Bu yüzden tercih edilen haberleşme protokolunun, optocoupler elemanları ile doğru bir şekilde çalışabilir olması göz önünde bulundurulmalıdır.

Malzeme seçimi konusunda seçilen malzemelerin ağırlıkları göz önünde bulundurulmalıdır. Üretim için seçilen malzemelerin daha hafif maddelerden seçilmesi sistemin daha rahat çalışmasına olanak tanıyacaktır. Ağır malzemelerin seçilmesi durumunda motorların aşağı yöndeki hareketinde daha fazla baskı uygulayıp uç sınırlardaki limitleme elemanlarına vurma gerçekleşebilir. Bununla birlikte yukarı yöndeki hareketlerde de motorlar fazla güç harcayıp, ağırlıklardan kaynaklı hareketler yavaşlık meydana gelebilir. Bu durumda PID kullanılarak ayarlanan PWM değerleri tam olarak motor hareketlerinde görülemeyebilir.

Motorlara, sürücüler aracılığı ile giden enerjiyi acil durumlarda kesebilmek adına sisteme bir acil durum butonu ekleyerek ana enerji kaynağından motorlara giden elektrik enerjisi, yüksek gerilim altında çalışabilen bir mosfet veya röle ile aracılığı ile gönderilmelidir. Bu sayede hem program hazırlanmadan, motorlara giden enerji kesilmiş olur ve normal olmayan hareket engellenmiş olunur hem de acil durumda direk olarak enerji kesilerek sistemin zarar görmesi engellenebilir.

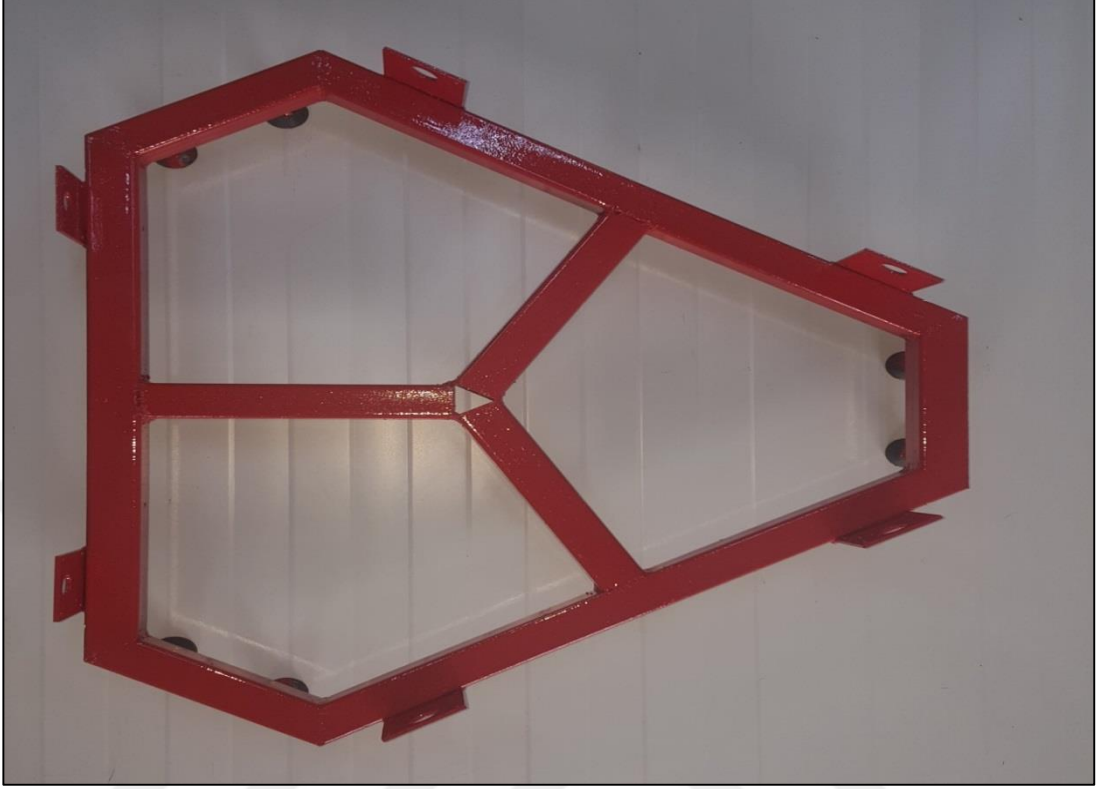
KAYNAKLAR

- [1] Volkaner B., Sozen S. N. and Omurlu V. E., Realization of a Desktop Flight Simulation System for Motion-cueing Studies, *International Journal of Advanced Robotic Systems*, 2015, DOI: 10.5772/63239
- [2] Yadav. S. K., DC Motor Position Control Using Fuzzy Proportional-Derivative Controllers With Different Defuzzification Methods, *IOSR Journal of Electrical and Electronic Engineering*, 2015, **10**(1), 37-47
- [3] Lerman K., Galstyan A., A General Methodology for Mathematical Analysis of Multi-Agent Systems, *Information Sciences Institute Univ. Of Southern California Marina del Rey*, 2001, CA-90292-6695, Technical Report ISI-TR-529
- [4] D. Stewart, A platform with six degrees of freedom, *Proceedings of the Institution of Mechanical Engineers*, 1965, **180**(1), 371-386, DOI:10.1243/PIME_PROC_1965_180_029_02.
- [5] https://people.maths.bris.ac.uk/~madjl/course_text.pdf (Ziyaret Tarihi: 19.08.2020)
- [6] Huang G., Lee S., PC-based PID speed control in DC motor, *International Conference on Audio, Language and Image Processing*, 2008, DOI: 0.1109/ICALIP.2008. 4590052,
- [7] KOEKEBAKKER S. H., Model based control of a flight simulator motion system, Doctoral Thesis, Delft University of Technology, Delft/Netherlands, 2001,
- [8] <https://calisphere.org/item/33c58cbde175ca0b35770e561d983cad/> (Ziyaret Tarihi: 20.08.2020)
- [9] <https://calisphere.org/item/fe0202f9f60ef905d96c9e718a7d9f51/> (Ziyaret Tarihi: 20.08.2020)
- [10] <http://www.analogmuseum.org/english/examples/lfsim/> (Ziyaret Tarihi: 25.08.2020)
- [11] <https://history.nasa.gov/SP-4302/ch3.8.htm> (Ziyaret Tarihi: 20.09.2020)
- [12] Sinha, R., Paredis, C. J. J., Liang, V., and Khosla, P. K., Modeling and Simulation Methods for Design of Engineering Systems, *J. Comput. Inf. Sci. Eng.*, 2000, **1**(1), DOI:10.1115/1.1344877

- [13] Caro P. W., Aircraft Simulators and Pilot Training, *Human Factors*, 1973, **15**(6), 502-509, DOI:10.1177/001872087301500602
- [14] Miyoshi T., Nakayasu H., Ueno Y., Patterson P., An emergency aircraft evacuation simulation considering passenger emotions, *Computers and Industrial Engineering*, 2012, **62**(3), 746-754,
- [15] Orlansky J., String J., Chatelier P. R., The Cost-Effectiveness of Military Training, *Institute For Defense Analyses Alexandria VA*, ADP000168, 1982
- [16] https://www.airteamimages.com/boeing-777__cathay-pacific-airways_62043.html (Ziyaret Tarihi: 01.12.2020)
- [17] http://web.physics.ucsb.edu/~fratus/phys103/Disc/disc_notes_3_pdf.pdf (Ziyaret Tarihi: 21.09.2020)
- [18] Filiposka M. Z., Djuricb A. M., ElMaraghya W., Complexity analysis for calculating the Jacobian matrix of 6DOF reconfigurable machines, *Procedia CIRP*, 2014, **17**, 218-223, DOI:10.1016/j.projir.2014.02.051
- [19] Baranowski L., Equations of Motion of a Spin Stabilized Projectile for Flight Stability Testing, *Journal of Theoretical and Applied Mechanics*, 2013, **51**(1), 235-246,
- [20] Qazani M. R. C., Pedrammehr S., Rahmani A., Danaei B., Ettefagh M. M., Rajab A. K. S. and Abdi H., Kinematic analysis and workspace determination of hexarot-a novel 6-DOF parallel manipulator with a rotation-symmetric arm system, 2014, *Robotica*, **33**(8), 1686-1703, DOI:10.1017/S0263574714000988
- [21] <https://mememememememe.me/post/stewart-platform-math/> (Ziyaret Tarihi: 11 Ağustos 2020)
- [22] Teia L., Geometry of the 3D Pythagoras' Theorem, *Journal of Mathematics Research*, 2016, **8**(8), 78, DOI: 10.5539/jmr.v8n6p78
- [23] <http://thetrig.blogspot.com/2011/11/m-sin-x-n-cos-x.html> (Ziyaret Tarihi: 11 Ağustos 2020)
- [24] J. Holtz, Pulsewidth modulation for electronic power conversion, *Proceedings of the IEEE*, 1994, **82**(8), 1194-1214, DOI: 10.1109/5.301684
- [25] El-Nagar A. M., El-Bardini M., Practical Implementation for the interval type-2 fuzzy PID controller using a low cost microcontroller, *Ain Shams Engineering Journal*, 2014, 475-487,
- [26] Kenneth W., GoffJerry K., Fenstermacher G. J., Closed loop control of PWM duty cycle, 1999, U.S. Patent No : US6307336B1, *Parker Intangibles LLC*
- [27] H. Chen, An H-bridge driver using gate bias for DC motor control, *IEEE International Symposium on Consumer Electronics (ISCE)*, 2013, pp. 265-266, DOI: 10.1109/ISCE.2013.6570219.



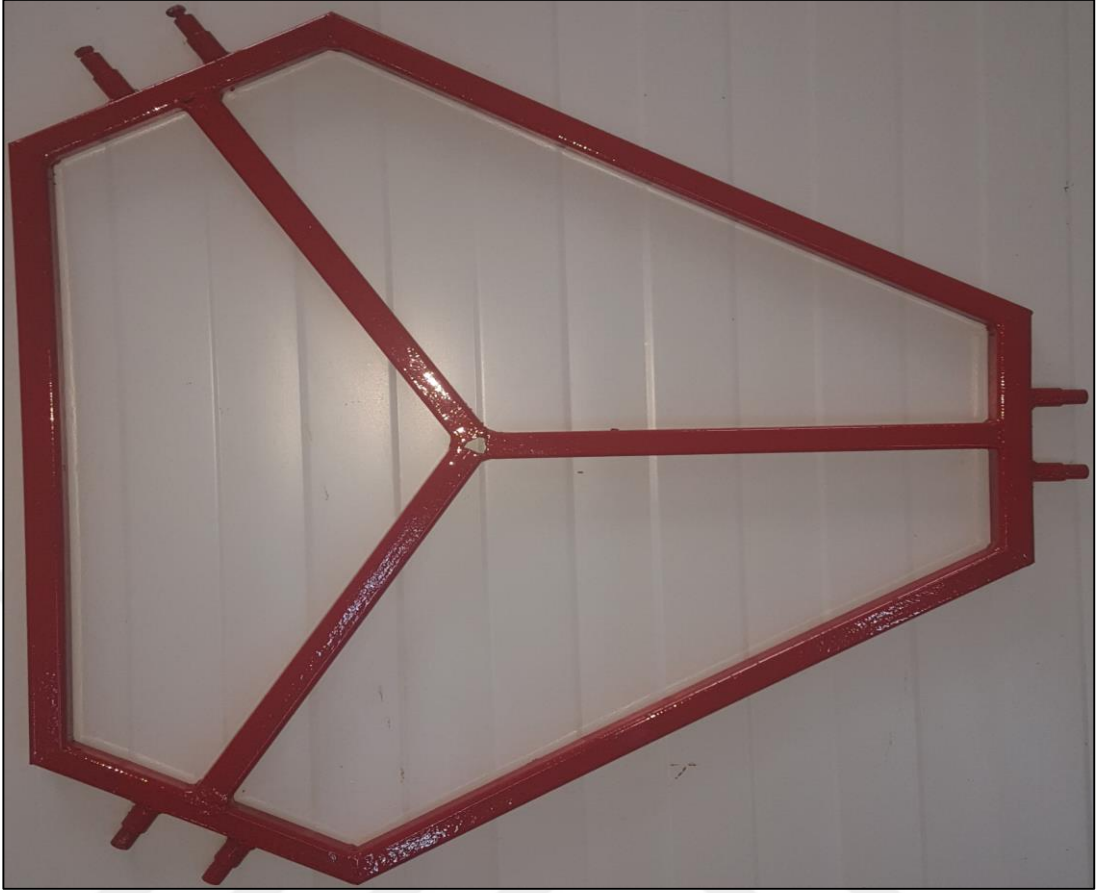
EK-A



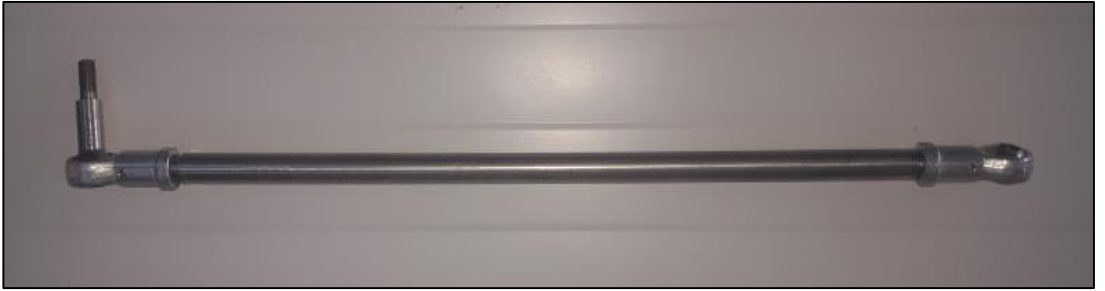
Şekil A.1. Üretilen Alt Tabla



Şekil A.2. Üretilen alt tablanın yandan görünüşü



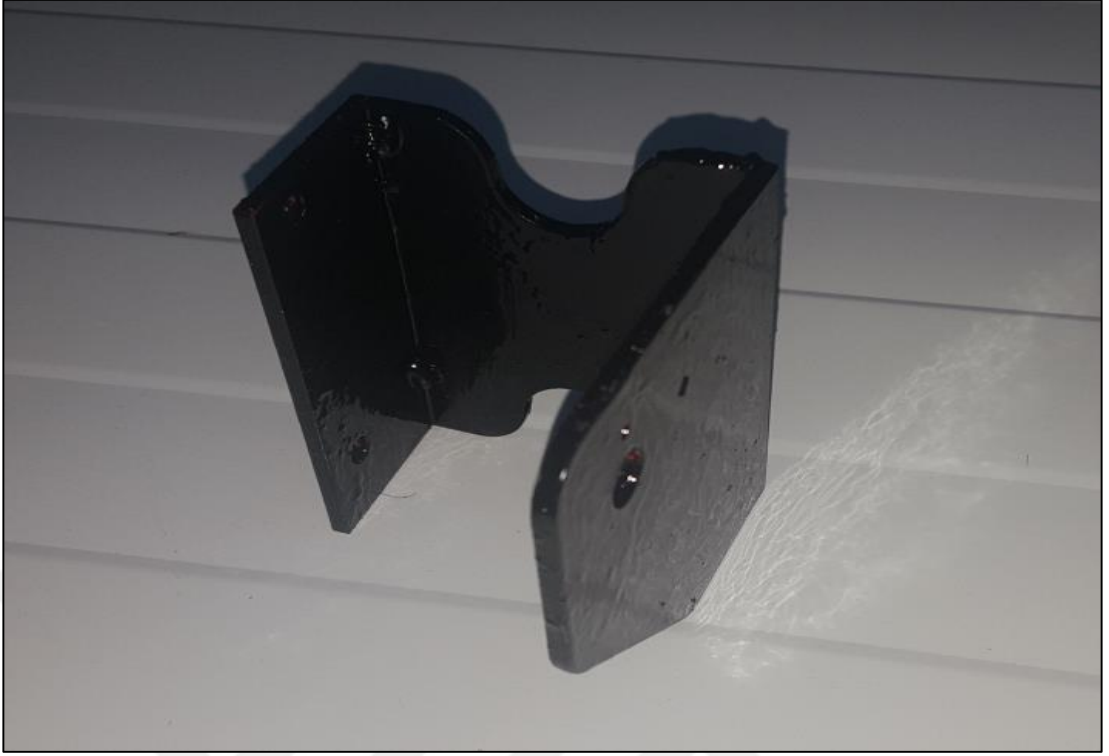
Şekil A.3. Platformun hareketli kısmı



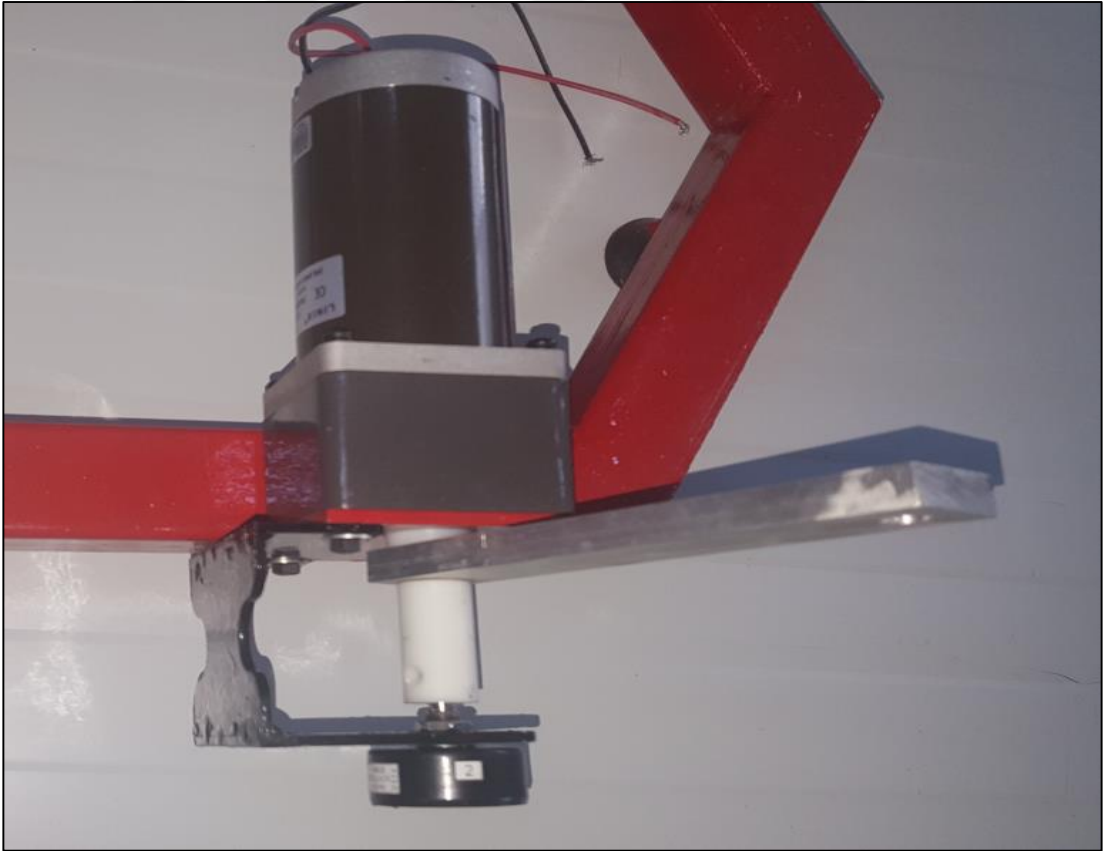
Şekil A.4. Üretilen bağlantı rodu



Şekil A.5. Üretilen motor-rod bağlantı kolu



Şekil A.6. Üretilen motor-sensör bağlantı aparatı



Şekil A.7. Motor kol bağlantısı

KİŞİSEL YAYIN VE ESERLER

- [1] **Tufan, B. H.**, Ürgün, S., Düşük Maliyetli 6 Eksen Hareketli Uçuş Simülatörü Prototip Tasarımı. *Avrupa Bilim ve Teknoloji Dergisi*, 2020 , (20) , 299-310. DOI: 10.31590/ejosat.783420



ÖZGEÇMİŞ

İlk ve orta öğrenimi Erdemli ilçesinde tamamladı. Lise eğitimini Bolu Dörtdivan Anadolu Öğretmen Lisesinde tamamladı. Üniversite eğitimimi Kocaeli Üniversitesi Sivil Havacılık Yüksekokulu, Uçak Elektrik Elektronik bölümünde tamamladım. Aynı yıl Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Havacılık Teknolojileri bölümünde yüksek lisans eğitimine başladı ve hala devam etmekte. Üniversite eğitiminin son iki yılında yazılım sektöründe çalıştı. 2014 ve 2016 yılları arasında Nacsoft Yazılım LTD. şirketinde yazılım geliştirme uzmanı olarak, 2016 ve 2017 yıllarında Visnet Bilişim Sistemleri ve ARGE Danışmanlığı firmasında çalıştı. 2017 yılında üniversite mezuniyetinden sonra çalışmaya başladığı Türk Hava Yolları Teknik A.Ş.'de meslek hayatına devam etmektedir. Aynı zamanda kendisine ait olan thecodeprogram.com üzerinde 2015 yılından bu yana blog yazarlığı yapmaktadır.