T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SİMÜLATÖR İÇİN HAREKET ALGI ALGORİTMASI GELİŞTİRİLMESİ

BERKAY VOLKANER

YÜKSEK LİSANS TEZİ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI
MEKATRONİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
DOÇ. DR. VASFİ EMRE ÖMÜRLÜ

İSTANBUL, 2016

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SİMÜLATÖR İÇİN HAREKET ALGI ALGORİTMASI GELİŞTİRİLMESİ

Berkay VOLKANER tarafından hazırlanan tez çalışması 27.04.2016 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Vasfi Emre ÖMÜRLÜ

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Vasfi Emre ÖMÜRLÜ

Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. Mehmet Selçuk ARSLAN

Yıldız Teknik Üniversitesi

Prof. Dr. Halil Rıdvan ÖZ

Aydın Üniversitesi

Bu çalışma, 0563.STZ.20132 numaralı SANTEZ projesi olup, Sanayi Bakanlığı ve HAVELSAN A.Ş. tarafından desteklenmiştir.

ÖNSÖZ

Mühendislik mesleğinin öğrenilmesinde üst kademelerden biri olan yüksek lisans eğitimimin sonuna gelmiş bulunmaktayım. Öğrendiğim bilgileri ve edindiğim mühendislik yeteneklerimi uygulamaya dökerken birçok zorlukla karşılaşacağımın farkında olarak;

Lisans eğitimimden bu yana bana desteğini esirgemeyen, engin bilgisi ve yol göstericiliği ile bana ışık tutan, hem tez danışmanım, hem de hocam değerli insan Doç. Dr. Vasfi Emre ÖMÜRLÜ hocama,

Derslerim ve dönem projelerimde bana destek olan diğer hocalarıma, bu tez için her türlü maddi ve manevi destekte bulunan Sanayi Bakanlığı ve HAVELSAN A.Ş.'deki değerli çalışanlar Mehmet Can BAŞKAN, Mehmet Murat AYGÜN, Mustafa ALTUNAY ve Hakan KARACAOVALI'ya, proje boyunca her türlü imkânı sunan AMEGA A.Ş.'de çalışan Ömer Faruk ÖZDEMİR'e ve aynı çalışma ortamımızı paylaştığımız yüksek lisans öğrencisi Şemsettin Numan SÖZEN ile lisans öğrencileri Hakan ALTUNTAŞ, Ece Ebru KAYA, Umut TEKİN ve Nejat SÖYLEMEZ arkadaşlarıma,

Bugünlere gelmemde en büyük katkıya sahip olan aileme, dostlarıma ve okul hayatım boyunca her zaman yanımda olan Kübra GÜNDÜZ'e teşekkürlerimi sunarım.

Mart, 2016

Berkay VOLKANER

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ v
KISALTMA LİSTESİ viii
ŞEKİL LİSTESİ ix
ÇİZELGE LİSTESİ xii
ÖZET xiii
ABSTRACT xv
BÖLÜM 1
GİRİŞ 1
1.1 Literatür Özeti 1
1.1.1 Simülatörlerin Yapısı ve Geçtiğimiz Dönemlerde Yapılan Çalışmalar 1
1.1.2 Hareket Algı Yazılımı Çalışmaları 10
1.1.2.1 Klasik Hareket Algı Algoritması 10
1.1.2.2 Uyarlamalı Hareket Algı Algoritması 13
1.1.2.3 İnsan Hareket Algı Sistemi, Yapısı ve Çalışma Prensibi 14
1.1.2.3.1 Yarı Dairesel Kanallar 14
1.1.2.3.2 Otolit Taşları 16
1.1.2.4 Optimal Kontrollü Hareket Algı Algoritması Çalışmaları 19
1.1.2.5 Model Öngörülü Hareket Algı Yazılımı Çalışmaları 20
1.2 Tezin Amacı 22
1.3 Hipotez 22
BÖLÜM 2
SİMÜLATÖRÜN MEKANİK VE ELEKTRONİK YAPISI 24
2.1 Simülatörün Mekanik Yapısı 25
2.2 Simülatörün Elektronik Yapısı 29
2.3 Uçuş Simülatörü Genel Şeması 31
BÖLÜM 3
STEWART PLATFORMUNUN KİNEMATİK ANALİZİ 32

3.1 Platformun Ters Kinematik Analiz 33
3.2 Platformun Düz Kinematik Analizi 34
3.3 Koordinat Eksenleri Dönüşümü 38
BÖLÜM 4
HAREKET ALGI ALGORİTMASI ÇALIŞMALARI 42
4.1 Klasik Sıfırlamalı Hareket Algı Algoritması 42
4.2 Model Öngörülü Hareket Algı Algoritması 47
4.2.1 Kontrol Yapısı ve Durum Uzay Modeli 48
4.2.2 İnsan İç Kulak Sistemi Yapısı, Elemanları ve Durum Uzay Modeli ... 51
4.2.2.1 Yarı Dairesel Kanal Sistemi Matematiksel Modellenmesi .. 51
4.2.2.2 Otolit Taşlarının Matematiksel Modellenmesi 53
4.2.3 Bütünleşik Durum Uzay Denklemleri 55
4.2.4 Eğim Etkisinin Durum Uzay Matrislerine Eklenmesi 56
4.2.5 Genel Sistemin Durum Uzay Modellerinin Oluşturulması 57
4.2.6 Maliyet Fonksiyonu ve Sistemin Optimizasyon Aşamaları 61
4.2.6.1 Hildreth Kuadratik Programlama Algoritması 65
4.2.7 Model Öngörülü Kontrolcünün Parametrelerinin Ayarlanması 73
BÖLÜM 5
SONUÇLAR 76
5.1 Klasik Algoritma Yazılımına Ait Pilot Denemeleri Sonuçları 78
5.1.1 Birinci Pilota Ait Denemelerin Grafikleri 81
5.1.2 İkinci Pilot Ait Denemelerin Grafikleri 85
5.1.3 Klasik Sıfırlamalı Algoritmaya Ait Genel Sonuçlar 89
5.2 Model Öngörülü Kontrol Algoritmasına Ait Pilot Denemeleri Sonuçları 91
5.2.1 Birinci Pilot Ait Denemelerin Grafikleri 93
5.2.2 İkinci Pilot Ait Denemelerin Grafikleri 99
5.2.3 Model Öngörülü Kontrole Ait Genel Sonuçlar 105
KAYNAKLAR 107
EK-A
DÜZ KİNEMATİK ALGORİTMASINA AİT YAZILIM DOSYASININ İÇERİĞİ 110
EK-B
KLASİK HAREKET ALGI YAZILIMININ PARAMETRE DOSYASI 113

EK-C
HILDRETH'İN KUADRATİK PROGRAMLAMA ÇÖZÜCÜSÜNÜN KODLARI..... 114
EK-D
MODEL ÖNGÖRÜLÜ HAREKET ALGI YAZILIMININ KODLARI 116
ÖZGEÇMİŞ 124

SİMGE LİSTESİ

$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$	Bütünleşik (Augmented) modele ait durum uzay matrisleri
$\boldsymbol{A_e}$	MPC formülasyonunda sıfırlama için kullanılan ek durum matrisi
$\boldsymbol{A_v}, \boldsymbol{B_v}, \boldsymbol{C_v}$	Vestibüler sistemin durum uzay matrisleri
$\boldsymbol{A_m}, \boldsymbol{B_m}, \boldsymbol{C_m}$	Eğim koordinasyonuyla beraber vestibüler sisteme ait durum uzay matrisleri
a_x, a_y, a_z	Uçuş simülasyonu öteleme ivme değerleri
$\boldsymbol{B_p}$	Simülatör bacaklarının, alt platform bağlantı noktalarının başlangıç koordinatları
$\boldsymbol{CV}$	Sınırlandırılmış (Constrained) sistem değişkenleri
dTH	Otolit taşlarının öteleme hareketini algılama sınırı
$\boldsymbol{F}, \boldsymbol{\varphi}$	Model öngörü durum uzay matrisleri
Fr_H	Simülatörün yere bağlantısının bulunduğu koordinat eksen takımı.
Fr_I	Simülatörün üst platformunun merkez noktasının koordinat eksen takımı
Fr_{PH}	Görsel simülasyondaki pilotun baş bölgesinin koordinat ekseni
Fr_S	Simülatörün üst platformunun merkez noktasının koordinat eksen takımı
Fr_{PS}	Simülatördeki pilotun baş bölgesine ait koordinat ekseni
Fr_{CG}	Helikopterin ağırlık merkezidir
$\boldsymbol{f}$	Görsel simülasyondaki aracın üzerine etki eden kuvvetler
$\hat{\boldsymbol{f}}$	Hareket sonucunda insanın algıladığı öteleme yönündeki ivmeler
g	Yerçekimi ivmesi
$\boldsymbol{H}$	Yer çekimi etkisinin sisteme dâhil edilmesinde kullanılan matris
$\boldsymbol{I}$	Birim matris
i	Sayaç parametresi
J	Maliyet fonksiyonu
$K, \tau_a, \tau_L, \tau_s$	Otolit taşları matematiksel modeline ait sistem parametreleri
K_x, K_y	Sınırlama olmadan hesaplanan maliyet fonksiyonu katsayıları
k	Örnekleme zamanı
$\boldsymbol{M}, \boldsymbol{b}$	Maliyet fonksiyonu hesaplamaları için kullanılan sınır şartları
$\boldsymbol{MV}$	Manipüle edilmiş (Manipulated) değişkenler
N_p	Sonlu zaman düzlemi boyunca öngörülen durum sayısı
N_u	Sonlu zaman düzlemi boyunca öngörülen kontrol girişi sayısı
p	Sistemin çıkış sayıları
p_c	Sistemin kontrol girişi sayıları
p_x, p_y, p_z	Görsel simülasyondaki araca ait konumlar
$\boldsymbol{p}_S^{Ps}$	Üst platformun orta noktası ile pilotun göz noktası arasındaki x, y ve z yönlerindeki mesafeleri içeren vektör

$\boldsymbol{P}$	Simülatör bacaklarının, üst platform bağlantı noktalarının başlangıç koordinatları
$\boldsymbol{Q}$	Maliyet fonksiyonu hata katsayı matrisi
$\boldsymbol{R}$	Maliyet fonksiyonu kontrol girişi değişimi katsayı matrisi
$\overline{\boldsymbol{R}}$	N_p öngörü düzlemi boyunca tahmin edilen referans bilgilerini içeren vektör
$\boldsymbol{R_B^P}$	Simülatörde üst platformun alt platforma göre rotasyon matrisi
$\boldsymbol{R_{PS}^S}$	Üst platformun orta noktası ile pilotun göz noktası arasındaki rotasyon matrisi
$\boldsymbol{R_z, R_y, R_x}$	Üst platformun x,y ve z koordinat eksenleri etrafında rotasyon matrisi
$\boldsymbol{r}$	Uçuş simülasyonundan elde edilen insan algısı referans bilgisi
$\boldsymbol{S}$	Maliyet fonksiyonu kontrol girişi katsayı matrisi
s	Frekans domeini
$\boldsymbol{T_a, T_L, T_s}$	Yarı dairesel kanalların matematiksel modeline ait zaman sabitleri
$\boldsymbol{T_{PS}^S}$	Üst platformun orta noktası ile pilotun göz noktası arasındaki dönüşüm matrisi
$\boldsymbol{Tilt_a}$	Eğim koordinasyonuna (Tilt coordination) ait ek durum ve çıkış matrisi
$\boldsymbol{U}$	N_u öngörü düzlemi boyunca tahmin edilen kontrol giriş vektörü
$\boldsymbol{u}$	Platforma gönderilen kontrol girişi
v_x, v_y, v_z	Görsel simülasyondaki araca ait hızlar
ω	Görsel simülasyondaki araçtan alınan açısal hızlar
ω_h	Yüksek geçiren filtre frekansı
ω_l	Düşük geçiren filtre frekansı
$\widehat{\omega}$	Hareket sonucunda insanın algıladığı açısal hızlar
$\boldsymbol{x_m}$	Vestibüler sisteme ait durum değişkenleri
$\boldsymbol{Y}$	N_p öngörü düzlemi boyunca tahmin edilen sistem çıkışı vektör
$\boldsymbol{y}$	Simülasyon sonucunda elde edilen insan algısı referans bilgisi
ρ	Üst platformun birbirlerine yakın 2 mafsal noktası arasındaki ayırma açısı
σ	Alt platformun birbirlerine yakın 2 mafsal noktası arasındaki ayırma açısı
ϵ	Vestibüler model sistem çıkışına ait hata vektörü
λ	Lagrange çarpanı
δTH:	Yarı dairesel kanallarının açısal hareketi algılama sınırı
ϕ, θ, Ψ	Simülatörün yuvarlanma (roll), yunuslama (pitch) ve sapma (yaw) açısal konumları
$\varphi_{x,y}$	Platformun x ve y açıları etrafındaki eğim koordinayonu açısı
$\Delta\boldsymbol{U}$	N_u öngörü düzlemi boyunca tahmin edilen kontrol girişi değişimi vektörü
$\Delta\boldsymbol{u}$	Kontrol girişi değişimi
$\Delta\boldsymbol{x_m}$	Vestibüler sisteme ait durum değişkenlerinin değişimi
$\Delta\boldsymbol{MV}$	Manipüle edilmiş değişkenlerin değişimi
$\boldsymbol{U_{max}}$	Platforma gönderilen kontrol girişlerinin üst limiti
$\boldsymbol{U_{min}}$	Platforma gönderilen kontrol girişlerinin alt limiti
$\Delta\boldsymbol{U_{max}}$	Platforma gönderilen kontrol girişleri değişiminin üst limiti
$\Delta\boldsymbol{U_{min}}$	Platforma gönderilen kontrol girişleri değişiminin alt limiti
$\boldsymbol{Y_{max}}$	Platformun açı, pozisyon ve öteleme hızlarının üst limitleri
$\boldsymbol{Y_{min}}$	Platformun açı, pozisyon ve öteleme hızlarının alt limitleri.

$( )_{SCC}$	Yarı dairesel kanalların matematiksel modeline ait parametre ve değişkenler
$( )_{oto}$	Otolit taşları matematiksel modeline ait parametre ve değişkenler
$( )_x, ( )_y, ( )_z$	Eksenlere ait parametre ve değişkenler

KISALTMA LİSTESİ

CV	Sınırlandırılmış veriler (Constrained Variables)
HAY	Hareket algı yazılımı
HS	Hareket sistemi
kHAY	Klasik hareket algı yazılımı
KK	Kumanda kolu
MPC	Model öngörülü kontrol (Model Predictive Control)
MIMO	Çoklu giriş, çoklu çıkış sistemleri (Multi Input Multi Output)
MOHAY	Model öngörülü hareket algı yazılımı
MV	Manipüle edilmiş veriler (Manipulated Variables)
SF	Sıfırlama filtreleri
SISO	Tek giriş, tek çıkış sistemleri (Single Input Single Output)
USY	Uçuş simülasyonu yazılımı
VS	Vestibüler sistem
VSg	Vestibüler sistem girdisi

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 IKK ve MARS simülatörü. 3
Şekil 1.2 VTI simülatörü. 3
Şekil 1.3 TRL ve JARI simülatörü. 4
Şekil 1.4 Iowa sürüş simülatörü ve NADS sürüş simülatörü. 4
Şekil 1.5 CARD ve ULTIMATE simülatörü. 5
Şekil 1.6 Leed Üniversitesine ait simülatör. 6
Şekil 1.7 DLR ve Desdemona 7
Şekil 1.8 Toyota sürüş simülatörü 7
Şekil 1.9 Cyber Motion simülatörü 8
Şekil 1.10 Padova Üniversitesinde yapılan model öngörülü kontrollü sürüş simülatörü. 9
Şekil 1.11 Max Planck Enstitüsünde yapılan simülatör çalışmaları [20] 9
Şekil 1.12 Helikopterde x yönünde ivmelenirken platformda oluşan eğim. 11
Şekil 1.13 Platformda oluşan açı ile elde edilen eğim etkisi. 12
Şekil 1.14 Klasik algoritmanın akış diyagramı. 12
Şekil 1.15 Uyarlamalı algoritmanın akış şeması [26]. 13
Şekil 1.16 Yarı dairesel kanalların hareket sırasındaki davranışı 15
Şekil 1.17 Yarı dairesel kanalların çalışması sırasında oluşan sinyal değişimleri a) insan başı saat yönünde döndüğü sırada, b) İnsan başı saat yönünün tersinde döndüğü sırada. 16
Şekil 1.18 Makülanın yapısı. 17
Şekil 1.19 Otolit taşları sinyal düzeyleri a) insan başı dik konumda, b) insan başı yüz istikametinde ivmelenirken, c) insan yüz istikametinin tersi yönünde hareket ederken 18
Şekil 1.20 Optimal hareket algı algoritması yazılımı [27] 20
Şekil 1.21 Model öngörülü kontrolün çalışma prensibi. 21
Şekil 2.1 Uçuş simülatörü genel gösterimi. 24
Şekil 2.2 Platformun hareketli pistonları 25
Şekil 2.3 USY bilgisayarı 26
Şekil 2.4 Kumanda kolu, hız kolu ve pedallar 27
Şekil 2.5 Yunuslama (Pitch) hareketinin açısal limiti. 28
Şekil 2.6 Yuvarlanma (Roll) hareketinin açısal limiti. 28
Şekil 2.7 Sapma (Yaw) hareketinin açısal limiti. 29
Şekil 2.8 Bacakların tahrikini sağlayan Siemens kontrolcüleri. 30
Şekil 2.9 Simülatör sisteminin elektronik yapısı. 30
Şekil 2.10 Genel sistemin şeması. 31
Şekil 3.1 Helikopterin yön bilgileri. 33

Şekil 3.2 Platformun düz kinematik yazılımının simulink görseli 35
Şekil 3.3 Platformun düz kinematik algoritması [31]. 36
Şekil 3.4 Düz kinematik analizin klasik algoritmalı modeldeki performansı. 37
Şekil 3.5 Düz kinematik analizin, model öngörülü kontrol algoritması modelindeki performansı. 38
Şekil 3.6 Simülatör ve görsel simülasyona ait koordinat eksenleri bilgisi 39
Şekil 3.7 Pilotun göz hizası – üst platform orta noktası arasındaki dönüşüm matrisi. ... 40
Şekil 3.8 Dönüşüm bloğu içyapısı. Platformun hareket eksenlerinin ayarlanması. 41
Şekil 4.1 Klasik sıfırlamalı algoritmanın kalibrasyon aşamaları 43
Şekil 4.2 kHAY filtrelerinin basamak giriş durumundaki cevapları, a) basamak giriş, b) Sinyalin yüksek geçiren filtre cevabı, c) Sinyalin düşük geçiren filtre cevabı, d) Yüksek ve düşük geçiren filtrelerin cevaplarının toplamı. 45
Şekil 4.3 kHAY'nın MATLAB/Simulink yapısı. 47
Şekil 4.4 Yarı Dairesel Kanallara Ait Blok Diyagram 51
Şekil 4.5 Otolit Taşları Yapısının Blok Diyagram 53
Şekil 4.6 Model Öngörülü Kontrol Sistemi Akış Şeması 66
Şekil 4.7 Model öngörülü hareket algı algoritmasının MATLAB/Simulink yapısı 67
Şekil 5.1 Görsel simülasyondan, kontrolcü bilgisayarına helikopter verilerinin aktarılması ve verilerin birim dönüşümü 77
Şekil 5.2 Simulink modelinin genel görüntüsü 77
Şekil 5.3 Platformun bacakları ve pilotun oturuş pozisyonuna ait görsel 78
Şekil 5.4 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri 81
Şekil 5.5 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma açısal hız verileri 82
Şekil 5.6 Birinci pilot için, hareketli platformun, uçuş senaryosu boyunca öteleme ve dönme konumları ile oluşan hatalı hareketlerin gösterimi 83
Şekil 5.7 Birinci pilot için uçuş senaryosu boyunca platformun bacak uzunlukları. 84
Şekil 5.8 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri 85
Şekil 5.9 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma açısal hız verileri 86
Şekil 5.10 İkinci pilot için, hareketli platformun, uçuş senaryosu boyunca öteleme ve dönme konumları ile oluşan hatalı hareketlerin gösterimi 87
Şekil 5.11 İkinci pilot için uçuş senaryosu boyunca platformun bacak uzunlukları. 88
Şekil 5.12 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri 93
Şekil 5.13 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma yönlerine ait açısal hız verileri 94
Şekil 5.14 Birinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı x-y-z yönü ivme hissiyatı 95
Şekil 5.15 Birinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı yuvarlanma – yunuslama – sapma yönlerindeki açısal hız hissiyatı 96
Şekil 5.16 Birinci pilota ait platforma gönderilen ve platformun yapabildiği hareketlerin grafiği. 97

Şekil 5.17 Birinci pilota ait, platformun eğim etkisi (Tilt Coordination) sonuçları 98
Şekil 5.18 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri 99
Şekil 5.19 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma yönlerine ait açısal hız verileri 100
Şekil 5.20 İkinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı x-y-z yönü ivme hissiyatı 101
Şekil 5.21 İkinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı yuvarlanma – yunuslama – sapma yönlerindeki açısal hız hissiyatı 102
Şekil 5.22 İkinci pilot için, hareketli platformun, uçuş senaryosu boyunca öteleme ve dönme konumları 103
Şekil 5.23 İkinci pilota ait, platformun eğim etkisi sonuçları 104

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Platformun hareket limitleri 28
Çizelge 4.1 Klasik algoritmada kullanılan filtrelerin denklemleri 43
Çizelge 4.2 Filtrelerin çalışma prensibi [33] 44
Çizelge 4.3 Klasik algoritma filtrelerinde kullanılan parametrelerin listesi. 46
Çizelge 4.4 Yarı dairesel kanal transfer fonksiyonu parametreleri [28] 52
Çizelge 4.5 Otolit taşları matematiksel modele ait parametre değerleri [28] 53
Çizelge 4.6 Sonuç matrislerinde kullanılan parametrelerin içerikleri 58
Çizelge 4.7 Maliyet fonksiyonunda kullanılan parametreler ve açıklamaları 68
Çizelge 4.8 Hata ağırlık matrisinin parametre açıklamaları 69
Çizelge 4.9 Kontrol girişi ve kontrol girişi değişimi ağırlık matrislerinin parametre açıklamaları 70
Çizelge 4.10 Simülasyon durum uzay modelinin çıkış sınırları 71
Çizelge 4.11 Platforma gönderilen kontrol girişi ve kontrol girişi değişimlerinin sınırları 72
Çizelge 4.12 Hata ağırlık matrisinin köşegen parametreleri 74
Çizelge 4.13 Kontrol girişine ait ağırlık matrisinin köşegen parametreleri 74
Çizelge 4.14 Kontrol girişi değişimine ait ağırlık matrisinin köşegen parametreleri 74
Çizelge 4.15 Sistem çıkışlarına ait kontrolcü sınırları 75
Çizelge 4.16 Kontrol girişlerine ait sınırlar 75
Çizelge 4.17 Kontrol girişleri değişimlerine ait sınırlar 75
Çizelge 5.1 Pilotlar ile ilgili bilgiler 76
Çizelge 5.2 Klasik sıfırlamalı algoritma için uçuş senaryoları ve puanları 79
Çizelge 5.2 Klasik sıfırlamalı algoritma için uçuş senaryoları ve puanları (devamı) 80
Çizelge 5.3 Model öngörülü kontrol algoritması için uçuş senaryoları ve puanları 91
Çizelge 5.3 Model öngörülü kontrol algoritması için uçuş senaryoları ve puanları (devamı) 92

ÖZET

SİMÜLATÖR İÇİN HAREKET ALGI ALGORİTMASI GELİŞTİRİLMESİ

Berkay VOLKANER

Mekatronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Vasfi Emre ÖMÜRLÜ

Pilotların gerçek uçuş denemeleri, maliyetli ve riskli olmaktadır. Bu dezavantajları ortadan kaldırmak ve güvenli bir uçuş eğitimi verilebilmesi için uçuş simülatörleri kullanılmaktadır. Kullanılan taşıt ve şartlara göre farklı tipte simülatörler vardır. Bunlar seri veya paralel robot sistemleridir. Simülatör kullanımında amaç, görsel ortamda gerçekleşen uçuş senaryosunun, simülatörü kullanan pilotun üzerinde gerçekçi bir şekilde hissettirilmesini sağlamaktır. Bunu yapabilmek için, klasik yöntemlerle beraber insan algısını içeren kontrol yapılarına sahip yazılımlar da kullanılmaktadır.

Sürüş ve uçuş simülatörleri sürücünün hissiyatı ile ilgili çalışmalar açısından kullanışlıdır. Bu nedenle, üniversiteler ve endüstriyel laboratuvarlar yeni nesil prototipler ve araç dinamiği modelleri için araştırmalar yapmaktadırlar. Trafik kazalarındaki artış birçok kamu kuruluşunu yollardaki güvenliği arttırmak amacı ile farklı taşımacılık ve araç tasarımı yapmaları için teşvik etmiştir ve sürüş simülatörleri sürüş anında insan davranışının nasıl olduğunu gerçeğe yakın bir şekilde daha iyi anlamamızı sağlayan önemli unsurlardandır. Benzer şekilde, daha yüksek bir risk unsuru taşıyan havacılık için de uçuş simülatörleri pilot eğitimleri için vazgeçilmez bir ihtiyaç haline gelmeye başlamıştır. Havacılık sektöründeki firmalar bu konudaki tasarım ve Ar-Ge çalışmalarını yoğunlaştırmışlardır. Sanal sürüşün mümkün olduğu kadar gerçek sürüşe yakın olması için simülatörün ses, sanal görsellik, kabin yapısı, dış ortamdan soyutlaştırma ve eylemsizlik etkileri ile donatılması gerekmektedir. Bu nedenle simülatör içerisindeki pilot hislerini gerçekte algılanan hisler ile en iyi şekilde benzetim yapabilecek şekilde yeniden oluşturabilmek için mobil platformlar ekranlar ile birleştirilmiştir ki bu da

performansı arttırmaya ve sanal ortamı daha geniş bir şekilde gerçekleştirmeye olanak sağlamaktadır.

Bu projede imalatı yapılan ve gerçek standartlara uygun şekilde hazırlanan uçuş simülatörü için hareket algı yazılımı (HAY) oluşturulmuştur. Yazılım olarak klasik ve model öngörülü hareket algı algoritması oluşturulmuştur. Model öngörülü algoritma için insan iç kulak sistemine, diğer adıyla vestibüler sistem'e (VS) ait matematiksel denklemler kullanılmıştır. İyileştirme yöntemi olarak Hildreth'in kuadratik programlama çözücüsü kullanılmıştır. Oluşturulan modeller simülatörle birleştirilip pilotlar aracılığı ile deneysel çalışma yapılmıştır. Her iki algoritma için de karşılaştırmalar yapılmıştır.

Anahtar Kelimeler: Model öngörülü kontrol, simülatörlerde hareket algısı, vestibüler sistem, uçuş simülatörü, klasik sıfırlamalı algoritma

ABSTRACT

DEVELOPMENT OF MOTION CUEING ALGORITHM FOR THE SIMULATOR

Berkay VOLKANER

Department of Mechatronic Engineering

MSc. Thesis

Adviser: Assoc. Prof. Dr. Vasfi Emre ÖMÜRLÜ

Flight experience of pilots are very costly and risky. Due to these reasons, simulators are widely using in flight trainings. There are many different kind of platforms, as like parallel and serial robot, according to vehicle and environment conditions. The first goal of using simulators are making feel the virtual flight on pilots perception. In order to realize, apart from the conventional methods, also different control algorithms including human perception are widely used.

Flight and driving simulators are widely using in studies on the pilot's perception. Therefore, universities and industrial laboratories are required to research for a new generation of prototypes and vehicle dynamic models. The increase of traffic accidents is encouraged many public organizations to make a difference in the transport technology and vehicle design with the aim to improve road safety and so, driving simulators are big elements in determine driver behaviors realistically during driving. Similarly, due to the higher risk factor in aviation, the flight simulators are become indispensable need for pilot training. Companies in the aviation sector have increased their design and R & D works in this regard. To be similar with the real flight, simulators must be equipped with the well designed sound, virtual reality, cabinet structure, seperate from external environment and inertial effects. For this case, in order to rebuild the best simulation with pilot perception in simulator to real time flight, screens are combined together with the platform and so, this case improves the flight performance and using the virtual space in the larger workspace area.

In this project, classical and model predictive controlled motion cueing algorithm are designed. Human vestibular system matematical model is used for theorical model in model predictive control. Hildreth's Quadratic Programmer solver is used in order to

solve the cost function of model predictive controller. These models were combined with simulator and tested with pilots. Experimental results compared between both two algorithms.

Keywords: Model predictive control, classical washout filters, Stewart Platform, pilot perception in simulators, vestibular system, flight simulator.

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

1.1.1 Simülatörlerin Yapısı ve Geçtiğimiz Dönemlerde Yapılan Çalışmalar

Uçuş simülatörlerinin yoğunlukla kullanıldığı alan eğitim alanıdır. Bunun dışında yolcu uçakları ve uzay araçları eğitim simülasyonu olarak da çokça kullanılır. 2. Dünya savaşı için birçok sayıda pilotun eğitimi ve yeteneklerinin geliştirilebilmesi için askeri uçuş simülatörlerinin varlığına ihtiyaç duyulmuştur. Elektronik dünyasının da bu dönemde gelişmesiyle analog bilgisayarların kullanımı artmıştır ve bu durumun simülatör teknolojisinin ortaya çıkmasında katkısı olmuştur. Aerodinamik yapı ve uçuş denklemlerinin modellenmesi gibi özellikler simülatörlerin hem yazılım hem de donanım bölümüne dâhil edilebilmiştir. Örneğin, analog bilgisayarlar sayesinde o dönemde denenen bir deniz uçağının yükseklik kontrolü çalışması yapılmış fakat yeni bir teknoloji olmasından dolayı sağlıklı bir sonuç alınamamıştır.

Uçuş simülatörleri için kullanılan algoritmalar hemen hemen birbirleriyle benzer amaçlar içermektedir. Tek amaç, pilota gerçekçi bir uçuş hissini sağlamaktır. Yapı ve kullanılan metot olarak farklılıklar göstermektedirler. Gerçek ve sanal verilerde oluşan farklılıkların hangi faktörden dolayı oluştuğunun kolaylıkla incelenmesi gerekmektedir. Genel olarak bu faktörler 3 kategoride toplanıyor. Bunlar insan hareket algı sistemi, sanal görüntüde oluşabilecek problemler ve ses sistemidir. İnsan algısı iç kulak, literatürdeki ismiyle vestibüler sistem sayesinde oluşturulmaktadır [1].

Simülatörler hareket bilgisinin geri beslenmesi ile kişiye gerçekçi bir sürüş hissi yaratır. Hareket algı algoritması uçuş simülasyonu yazılımının (USY) mevcut öteleme ivmesi ile açısal

hız verilerini kullanır [2]. Uçuş simülatörleri tamamen güvenilir, gerçeklenebilir yapılardır. Yazılım, donanımsal ve aynı zamanda simülatörün fiziksel yetenekleri artırılarak bu gerçekliğin etkisi artırılmaya çalışılmıştır. Fakat fiziksel limitlerden dolayı bu hareket hissinin uygulanması birtakım zorlu durumları da beraberinde getirmektedir. Gerçek sürüş etkisiyle benzer olması için simülatör üzerinde yapılması gereken birtakım yazılım ayarları vardır. Hareket algı algoritmasında bu yapılanmanın önemi bir hayli yüksektir [3].

Simülatör bir insan makine ara yüzüdür. Simülasyon süreci, pilotun kendisini de içerdiğinden kontrol döngüsünün bir parçası olarak kabul etmektedir. Sistem yoğun bir kod algoritması içermektedir. Örneğin, D seviye bir simülatörde çok yüksek sayıda kod dizisi bulunmaktadır ve fiyatı yüksek miktarlardadır. Kompleks yapı ve pahalı olmasından dolayı bir simülatör tasarımı yapmak oldukça zorlu bir iştir. Simülatörlerin üretiminde, ihtiyaca göre serbestlik dereceleri ve kapasiteleri belirlenmeli, algı yazılımı bu belirlenen serbestlik derecesine göre modellenmelidir. Böylelikle hareket olmayan eksenler, sistemden çıkarılmış ve ekonomik boyutlarda simülatörlerin üretimi gerçekleştirilmiş olmaktadır. İhtiyaç ve özel isteklere göre simülatörler gerçeklik kapsamı içerisinde farklı niteliklere sahip olabilmektedirler [4].

Yapılan literatür taraması sonucunda çok çeşitli simülatör modelleri ve yazılımları tespit edilmiştir. Yazının devamında bulunan simülatörler ile ilgili öz açıklamalar yapılacaktır. Literatür taraması genel olarak bütün çalışmaların arşivlendiği [1] literatürü incelenerek yazılmıştır.

Avrupa'da, ilk önemli bilgisayar tabanlı sürüş simülatörü 1970'lerin başlarında Alman araç üreticisi Volkswagen tarafından geliştirilmiştir. Tek bir düz ekrana sahip yuvarlanma, yunuslama ve sapma (roll, pitch, yaw) yönünde sınırlı hareket sağlayan bu sistem, bir araç kabini olmadan oluşturulmuştur.

1984 yılında ise Hamburg Otomotiv Mühendisliği ve Pistonlu Motorlar Enstitüsü tarafından bu bahsedilen ilk model geliştirilmiştir. Bir Volkswagen Golf taksi, şoförü önünde görüntünün dar bir alan ile ve sadece yanal hareket sağlayan bir kızak üzerine monte edilmiştir. On yıl sonra, simülatörler Modüler Otomobil Yol Simülatörü, MARS [5] olmak üzere düzenlenmiştir. Bu düzenlemeyle kızakların üstüne 6 serbestlik dereceli platformun bağlantısı yapılmıştır (Şekil 1.1).



Şekil 1.1 IKK ve MARS simülatörü.

İsveç Ulusal Karayolu ve Ulaştırma Araştırma Enstitüsü, kendi simülatörünün yapımını tamamlamıştır. 120°'lik görüş alanına sahip, 3 serbestlik dereceli, ± 24° yunuslama ve sapma, yönünde, yanal yer değiştirmesi ise 7 metre hareket kapasitesine sahiptir. Ulaşabileceği maksimum ivme değeri ise 0.4g'dir (Şekil 1.2).



Şekil 1.2 VTI simülatörü.

1990'larda bir çok araba üreticileri simülasyon alanındaki çalışmalarını yoğunlaştırmıştır. Mazda [6], General Motors [7], Ford [8], Renault [9] ve BMW tam hareket sistemlerine (HS) sahip simülatörler üzerine yönelmiştir.

Bunlarla birlikte, araştırma kurumları da bu konu üzerinde çalışmalar yapmaya başlamış ve küçük ölçekli tasarımlarını geliştirme yönünde ilerlemiştir. 1990'da Japon Otomobil Araştırma Enstitüsünde, simülatör üzerinde çalışmalar başlamış ve 1995'te kullanıma geçilmiştir [10]. Sistem, altı serbestlik dereceli hareket yeteneği ile ilk Japon simülatörü olmuştur. Honda Civic kabininin 2001 yılında geliştirilmesinden sonra, TRL araç simülatörü (Şekil 1.3) 2004 yılında İngiltere'nin ilk sürüş simülasyon merkezini oluşturmuştur.



Şekil 1.3 TRL ve JARI simülatörü.

1994 yılında Iowa Üniversitesi (IDS) [11] tarafından sürüş simülatörü geliştirilmiştir (Şekil 1.4). Simülatör, Boeing 737 uçuş simülatörü olarak dizayn edilmiştir. Araç kabini ise Ford Taurus Sedan'dır ve 140°'lik bir görüş alanı sağlanmıştır.

NADS (National Advanced Driving Simulator), uzunlamasına ve yanal hareket serbestliğine sahip, 4 farklı ve değiştirilebilir araç kabinine sahiptir. Sapma ekseninde 330°'lik dönme serbestliğine sahiptir. Bu yapı aynı zamanda ilk 9 serbestlik dereceli sistem olma özelliğine de sahiptir (Şekil 1.4).



Şekil 1.4 Iowa sürüş simülatörü ve NADS sürüş simülatörü.

21. yüzyıla doğru, elektriksel sürüş HS'lerinin performansı hidrolik sistemlerle rekabete girmeye başlamıştır. Bu, daha hesaplı ve etkin hareketli platform çağının başlangıcı olmuştur.

Bunların ilki olma özelliğine sahip Renault ULTIMATE, 3 kanallı 150° görüş alanlı, 1 tonluk donanıma sahiptir (Şekil 1.5). Kızak, 7 metrelik yanlamasına ve uzunlamasına hareket edebilmekte ve eyleyiciler 500mm'lik hareket özelliğine sahiptir [12]. Bunun dışında CARD isimli simülatör üzerinde de çalışmalar yapmışlardır (Şekil 1.5). CARD, tekerlerinde ve pedalında kuvvet geri besleme sistemi bulunan, 5 m/s²'lik ivmelenme kapasitesi olan ±30 cm'lik çalışma alanı içerisinde hareket eden bir sistemdir.



Şekil 1.5 CARD ve ULTIMATE simülatörü.

CARD, ağırlığı 1000 kg olan elektromekanik, ±30 cm ve ±30°, 6 serbestlik dereceli çalışma uzayına sahip, pedal ve tekerlerde kuvvet geri besleme birimlerine sahiptir. CRT 808S tarafından görüntülenen, ön ve arka olmak üzere 60 Hz gerçek zamanlı görüntü oluşumuna sahip 6 bilgisayardan oluşmaktadır (Pentium Windows xp 3 Ghz). ULTIMATE simülatörü ise daha hafif kokpit ve simülasyon ekranına sahiptir. ±7 m/s² ivmelenme değerine kadar çıkmaktadır. 30°'nin üzerinde hareket (3 açısal eksen boyunca) ve 300 deg/s hıza sahiptir.

3 yıl sonra, Leeds Üniversitesi [13] ve Peugeot [14] aynı simülatör üzerinde çalışmış ve yükü 2.5 tona kadar çıkmıştır. Fakat yatay yöndeki ilerleme kapasitesi 5 m'ye düşmüştür (Şekil 1.6).



Şekil 1.6 Leed Üniversitesine ait simülatör.

Hanover, Alman Havacılık ve Uzay Merkezi Ulaştırma Araştırmaları Enstitüsü'nde DLR simülatörü üretilmiştir [15] (Şekil 1.7). Bu simülatörün farkı, pistonlarda aşağı asılı bir şekilde duran kabine sahip olmasıdır. Bunun avantajı, hiçbir sistem kaybı yaratmadan, dönme noktalarının ve hareketin sürücüde oluşturduğu hissinin, sürücünün başına daha yakın olmasıdır.

Hollanda'da Desdemona isimli simülatör üretilmiştir. 2m'lik dikey hareketle sınırlı ve eksenler çevresinde limitsiz dönüş özelliğine sahiptir. Başta askeri amaçlı üretilmiştir ve küçük sürüş simülatörleriyle benzer özelliklere sahip bir yapıdır. Geniş bir sapma hareketi oranına sahiptir ve yanal hızlanmayı sürdürmek için agresif manevralar içeren sürüş uygulamaları bulunmaktadır (Şekil 1.7) [16].

Yere sabit simülatörlerde öteleme ivmelerinin tam olarak hissettirilemeyeceği referanslarla belirtilmiştir. Stewart platformunun bir kızağın üstünde hareket etmesiyle araba simülatörlerinde daha verimli hareket algısı oluştuğu iddia edilmiştir. Örnek bilgi olarak, "Eğim koordinasyonu" algoritmasının daha çok uçuş simülatörleri için verimli olduğu, araç simülatörlerinde ise daha az etkin olduğu belirtilmiştir.



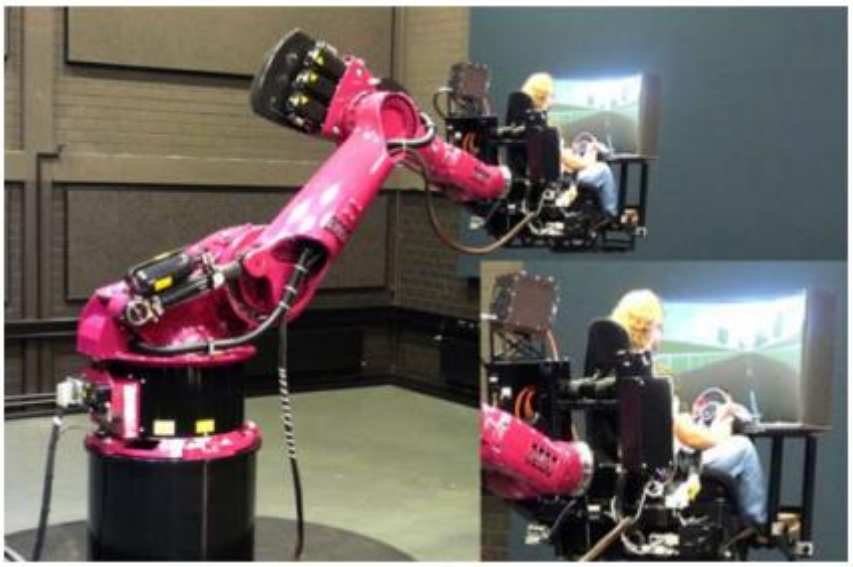
Şekil 1.7 DLR ve Desdemona

NADS tasarımından etkilenilerek, yanal 25 m ve uzunlamasına 35 m harekete sahip Higashi-Fuji Teknik Merkezi'nde devasa Toyota sürüş simülatörü üretilmiştir (Şekil 1.8).



Şekil 1.8 Toyota sürüş simülatörü

Seri bir robotun kullanımıyla da simülatör sistemi denenmiştir. Merkezkaç ivme odaklı bir çalışma olduğundan Stewart platformuna göre üstün olduğu iddia edilmiştir. Bu çalışmada eğik ekran sistemi kullanılmıştır. Kullanıcı ekrana 73 cm uzaklıktadır. 1152x450 pixel çözünürlükte 42 ms. gecikme ile görüntü sistemi kullanılmıştır. Görüntü sistemi için Max Plank Enstitüsü'nde geliştirilen MPI VeLib kullanılmıştır (Şekil 1.9).

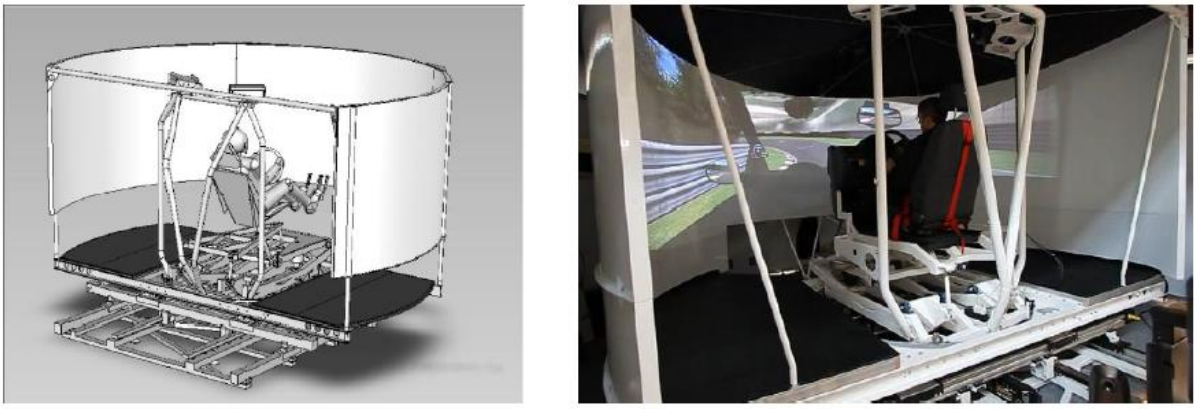


Şekil 1.9 Cyber Motion simülatörü

[17] çalışmasında doğrusal kuadratic optimal kontrolcü kullanılarak model tabanlı hareket algı kontrolcüsü ile ATMOS sürüş simülatörü üzerinde çalışılmıştır. Elde edilen sonuçlarda klasik yapıya sahip algoritmalara kıyasla daha gerçeğe uygun sonuçların elde edildiği belirtilmiştir. Kontrolcü yapısına sahip olduğundan ve simülatörün sınır durumlarını içerdiğinden simülatör kendi çalışma uzayını da tasarruflu kullanmaktadır.

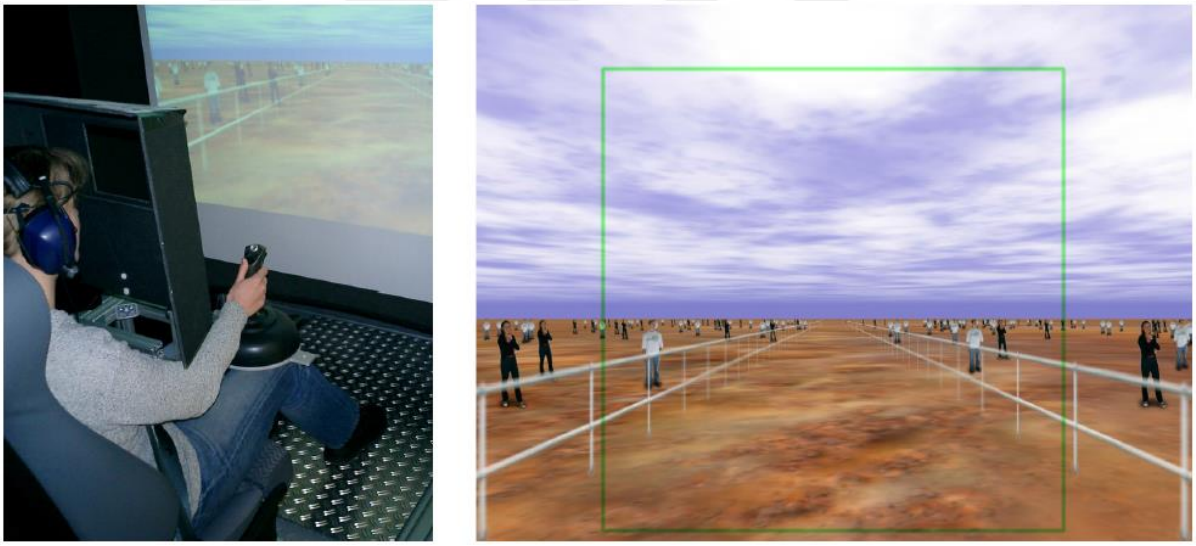
[18] çalışmasında sürüş simülatörleri için eğri sürüş sırasında oluşan hareket hatalarını azaltma üzerinde çalışmalar yapılmıştır. Bunu yaparken hem öteleme yönündeki hem de viraj bölgelerinde aracın yanal ekseni üzerindeki hatalar azaltılmaya çalışılmıştır.

[19] çalışmasında sürüş simülatörü üzerinde model öngörülü kontrol çalışması yapılmıştır. Model üzerinde insan algısı denklemleri kullanılmıştır. (Şekil 1.10) maliyet fonksiyonu çözücüsü olarak kuadratik programlama kullanılmıştır. Klasik algoritma ile sonuçları karşılaştırılmış ve model öngörülü kontrolcünün sonuçlarının gerçeğe çok daha yakın bir cevap verdiği tespit edilmiştir.



Şekil 1.10 Padova Üniversitesinde yapılan model öngörülü kontrollü sürüş simülatörü

[20] Çalışmasında, aracın ileri yöndeki ivmelenmesi sırasında platformun eşdeğer yöndeki ilerlemesi ve bu harekete bağlı eğim hareketinin şiddeti üzerinde çalışmalar yapılmıştır (Şekil 1.11)



Şekil 1.11 Max Planck Enstitüsünde yapılan simülatör çalışmaları [20]

Simülatörlerde hareket etkisi kadar, USY'nın etkisi ve ses düzeyi de önem taşımaktadır. [21] çalışmasında görsel etki ve insan hareket algısı bir arada kullanılarak sistemin modellemesi yapılmıştır. Yapılan denemelerin sonunda, simülatörün çalışması için ne kadarlık bir hareket sığasına sahip olması gerektiği tespit edilmiştir.

1.1.2 Hareket Algı Yazılımı Çalışmaları

Simülatörlerde kullanılan hareket algı yazılımının genel adı "Sıfırlama Filtre (SF)"dir. Burada "Sıfırlama" denmesinin sebebi platformun sınırlı hareket uzayında, hissiyatı sağlayacak kadar hareketi yaptıktan sonra tekrar sıfır pozisyonu olarak isimlendirilen başlangıç pozisyonuna dönmesi durumudur. Bunu yaparken basit filtre yapılarının yanı sıra, kontrol tekniği içeren algoritmalar da kullanılmaktadır. Literatürde yer alan algoritmaların temeli klasik algoritma yapısına dayanmaktadır. Bunun dışında yoğunluklu olarak uyarlamalı, optimal ve model öngörülü hareket algı algoritmaları da bulunmaktadır. Bu çalışmada öncelikle klasik hareket algı yazılımı (kHAY) kullanılacak, sonrasında model öngörülü hareket algoritması üzerinde çalışılacaktır.

1.1.2.1 Klasik Hareket Algı Algoritması

Klasik algoritmanın oluşumu ilk NASA Ames Araştırma Merkezinde yapılmıştır [22]. Algoritmada 1. ve 2. dereceden yüksek ve düşük geçiren filtreler bulunmaktadır. USY'den alınan açısal hız ve öteleme ivmeleri bu filtre yapılarından geçerek frekanslarına göre sinyaldeki genlikler üzerinde oynanmakta ve platformun hareketi sağlanmaktadır.

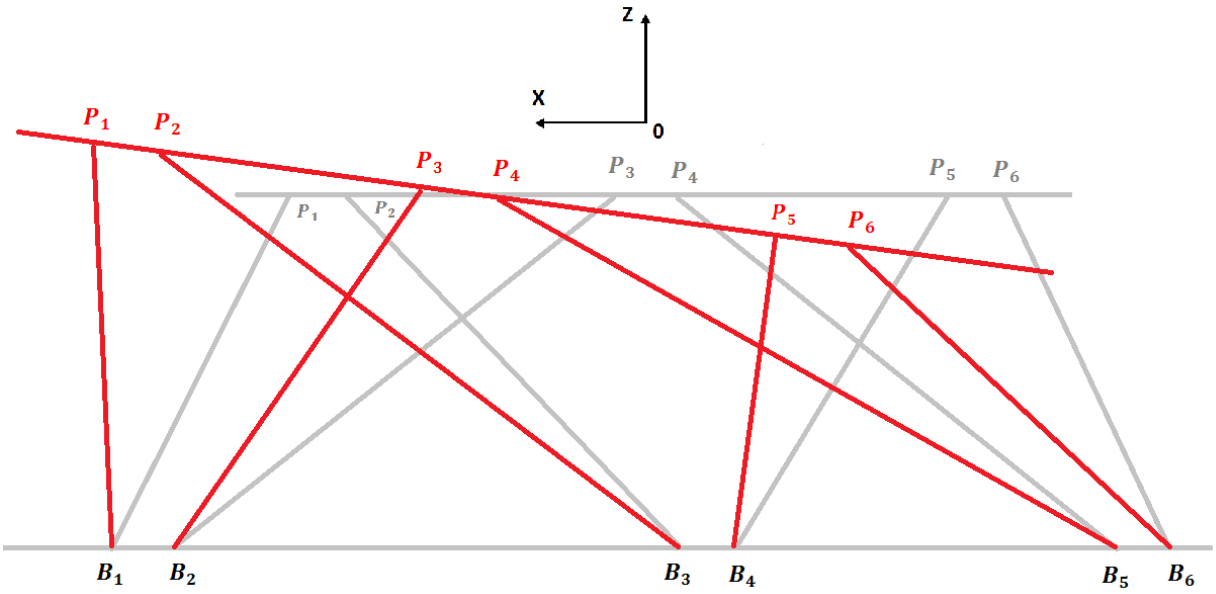
Özetle, algoritmanın genel özellikleri aşağıdaki maddeler ile açıklanabilir;

- Algoritma genel olarak yüksek ve alçak geçiren filtrelerin birleşiminden oluşmaktadır. Kesme frekansı ve sönüm değeri başta ayarlanır.
- İvme etkisi incelenirken dikey yöndeki ivme değişimine yerçekimi etkisinin de dâhil edilmesi önemli bir noktadır.
- Öteleme ivmeleri ve açısal hızlar öncelikle ölçeklendirilir. Öteleme ivmeleri de eğim etkisinin yaratılması için düşük geçiren filtreler ile filtrelenir ve sınırlandırılır. Eğim etkisinin oluşturulmasındaki en önemli etken ise yerçekiminin varlığıdır ve bu düşük frekans kuvvet işaretleriyle bağlantılıdır.
- Platformun sıfır pozisyonuna geri dönmesi için ikinci veya üçüncü dereceden filtreler kullanılabilir. Eğer ki etkiler ölçeklenmiş ve sınırlandırılmışsa, bu olayın gerçekleşmesi için sadece 2. Dereceden filtreler yeterli olacaktır [23].

Simülatörler için yapılan klasik filtre algoritma çalışmalarından yakın bir zaman sonra eğim koordinasyonu filtresi oluşturulmuştur [24]. Eğim koordinasyonu, uçuş esnasındaki öteleme

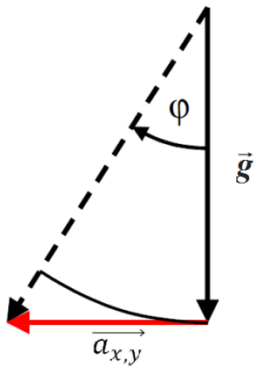
ivmelerinin platformda eğim hareketi ile pilota hissettirilmesini sağlamak amacıyla kullanılmaktadır. Bunun için, araçta ivme olduğu sürece bu ivmenin yerçekimine eşdeğer olacak şekilde platformun eğimi sağlanmaktadır (Şekil 1.12). Uçağın öteleme ivmesinin, yerçekimi ivmesine oranının tanjantı, bu ivme değerine eşdeğer olan eğim etkisini vermektedir. Klasik algoritmada eğim etkisi kanalında (1.1) denklemi kullanılacaktır.

$$\varphi_{x,y} = arctan\left(\frac{|\vec{a}_{x,y}|}{|\vec{g}|}\right) \tag{1.1}$$



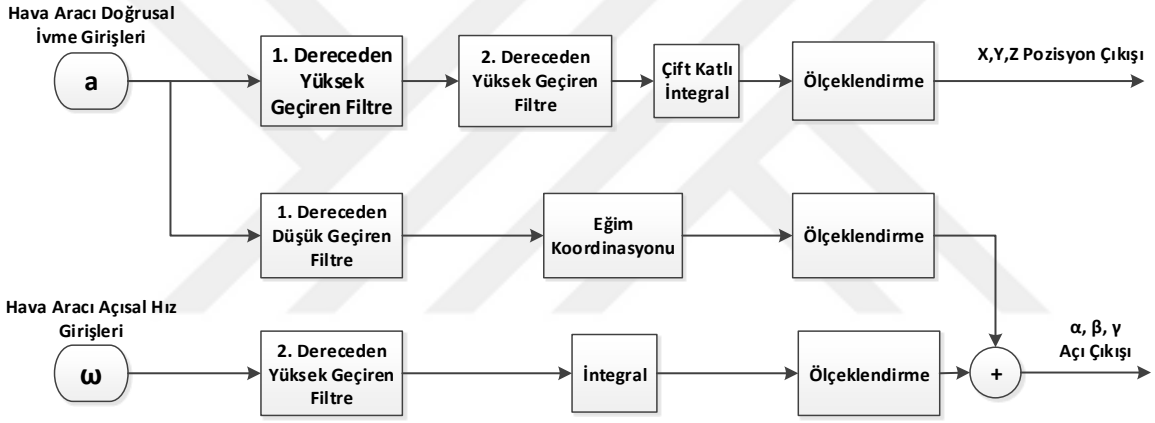
Şekil 1.12 Helikopterde x yönünde ivmelenirken platformda oluşan eğim.

Şekil 1.12'de P noktaları platformun üst bağlantı noktaları, B noktaları platformun alt bağlantı noktalarıdır. Hava aracı x yönünde ivmelenmektedir ve bu sırada platform bu ivme etkisini hissettirmek için platformu y ekseni etrafında eğmektedir. Bu sırada platform üzerinde Şekil 1.13'te anlatıldığı gibi -x yönü doğrultusunda bir yerçekimi etkisi oluşmaktadır.



Şekil 1.13 Platformda oluşan açı ile elde edilen eğim etkisi.

Filtrelerin çıkışlarından sonra elde edilen filtrelenmiş değerler öteleme ivmeleri için çift integral, açısal hızlar için tek integral alınarak platform kontrolcüsüne platformun koordinat bilgileri gönderilmektedir. Klasik algoritmanın akış diyagramı Şekil 1.14'te gösterilmiştir.



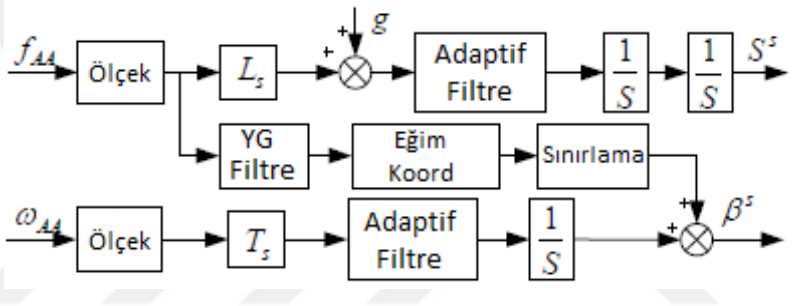
Şekil 1.14 Klasik algoritmanın akış diyagramı.

Klasik algoritmanın dezavantajı, sistemde lineer elementler kullandığından simülatörün kapasitesini tam olarak kullanamamasıdır. Doğrusal olmayan işaretlerin olmasından dolayı insan algısında da kısıtlı bir durum söz konusu olmaktadır. Bu yüzden klasik algoritmanın geliştirilip kötü şartlar altındaki uçuş durumlarının da algoritma hesabına katılması gerekir. Bundan ötürü farklı algoritma tiplerine ihtiyaç duyulmaktadır.

Literatürde 3,6 ve 8 serbestlik dereceli (DOF) platformlar üzerinde de klasik algoritmanın çalışmaları ve aralarındaki performans karşılaştırmaları yapılmıştır. 3 DOF platformlarda bu sistemin öteleme hareketinin geri beslemesi oldukça iyi olmuştur. 6 DOF platformlarda ise hareket serbestliği daha iyi olduğundan geniş hareket işaretleri daha yoğun hissedilebilmiştir. 8 DOF platformlarda da yine 6 DOF platforma benzer performans etkileri gözlemlenmiştir [2].

1.1.2.2 Uyarlamalı Hareket Algı Algoritması

Klasik algoritmanın kullanımı platform sınırları için yetersiz olmaktadır. Bu sorunu ortadan kaldırmak için Parrish, Dieudonne, Bowles ve Martin [25] uyarlamalı algoritmayı geliştirmişlerdir. Uyarlamalı algoritmanın avantajı, maksimum aktüatör uzantılarında bulunan yanlış işaretlerin etkisiz hale getirilmesi ve pürüzsüz bir simülasyonun oluşmasını sağlamaktır. Yeni algoritma, klasik algoritmadan türetilmiştir. Her örnekleme zamanında ayarlarını maliyet fonksiyonunu azaltmak için ayarlamaktadır. Klasik algoritmadan çok daha iyi bir çözüm olmasıyla beraber, yaklaşım yapma ve maliyet fonksiyonunu azaltma durumundan dolayı algoritmanın sistemdeki yükü de yüksektir. Ayrıca sistemde herhangi bir sınırlama prosedürü uygulanmadığından platform sınırlarına ulaşma durumunda hissiyattaki kayıplar önlenemeyebilir. Uyarlamalı algoritma ile ilgili bir şematik görsel Şekil 1.15'te gösterilmiştir.



Şekil 1.15 Uyarlamalı algoritmanın akış şeması [26].

Pilotta oluşan görsel etki ve sesteki 3 boyutlu etkileşimin insan algısı üzerindeki etkisi oldukça fazladır. Fakat platformun hareketlerinin de insan algısı için uyumlu olması gerekmektedir. İnsanın baş bölgesindeki hareketin hissedilmesinde iç kulakta bulunan yarı dairesel kanallar ve otolit taşları görev almaktadır. Bu organların matematiksel olarak modellenmesi, yapının platformun kontrol sistemine eklenmesine olanak sağlamaktadır. Bu şekilde, platformun hareketi insan algı modeli temel alınarak kontrol edilir ve hem çalışma uzayı içerisinde bir hareket sağlanır, hem de görsel ve ses etkisiyle beraber hareket de insan algısıyla uyumlu hale getirilir. Bunun için Optimal ve Model Öngörülü hareket algı algoritması çalışmaları yapılmıştır. Pratik çalışmalarda yer alan model öngörülü kontrole ait denklem yapıları da literatürlerde bulunan modellerin birleşimi ile oluşturulmuştur.

1.1.2.3 İnsan Hareket Algı Sistemi, Yapısı ve Çalışma Prensibi

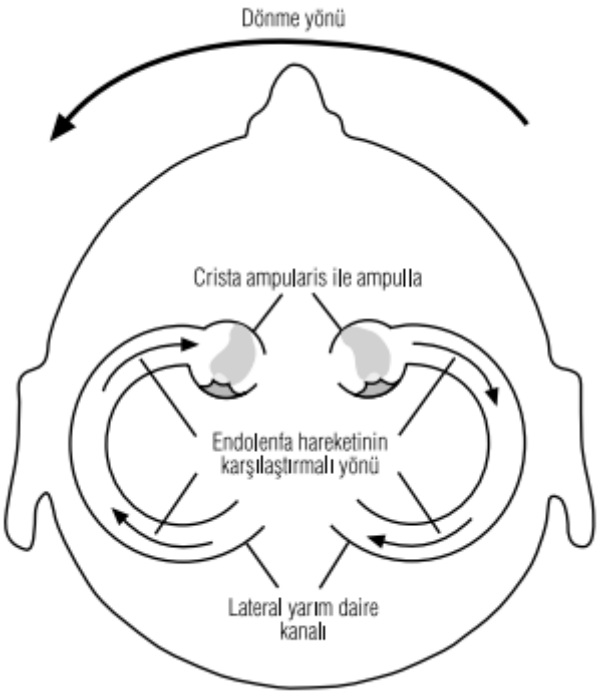
Optimal ve Model Öngörülü Kontrol çalışmalarından önce hareket algı sisteminin nasıl bir yapıya sahip olduğu bilinmesi gerekmektedir. Hareket algı sisteminin, insan kulağının iç kısmında yer almakta ve yarı dairesel kanal ile otolit taşlarının birleşiminden oluştuğunu önceki başlıklarda belirtmiştik. Bu yapı, hareket algısı konusunda insan vücudundaki en etkili yapıdır. Bu başlık altında, insan algısına etki eden önemli faktörler anlatılacaktır.

İnsanın görsel hareket algısı, ivmelenmeden çok hızlanmaya yönelik ayarlanmıştır. VS içinde, kırbacık ve kesecik ikilisiyle birlikte otolit organlarını oluşturan, saç teli şekline sahip küçük keseler vardır. İnsan kafası, yerçekimine göreli harekette bulunduğunda veya ivmelendiğinde kıllar bükülür ve merkezi sinir sistemine bilgi gönderir. Otolitler ile hem doğrusal hızlanma hem de eğim etkisi aynı anda gerçekleşir.

1.1.2.3.1 Yarı Dairesel Kanallar

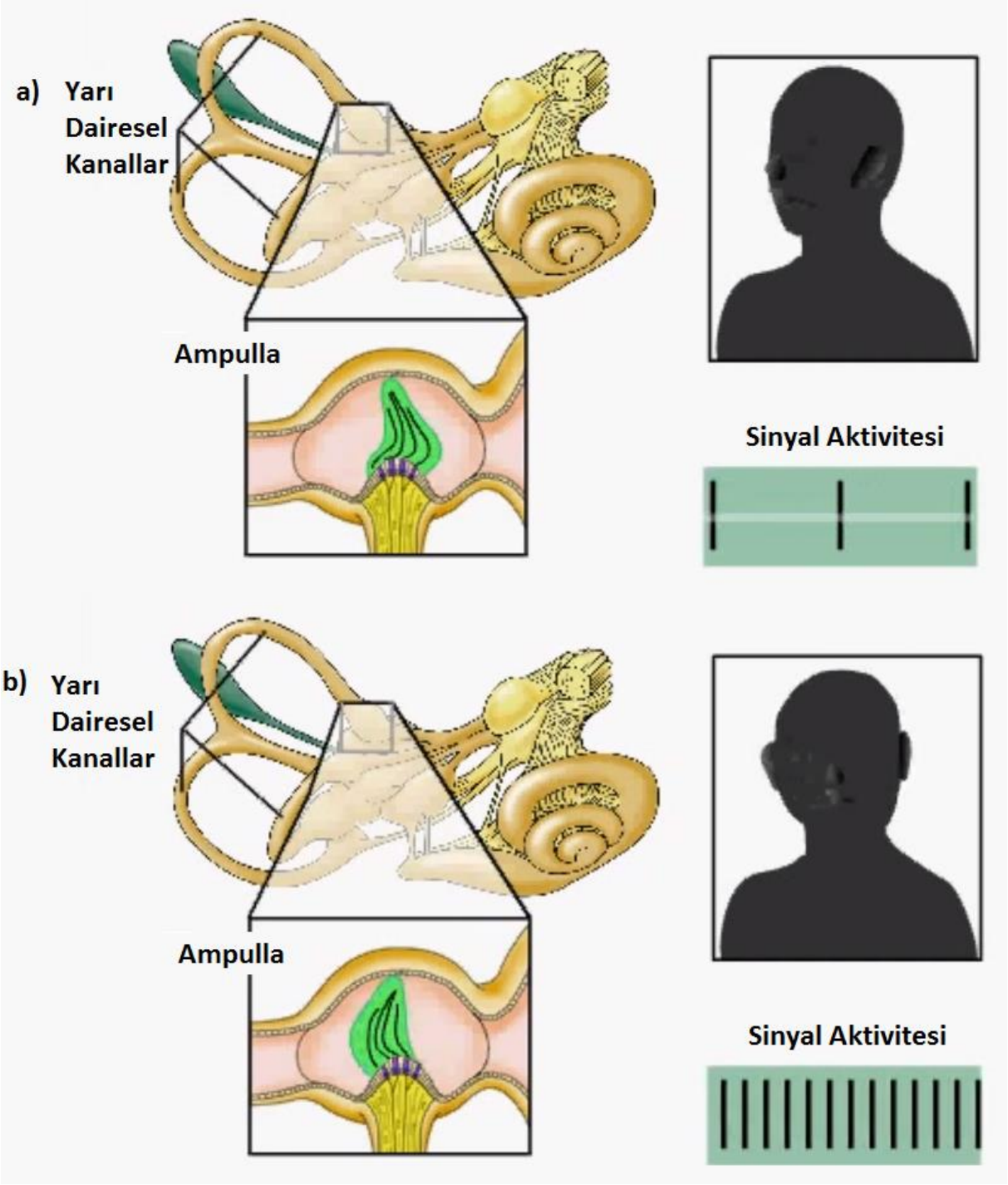
Yarı dairesel kanallar kişinin üzerinde oluşan açısal etkilerin hissedilmesinde etkilidir. Yapı, 3 adet yarı daire şeklinde uzantılara sahiptir ve hareket esnasında bu uzantılar eksenel hareket yaparak dönme eğilimi gösterir (Şekil 1.16).

Ampulla, krista ve kupula başın açısal hızlanma ya da yavaşlama değişimlerine yanıt verir. Buradaki mekanizmalara etki eden yapısal özellikler yarım daire kanallarındaki simetrik yapı ve ortogonal organizasyon ile kanal hareketleriyle endolenfa akımı arasındaki ivme farkıdır. İnsan başına dik koordinat ekseni üzerinden bakacak olursak, saat yönünün tersinde bir hareket olması durumunda ampulla yapısı içinde bulunan kupula cisimciği içeride oluşan akıştan dolayı bir yönelme gösterir. Saat yönünde dönmesi durumunda ise bunun tam tersi geçerlidir. Yavaşlama sırasında ise hızlanmanın tersi olaylar gelişir.



Şekil 1.16 Yarı dairesel kanalların hareket sırasındaki davranışı

Yarı dairesel kanallardaki sinyal akışlarını daha iyi anlatabilmek 3 farklı durum Şekil 1.17'de sırasıyla gösterilmiştir. Burada, kişi saat yönünde hareket ettiği sırada kanallar içerisinde bulunan endolenfa akımı kupulayı tersi yönde bükmektedir. Bu sırada sinyal aktivitesinde azalma meydana gelmektedir. Kişi saat yönünün tersinde döndüğü sırada ise bu durumun tam tersi gerçekleşmekte ve sinyal aktivitesinde yükselme olmaktadır. Böylelikle insan beyni bu hareketlerin oluşturduğu sinyal etkisiyle kişinin hangi yönde harekette bulunduğunu anlamaktadır.



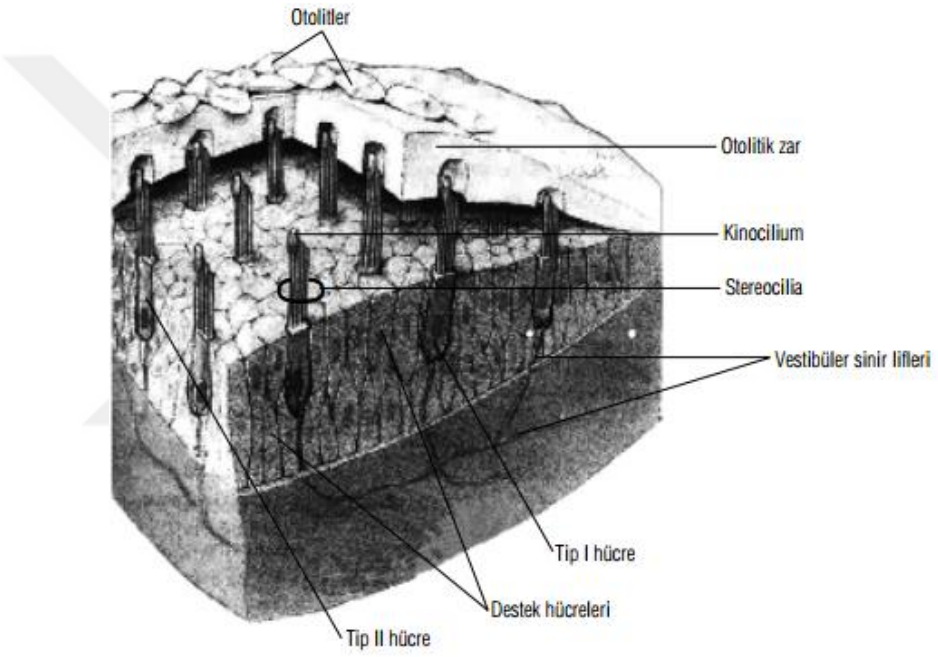
Şekil 1.17 Yarı dairesel kanalların çalışması sırasında oluşan sinyal değişimleri a) insan başı saat yönünde döndüğü sırada, b) İnsan başı saat yönünün tersinde döndüğü sırada.

1.1.2.3.2 Otolit Taşları

Otolit taşları yapısı, insanın doğrusal yöndeki hareketleri algılanmasında görevlidir. Şekil 1.18'de görsel olarak anlatılmıştır.

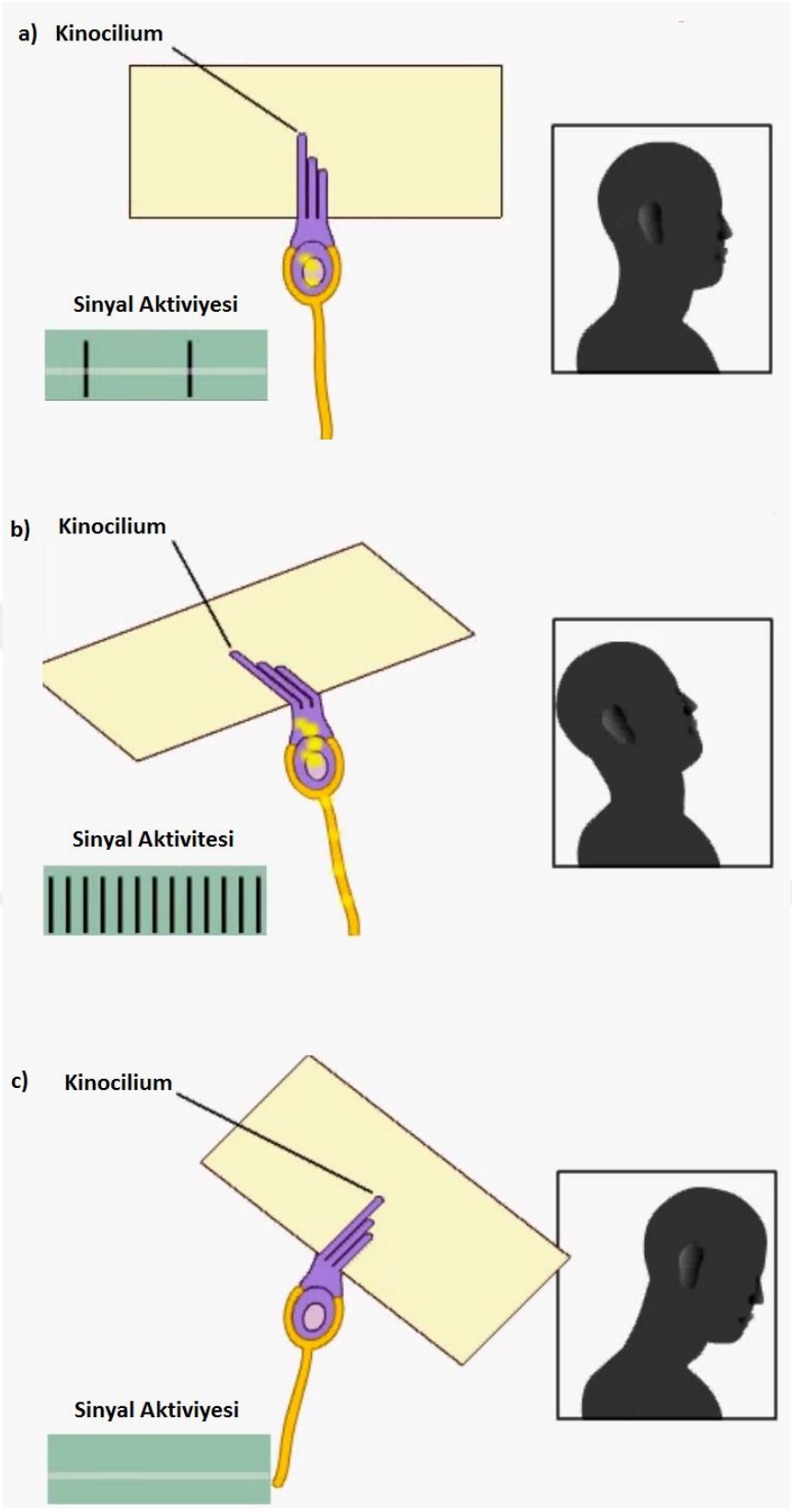
Utriculus ve sacculus'ta bulunan ve taban bölgesine uzanan stereocilia, endolenfa içinde bulunan ve otolitik zar adı verilen bir kütle içine gömülüdür. Bu yapı otolit isimli kalsiyum karbonat kristalleri ile çevrili yassı bir maddeden oluşur. Özgül ağırlığı fazla olduğundan yerçekimi etkisine açıktır.

Stereocilia ucunda bulunan kinocilium isimli yapılar, insan başındaki harekete göre bükülmektedir. Bu bükülme ile uç bölgesinde bulunan sinirsel yapı ile beyne sinyaller gönderilmektedir ve insan, öteleme yönündeki hareketlerin şiddetini bu şekilde algılayabilmektedir. Makülla isimli yapı içerisinde bulunan bu elemanların gösterimi Şekil 1.18'da gösterilmektedir.



Şekil 1.18 Makülanın yapısı.

İnsan başı dik konumdayken, beyne ulaşan beyin sinyalleri Şekil 1.19a'da gösterilmiştir. Kişinin ivmelenme sırasında Stereocilia cisminin ucunda bulunan Kinocilium isimli yapı eğim almaya başlar. Bu sırada cisim üzerinde oluşan sinyallerde artış olmaktadır. Sinyal artışıyla beraber insana yüz önü istikametindeki ivme hareketi hissettirilir (Şekil 1.19b). Benzer şekilde kişi sırt yönüne doğru ivmelendiğinde Kinocilium yapısı öne doğru pozisyon alır ve sinyallerde ciddi anlamda azalma meydana gelir (Şekil 1.19c). Beyin bu sinyal akışındaki azalmayı kişinin sırt yönünde ivmelenmesi şeklinde algılamaktadır. Benzer sinyal akışları, kişinin yanal hareketinde de olmaktadır.



Şekil 1.19 Otolit taşları sinyal düzeyleri a) insan başı dik konumda, b) insan başı yüz istikametinde ivmelenirken, c) insan yüz istikametinin tersi yönünde hareket ederken

İnsan algısı, kişiden kişiye değiştiğinden net bir model oluşturulamamaktadır. Bu yüzden yapılan araştırmalarda yaklaşım metodu ile ortalama VS matematiksel modelleri oluşturulmaktadır.

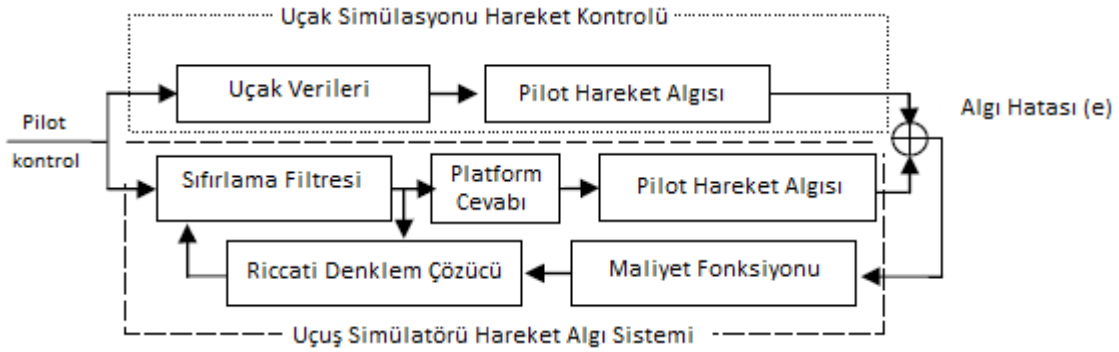
Bu anlatılan insan algısı modeli kullanılarak literatürdeki çalışmalarda optimal, model öngörülü ve daha bir çok farklı prensibe sahip kontrol algoritmaları kullanılmıştır. Yoğunluklu olarak bu iki algoritma kullanıldığından literatür içerisinde bunlardan bahsedilecektir ve belirlenen bir algı modeli ile platformun kontrolü tamamlanıp algı sonuçları ve platformun hareket durumları elde edilecektir.

1.1.2.4 Optimal Kontrollü Hareket Algı Algoritması Çalışmaları

Literatürdeki optimal algoritma çalışmalarının yapısında bir önceki bölümde anlatılan hareket algı sisteminin etkisi göz önünde bulundurulmuştur. İnsan algısına uygun bir şekilde hareket algı yapısı geliştirilmiştir.

Yarı dairesel kanallar giriş sinyali olarak baş bölgesinin açısal hız verisini kullanır, çıkış parametresi olarak hissedilen açısal hız cevabını verir. Otolit taşları ise sistemin özelleştirilmiş kuvvet (ivme) bilgisinin algılanmasında rol oynar ve çıkış parametresi olarak algılanan ivme hissini verir. Bu kontrol sistemi hem referans, hem de platformdaki pilotun algıladığı his çıkışları arasındaki farkı kullanarak algıda oluşan hatanın azaltılmasında rol oynamaktadır. Bunu yaparken, belirlenen bir maliyet fonksiyonunun değerinin düşürülmesini sağlayarak optimal SF'lerinin oluşmasını sağlar.

Bu algoritmada Riccati denklemi gerçek zamanlı hesaplanır. Optimal algoritmadaki filtreler için en büyük problem optimal filtrelerin tespit edilmesidir. Bu filtreler istenen platform konumunun elde edilmesinde kullanılır ve pilotun his hatasını düşürür.



Şekil 1.20 Optimal hareket algı algoritması yazılımı [27]

[27] Çalışmasından örnekle, Şekil 1.20'de gösterilen modelde pilot kontrolü aracılığı ile sanal ortamdaki hava aracının ivme ve açısal hız cevabı elde edilir. Elde edilen bu cevaplar vestibüler sistem girdisi (VSg) olarak VS matematiksel modelinden geçirilir ve sanal ortamdaki pilotun algı modeli elde edilir. Bunun haricinde, karşılaştırma yapmak için teorikte veya pratik yolla simülatör üzerindeki pilotun baş bölgesindeki öteleme ivmeleri ve açısal hız bilgisi, VS transfer fonksiyonlarından geçirilir. Hem referans hem de pratikteki değerler arasındaki fark algı hatasını oluşturur. Bu algı hatası, maliyet fonksiyonundan sonra Ricatti denklem çözücüsünden geçirilir ve optimal SF'ler elde edilerek sisteme eklenir. Bu filtreler sayesinde sistemin optimal kontrolü yapılmış olunur. Model öngörülü algoritma da benzer bir yapıya sahip olup Ricatti denklem çözücüsü yerine kuadratik programlama yöntemi kullanılır.

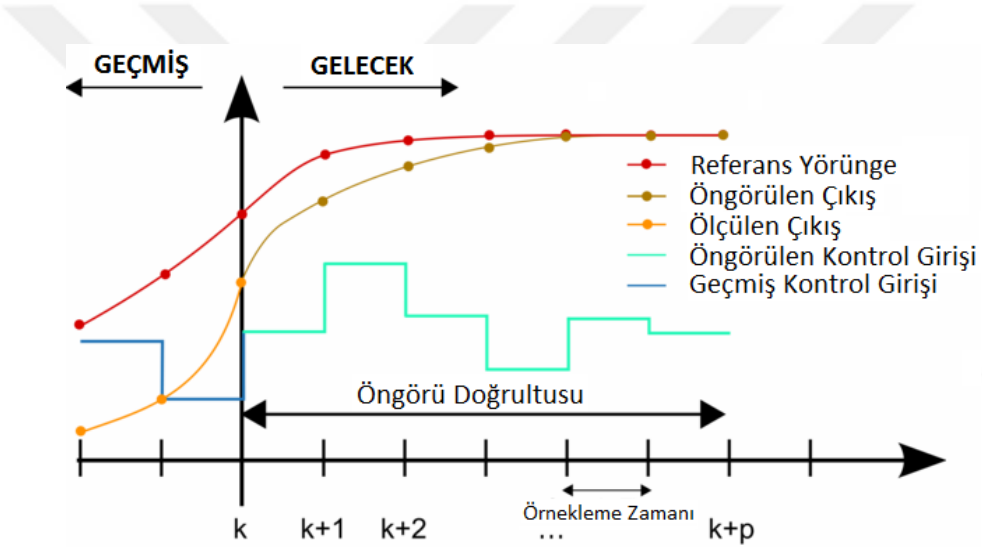
1.1.2.5 Model Öngörülü Hareket Algı Yazılımı Çalışmaları

İnsan VS'i, transfer fonksiyon olarak modellenmesi durumunda yapısı yüksek ve düşük geçiren filtrelere benzediği görülmektedir. Bu durumdan dolayı akla ilk olarak klasik algoritma yapısının aslında VS yapısı örnek alınarak oluşturulduğu gelmektedir. Fakat basit bir yapıya sahip olduğundan matematiksel formu üzerinde oynamalar yapılmasına olanak sağlamaktaydı. Bu durum insan algısının tam olarak ölçülmeden, platform hareketinin sadece yaklaşım ile ve çalışma uzayı içerisinde kalabilecek şekilde olmasını sağlamaktadır.

Literatürdeki kısaltma adı MPC (Model Predictive Control) olan model öngörülü kontrol sisteme çoklu giriş - çoklu çıkış özelliğini ekler. Lineer olmayan modellerin de çözümlenmesine olanak sağlar. Model öngörülü hareket algı yazılımında (MOHAY), optimal kontrolden farklı olarak kontrol nesnelerinin sistemin belirlenen bir öngörü düzlemi üzerinde

tahmin yapmasına olanak sağlar. Böylelikle sistem ne kadar ilerisini öngörürse, platform da hareket anındaki örnekleme zamanında o kadar doğru bir cevap verir. Bu hareket stratejisi aynı zamanda sistemin sınırlamaları da algoritmaya tanıtılarak yapılabilmektedir. Maliyet fonksiyonu, hesaplamalar sonrasında "kuadratik yapı" denilen bir denklem yapısına dönüşmektedir. Bu yapı normal bir maliyet fonksiyonundan farklı olmayıp, sadece çözüm yöntemi olarak farklılık göstermektedir. Sistemin öngördüğü bütün durum ve kontrol girişleri de hesaplandığından, çözümü de bir o kadar karışık ve yoğun olmaktadır [28].

Kontrolcüde platformun müsaade edilebilen en yüksek çalışma alanı kullanılması amaç edilmiştir. MPC yatay düzlem kontrolünü kullanır. Her sistem basamağında maliyet fonksiyonunu minimize etmek için yatay düzlem üzerinde bir kontrol dizisi hesaplar.



Şekil 1.21 Model öngörülü kontrolün çalışma prensibi.

Şekil 1.21'de gösterildiği gibi, sistem sıfır anında herhangi bir öngörü durumunda bulunmamaktadır. Çalışma başladığı anda ilk hata değeri elde edilip platformun kontrol algoritmasına gönderilmektedir. Bu hata ile ilk hesaplamada belirtilen örnekleme zamanı sayısı kadar platform için kontrol girişi ve sistem çıkışı öngörülerinde bulunur. k. örnekte öngörülen kontrol girişlerinin k+1'deki değeri platforma gönderilir ve bu hareket sonucunda hesaplanan sistem çıkışına bakılır. Sistem çıkışı ile referans arasındaki hata ölçülerek tekrardan maliyet fonksiyonuna gönderilir ve sisteme gönderilecek olan yeni kontrol giriş değişimleri ile öngörülen sistem çıkışları hesaplanır. Bu döngü her örnekte tekrarlanır.

Sistemin hareketlerinin sınırlandırılması veya genlikleri üzerinde oynanabilmesi durumu MOHAY'ın avantajlı özelliklerinden birisidir. [29] çalışmasında Renault'un simülatörü üzerinde öteleme hareketi ve beraberindeki eğim hareketi üzerinde çalışılmıştır. Karmaşık sınır koşulları altında sağlıklı bir hareketin yapılması amaçlanmıştır. Yapılan çalışma sonunda MOHAY'ın, klasik yapıdaki bir algoritmaya kıyasla çok daha iyi sonuçlar verdiği gözlemlendiği belirtilmiştir.

Bu projede sınırlamaların sisteme eklenmesi sırasında Hildreth'in kuadratik programlama çözücüsü kullanılacaktır. Sistemin bütün denklemleri ve parametrelerinin ne olduğu, matematiksel modelin oluşumu sırasında anlatılacaktır. Bu denklemlerin oluşumu sırasında Chalmer Üniversitesi'nin [28] çalışması temel alınmış ve model öngörülü kontrol çalışması ile ilgili [30] çalışmasından faydalanılmıştır.

1.2 Tezin Amacı

Literatürde uçuş ve sürüş simülatörleri için birçok algı algoritması oluşturulmasına rağmen, model öngörülü algı algoritması geliştirilmesi konusunda eksikler bulunmaktadır. Özellikle uçuş simülatörlerinde bu kontrol sistemi farklı yöntemlerle uygulanmaya çalışılmıştır. Bu tezde amaç, öncelikle simülatörlerde hareket yazılımı mimarisinin temelini oluşturan klasik sıfırlamalı algoritmanın oluşturulması, ardından model öngörülü hareket algı algoritmasının oluşturulmasıdır. Her iki algoritmanın sonuçları gerçek sistem üzerinde karşılaştırılıp hangisinin insan algısına daha uygun olduğu belirlenecektir.

1.3 Hipotez

Bu çalışmada model öngörülü algoritma, simülatörün durum değişkenleri ve kontrol girişlerinin devam eden hareketler için öngörülmesi ve bu öngörülen verilerin sonraki örnekleme zamanlarında iyileştirilerek sistemin kontrol edilmesi esasına dayanır. Genel olarak, kontrolcünün etkin çalışabilmesi için, platforma gönderilen helikopterin açısal hız ve öteleme ivme verilerinin önceden belirlenmesi gerekir. Bu çalışmada sabit referans değerleri kullanılarak platformun model öngörülü algoritma ile çalıştırılması sağlanmıştır. İnsan algısı ile beraber, platformun hareketini tamamladıktan sonra başlangıç pozisyonuna dönmesi gerekmektedir. Bu da yine kontrolcü aracılığı ile yapılmıştır. Algoritma oldukça gelişmiş bir

yapıya sahip olduğu kadar, yoğun matematiksel işlemler içeren iyileştirme sistemine de sahiptir. Dolayısıyla çözüm yönteminin basitleştirilmesi üzerinde de çalışılmıştır.

BÖLÜM 2

SİMÜLATÖRÜN MEKANİK VE ELEKTRONİK YAPISI

Simülatör 6 bacaktan oluşmakta ve her bir bacak için üst platformda 6, alt platformda 6 adet farklı bağlantı noktaları bulunmaktadır (Şekil 2.1). Yapı, paralel mekanizma olup her bacağın hareketi diğer bacakların durumunu da etkilemektedir. Kapasitesi 1.5 ton olup 0563.STZ.20132 Santez projesi kapsamında üretilmiştir. Tahrik mekanizmasında 6 adet AC servo motor kullanılmıştır. Platformun hareket komutları, sürücülerle beraber kontrolcü sistemi tarafından yönetilmektedir. Bu başlık altında simülatörün mekanik ve elektronik elemanları ile ilgili bilgiler verilecektir.



Şekil 2.1 Uçuş simülatörü genel gösterimi.

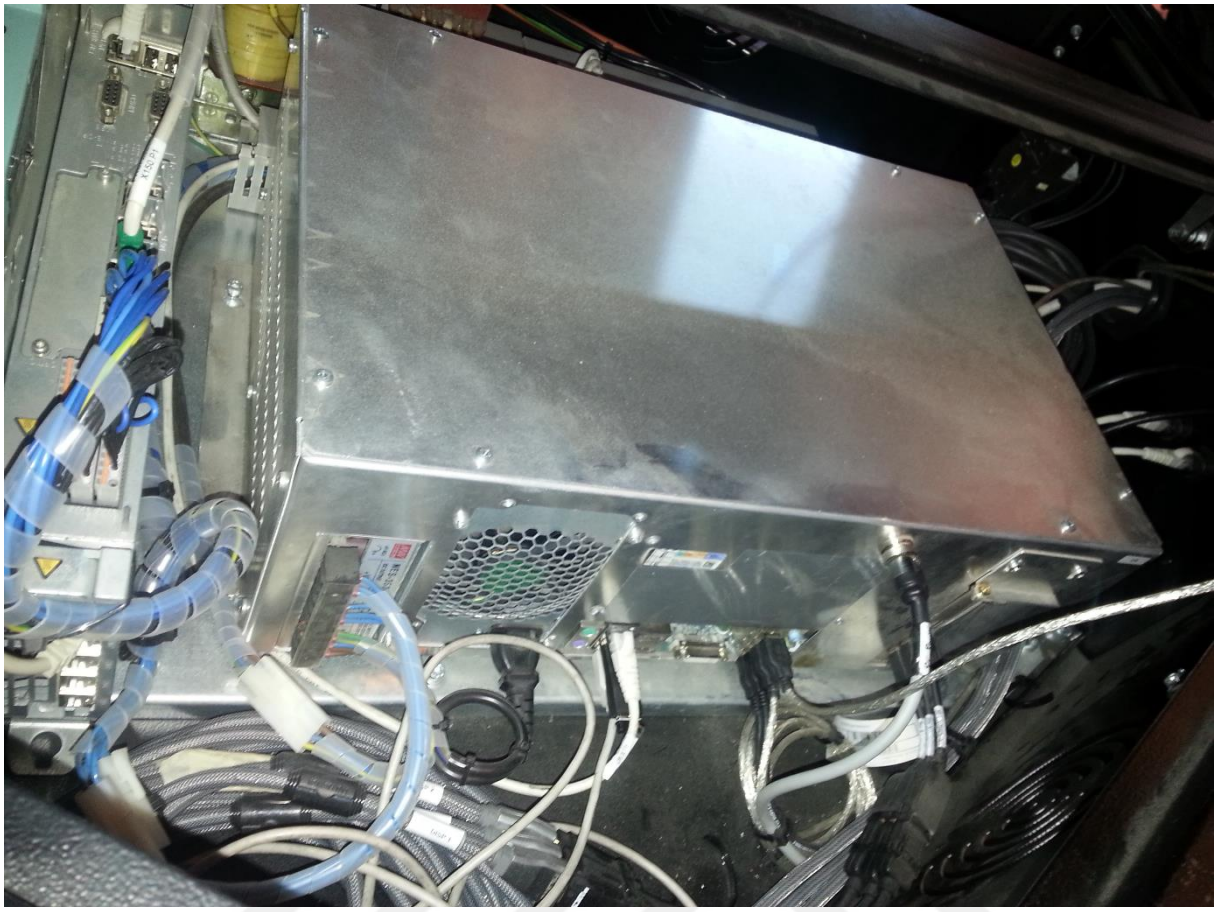
2.1 Simülatörün Mekanik Yapısı

Uçuş simülasyonu sırasında, pilot tarafından kumanda kolu (KK) ile gönderilen yönelim emirleri, USY'ye gönderilmektedir. Helikopterden alınacak hareket verileri Siemens hareket kontrolcüsü aracılığı ile MATLAB modeline aktarılmaktadır. Hareket kontrolcüsü üç motor sürücüsüne hareket emirlerini göndermektedir. Her sürücü 2 farklı bacağı sürmektedir. Platformun ayaklarının hem zemine, hem de üst platforma olan bağlantıları her bacak için 2'şer adet kardan kavraması ile yapılmıştır. Bacakların toplam stroku 570 mm'dir (Şekil 2.2).



Şekil 2.2 Platformun hareketli pistonları

Platformda 2 adet bilgisayar kullanılmıştır. Biri USY'nin koşturulduğu ve platformun alt bölümünde, kontrolcülerin yanında yer alan HP Z230 model iş istasyonu (Şekil 2.3), ikincisi ise sistemin hareket algı algoritmasını içeren MATLAB yazılımının koşturulduğu ve sistemin dış bölümünde bulunan kontrolcü bilgisayarıdır.



Şekil 2.3 USY bilgisayarı.

Uçuş benzetiminin gerçekçi yapılabilmesi için, uçak ve helikopterlerde bulunan KK ile pedal elemanları eklenmiştir (Şekil 2.4). KK ve hız koluna ait düğmelerin ve kumanda fonksiyonlarının özellikleri;

- 3D manyetik sensör,
- 16 bit kumanda kolu çözünürlüğü ile 19 aktif buton,
- 14 bit gaz kelebeği çözünürlüğü ile 17 aktif buton,
- USB bağlantılı,
- 2 adet gaz kolu.

Pedalların özellikleri;

- Pedallar direnç etkisi gösteren ayarlı sisteme sahiptir,
- Pedallar uçağın sapma hareketini gerçekleştirebilmek için ileri-geri yönde hareket etmektedir,

- Uçağın frenlenmesi için pedallara dik yönde basılması gerekmektedir.



Şekil 2.4 Kumanda kolu, hız kolu ve pedallar.

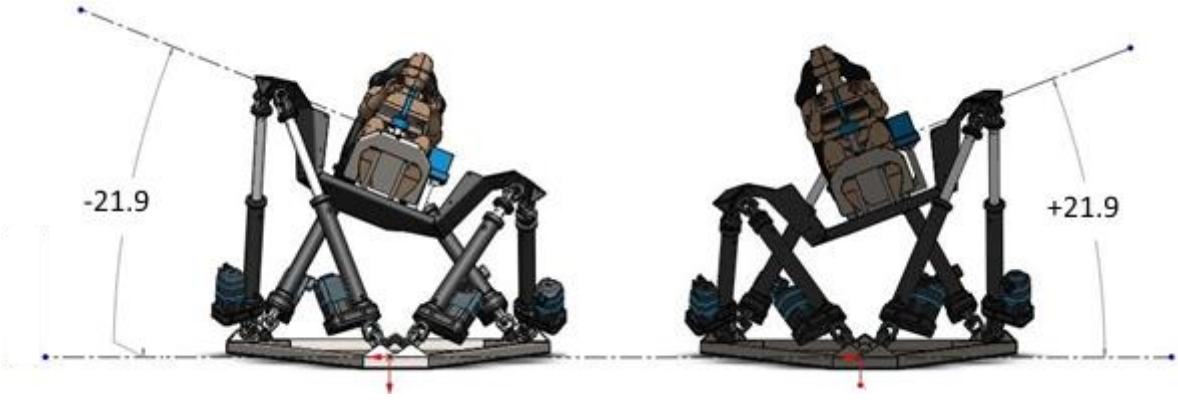
Platformun çalışma uzayı sınırları Çizelge 2.1'de gösterildiği gibidir. Şekilsel gösterimi Şekil 2.5, Şekil 2.6 ve Şekil 2.7'de gösterilmiştir. Hem kHAY'da, hem de MOHAY'da bu sınırlamalar kullanılacaktır. kHAY'da sınırlamalar MATLAB'de bulunan sınırlama bloğu kullanılarak, MOHAY'da ise iyileştirme fonksiyonunun bir parçası olan sınırlama denklemi kullanılarak yapılacaktır. Sınırlamalarda amaç, platformun çalışma uzayı içerisinde güvenli bir şekilde çalışması ve duraklamalardan veya ani hareketlerden dolayı oluşabilecek his hatasını azaltmaktır. İhtiyaç duyulması hâlinde sınırlar içerisinde kalınması için platforma gönderilen veriler ölçeklenecektir.

Çizelge 2.1 Platformun hareket limitleri.

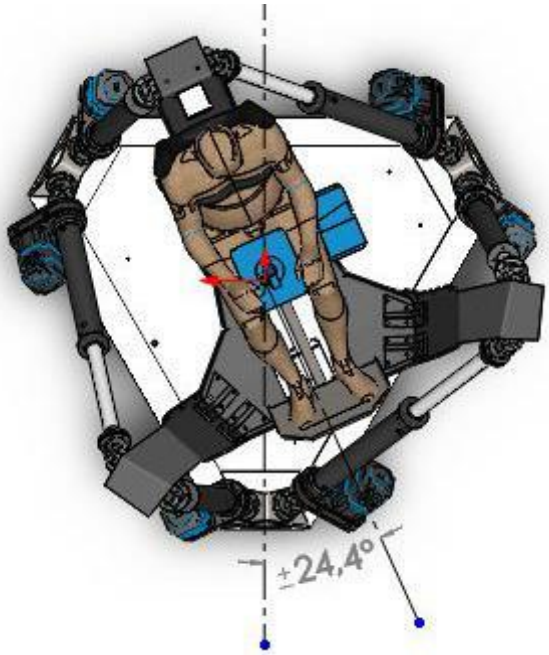
		Alt Sınırlar (mm)	Üst Sınırlar (mm)
Öteleme	**x Ekseni**	-605	425
	y Ekseni	-460	460
	z Ekseni	-359	321
		Alt Sınırlar (deg)	**Üst Sınırlar (deg)**
Açısal	**Yunuslama**	-21.5	21.5
	Yuvarlanma	-21.9	21.9
	Sapma	-24.4	24.4



Şekil 2.5 Yunuslama (Pitch) hareketinin açısal limiti.



Şekil 2.6 Yuvarlanma (Roll) hareketinin açısal limiti.



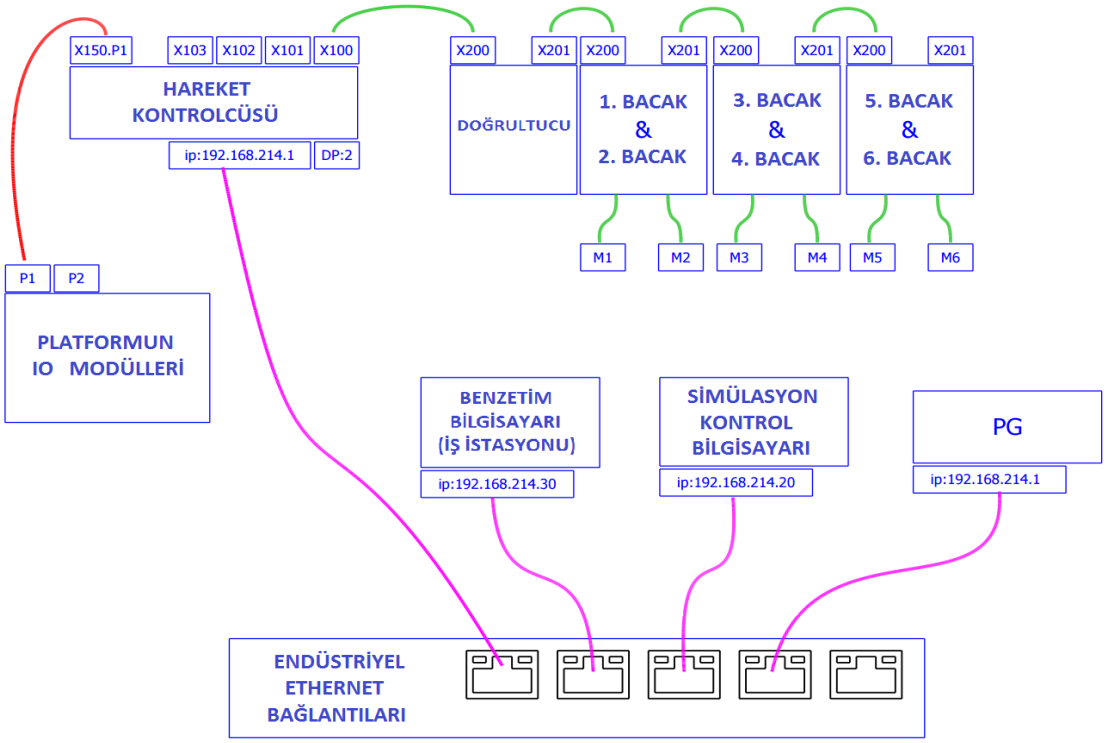
Şekil 2.7 Sapma (Yaw) hareketinin açısal limiti.

2.2 Simülatörün Elektronik Yapısı

Bütün elektronik elemanlar endüstriyel ağ bağlantısı modülü üzerinden haberleşmektedir. Hareket kontrolcüsü, sürücü sistemine bağlıdır. Sürücü sisteminde her sürücü iki bacağı tahrik etmekle görevlidir. Kontrolcüye platformun merkez pozisyonu olarak 3 doğrusal ve 3 açısal konum gönderilir ve kontrolcü içerisinde bu veriler platformun ters kinematik analizi yapılarak bacak boylarına çevrilmektedir. Elde edilen bacak boyu sistemden takip edilebilmektedir. Kontrolcülerin resimleri Şekil 2.8'de gösterilmiştir. Elektronik sistemin genel şeması ise Şekil 2.9'da gösterilmiştir.



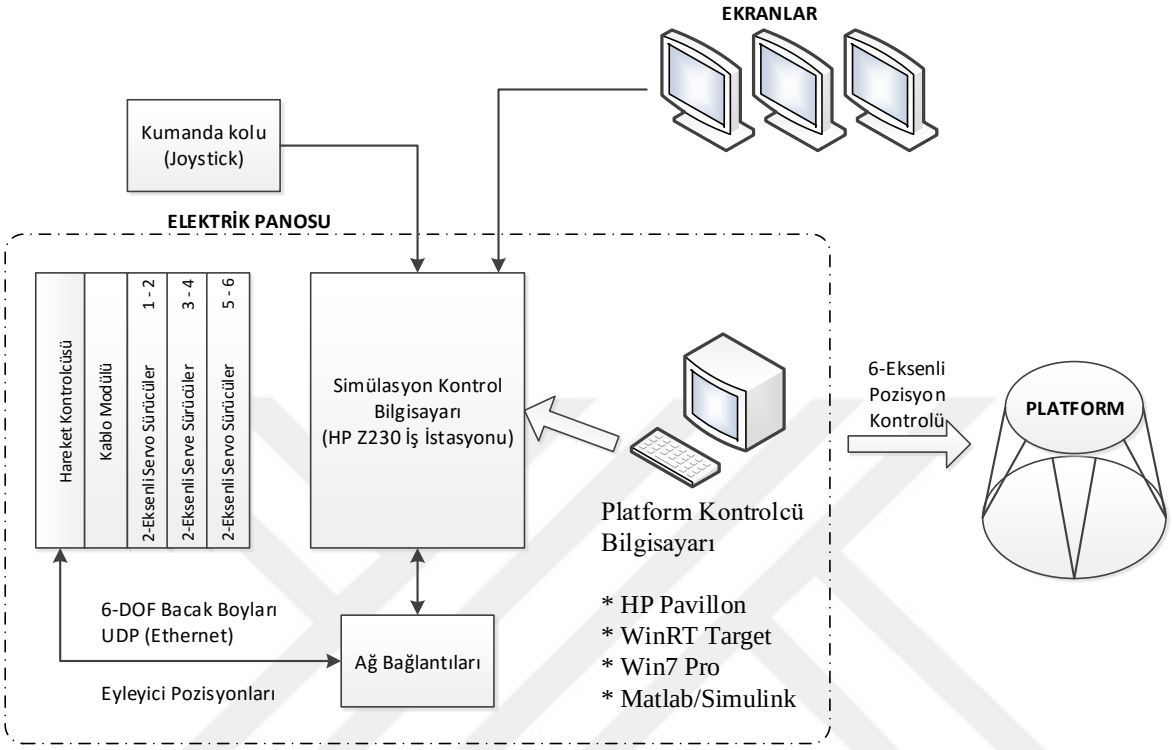
Şekil 2.8 Bacakların tahrikini sağlayan Siemens kontrolcüleri.



Şekil 2.9 Simülatör sisteminin elektronik yapısı.

2.3 Uçuş Simülatörü Genel Şeması

Platform, kontrolcü ve görüntü sistemi ile diğer sistem parçalarının birleşimiyle oluşan genel sistem şeması Şekil 2.10'daki gibidir.



Şekil 2.10 Genel sistemin şeması.

Uçuş kontrolcüsü KK olup kullanıcı tarafından sisteme gönderilen ilk sinyal bilgisidir. Benzetim yazılımı tarafından sanal ortamda uçurulan hava aracından alınan uçağın açısal hız ve öteleme ivmesi verileri kontrolcü bilgisayarı tarafından MATLAB/Simulink ortamında HAY'dan geçirilmektedir. Elde edilen veriler Siemens kontrolcülerine UDP haberleşme blokları aracılığı ile gönderilir ve platformun hareketi sağlanır. Geri besleme ile alınan platform bacak boyları ilerideki başlıklarda anlatılacak olan düz kinematik algoritmasından geçirilir ve üst platformun merkez noktası konumları hesaplanır. Bu veri platformdan elde edilen ivme değerinin gözlemlenmesinde kullanılmaktadır.

BÖLÜM 3

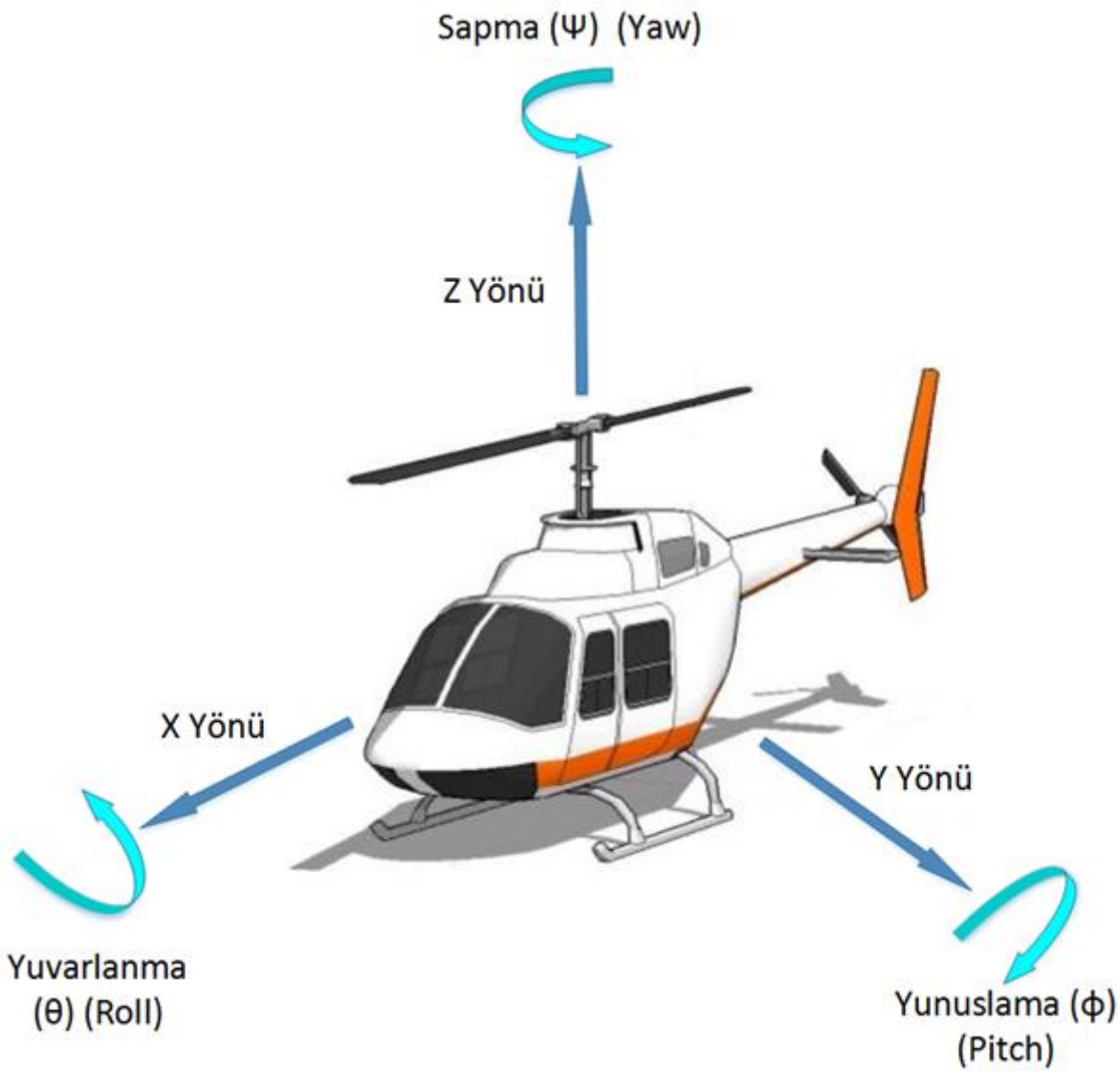
STEWART PLATFORMUNUN KİNEMATİK ANALİZİ

İnsan algısı modeli ile hareket algı algoritması oluşturulmasından önce platform sisteminin yapı taşlarından biri olan ters ve düz kinematik analizleri yapılmalıdır. Bu başlık altında temel olarak kullanılan denklemler verilecektir. Bu denklemler platformun hareketlerinin oluşturulmasında kullanılmaktadır. Hareket algı algoritmasından elde edilen sonuçlar platformun kontrolcüsüne üst platformun orta noktası koordinatları şeklinde gönderilmektedir. Platform kontrolcüsü içinde bu veriler ters kinematik analiz ile platformun bacak uzunluklarına dönüştürülmektedir. Elde edilen bacak boyu değişimleri ile kontrolcünün çıkışında, MATLAB/Simulink bölümünde düz kinematik analiz yapılarak platformun ivmesi analitik olarak tespit edilmektedir. Her iki kinematik analizde de kullanılmak üzere, rotasyon matrisi çıkarımı;

$$\boldsymbol{R_B^P} = \boldsymbol{R_z}(\Psi)\boldsymbol{R_y}(\theta)\boldsymbol{R_x}(\phi) = \begin{pmatrix} C_\Psi & -S_\Psi & 0 \\ S_\Psi & C_\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi & C_\phi \end{pmatrix}$$
$$= \begin{pmatrix} C_\Psi C_\theta & C_\Psi S_\theta S_\phi - S_\Psi C_\phi & S_\Psi S_\phi + C_\Psi S_\theta C_\phi \\ S_\Psi C_\theta & S_\Psi S_\theta S_\phi + C_\Psi C_\phi & S_\Psi S_\theta C_\phi - C_\Psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix} \quad (3.1)$$

$$\boldsymbol{R_B^P} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (3.2)$$

Şeklindedir. Eksenlerin helikopter görseli üzerinde gösterimi Şekil 3.1'de gösterilmektedir.



Şekil 3.1 Helikopterin yön bilgileri.

3.1 Platformun Ters Kinematik Analiz

Stewart platformunun ters kinematik analizi, üst platformun orta noktasının koordinatları kullanılarak platformun bacak boylarının bulunmasıdır. Bu hesaplama için, platformda kullanılacak rotasyon matrisi (3.1) denkleminde verilmiştir. Platformun bacak boyu ters kinematik analiz ile (3.3) denklemi ile hesaplanmaktadır. Bunun dışında platformun bacaklarının iki uç noktasının koordinatları da (3.4) ve (3.5) denklemlerinde hesaplanarak ana denklemde kullanılmıştır.

$$l_i{}^2 = x^2+y^2+z^2 + R_p{}^2+R_b{}^2 + 2.\left(R_{11}\,.\,P_{ix} + R_{12}\,.\,P_{iy}\right).(x - B_{ix}) + 2\,.\left(R_{21}.\,P_{ix} + R_{22}\,.P_{iy}\right).\left(y - B_{iy}\right) + 2.\left(R_{31}.P_{ix} + R_{32}.P_{iy}\right).(z) - 2.(x.B_{ix} + y.B_{iy}) \quad (3.3)$$

Üst platform bağlantı noktaları (3.4) denkleminde, alt platform bağlantı noktaları (3.5) denkleminde gösterilmiştir. Bu denklemlerdeki σ ve ρ üst ve alt platformun bacakları arasındaki ayırma açılarıdır.

$$\boldsymbol{P} = [R_p\cos(\sigma)\,;R_p\sin(\sigma)\,;0] \quad (3.4)$$

$$\boldsymbol{B_p} = [R_b\cos(\rho)\,;R_b\sin(\rho)\,;0] \quad (3.5)$$

3.2 Platformun Düz Kinematik Analizi

Düz kinematik analiz, ters kinematik analizin tersi olup daha yüksek işlem yoğunluğuna sahiptir. Ters kinematik analizde tek bacak boyu denkleminden çözüm yapılırken, düz kinematik analizde yaklaşıma bağlı bir hesaplama sistemi bulunmaktadır. Bu yaklaşım Newton Raphson yöntemi ile yapılmaktadır.

Platformun eksenlere bağlı bacak boyu değişimi jakobiyen matrisini oluşturmaktadır. Jakobiyen matrisi ile beraber, platformun bir önceki bacak boyu referans alınarak yaklaşım yöntemi sayesinde bir sonraki bacak boyu için sonuç çıkarılır. Çıkarılan sonuç, kullanıcı tarafından belirlenen ihmal edilebilen bir hata payı ile karşılaştırılır. Eğer sonuç hata payının altında ise yeni üst platform pozisyonu olarak tayin edilir. Sistemin denklem yapısı sırasıyla aşağıdaki gibi olmaktadır;

$$F_i(\boldsymbol{X}) = x^2 + y^2 + z^2 + r_p{}^2 + r_b{}^2 + 2\left(R_{11}p_{ix} + R_{12}p_{iy}\right)(x - b_{ix}) + 2\left(R_{21}p_{ix} + R_{22}p_{iy}\right)\left(y - b_{iy}\right) + 2\left(R_{31}p_{ix} + R_{32}p_{iy}\right)(z) - 2\left(xb_{ix} + yb_{iy}\right) - l_i{}^2 = 0 \quad (3.6)$$

$$\boldsymbol{X} = [x, y, z, \alpha, \beta, \gamma] \quad , \qquad i = 1,2,\ldots,6$$

Burada $\mathrm{F_i}(\mathbf{X})$ hesaplanan bacak boyu pozisyonu hatasının karesidir. Her iterasyonun başında ve sonunda bu hesaplama yapılmaktadır. $\mathbf{X}$ üst platformun orta noktasına ait pozisyon değerleridir. Jakobiyen matrisi hesabı denklem (3.7)'de gösterilmiştir.

$$\boldsymbol{J}_{(ij)} = \frac{dF_i}{d\boldsymbol{X_j}}(\boldsymbol{X}), \qquad \forall\{i,j\}\,\epsilon\,\{1,..6\} \quad (3.7)$$

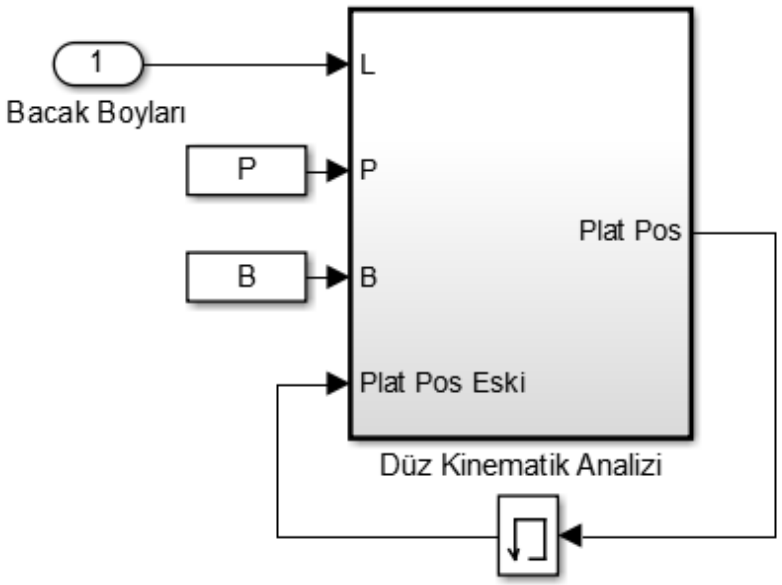
Bir sonraki üst platform koordinatı;

$$\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} - \left[\mathbf{J} \cdot \left(\mathbf{X}^{(n)}\right)\right]^{-1} \cdot F\left(\mathbf{X}^{(n)}\right) \tag{3.8}$$

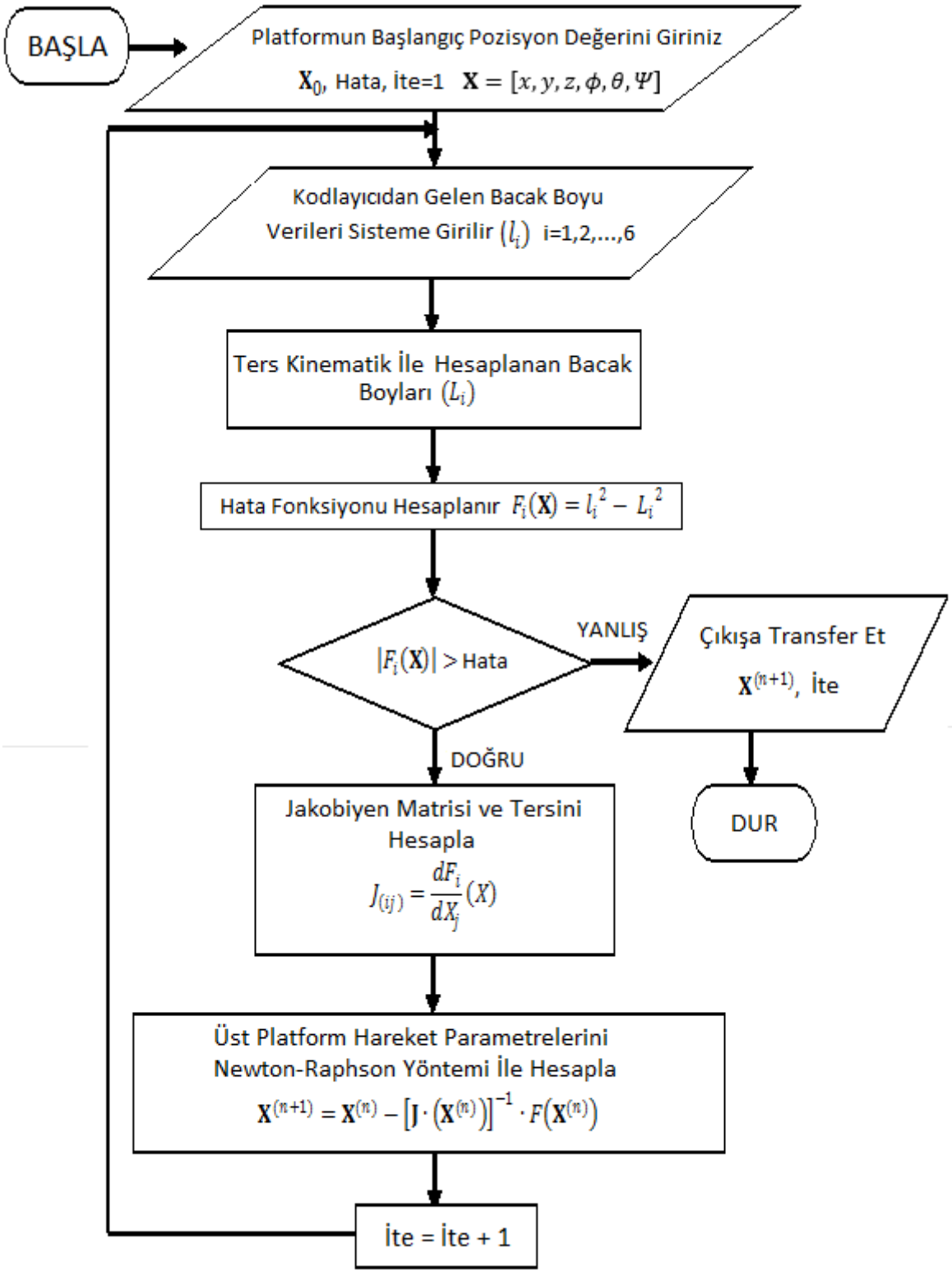
(3.8) denkleminde, $\mathbf{X}^{(n)}$ üst platformun bir önceki örnekleme zamanına ait konumu, $\mathbf{X}^{(n+1)}$ bir sonraki örnekleme zamanına ait üst platform konumudur. Yeni elde edilen pozisyonun doğruluğu (3.9) denklemi ile kontrol edilir ve belirlenen hata değerinin altında ise döngü tamamlanır.

$$|F_i(\boldsymbol{X})| \leq \varepsilon \tag{3.9}$$

Burada ε referans alınan pozisyon hatasının karesidir. Denklem (3.9), belirlenen bir tolerans değer için çözüm yapmaktadır. Düz kinematik algoritmasının simulink yapısı Şekil 3.2'de, program akış şeması Şekil 3.3'te gösterilmiştir. Düz kinematiğe ait Matlab kod satırları Ek-A'da gösterilmiştir.

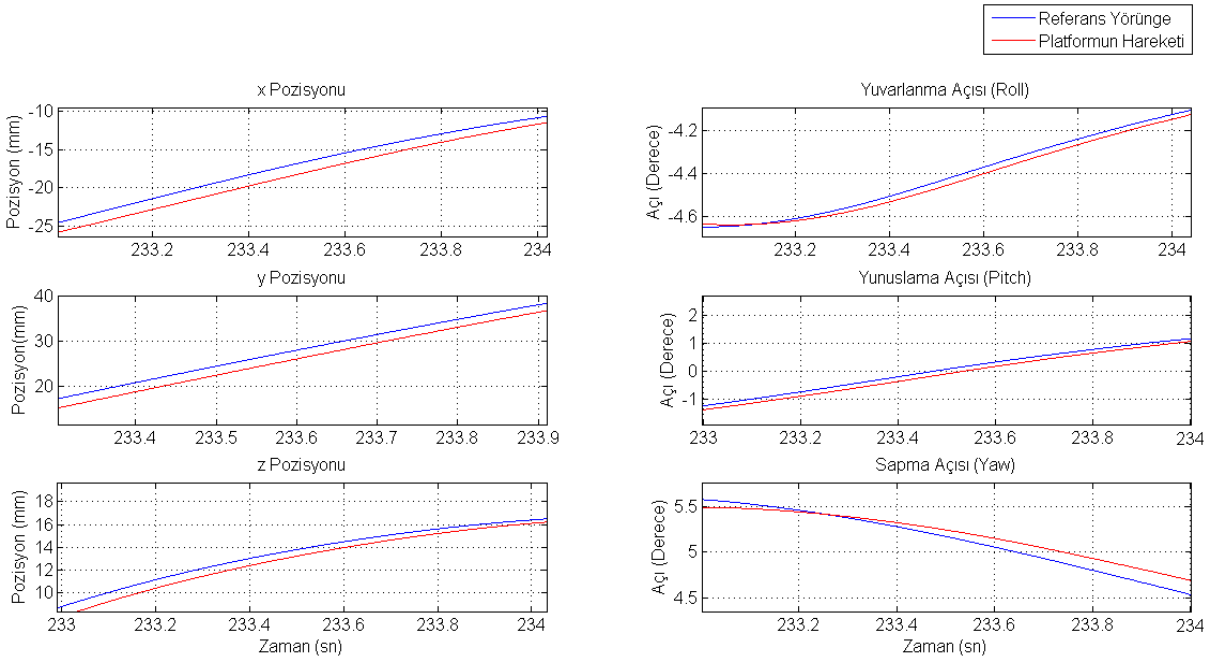


Şekil 3.2 Platformun düz kinematik yazılımının simulink görseli



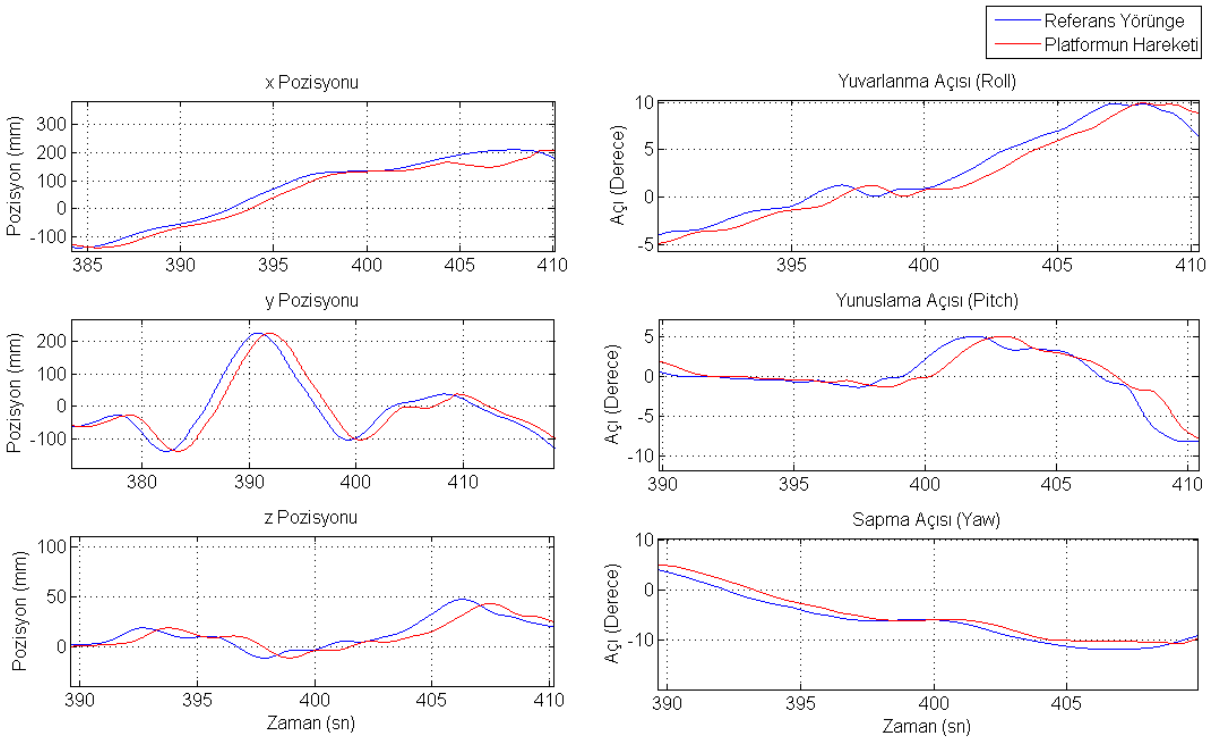
Şekil 3.3 Platformun düz kinematik algoritması [31].

Şekil 3.3'teki algoritma kullanılarak oluşturulan yazılımın performansı Şekil 3.4 ve Şekil 3.5'te hareket algı algoritmalarına göre gösterilmiştir.



Şekil 3.4 Düz kinematik analizin klasik algoritmalı modeldeki performansı.

Şekil 3.4'te gösterildiği gibi, klasik algoritma devrede iken bütün eksenlerin konum hesaplamalarında düz kinematik analizin gecikme payı 0.1 saniye olmaktadır. Görsel simülasyon, algı yazılımı, düz kinematik hesaplama ve verilerin transfer edilmesindeki gecikmeler, hesaplamadaki bu faz farkına sebebiyet vermektedir.

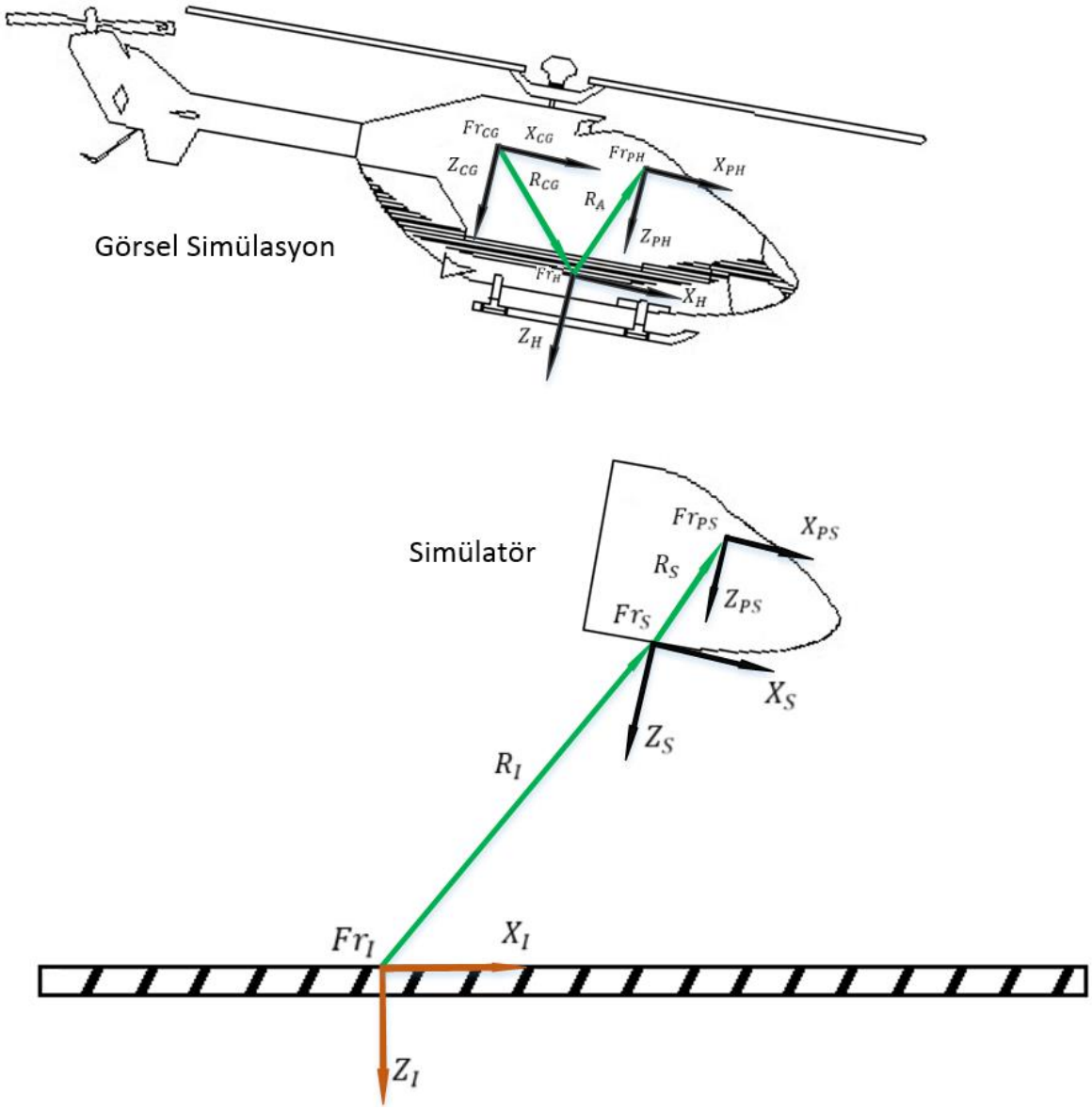


Şekil 3.5 Düz kinematik analizin, model öngörülü kontrol algoritması modelindeki performansı.

Şeki 3.5'te model öngörülü kontrol algoritması kullanıldığında düz kinematik hesaplamadaki gecikmeler gösterilmiştir. Bu algoritmada kinematik hesaplama yaklaşık olarak 1 saniye civarında gecikmeye sebep olmaktadır. İki algoritmanın da gecikmeleri karşılaştırıldığında bu gecikmenin düz kinematik yazılımının dışında, hareket algısı denklemlerinin oluşturduğu yoğunluğun da etkisinin olduğu gözlemlenmiştir. İstendiği takdirde yaklaşım sayısının azaltılması için, düz kinematik algoritmasının hata hassasiyeti düşürülebilir fakat bu durumda gecikme azalırken, pozisyon hatalarında artma gözlemlenir. Bu gecikme problemi, simulink ara yüzünün daha efektif kullanılması, algoritmalardaki işlem yüklerinin azaltılması veya daha yüksek performansa sahip kontrol bilgisayarlarının kullanılması gibi bir takım düzenlemeler ile iyileştirilebilmektedir.

3.3 Koordinat Eksenleri Dönüşümü

Görsel simülasyondan elde edilen değerler, pilotun helikopter içindeki göz noktası hizasında ve kendi eksenine ait hareket bilgilerini içermektedir. Dolayısıyla platforma gönderilmeden önce, koordinat eksenleri arasında bir dönüşüme ihtiyaç duyulmaktadır. Şekil 3.6'da helikopter ve simülatörün pilot hizasıyla beraber koordinat eksenleri tanımlanmıştır.



Şekil 3.6 Simülatör ve görsel simülasyona ait koordinat eksenleri bilgisi

Şekil 3.6'da, Fr_I simülatörün alt mafsal merkezlerinden geçen dairenin merkezindeki simülatör sabit koordinat eksen takımı, Fr_S simülatörün üst mafsal merkezlerinden geçen dairenin merkezindeki simülatör hareketli koordinat eksen takımı, Fr_{PS} simülatördeki pilotun baş bölgesine ait koordinat eksen takımı, Fr_{PH} görsel simülasyondaki pilotun baş bölgesinin koordinat eksen takımı, Fr_H görsel simülasyondaki pilot kabininin taban noktasının koordinat eksen takımı ve Fr_{CG} helikopterin ağırlık merkezi eksen takımıdır. Model öngörülü kontrol için, görsel simülasyondaki helikopterin ağırlık merkezinden alınan ivme ve açısal hızlar, vestibüler sistemde kullanılmak üzere yine görsel simülasyondaki pilotun göz hizasına dönüşüm matrisleri ile taşınması gerekmektedir. Havelsan ile yapılan

çalışmada, kendilerine ait dinamik modelde bu dönüşüm yapılmaktadır. Fakat platforma gönderilen veriler üst platformun merkez noktasına ait olduğundan, hesaplamalar sonucunda tekrardan üst platformun merkez noktasına dönüştürülmesi gerekmektedir. [32] çalışması referans alınarak, pilotun göz hizasındaki pozisyon ve açı verileri platformun orta noktasına indirgenmektedir.

$$\boldsymbol{T}_{Ps}^{S} = \begin{bmatrix} \boldsymbol{R}_{Ps}^{S} & 0_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \begin{bmatrix} I_{3x3} & \boldsymbol{p}_{S}^{Ps} \\ 0_{1x3} & 1 \end{bmatrix} \tag{3.10}$$

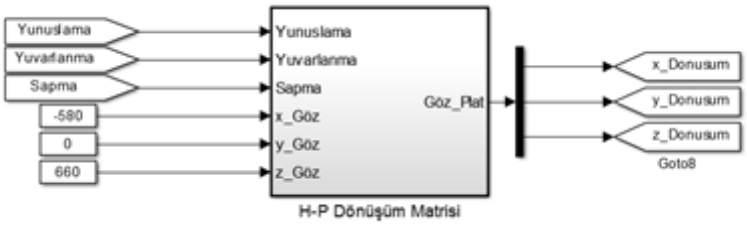
Bu denklemde $\boldsymbol{T}_{Ps}^{S}$ matrisi, simülatörde üst platformun merkez koordinat takımı Fr_S ile, simülatördeki pilotun göz hizasındaki koordinat takımı Fr_{PH} arasındaki dönüşüm matrisi olmaktadır. $\boldsymbol{R}_{Ps}^{S}$ rotasyon matrisi, $\boldsymbol{p}_{Ps}^{S}$ iki nokta arasındaki kartezyen uzaklık olmaktadır ve vektörel olarak üst platformun merkez noktasından, pilotun göz hizasına doğrudur. $\boldsymbol{p}_{S}^{Ps}$'in içeriği (3.11) denkleminde gösterilmiştir ve birimi mm'dir.

$$\boldsymbol{p}_{S}^{Ps} = \begin{bmatrix} -580 \\ 0 \\ 660 \end{bmatrix} \tag{3.11}$$

İki noktanın açısal hareketleri aynı olduğundan, sadece x,y ve z yönlerine ait pozisyonların dönüşümü yapılmıştır. Bunun için üst platformun merkez noktasının koordinat ekseni sabit olduğu varsayılarak pilotun göz noktasının koordinat takımının değiştiği düşünülmüştür. (3.2) denkleminin tersi alınarak, $\boldsymbol{R}_{Ps}^{S}$ matrisi elde edilmiştir.

$$\boldsymbol{R}_{Ps}^{S} = \boldsymbol{R}_{B}^{P^{-1}} \tag{3.12}$$

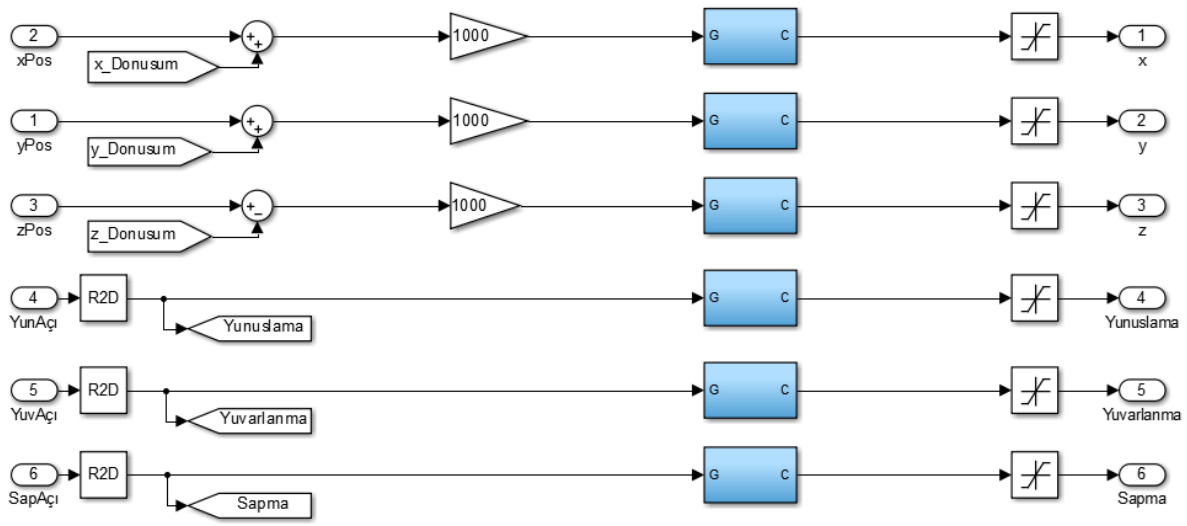
Dönüşüm matrislerinin simulink üzerindeki gösterimi Şekil 3.7'de belirtilmiştir.



Şekil 3.7 Pilotun göz hizası – üst platform orta noktası arasındaki dönüşüm matrisi.

Şekil 3.7'nin yer aldığı simulink alt sistemi Şekil 3.8'de belirtilmiştir. Açısal konumlar şekilde gösterilen sondaki 3 açısal konum kanalından alınmaktadır. Şekil 3.7'de belirtilen hesaplama sonucunda elde edilen pozisyon farklı Şekil 3.8'de gösterilen üstteki ilk 3 kanala

beslenmektedir. Metre – milimetre dönüşümü yapıldıktan sonra ani değişimlerin yumuşatılması için basit bir filtreleme işleminden geçirilen platform pozisyonları güvenlik açısından platform sınırlarından geçirilerek platform kontrolcüsüne gönderilmektedir. Bu blok şeması platformun genel blok şemasında "Dönüşüm Bloğu" alt sistemi olarak gösterilecektir.



Şekil 3.8 Dönüşüm bloğu içyapısı. Platformun hareket eksenlerinin ayarlanması.

BÖLÜM 4

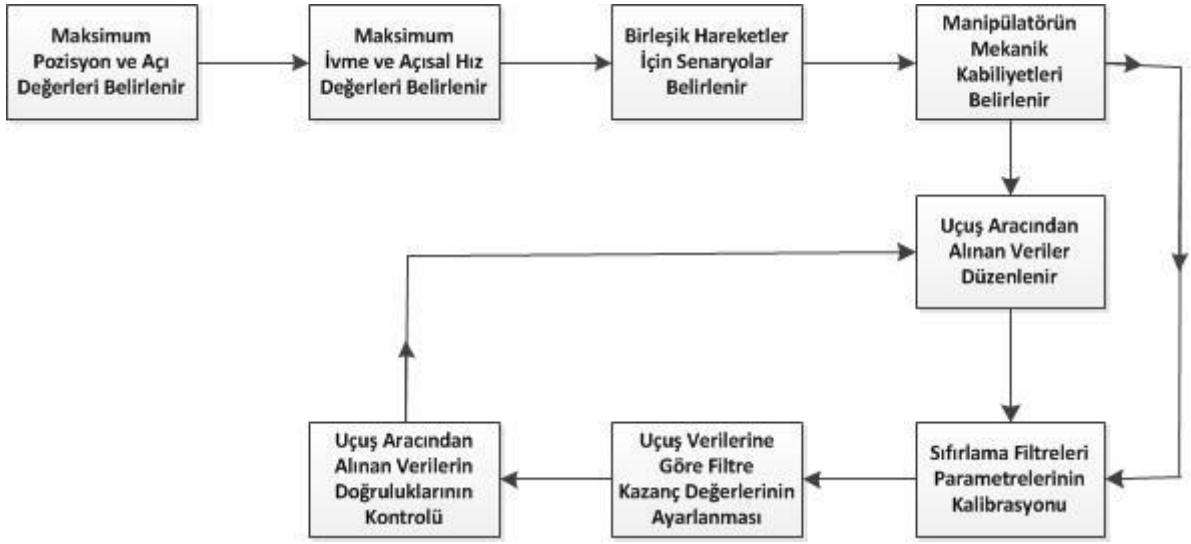
HAREKET ALGI ALGORİTMASI ÇALIŞMALARI

Bu başlık altında, yapılacak pratik uygulamalar için matematiksel modeller oluşturulacaktır. Çalışmalara öncelikli olarak klasik algoritma ile başlanmıştır. Ardından insan algısı için matematiksel modellemeler yapılmış ve MOHAY ile birleştirilerek daha gelişmiş bir algı algoritması geliştirilmiştir.

4.1 Klasik Sıfırlamalı Hareket Algı Algoritması

Klasik algoritma, yüksek ve düşük geçiren filtrelerden oluşmaktadır ve filtre parametreleri kullanıcı tarafından tek seferlik belirlenmektedir. Filtre girişleri hava aracının öteleme ivmeleri ve açısal hızları olmaktadır. Filtrelenme sonrasında verilerin integralleri alınarak pozisyon verisi olarak platforma gönderilmektedir. Filtre parametreleri deneysel çalışmayla belirli bir prosedür takip edilerek belirlenmiştir. Burada asıl amaç, platformun pilota hissettirmesi gereken hareketleri tamamladıktan sonra insan algısının altında kalacak şekilde başlangıç konumuna geri dönmesidir.

Filtrelerin denklemleri Çizelge 4.1'de, çalışma prensibi parametrelere göre Çizelge 4.2'de gösterilmiştir. Bu denklemlerde s laplace değişkenini, ω_l ve ω_h filtre frekanslarını ve ξ filtrenin sönüm oranını belirtmektedir. Klasik sıfırlamalı algoritmanın kalibrasyon aşamaları Şekil 4.1'de belirtilmiştir. Filtrelerin basamak girişe cevapları Şekil 4.2'deki gibi gösterilebilir.



Şekil 4.1 Klasik sıfırlamalı algoritmanın kalibrasyon aşamaları

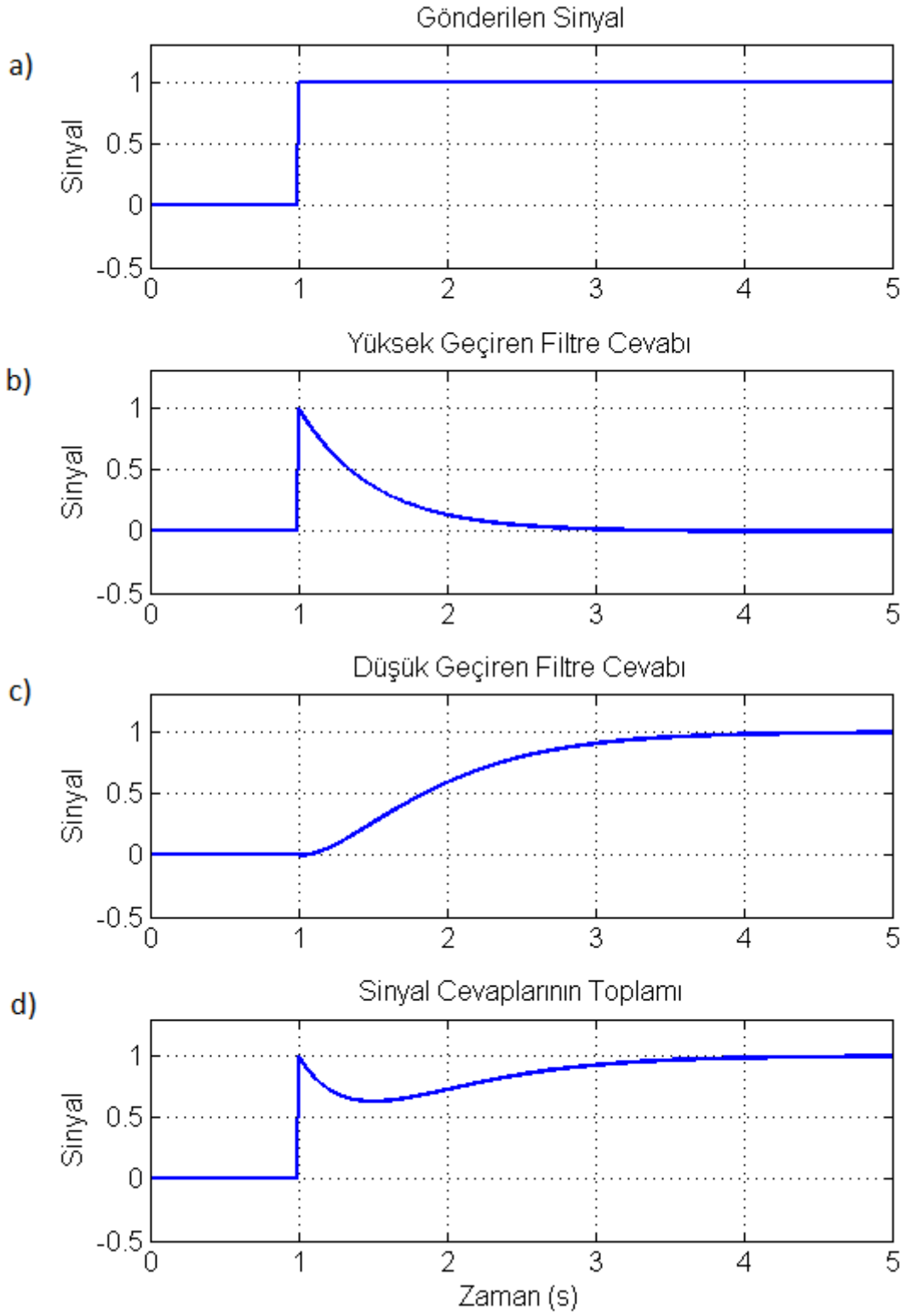
Kalibrasyonda ilk olarak, platformun ulaşabileceği en yüksek pozisyon ve açı değerleri ile uçuş aracı ve platformun en yüksek ivme ile açısal hızları belirlenir. Simülatörün birleşik hareketi de göz önüne alınarak sınırları sisteme tanıtılır. Sıfırlama filtrelerinin parametreleri öncelikle sinyalin genlik ayarı için 1. dereceden filtreler, ardından sıfırlama için 2. dereceden filtrelerden geçirilir. Platformun hareket kapasitesine göre simülatöre gönderilmeden önce platform konumları üzerinde ölçeklendirme yapılır. Hareket sırasında pilotun yorumu ile kalibrasyon işlemi tekrar edilir.

Çizelge 4.1 Klasik algoritmada kullanılan filtrelerin denklemleri

Filtre Çeşidi	**Denklemi**
1. Dereceden Yüksek Geçiren Filtre	$\frac{s}{s+\omega_h}$
2. Dereceden Yüksek Geçiren Filtre	$\frac{s^2}{s^2+2\omega_h\xi s+{\omega_h}^2}$
1. Dereceden Düşük Geçiren Filtre	$\frac{\omega_l}{s+\omega_l}$
2. Dereceden Düşük Geçiren Filtre	$\frac{{\omega_l}^2}{s^2+2\omega_l\xi s+{\omega_l}^2}$

Çizelge 4.2 Filtrelerin çalışma prensibi [33].

		Yükseltme	Azaltma	Max. Değer	Min. Değer
Yüksek Geçiren Filtreler (High Pass Filter)	1. Dereceden Filtre Parametreleri	ω_h	Düşük frekanslı sinyalleri azaltır. Platformun ani hareketlerini güçlendirir.	Düşük frekanslı sinyalleri yükseltir. Platformun ani hareketlerini azaltır.	Yok	Yok
	2. Dereceden Filtre Parametreleri	ω_h	Sıfırlama hızını arttırır	Sıfırlama hızını azaltır.	Yok	Yok
		ξ	Diğer parametrelerden bağımsız olarak, platform lineer hareketlerini yavaş tamamlar.	Diğer parametrelerden bağımsız olarak, platform lineer hareketlerini hızlı tamamlar.	1,7-1,5	0,8
Düşük Geçiren Filtreler (Low Pass Filter)	2. Dereceden Filtre Parametreleri	ω_l	Düşük frekanslı sinyallerin etkisini yükseltir. Yüksek frekanslı sinyallerin etkisini azaltır.	Düşük frekanslı sinyallerin etkisini azaltır. Yüksek frekanslı sinyallerin etkisini artırır.	Yok	Yok
		ξ	Diğer parametrelerden bağımsız olarak, platform lineer hareketlerini yavaş tamamlar.	Diğer parametrelerden bağımsız olarak, platform lineer hareketlerini hızlı tamamlar.	1,2	0,8



Şekil 4.2 kHAY filtrelerinin basamak giriş durumundaki cevapları, a) basamak giriş, b) Sinyalin yüksek geçiren filtre cevabı, c) Sinyalin düşük geçiren filtre cevabı, d) Yüksek ve düşük geçiren filtrelerin cevaplarının toplamı.

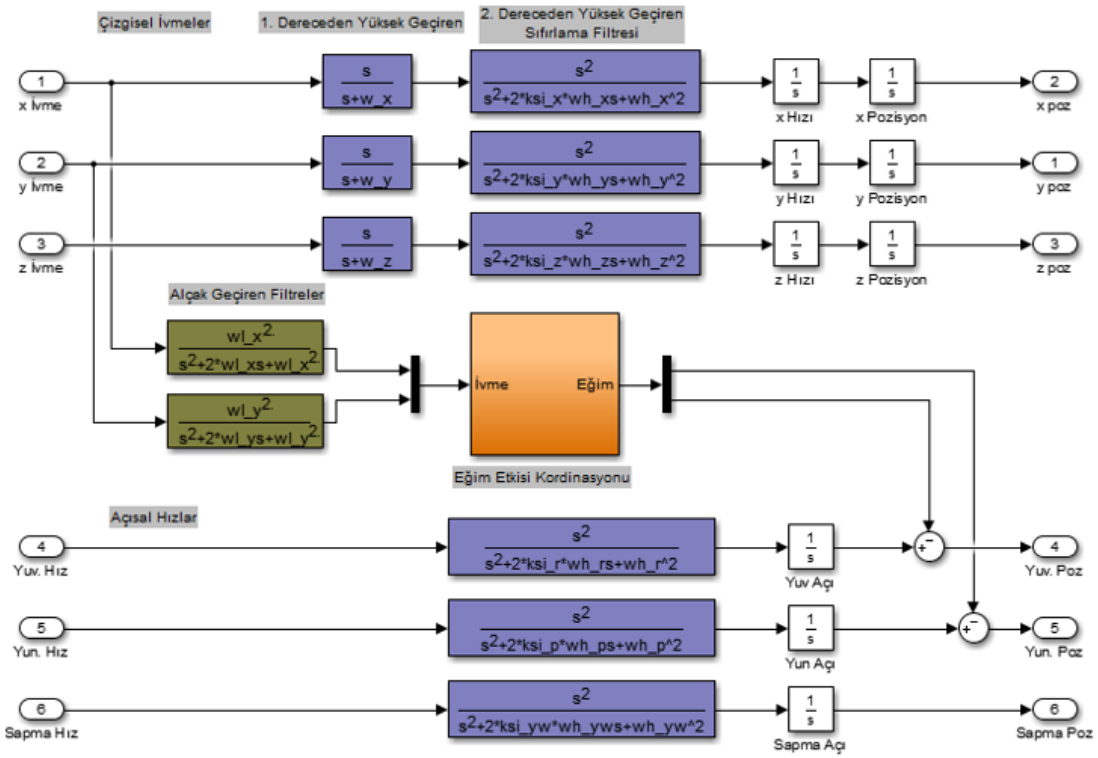
Şekil 4.2'deki grafiklerden de görüleceği üzere, yüksek geçiren filtreler sinyalin anlık genliğini verdikten kısa bir süre sonra sinyali sıfırlamaya başlamaktadır. Bu, simülatör sistemleri için hem sinyaldeki gürültünün azaltılmasında, hem de 2. dereceden filtre kullanılması durumunda negatif ivme oluşturup platformun başlangıç pozisyonuna geri dönmesinde çok etkilidir. USY'den alınan öteleme ivmeleri ve açısal hızlar için bu filtre yapıları kullanılacaktır.

Düşük geçiren filtrelerde hareket sinyal çıkışı sıfırdan yükselmekte ve istenen değere ulaşmaktadır. Bu özellik de pilota eğim etkisini oluşturarak öteleme ivmelerindeki hissiyatı açısal hareket ile hissettirmek için kullanılacaktır. Her iki filtre cevabının toplamında oluşan sarkmaların olabildiği kadar düşük olması beklenmektedir (Şekil 4.2d) Klasik sıfırlamalı algoritmanın MATLAB – Simulink yapısı Şekil 4.3'teki gibi olmaktadır.

Algoritma içindeki filtreler dışında, sistemin kararlı ve düzgün çalışması açısından sistemin MATLAB modelinde bazı noktalarda ek filtrelemeler kullanılmıştır. Platformun hareketi için seçilen yüksek geçiren ve düşük geçiren filtre parametreleri, eksenlerine göre Çizelge 4.3'te gösterilmiştir. EK-B'de klasik sıfırlamalı hareket algı yazılımının MATLAB kodları belirtilmiştir.

Çizelge 4.3 Klasik algoritma filtrelerinde kullanılan parametrelerin listesi.

Parametre sınıfı	**Bağlı olduğu eksen**	**ω_h - ω_l Frekans (1/s)**	**ξ - Sönüm**
1. Dereceden Yüksek Geçiren Filtre Parametreleri	x ekseni	1.4	Yok
	y ekseni	1.2	Yok
	z ekseni	1.6	Yok
2. Dereceden Yüksek Geçiren Filtre	x ekseni	0.4	1.2
	y ekseni	0.4	1.2
	z ekseni	0.4	1.2
	Yuvarlanma	0.12	0.8
	Yunuslama	0.12	1
	Sapma	0.12	0.8
2. Dereceden Düşük Geçiren Filtre	x ekseni	0.9	1
	y ekseni	1.5	1



Şekil 4.3 kHAY'nın MATLAB/Simulink yapısı.

4.2 Model Öngörülü Hareket Algı Algoritması

Model öngörülü kontrol sistemi, optimal kontrol teorisinin özel bir dalı olup, her bir örnekleme zamanında sistem çıkışlarının iyileştirilmesi için öngörülerde bulunmaktadır. Diğer kontrol sistemlerine nazaran, sistemi tam kapsamlı olarak kullanabilme özelliğine sahiptir [29]. Sistem modelini temel alarak, sistemin çalışma uzayı içerisinde kontrolünü sağlamaktadır [34]. Bu kontrol sisteminde asıl amaç platforma gidecek kontrol girişi vektörünün, u_k, sistemin geçmiş durumunu ele alarak doğru bir şekilde hesaplanmasına dayanmaktadır. MOHAY, genel olarak aşağıdaki maddeleri içermektedir.

- **Maliyet Fonksiyonu:** Bir sistemin maliyeti, o sistemin referans ve çıkış değerleri arasındaki farkın bir ölçüsüdür. Bu farkın, ideal sistemlerde sıfır olması gerekmektedir. Dolayısıyla uygulamalarda da maliyet fonksiyonunun düşük olması beklenmektedir. Bu referans değerleri sistemin durum vektörü, sistemin girişleri veya çıkışları olabilmektedir. Maliyet fonksiyonu genel olarak J_k şeklinde gösterilir. Burada k örnekleme anıdır.

- **Kısıtlamalar:** MOHAY'ın geri planında, sistemin düzenli bir şekilde çalışması için kısıtlamalar bulunmaktadır. Bunlar sistemin giriş genliğindeki kısıtlamalar olabileceği gibi, girişin değişiminde yapılan kısıtlamalar sistemin istenen bir şekilde, düzenli olarak çalışmasını sağlar.
- **Öngörülecek Model:** MOHAY'ın yapı taşı olan öngörü mekanizması, sistemin davranışlarının belirlenmesindeki en temel unsurdur. Bunun mümkün olabilmesi için öngörülecek modelin, yani sistem modelinin tanımlanmış olması gerekmektedir. Dolayısıyla sistemin giriş ve çıkışlarının da belirlenmiş olduğu durum uzay formunun elde edilmiş olması gerekmektedir.

MOHAY, MIMO yapısına sahip sistemlere kolaylıkla uyarlanabilen bir kontrol çeşididir.

4.2.1 Kontrol Yapısı ve Durum Uzay Modeli

MOHAY, bir kontrol düzlemi boyunca belirlenen tahmin edilecek durum sayısı ve bunları iyileştirme görevi gören maliyet fonksiyonundan oluşmaktadır. Sistem denklemleri [17]'den destek alınarak sırasıyla aşağıdaki gibi oluşturulmuştur. İnsan iç kulak sistemi ise [28] temel alınarak oluşturulmuş ve bütün denklemler ortak çözüm ortamına alınmıştır. Hata denklemi,

$$\boldsymbol{\epsilon}(k+i|k) = \boldsymbol{r}(k+i|k) - \boldsymbol{y}(k+i|k), \qquad i = 1, \dots, N_p \tag{4.1}$$

olmaktadır. Burada k örnekleme zamanı, i bir sonraki örnekleme zamanı, **r** sistemin referans pozisyonu vektörü, **y** hareketin sonunda elde edilen pozisyon değeri ve N_p öngörülen sistem çıkışları değeridir. Yukarıdaki gösterimde "|" ifadesinden sonra gelen k değeri, sistemin o andaki örnekleme zamanını temsil etmektedir. "k+i" ifadesi sistemin "k" anındaki örnekleme zamanında öngörülen "k+i" zamanına ait değer olmaktadır.

Bu yeni değer, sistemin çıkışında bulunan maliyet fonksiyonu ile belirlenmektedir. i değeri tahmini yapılan örnekleme zamanı olup, kullanıcı tarafından kaç örnekleme zamanı için öngörü yapılacağı belirlenecektir. Kontrol girişi ise;

$$\boldsymbol{u}(k+i|k) = \boldsymbol{u}(k|k) + \Delta\boldsymbol{u}(k+i|k), \qquad i = 0, \dots, N_u - 1 \tag{4.2}$$

şeklindedir. N_u öngörülen kontrol girişi sayısı olmaktadır. MOHAY'da göz önüne alınması gereken maddeler;

- Sistem sınırlı çalışma uzayında bulunduğundan gerçek zamanlı kısıtlamalar olacaktır.
- Doğrusal ayrık zamanlı durum uzay modeli oluşturulacaktır.
- Bir sonraki hareket için sistem, o andan bir önceki örnekleme zamanına ait hareket bilgisini kullanacak ve buna göre yeni bir yörünge belirleyecektir. Bunu öngörü düzlemi boyunca istenen öngörü miktarı kadar yapacaktır.

Durum uzay modeli, insan algısını ve platform durumlarını da içereceğinden birleşik bir sistem olacaktır. Bunu aşağıdaki denklem ile gösterebiliriz.

$$\begin{aligned} \boldsymbol{X_m}(k+1) &= \boldsymbol{A_m X_m}(k) + \boldsymbol{B_m u}(k) \\ \boldsymbol{Y}(k) &= \boldsymbol{C_m X_m}(k) \end{aligned} \tag{4.3}$$

İlk denklem gibi, öngörü düzlemi boyunca belirlenen durum uzay modellerini alt alta sıralarsak;

$$\begin{aligned} \boldsymbol{X_m}(k+1) &= \boldsymbol{A_m X_m}(k) + \boldsymbol{B_m u}(k) \\ \boldsymbol{X_m}(k) &= \boldsymbol{A_m X_m}(k-1) + \boldsymbol{B_m u}(k-1) \end{aligned} \tag{4.4}$$

Bu denklemlerin sırayla birbirleri arasındaki fark alındığında,

$$\Delta \boldsymbol{X_m}(k+1) = \boldsymbol{A_m} \Delta \boldsymbol{X_m}(k) + \boldsymbol{B_m} \Delta \boldsymbol{u}(k) \tag{4.5}$$

olmaktadır. Burada $\Delta \boldsymbol{X_m}(k+1) = \boldsymbol{X_m}(k+1) - \boldsymbol{X_m}(k)$ bir önceki durum verisinden mevcut durum verisinin farkıdır. Benzer olarak kontrol girişinde de aynı durum mevcuttur. Sistem çıkışları için;

$$\boldsymbol{y}(k+1) - \boldsymbol{y}(k) = \boldsymbol{C_m} \underbrace{(\boldsymbol{X_m}(k+1) - \boldsymbol{X_m}(k))}_{\Delta \boldsymbol{X_m}(k+1)} \tag{4.6}$$

$$\boldsymbol{y}(k+1) - \boldsymbol{y}(k) = \boldsymbol{C_m A_m} \Delta \boldsymbol{X_m}(k) + \boldsymbol{C_m B_m} \Delta \boldsymbol{u}(k) \tag{4.7}$$

Denklem (4.5), (4.6) ve (4.7)'nin birleşimi ile genişletilmiş model;

$$\begin{aligned} \boldsymbol{X}(k+1) &= \underbrace{\begin{bmatrix} \boldsymbol{A_m} & 0 \\ \boldsymbol{C_m A_m} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{X}(k) + \underbrace{\begin{bmatrix} \boldsymbol{B_m} \\ \boldsymbol{C_m B_m} \end{bmatrix}}_{\boldsymbol{B}} \Delta \boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \underbrace{[0 \quad I]}_{\boldsymbol{C}} \boldsymbol{X}(k) \end{aligned} \tag{4.8}$$

olmaktadır. $\Delta \boldsymbol{u}(k)$ kontrol girişi değişimidir. Burada $\boldsymbol{X}(k)$terimi;

$$\boldsymbol{X}(k) = \begin{bmatrix} \Delta \boldsymbol{X_m}(k) \\ \boldsymbol{y}(k) \end{bmatrix} \tag{4.9}$$

olmaktadır. Elde edilen bu model, tahmin düzlemi boyunca çoğaltılırsa istenilen MPC durum uzay modeli oluşmaktadır. Bir önceki durum denklemi bir sonraki durum denklemine eklenirse oluşan durum denklemleri:

$$\begin{aligned}
&\boldsymbol{X}(k+1|k) = \boldsymbol{A}\boldsymbol{X}(k) + \boldsymbol{B}\Delta\boldsymbol{u}(k) \\
&\boldsymbol{X}(k+2|k) = \boldsymbol{A}\boldsymbol{X}(k+1|k) + \boldsymbol{B}\Delta\boldsymbol{u}(k+1) = \boldsymbol{A^2}\boldsymbol{X}(k) + \boldsymbol{A}\boldsymbol{B}\Delta\boldsymbol{u}(k) + \boldsymbol{B}\Delta\boldsymbol{u}(k+1) \\
&\qquad\qquad\qquad

N_P öngörü sayısı kadardır. F öngörü modelinin durum matrisi, φ öngörü modelinin kontrol matrisidir.

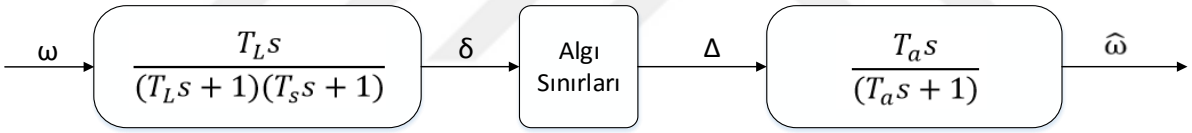
4.2.2 İnsan İç Kulak Sistemi Yapısı, Elemanları ve Durum Uzay Modeli

İnsanın iç kulak yapısı literatürlerde aynı zamanda vestibülar sistem olarak geçmektedir. Yapı, insanın açısal ve öteleme hareketlerini algılayabilecek şekildedir. Temel olarak 2 parçadan oluşur. Bunlar yarı dairesel kanallar ve otolit taşlarıdır.

Vestibülar sistem sayesinde platformun düz kinematik çıkış pozisyonlarının insan algısına uygun hale getirilmesi sağlanacaktır. Bunun için insan algı sisteminin matematiksel modelleri kullanılacaktır. Bu matematiksel modeller [28] çalışmasından alınmış olup transfer fonksiyonlardan oluşmaktadır. İnsan algısı, hareketin belli bir değerinin üstünde uyarılmaktadır.

4.2.2.1 Yarı Dairesel Kanal Sistemi Matematiksel Modellenmesi

Yarı dairesel kanallara ait transfer fonksiyonlar Şekil 4.4'te gösterilmiştir.



Şekil 4.4 Yarı Dairesel Kanallara Ait Blok Diyagram

Blokların ilki, yarı dairesel kanal sisteminin dinamiğini oluşturmaktadır ve platformdan alınan değer ilk olarak bu bloktan geçerek filtrelenir. Böylelikle insan algısına yakın bir düzeye getirilir. 2. Blok insan algısının eşik değerini belirtir. Burada ilgili açısal hızlar insan algısının hissedemeyeceği düzeyde ise sistemden geçirilmez. Dolayısıyla sistem bu sırada sıfır açısal hız gönderir. Son blok ise sisteme yeni bir yüksek geçiren filtreyi dâhil ederek hareketin sıfırlanmasını belirtir. Böylelikle insan algısı için açısal hızlar oluşturulmuş olur. Transfer fonksiyonun parametreleri Çizelge 4.4'te gösterilmiştir.

ω – Pilota/sürücüye etki eden açısal hız değeri.

δ – Kupula yer değişimi sonucu elde edilen değer.

Δ – İnsan algı sınırlaması sonucu elde edilen ivme değeri.

$\hat{\omega}$ – Sıfırlama bloğundan çıktıktan sonraki insanın hissettiği açısal hız.

Çizelge 4.4 Yarı dairesel kanal transfer fonksiyonu parametreleri [28]

	Yuvarlanma (x)	**Yunuslama (y)**	**Sapma (z)**
$T_L\ [s]$	6,1	5,3	10,2
$T_s\ [s]$	0,1	0,1	0,1
$T_a\ [s]$	30	30	30
$\delta TH\ [\mathrm{deg/s}]$	3	3,6	3,6

Transfer fonksiyonlardan sistemin durum uzay modelini oluşturacak olursak;

$$\frac{\widehat{\omega}}{\omega} = \frac{T_a T_L s^2}{(T_L s + 1)(T_s s + 1)(T_a s + 1)} \tag{4.16}$$

$$\frac{\widehat{\omega}}{\omega} = \frac{\frac{s^2}{T_s}}{s^3 + \left(\frac{1}{T_a} + \frac{1}{T_s} + \frac{1}{T_L}\right)s^2 + \left(\frac{1}{T_s T_a} + \frac{1}{T_L T_a} + \frac{1}{T_s T_L}\right)s + \left(\frac{1}{T_L T_s T_a}\right)} \tag{4.17}$$

$$a_1 = \frac{1}{T_a} + \frac{1}{T_s} + \frac{1}{T_L}, \quad a_2 = \frac{1}{T_s T_a} + \frac{1}{T_L T_a} + \frac{1}{T_s T_a}, \quad a_3 = \frac{1}{T_L T_s T_a} \tag{4.18}$$

$$b_0 = 0, \quad b_1 = \frac{1}{T_s}, \quad b_2 = 0, \quad b_3 = 0 \tag{4.19}$$

$$\beta_0 = 0, \quad \beta_1 = b_1, \quad \beta_2 = -a_1 b_1, \quad \beta_3 = {a_1}^2 b_1 - a_2 b_1 \tag{4.20}$$

$$x_1 = \widehat{\omega} - \beta_0 \omega = \widehat{\omega} \tag{4.21}$$

$$x_2 = \dot{x}_1 - \beta_1 \omega = \dot{x}_1 - b_1 \omega \tag{4.22}$$

$$x_3 = \dot{x}_2 - \beta_1 \dot{\omega} - \beta_2 w = \dot{x}_2 - b_1 \dot{\omega} + a_1 b_1 \omega \tag{4.23}$$

$$\dot{x}_1 = x_2 + \beta_1 \omega \tag{4.24}$$

$$\dot{x}_2 = x_3 + \beta_2 \omega = x_3 - a_1 b_1 \omega \tag{4.25}$$

$$\dot{x}_3 = -\frac{1}{T_L T_s T_a} x_1 - \left(\frac{1}{T_s T_a} + \frac{1}{T_L T_a} + \frac{1}{T_L T_s}\right) x_2 - \left(\frac{1}{Ta} + \frac{1}{T_s} + \frac{1}{T_L}\right) x_3 + ({a_1}^2 b_1 - a_2 b_1)\omega \tag{4.26}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{-1}{T_L T_s T_a} & -\left(\frac{1}{T_s T_a} + \frac{1}{T_L T_a} + \frac{1}{T_s T_L}\right) & -\left(\frac{1}{T_a} + \frac{1}{T_s} + \frac{1}{T_L}\right) \end{bmatrix}}_{A_{SCCX,Y,Z}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} b_1 \\ -a_1 b_1 \\ ({a_1}^2 b_1 - a_2 b_1) \end{bmatrix}}_{B_{SCCX,Y,Z}} \omega \tag{4.27}$$

$$\boldsymbol{C_{SCCX}} = \boldsymbol{C_{SCCY}} = \boldsymbol{C_{SCCZ}} = [1 \quad 0 \quad 0] \tag{4.28}$$

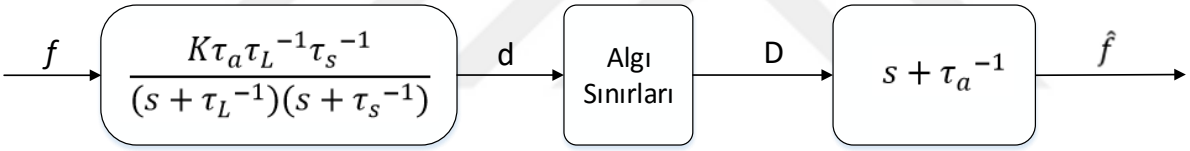
Yarı dairesel kanallarının matematiksel modelinin bütün açısal eksenler için için uyarlanmış hali;

$$\boldsymbol{A_{SCC}} = \begin{bmatrix} \boldsymbol{A_{SCCX}} & \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{A_{SCCY}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x3}} & \boldsymbol{A_{SCCZ}} \end{bmatrix}, \quad \boldsymbol{B_{SCC}} = \begin{bmatrix} \boldsymbol{B_{SCCX}} & \boldsymbol{0_{3x1}} & \boldsymbol{0_{3x1}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x1}} & \boldsymbol{B_{SCCY}} & \boldsymbol{0_{3x1}} & \boldsymbol{0_{3x3}} \\ \boldsymbol{0_{3x1}} & \boldsymbol{0_{3x1}} & \boldsymbol{B_{SCCZ}} & \boldsymbol{0_{3x3}} \end{bmatrix},$$
$$\boldsymbol{C_{SCC}} = \begin{bmatrix} \boldsymbol{C_{SCCX}} & \boldsymbol{0_{1x3}} & \boldsymbol{0_{1x3}} \\ \boldsymbol{0_{1x3}} & \boldsymbol{C_{SCCY}} & \boldsymbol{0_{1x3}} \\ \boldsymbol{0_{1x3}} & \boldsymbol{0_{1x3}} & \boldsymbol{C_{SCCZ}} \end{bmatrix}, \quad \boldsymbol{D_{SCC}} = [0], \boldsymbol{x_{SCC}} = \begin{bmatrix} \boldsymbol{x_{SCCX}} \\ \boldsymbol{x_{SCCY}} \\ \boldsymbol{x_{SCCZ}} \end{bmatrix} \tag{4.29}$$

Oluşturulan bu insan algısı durum uzay modelleri, MATLAB yapısını oluştururken MOHAY'ın birleşik durum uzay formuna eklenecektir.

4.2.2.2 Otolit Taşlarının Matematiksel Modellenmesi

Sistemin matematiksel modeli Şekil 4.5'teki gibidir. Transfer fonksiyonun parametreleri Çizelge 4.5'te verilmiştir.



Şekil 4.5 Otolit Taşları Yapısının Blok Diyagram

f → Pilota/sürücüye etki eden özelleştirilmiş kuvvet değeri.

d → Otolit taşlarının dinamiğinden çıkan ivme değeri.

D → İnsan algı sınırlaması sonucu elde edilen ivme değeri.

$\hat{f}$ → Nöral aktivite sonrasında oluşan insan algısı.

Çizelge 4.5 Otolit taşları matematiksel modele ait parametre değerleri [28]

	$\boldsymbol{a_x}$	$\boldsymbol{a_y}$	$\boldsymbol{a_z}$
$\tau_l[s]$	5	5	5
$\tau_s[s]$	0,66	0,66	0,66
$\tau_a[s]$	10	10	10
K	0,4	0,4	0,4
$dTH\ [m/s^2]$	0,17	0,17	0,28

Yarı dairesel kanal sisteminde olduğu gibi, otolit taşları için de durum uzay modeli çıkarılacak olursa;

$$\frac{\hat{f}}{f} = \frac{K\tau_a {\tau_L}^{-1}{\tau_s}^{-1}(s + {\tau_a}^{-1})}{s^2 + ({\tau_L}^{-1} + {\tau_s}^{-1})s + {\tau_L}^{-1}{\tau_s}^{-1}} \tag{4.30}$$

$$a_1 = {\tau_L}^{-1} + {\tau_s}^{-1}, \quad a_2 = {\tau_L}^{-1}{\tau_s}^{-1} \tag{4.31}$$

$$b_1 = K\tau_a {\tau_L}^{-1}{\tau_s}^{-1}, \quad b_2 = K{\tau_L}^{-1}{\tau_s}^{-1} \tag{4.32}$$

$$\beta_0 = 0, \quad \beta_1 = b_1, \quad \beta_2 = b_2 - a_1 b_1 \tag{4.33}$$

$$x_1 = \hat{f} - \beta_0 f = \hat{f} \tag{4.34}$$

$$x_2 = \dot{\hat{f}} - \beta_0 \dot{f} - \beta_1 f = \dot{\hat{f}} - \beta_1 f \tag{4.35}$$

$$\dot{x}_1 = x_2 + \beta_1 f = x_2 + K\tau_a {\tau_L}^{-1}{\tau_s}^{-1} f \tag{4.36}$$

$$\dot{x}_2 = -a_2 x_1 - a_1 x_2 + \beta_2 f \tag{4.37}$$

$$h = (K{\tau_L}^{-1}{\tau_s}^{-1}) - ({\tau_L}^{-1} + {\tau_s}^{-1})(K\tau_a {\tau_L}^{-1}{\tau_s}^{-1}) \tag{4.38}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ (-{\tau_L}^{-1}{\tau_s}^{-1}) & -({\tau_L}^{-1} + {\tau_s}^{-1}) \end{bmatrix}}_{\boldsymbol{A_{oto}}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} K{\tau_L}^{-1}{\tau_s}^{-1} \\ h \end{bmatrix}}_{\boldsymbol{B_{oto}}} \tag{4.39}$$

$$\boldsymbol{C_X} = \boldsymbol{C_Y} = \boldsymbol{C_Z} = [1 \quad 0] \tag{4.40}$$

Platformda oluşan açısal hareketler, pilot üzerinde yer çekimi etkisinden dolayı eksenel yönde ivme etkisi oluşturacaktır. Bu etki sistemde eğim etkisinin (Tilt Coordination) de eklenmesi ile artmaktadır. Dolayısıyla, yerçekiminin insan üzerinde oluşturduğu eksenel yöndeki histen dolayı platform eksenele olarak ters yöne doğru sıfırlama durumuna geçecektir. Böylelikle platform sonraki hareket planı için hazır halde olacaktır. Platform eğiminin otolit taşları modeline eklenmesi aşağıdaki gibi gösterilmiştir. $\boldsymbol{f}$ özelleştirilmiş kuvvet vektörü, platformun açısal konumundan dolayı yerçekiminin, insan hissiyatına olan etkisini göstermektedir.

$$\boldsymbol{f} = \boldsymbol{H}[\,\phi\ \theta\ \Psi\ a_x\ a_y\ a_z]^T \tag{4.41}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & g & 0 & 1 & 0 & 0 \\ g & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.42}$$

(4.41) denkleminin sonucu, $\boldsymbol{f}$ vektörünün açılımı aşağıdaki gibi,

$$\boldsymbol{f} = \begin{bmatrix} a_x + g\theta \\ a_y + g\phi \\ a_z \end{bmatrix} \tag{4.43}$$

şeklindedir. Bu eğim etkisi, platformun kontrol girişlerinden biridir. Hem yuvarlanma, hem de yunuslama hareketi için ayrı 2 açısal veri sistemi tahrik etmektedir. Otolit taşlarının bütün öteleme eksenlerine uyarlanmış hali aşağıdaki gibi olmaktadır. Burada platformun açısal konumları da gösterilmiştir. [28] çalışmasındaki gibi birtakım makalelerde platformun açısal konumunu, eğim etkisiyle birleştirmişlerdir. Bu yüzden, 3 açı ifadesi de bu tezde otolit taşları matris denklemlerine eklenmiştir.

$$\boldsymbol{A_{OTO}} = \begin{bmatrix} \boldsymbol{A_{otox}} & \boldsymbol{0_{2x2}} & \boldsymbol{0_{2x2}} & \boldsymbol{0_{2x3}} \\ \boldsymbol{0_{2x2}} & \boldsymbol{A_{otoy}} & \boldsymbol{0_{2x2}} & \boldsymbol{0_{2x3}} \\ \boldsymbol{0_{2x2}} & \boldsymbol{0_{2x2}} & \boldsymbol{A_{otoz}} & \boldsymbol{0_{2x3}} \\ \boldsymbol{0_{3x2}} & \boldsymbol{0_{3x2}} & \boldsymbol{0_{3x2}} & \boldsymbol{I_{3x3}} \end{bmatrix}, \boldsymbol{B_{OTO}} = \begin{bmatrix} \boldsymbol{0_{2x3}} & \boldsymbol{B_{otox}} & \boldsymbol{0_{2x1}} & \boldsymbol{0_{2x1}} \\ \boldsymbol{0_{2x3}} & \boldsymbol{0_{2x1}} & \boldsymbol{B_{otoy}} & \boldsymbol{0_{2x1}} \\ \boldsymbol{0_{2x3}} & \boldsymbol{0_{2x1}} & \boldsymbol{0_{2x1}} & \boldsymbol{B_{otoz}} \end{bmatrix},$$

$$\boldsymbol{C_{OTO}} = \begin{bmatrix} \boldsymbol{C_X} & \boldsymbol{0_{1x2}} & \boldsymbol{0_{1x2}} & \boldsymbol{0_{1x3}} \\ \boldsymbol{0_{1x2}} & \boldsymbol{C_Y} & \boldsymbol{0_{1x2}} & \boldsymbol{0_{1x3}} \\ \boldsymbol{0_{1x2}} & \boldsymbol{0_{1x2}} & \boldsymbol{C_Z} & \boldsymbol{0_{1x3}} \end{bmatrix}, \boldsymbol{D_{OTO}} = [0], \boldsymbol{x_{OTO}} = \begin{bmatrix} \boldsymbol{x_{otox}} \\ \boldsymbol{x_{otoy}} \\ \boldsymbol{x_{otoz}} \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{4.44}$$

4.2.3 Bütünleşik Durum Uzay Denklemleri

İnsan algı modellerinin birleşiminden oluşan durum uzay modelleri (4.46) denkleminde gösterilmiştir. Bu modeller uçağın 6 eksendeki hareketlerin tamamını vermektedir. Sistemin bu aşamaya kadar elde edilen sonuç vektörü ise (4.47)'de belirtilmiştir.

MPC algoritması için kullanılması gereken integral durumları için matrisler yeniden düzenlendiğinde, durum matrisine $\boldsymbol{A_e}$ matrisin eklenmesi gerekmektedir. Bu integral durumları, platformun sıfırlama durumuna geçmesi için kullanılmaktadır;

$$\boldsymbol{A_e} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.45}$$

Bütün eksenlerin birleşiminden sonra oluşan VS durum uzay modeli;

$$\boldsymbol{A_v} = \begin{bmatrix} \boldsymbol{A_{SCC}} & \boldsymbol{0_{9x6}} & \boldsymbol{0_{9x3}} & \boldsymbol{0_{9x6}} \\ \boldsymbol{0_{6x9}} & \boldsymbol{A_{OTO}} & \boldsymbol{0_{6x3}} & \boldsymbol{0_{6x6}} \\ \boldsymbol{0_{3x9}} & \boldsymbol{0_{3x6}} & \boldsymbol{0_{3x3}} & \boldsymbol{0_{3x6}} \\ \boldsymbol{0_{6x9}} & \boldsymbol{0_{6x6}} & \boldsymbol{0_{6x3}} & \boldsymbol{A_e} \end{bmatrix}, \boldsymbol{B_v} = \begin{bmatrix} & & \boldsymbol{B_{SCC}} & & & \\ & & \boldsymbol{B_{OTO}} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{C_v} = \begin{bmatrix} \boldsymbol{C_{SCC}} & \boldsymbol{0_{3x6}} & \boldsymbol{0_{3x6}} \\ \boldsymbol{0_{3x6}} & \boldsymbol{C_{OTO}} & \boldsymbol{0_{3x6}} \\ \boldsymbol{0_{9x6}} & \boldsymbol{0_{9x6}} & \boldsymbol{I_{9x9}} \end{bmatrix} \tag{4.46}$$

$$\boldsymbol{Y_v} = [\underbrace{\hat{\omega}_x\ \hat{\omega}_y\ \hat{\omega}_z}_{x_{scc}}\ \underbrace{\hat{f}_x\ \hat{f}_y\ \hat{f}_z\ \phi\ \theta\ \Psi}_{x_{oto}}\ \underbrace{p_x v_x\ p_y v_y\ p_z v_z}_{x_{MPC}}] \tag{4.47}$$

şeklindedir. Burada $\boldsymbol{x_{scc}}$ yarı dairesel kanalların etkisi sonucu oluşan hareket hissi, $\boldsymbol{x_{oto}}$ otolit taşlarının etkisi ile oluşan öteleme hareketi hissi ve platforma ait açısal konumlar, $\boldsymbol{x_{MPC}}$ MOHAY için eklenen platformun sıfırlanma hareketini yapmasını sağlayan simülasyondaki helikopterin eksenel yöndeki konum ve hızlarını belirtmektedir.

4.2.4 Eğim Etkisinin Durum Uzay Matrislerine Eklenmesi

Klasik algoritmada kullanılan eğim etkisi filtre yapısı, MOHAY'da otolit taşlarına ait filtre katsayılarıyla çarpılarak VS denklemine eklenmektedir. Bu durumda VS'e ait matris takımının son hali,

$$\boldsymbol{Tilt_A} = \begin{bmatrix} \mathbf{0_{9x1}} & \mathbf{0_{9x1}} \\ 0 & K{\tau_L}^{-1}{\tau_s}^{-1} \\ 0 & h \\ K{\tau_L}^{-1}{\tau_s}^{-1} & 0 \\ h & 0 \\ \mathbf{0_{11x1}} & \mathbf{0_{11x1}} \end{bmatrix}, \tag{4.48}$$

$$\boldsymbol{A_m} = \begin{bmatrix} \boldsymbol{A_v} & \boldsymbol{Tilt_A} \\ \mathbf{0_{2x24}} & \mathbf{0_{2x2}} \end{bmatrix}, \boldsymbol{B_m} = \begin{bmatrix} \boldsymbol{B_v} & \mathbf{0_{24x2}} \\ \mathbf{0_{2x6}} & \boldsymbol{I_{2x2}} \end{bmatrix}, \boldsymbol{C_m} = [\boldsymbol{C_{vest}} \quad \mathbf{0_{15x2}}] \tag{4.49}$$

4.2.5 Genel Sistemin Durum Uzay Modellerinin Oluşturulması

Kontrol sırasında kullanılacak olan model matrisleri (4.49)'da gösterildiği gibidir. Bu denklemdeki matrislerin tamamı sürekli zamanlı yapıdan, ayrık zamanlı yapıya MATLAB aracılığı ile dönüştürülmüştür. Modelin durum matrisi 41 adet durum değişkeninden oluşmaktadır (geçmiş değerlere ait 26 adet sistemin durum farkları (Δx_{vest}) ve 15 adet sistem çıkışları). Kontrol girişi 8 elemandan meydana gelmektedir (Öngörülen açısal ivmeler, çizgisel hızlar ve eğim etkisi).

VS'in durum uzay modeli, önceki bölümlerde anlatılan MPC formülasyonu şekline dönüştürülürse aşağıdaki gibi olmaktadır;

$$\begin{aligned} \boldsymbol{X_m}(k+1|t) &= \boldsymbol{A_m}\boldsymbol{X_m}(k|t) + \boldsymbol{B_m}\Delta\boldsymbol{u}(k|t) \\ \boldsymbol{y_m}(k|t) &= \boldsymbol{C_m}\boldsymbol{X_m}(k|t) \end{aligned} \tag{4.50}$$

Öngörü durum uzay modellerinden yola çıkarak nihai birleşik model öngörü sistemi (4.51) denkleminde belirtilmiştir;

$$\begin{aligned} &\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A_m} & \mathbf{0_{26x15}} \\ \boldsymbol{C_m}\boldsymbol{A_m} & \boldsymbol{I_{15x15}} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B_m} \\ \boldsymbol{C_m}\boldsymbol{B_m} \end{bmatrix}, \\ &\boldsymbol{C} = [\mathbf{0_{15x26}} \quad \boldsymbol{I_{15x15}}], \boldsymbol{X_m} = \begin{bmatrix} \Delta\boldsymbol{X_m} \\ \boldsymbol{y} \end{bmatrix} \end{aligned} \tag{4.51}$$

Çizelge 4.6 Sonuç matrislerinde kullanılan parametrelerin içerikleri

Roll	Pitch	Yaw	Otolit
$A_{SCCR1} = \frac{-1}{T_{LR}T_sT_a}$	$A_{SCCP1} = \frac{-1}{T_{LP}T_sT_a}$	$A_{SCCP1} = \frac{-1}{T_{LY}T_sT_a}$	$A_{OTO1} = -{\tau_L}^{-1}{\tau_s}^{-1}$
$A_{SCCR2} = \frac{-T_{LR} - T_s - T_a}{T_{LR}T_sT_a}$	$A_{SCCP2} = \frac{-T_{LP} - T_s - T_a}{T_{LP}T_sT_a}$	$A_{SCCY2} = \frac{-T_{LY} - T_s - T_a}{T_{LY}T_sT_a}$	$A_{OTO2} = -({\tau_L}^{-1} + {\tau_s}^{-1})$
$A_{SCCR3} = -\left(\frac{1}{T_a}\frac{1}{T_s}\frac{1}{T_{LR}}\right)$	$A_{SCCP3} = -\left(\frac{1}{T_a}\frac{1}{T_s}\frac{1}{T_{LP}}\right)$	$A_{SCCY3} = -\left(\frac{1}{T_a}\frac{1}{T_s}\frac{1}{T_{LY}}\right)$	$B_{OTO} = \begin{bmatrix} K{\tau_L}^{-1}{\tau_s}^{-1} \\ h \end{bmatrix}$
$a_{1R} = \frac{1}{T_a}\frac{1}{T_s}\frac{1}{T_{LR}}$	$a_{1P} = \frac{1}{T_a}\frac{1}{T_s}\frac{1}{T_{LP}}$	$a_{1Y} = \frac{1}{T_a}\frac{1}{T_s}\frac{1}{T_L}$	$h = (K{\tau_L}^{-1}{\tau_s}^{-1}) - ({\tau_L}^{-1} + {\tau_s}^{-1})(K{\tau_a}{\tau_L}^{-1}{\tau_s}^{-1})$
$a_{2R} = \frac{1}{T_sT_a}\frac{1}{T_{LR}T_a}\frac{1}{T_sT_{LR}}$	$a_{2P} = \frac{1}{T_sT_a}\frac{1}{T_{LP}T_a}\frac{1}{T_sT_{LP}}$	$a_{2Y} = \frac{1}{T_sT_a}\frac{1}{T_{LY}T_a}\frac{1}{T_sT_{LY}}$	
$b_1 = \frac{1}{T_s}$	$b_1 = \frac{1}{T_s}$	$b_1 = \frac{1}{T_s}$	

$$
\boldsymbol{A_m} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A_{SCCR1} & A_{SCCR2} & A_{SCCR3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & A_{SCCP1} & A_{SCCP2} & A_{SCCP3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & A_{SCCY1} & A_{SCCY2} & A_{SCCY3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K\tau_L^{-1}\tau_s^{-1}g \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{OTO1} & A_{OTO2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & hg \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K\tau_L^{-1}\tau_s^{-1}g & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{OTO1} & A_{OTO2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & hg & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{OTO1} & A_{OTO2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$\boldsymbol{B_m} = \begin{bmatrix}
b_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-a_{1R}b_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
({a_{1R}}^2 b_1 - a_{2R}b_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & b_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -a_{1P}b_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & ({a_{1P}}^2 b_1 - a_{2P}b_1) & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & b_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -a_{1Y}b_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & ({a_{1Y}}^2 b_1 - a_{2Y}b_1) & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & K{\tau_L}^{-1}{\tau_s}^{-1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & h & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & K{\tau_L}^{-1}{\tau_s}^{-1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & h & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & K{\tau_L}^{-1}{\tau_s}^{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & h & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$C_m = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} \widehat{\omega}_x \\ \widehat{\omega}_y \\ \widehat{\omega}_z \\ \hat{f}_x \\ \hat{f}_y \\ \hat{f}_z \\ \phi \\ \theta \\ \Psi \\ p_x \\ v_x \\ p_y \\ v_y \\ p_z \\ v_z \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ f_x \\ f_y \\ f_z \\ \phi_T \\ \theta_T \end{bmatrix}, \quad \boldsymbol{x_m} = \begin{bmatrix} x_{SCCX1} \\ x_{SCCX2} \\ x_{SCCX3} \\ x_{SCCY1} \\ x_{SCCY2} \\ x_{SCCY3} \\ x_{SCCZ1} \\ x_{SCCZ2} \\ x_{SCCZ3} \\ x_{OTOX1} \\ x_{OTOX2} \\ x_{OTOY1} \\ x_{OTOY2} \\ x_{OTOZ1} \\ x_{OTOZ2} \\ \phi \\ \theta \\ \Psi \\ p_x \\ v_x \\ p_y \\ v_y \\ p_z \\ v_z \end{bmatrix}$$

4.2.6 Maliyet Fonksiyonu ve Sistemin Optimizasyon Aşamaları

Genel olarak, MPC uygulamalarında kullanılan maliyet fonksiyonu terimi aşağıdaki skaler fonksiyon ile gösterilebilir:

$$J = \sum_{k=1}^{N_p} [\bar{\boldsymbol{R}}(k) - \boldsymbol{Y}(k)]^T \boldsymbol{Q_i} [\bar{\boldsymbol{R}}(k) - \boldsymbol{Y}(k)] + \sum_{k=1}^{N_u - 1} [\boldsymbol{U^T}(k) \boldsymbol{P_i} \boldsymbol{U}(k) + \Delta \boldsymbol{U}^T \boldsymbol{R_i} \Delta \boldsymbol{U}] \quad (4.52)$$

Bu fonksiyonda N_p sistem çıkışlarının grup olarak öngörü sayısı, N_u sistem giriş gruplarının öngörü sayısıdır. $\boldsymbol{Q_i} \in \mathbb{R}^{pxp}$, $\boldsymbol{P_i} \in \mathbb{R}^{p_c x p_c}$, ve $\boldsymbol{R_i} \in \mathbb{R}^{p_c x p_c}$ simetrik ve pozitif yarı belirli (semi definite) ağırlık matrislerdir. MPC yönteminde genellikle $\boldsymbol{Q_i} = q\boldsymbol{I_m}$, $\boldsymbol{P_i} = p\boldsymbol{I_r}$ ve $\boldsymbol{R_i} = r_0 \boldsymbol{I_r}$ seçilmektedir. Burada $I_{m,r}$ birim matris, q,p ve r_0 ise pozitif parametrelerdir. MIMO sistemde her bir değer için farklı kontrol kazançları gerekecektir. Denklemde görüldüğü üzere ilk değer referans pozisyon değerinin tahmin edilen pozisyon değeri ile arasındaki farkı, 2. değer öngörülen input değeri ve son değer, kontrol girişinin değişimi üzerinden maliyet fonksiyonunu hesaplamaktadır.

Denklem parametreleri, oluşan hataya ve sistem yapısına göre seçilmelidir. **Q**, **P** ve **R** matrisleri pozitif yarı belirli ve N_p öngörü sayısı yüksek seçilirse sistemin kontrol edilebilirliği de artmaktadır.

Sınırlar dâhil edilmeden, maliyet fonksiyonunun çözümü yapıldığında aşağıda belirtilen yol izlenmektedir. Burada maliyet fonksiyonunu minimize etmek için (4.52) denkleminin kontrol girişi değişimine göre türevi alınır ve sıfıra eşitlenir.

$$\frac{\partial J}{\partial \Delta \boldsymbol{U}} = -2\boldsymbol{\varphi}^T \boldsymbol{Q}[\bar{\boldsymbol{R}}(k) - \boldsymbol{F}\boldsymbol{X}(k)] + 2[\boldsymbol{\varphi}^T \boldsymbol{Q} \boldsymbol{\varphi} + \boldsymbol{R}] \Delta \boldsymbol{U} = 0 \quad (4.53)$$

Son olarak ΔU denklemde sol tarafa atılır ve maliyet fonksiyonundan elde edilen kontrol girişi farkı elde edilir:

$$\Delta \boldsymbol{U} = [\boldsymbol{\varphi}^T \boldsymbol{Q} \boldsymbol{\varphi} + \boldsymbol{R}]^{-1} \boldsymbol{\varphi}^T \boldsymbol{Q}[\bar{\boldsymbol{R}}(k) - \boldsymbol{F}\boldsymbol{X}(k)] \quad (4.54)$$

$$\boldsymbol{K_x} = [\boldsymbol{\varphi}^T \boldsymbol{Q} \boldsymbol{\varphi} + \boldsymbol{R}]^{-1} \boldsymbol{\varphi}^T \boldsymbol{Q} \quad (4.55)$$

$$\boldsymbol{K_y} = -[\boldsymbol{\varphi}^T \boldsymbol{Q} \boldsymbol{\varphi} + \boldsymbol{R}]^{-1} \boldsymbol{\varphi}^T \boldsymbol{Q} \boldsymbol{F} \quad (4.56)$$

$$\Delta \boldsymbol{U} = \boldsymbol{K_x} \bar{\boldsymbol{R}}(k) + \boldsymbol{K_y} \boldsymbol{X}(k) \quad (4.57)$$

Elde edilen (4.57) denklemi sistem sınırları belirtilmeden bulunan bir sonuçtur. Çalışmada [30] literatürü temel alınarak maliyet fonksiyonu kuadratik programlama

kullanarak çözümlenecektir. Modelimizde sınırlı çalışma uzayı olduğundan bu kuadratik yapı, sınırlamalara tabi tutulacaktır. Sistemin hareketi sınırlamalarla belirlenecek ve maliyet fonksiyonu da buna göre kendisini iyileştirecektir. Modelde maliyet fonksiyonu için, sistem hatası ve kontrol girişi farkı temel alınmıştır. Bunun için maliyet fonksiyonu kuadratik regülatör şekline dönüştürülecektir.

$$\boldsymbol{M}.\Delta\boldsymbol{U} \leq \boldsymbol{b} \tag{4.58}$$

Şeklindedir. Sınırlandırma yapılan veriler kontrol girişleri, kontrol girişlerinin değişimi ve çıkış vektörünün son 9 değeri olacaktır. Bunlar parametrik olarak aşağıdaki gibi gösterilebilir;

$$\boldsymbol{MV} = \boldsymbol{U} = \left[\omega_x\ \omega_y\ \omega_z\ a_x\ a_y\ a_z\ \dot{\phi}_a\ \dot{\theta}_a\right] \tag{4.59}$$

$$\Delta\boldsymbol{MV} = \Delta\boldsymbol{U} = \left[\Delta\omega_x\ \Delta\omega_y\ \Delta\omega_z\ \Delta a_x\ \Delta a_y\ \Delta a_z\ \Delta\dot{\phi}_a\ \Delta\dot{\theta}_a\right] \tag{4.60}$$

$$\boldsymbol{CV} = \left[\boldsymbol{MV}\ \Delta\boldsymbol{MV}\ \phi\ \theta\ \Psi\ p_x v_x\ p_y v_y\ p_z v_z\right] \tag{4.61}$$

Burada $\boldsymbol{MV}$ ve $\Delta\boldsymbol{MV}$ değerleri manipüle edilmiş parametreler olmaktadır (Manipulated Variables). Bu değerler aynı zamanda simülatöre giden kontrol girişi değişimlerine bağlı olan kontrol girişi değerleridir. Sınır şartındaki $\boldsymbol{M}$ ve $\boldsymbol{b}$ matrisleri denklem (4.62)'teki gibidir.

$$\begin{bmatrix} \boldsymbol{\delta} \\ -\boldsymbol{\delta} \\ \boldsymbol{I} \\ -\boldsymbol{I} \\ \boldsymbol{\varphi_c} \\ -\boldsymbol{\varphi_c} \end{bmatrix}.\Delta\boldsymbol{U} \leq \begin{bmatrix} \boldsymbol{U_{max}} - \boldsymbol{U_t} \\ -\boldsymbol{U_{min}} + \boldsymbol{U_t} \\ \Delta\boldsymbol{U_{max}} \\ -\Delta\boldsymbol{U_{min}} \\ \boldsymbol{Y_{max}} - \boldsymbol{F_c X} \\ -\boldsymbol{Y_{min}} + \boldsymbol{F_c X} \end{bmatrix} \tag{4.62}$$

Bu denklemde $\boldsymbol{U_{max}}, \boldsymbol{U_{min}}, \Delta\boldsymbol{U_{max}}, \Delta\boldsymbol{U_{min}}, \boldsymbol{Y_{max}}$ ve $\boldsymbol{Y_{min}}$ sistemi sınırlandıran elemanlar olup, platformun ve insan algısının sınırlamalarına göre belirlenecektir. (4.52) denklemindeki maliyet fonksiyonunda $\boldsymbol{Y}$ yerine MPC algoritması için hazırlanan (4.14) denklemi eklendiğinde ve sistemdeki hata ile kontrol girişi ve kontrol girişinin değişimi temel alındığında aşağıdaki gibi bir formül yapısı oluşmaktadır;

$$J = [\bar{\boldsymbol{R}}(k) - \boldsymbol{FX}(k)]^T[\bar{\boldsymbol{R}}(k) - \boldsymbol{FX}(k)] + \boldsymbol{U_t}^T\boldsymbol{SU_t} - 2\Delta\boldsymbol{U}^T[\boldsymbol{\varphi}^T\boldsymbol{Q}[\bar{\boldsymbol{R}}(k) - \boldsymbol{FX}(k)] + \boldsymbol{U_t}\boldsymbol{S\delta}] + \Delta\boldsymbol{U}^T[\boldsymbol{\varphi}^T\boldsymbol{Q\varphi} + \boldsymbol{R} + \boldsymbol{\delta}^T\boldsymbol{S\delta}]\Delta\boldsymbol{U} \quad (4.63)$$

Maliyet fonksiyonu Kuadratik Programlama formuna dönüşmüştür. Maliyet Fonksiyonunu minimize etmek için Lagrange Çarpanı metodu kullanılmıştır. Böylelikle minimize sırasında formüle platformun giriş ve çıkış verileri ile ilgili sınırlamalar da eklenmiştir. Öncelikle maliyet fonksiyonunu kuadratik şekliyle belirtecek olursak,

$$J = \frac{1}{2}\Delta\boldsymbol{U}^T \underbrace{2[\boldsymbol{\varphi}^T\boldsymbol{Q\varphi} + \boldsymbol{R} + \boldsymbol{\delta}^T\boldsymbol{S\delta}]}_{\boldsymbol{E}}\Delta\boldsymbol{U} + \Delta\boldsymbol{U}^T \underbrace{\boldsymbol{2}[\boldsymbol{\varphi}^T\boldsymbol{Q}[\boldsymbol{FX}(k) - \bar{\boldsymbol{R}}(k)] + \boldsymbol{U_t}\boldsymbol{S\delta}]}_{\boldsymbol{F_1}} + \underbrace{[\bar{\boldsymbol{R}}(k) - \boldsymbol{FX}(k)]^T[\bar{\boldsymbol{R}}(k) - \boldsymbol{FX}(k)] + \boldsymbol{U_t}^T\boldsymbol{SU_t}}_{c} \quad (4.64)$$

$$J = \frac{1}{2}\Delta\boldsymbol{U}^T\boldsymbol{E}\Delta\boldsymbol{U} + \Delta\boldsymbol{U}^T\boldsymbol{F_1} + c \quad (4.65)$$

Şeklinde olmaktadır. Sınırları ifade eden $\boldsymbol{M}\Delta\boldsymbol{U} \leq \boldsymbol{b}$ ifadesi eşitliğe dönüştürülerek $\boldsymbol{M}\Delta\boldsymbol{U} = \boldsymbol{b}$ şeklinde yazılır ve quadratik fonksiyona eklerken bir lagrange çarpanı ile eklenir. $\boldsymbol{\lambda}$ lagrange çarpanı olmak üzere sınırların sisteme dâhil edilmesiyle oluşan yeni kuadratik fonksiyon,

$$J = \frac{1}{2}\Delta\boldsymbol{U}^T\boldsymbol{E}\Delta\boldsymbol{U} + \Delta\boldsymbol{U}^T\boldsymbol{F_1} + c + \boldsymbol{\lambda}^T(\boldsymbol{M}\Delta\boldsymbol{U} - \boldsymbol{b}) \quad (4.66)$$

şeklindedir. Oluşan maliyet fonksiyonunun $\Delta\boldsymbol{U}$ ve $\boldsymbol{\lambda}$ cinslerinden kısmi türevleri alındığında,

$$\frac{\partial J}{\partial \Delta\boldsymbol{U}} = \boldsymbol{E}\Delta\boldsymbol{U} + \boldsymbol{F_1} + \boldsymbol{M}^T\boldsymbol{\lambda} = \boldsymbol{0} \quad (4.67)$$

$$\frac{\partial J}{\partial \boldsymbol{\lambda}} = \boldsymbol{M}\Delta\boldsymbol{U} - \boldsymbol{b} = \boldsymbol{0} \quad (4.68)$$

Denklemleri elde edilir. Her iki denklem de birbiri cinsinden çözüldüğünde, kontrol girişi değişiminin ifadesi aşağıdaki gibi olmaktadır.

$$\boldsymbol{\lambda} = -[\boldsymbol{ME^{-1}M}^T]^{-1}[\boldsymbol{b} + \boldsymbol{ME^{-1}F_1}] \quad (4.69)$$

$$\Delta\boldsymbol{U} = -\boldsymbol{E^{-1}}[\boldsymbol{M^T\lambda} + \boldsymbol{F_1}] \quad (4.70)$$

4.2.6.1 Hildreth Kuadratik Programlama Algoritması

Hildreth'in algoritması lagrange çarpanı olan λ parametresinin $\lambda \geq 0$ olmasını sağlamaktadır. Bunu yaparken, belirli bir iterasyon sayısı kadar, elde edilen sınır sayısı kadar λ değeri hesaplanır. Hesaplanan λ parametreleri eğer $\lambda < 0$ ise algoritma otomatik olarak $\lambda = 0$ değerini atar. Yine her iterasyonda elde edilen λ değerleri bir önceki yaklaşımdaki değer ile karşılaştırılır. Bu eşitlik, eğer belirtilen hassasiyet miktarı kadarını sağlıyor ise λ değeri atanır. Bu hesaplamadaki amaç, normal lagrange çarpanı metodu ile sınırlarda oluşan lineer bağımlılık ilkesinden dolayı elde edilemeyen λ çarpanını aktif ve pasif sınır şartları kullanarak elde etmektir. Her $\lambda = 0$ değeri için maliyet fonksiyonu içerisinde o değere ait sınır şartı elenir. Bunda amaç, sınır şartı aranan $\Delta \boldsymbol{U}(k)$ değeri için en kısıtlı aralığın belirlenmesi ve bu aralığın dışında olan değerlerin elenmesidir.

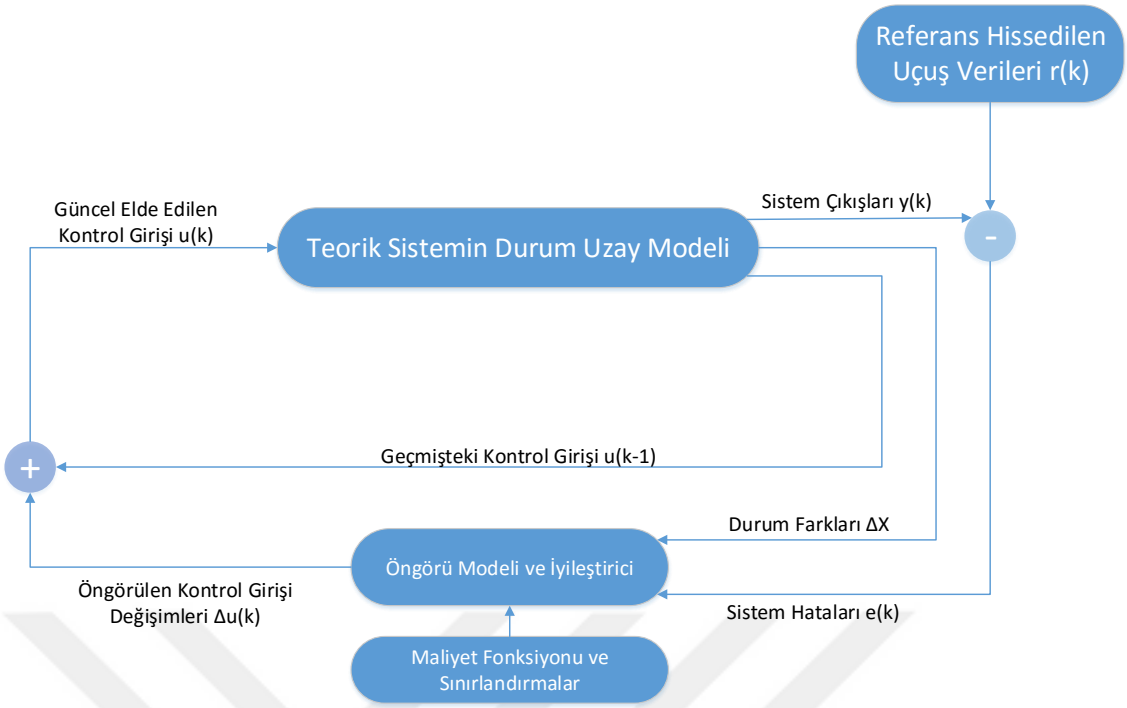
$$\boldsymbol{\lambda_i^{m+1}} = \boldsymbol{max}(\mathbf{0}, \boldsymbol{w_i^{m+1}}) \tag{4.71}$$

$$\boldsymbol{w_i^{m+1}} = -\frac{1}{h_{ii}}[k_i \sum_{\boldsymbol{j=1}}^{\boldsymbol{i-1}} h_{ij}\lambda_j^{m+1} + \sum_{\boldsymbol{j=i+1}}^{\boldsymbol{n}} h_{ij}\lambda_j^{m}] \tag{4.72}$$

Burada h_{ij} değeri, $\boldsymbol{H} = \boldsymbol{ME^{-1}M^T}$ matirisinin ij. elemanı, k_i değeri $\boldsymbol{K} = \boldsymbol{\gamma} + \boldsymbol{ME^{-1}F_1}$ matrisinin i. elemanı olmaktadır. Kullanıcı tarafından belirtilen iterasyon sayısı kadar hesaplama sonucunda, yine tasarımcı tarafından belirlenen hata payının altında kalacak şekilde λ_i^{m+1} hesaplanır. Bütün iterasyon tamamlandığında bütün belirlenen katsayıları içeren λ^* vektörü elde edilir ve kontrol girişi değişimi $\Delta \boldsymbol{U}$ elde edilir.

$$\Delta \boldsymbol{U} = -\boldsymbol{E}[\boldsymbol{F_1} + \boldsymbol{M^T}\boldsymbol{\lambda}^*] \tag{4.73}$$

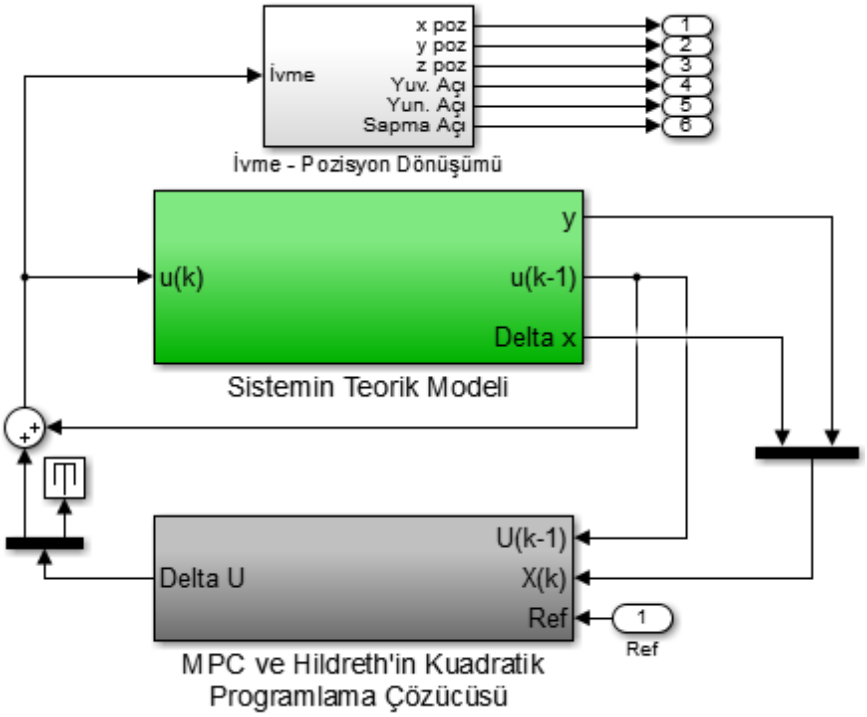
MATLAB üzerinde oluşturulacak simulink blok şemasına ait sembolik akış diyagramı Şekil 4.6'da gösterilmiştir.



Şekil 4.6 Model Öngörülü Kontrol Sistemi Akış Şeması

Model öngörülü kontrolün MATLAB/Simulink'te oluşturulan modeli Şekil 4.7'de gösterilmiştir. Hildreth'in kuadratik programlama çözücüsü yazılımı EK-C'de belirtilmiştir ve Matlab fonksiyon bloğu olarak Şekil 4.7'de bulunan alt sistemin içinde yer almaktadır. Burada Hildreth'in kuadratik programlama çözücüsüne helikopterden alınan uçuş verileri, insan algısı transfer fonksiyonlarından geçirilerek referans olarak gönderilmektedir. Her örnekleme zamanında bu işlem yeni referans değerleriyle tekrarlanmaktadır. Kuadratik programlama çözücüsü içinde gönderilen referans verileri, N_p kadar çoğaltılır ve çözücü sayesinde hesaplanır. (4.73) denklemi ile elde edilen kontrol girişi değişimleri, üst platform orta noktası koordinatı öteleme ivmeleri ve açısal hız değişimleri şeklinde N_u öngörü sayısı kadar $\Delta \boldsymbol{u}$ kontrol vektörü ile dışarı alınır. Bu vektörlerin sadece ilk örnekleme zamanındaki verileri kullanılır. Bu veriler, teorik model üzerinde önceden uygulanmış olan $\boldsymbol{u}(k-1)$ vektörü ile toplanarak yeni $\boldsymbol{u}(k)$ vektörü elde edilir. Mevcut kontrol girişi, teorik sistem üzerindeki insan algısı matematiksel modeline gönderilerek sistem çıkışlarını elde eder. Kontrol mekanizması bu teorik model üzerinden kendisini tekrarlarken, elde edilen $\boldsymbol{u}(k)$ kontrol girişlerinin integralleri alınarak platform kontrolcüsüne gönderilmektedir.

Model öngörülü kontrol yazılımının denklem yapısı, MATLAB kodu olarak EK-D'de belirtilmiştir. Simulink bloğunun içyapısı Şekil 4.7'deki gibi olmaktadır.



Şekil 4.7 Model öngörülü hareket algı algoritmasının MATLAB/Simulink yapısı

Maliyet fonksiyonunun çözülmesi sırasında ağırlık matrislerinin ayarlama işlemi yapılacaktır. Bunun dışında kullanıcı tarafından sisteme dâhil edilmesi gereken diğer parametreler de bulunmaktadır. Maliyet fonksiyonunda ayarlama yapılacak parametreler ve sistem sınırları tanımları Çizelge 4.7'de gösterilmiştir. Öngörü parametreleri, ağırlık matrisleri ve sistem sınırlamalarının açıklamaları Çizelge 4.8 ile Çizelge 4.11 arasında gösterilmiştir.

Çizelge 4.7 Maliyet fonksiyonunda kullanılan parametreler ve açıklamaları

Matris	Tanımı	Açıklama	Belirlenme Şekli
R	Input Değişimi Ağırlık Matrisi	Maliyet Fonksiyonundan çıkan input değişiminin genliğini ayarlar.	Kullanıcı belirler.
S	Input Ağırlık Matrisi	Sisteme giden inputların genliğini ayarlar.	Kullanıcı belirler.
Q	Hata Ağırlık Matrisi	Sistemden elde edilen çıkışlar ile referanslar arasındaki farkın, hatanın, miktarını ayarlar.	Kullanıcı belirler.
N_p	Öngörü Sayısı	MPC tarafından kaç örnekleme zamanı için sistem çıkışları tahminleri oluşturacağını belirler.	Kullanıcı belirler.
N_u	Input Öngörü Sayısı	MPC tarafından kaç örnekleme zamanı için kontrol girişi tahminleri oluşturulacağını belirler.	Kullanıcı belirler.
Y_{min}, Y_{max}	Max ve Min Platform Sınırları	Üst platform pozisyonlarının sınırlarını belirler.	Platform kapasitesine bağlıdır.
U_{min}, U_{max}	Max ve Min Kontrol Girişi Sınırları	Platforma gönderilen ivme ve açısal hız sınırlarını belirler	İnsan algı sınırlarına göre belirlenir.
ΔU_{min}, ΔU_{max}	Max ve Min Kontrol Girişi Değişimi Sınırları	Platforma gönderilen ivme ve açısal hız değişimi sınırlarını belirler	İnsan algı sınırlarına göre belirlenir.

Çizelge 4.8 Hata ağırlık matrisinin parametre açıklamaları.

	Matris Parametreleri																Aksiyon	Açıklama	Sınıf
Q - Genel Sistem Çıkışı Ağırlık Matrisi: Köşegen matristir. Köşegendeki her parametre o parametreye bağlı sistem çıkışı ile referans arasındaki hatayı düzenler.	Q_1																←	$\tilde{\phi}'$ın hata değerini düşürür.	İnsan hissiyatı ile ilgili katsayılar
		Q_2															←	$\tilde{\theta}'$ın hata değerini düşürür.	
			Q_3														←	$\tilde{\Psi}'$ın hata değerini düşürür.	
				Q_4													←	$\hat{a}_x{}'$in hata değerini düşürür.	
					Q_5												←	$\hat{a}_y{}'$nin hata değerini düşürür.	
						Q_6											←	$\hat{a}_z{}'$nin hata değerini düşürür.	
							Q_7										←	ϕ'nın sıfıra gitmesini zorlar.	Washout (Sıfırlama) katsayıları
								Q_8									←	θ'nın sıfıra gitmesini zorlar.	
									Q_9								←	Ψ'nın sıfıra gitmesini zorlar.	
										Q_{10}							←	$P_x{}'$in sıfırlanma hızını yükseltir.	
											Q_{11}						←	$V_x{}'$in sıfırlanma hızını yükseltir.	
												Q_{12}					←	$P_y{}'$nin sıfırlanma hızını yükseltir.	
													Q_{13}				←	$V_y{}'$nin sıfırlanma hızını yükseltir.	
														Q_{14}			←	$P_z{}'$nin sıfırlanma hızını yükseltir.	
															Q_{15}		←	$V_z{}'$nin sıfırlanma hızını yükseltir.	

Çizelge 4.9 Kontrol girişi ve kontrol girişi değişimi ağırlık matrislerinin parametre açıklamaları

	Matris Parametreleri								Aksiyon	Açıklama
R - Simülatör ΔU Ağırlık Matrisi: Platforma gönderilen açısal hız ve çizgisel ivmelerin değişimlerinin genliğini düzenleyen matristir.	R_1								↑	ΔU_ϕ'in genliğini düşürür.
		R_2							↑	ΔU_θ'in genliğini düşürür.
			R_3						↑	ΔU_ψ'in genliğini düşürür.
				R_4					↑	ΔU_x'in genliğini düşürür, $\Delta U_{\phi-Tilt}$'i yükseltir.
					R_5				↑	ΔU_y'nin genliğini düşürür, $\Delta U_{\theta-Tilt}$'i yükseltir.
						R_6			↑	ΔU_z'in değişiminin genliğini düşürür.
							R_7		↑	$\Delta U_{\theta-Tilt}$'in genliğini düşürür, ΔU_y'in genliğini yükseltir.
								R_8	↑	$\Delta U_{\phi-Tilt}$'in genliğini düşürür, ΔU_x'in genliğini yükseltir.
	Matris Parametreleri								**Aksiyon**	**Açıklama**
S - Simülatör Giriş Ağırlık Matrisi: Platforma gönderilen açısal hız ve çizgisel ivmelerin genliğini düzenleyen matristir.	S_1								↑	U_ϕ'in genliğini düşürür.
		S_2							↑	U_θ'in genliğini düşürür.
			S_3						↑	U_ψ'in genliğini düşürür.
				S_4					↑	U_x'in genliğini düşürür, $U_{\phi-Tilt}$'i yükseltir.
					S_5				↑	U_y'nin genliğini düşürür, $U_{\theta-Tilt}$'i yükseltir.
						S_6			↑	U_z'nin genliğini düşürür.
							S_7		↑	$U_{\theta-Tilt}$'i düşürür, U_y'yi artırır.
								S_8	↑	$U_{\phi-Tilt}$'i düşürür, U_x'i artırır.

Çizelge 4.10 Simülasyon durum uzay modelinin çıkış sınırları

Y_{max}	Y_{min}	Açıklama	Birimi
Y_{max1}	Y_{min1}	Üst platform orta noktasının ± ϕ yönünde ulaşabileceği maksimum açı değeri	rad
Y_{max2}	Y_{min2}	Üst platform orta noktasının ± θ yönünde ulaşabileceği maksimum açı değeri	rad
Y_{max3}	Y_{min3}	Üst platform orta noktasının ± Ψ yönünde ulaşabileceği maksimum açı değeri	rad
Y_{max4}	Y_{min4}	Üst platform orta noktasının ± P_x sınırları	m
Y_{max5}	Y_{min5}	Üst platform orta noktasının ± V_x sınırları	m/s
Y_{max6}	Y_{min6}	Üst platform orta noktasının ± P_ysınırları	m
Y_{max7}	Y_{min7}	Üst platform orta noktasının ± V_y sınırları	m/s
Y_{max8}	Y_{min8}	Üst platform orta noktasının ± P_z sınırları	m
Y_{max9}	Y_{min9}	Üst platform orta noktasının ± V_z sınırları	m/s

Y_{min}, Y_{max}: Platformun ulaşabileceği maks. ve min. hareket değerleridir.

Çizelge 4.11 Platforma gönderilen kontrol girişi ve kontrol girişi değişimlerinin sınırları

ΔU_{min}, ΔU_{max}: Üst platform orta noktasının açısal hız değişimi ve öteleme ivmelerinin değişiminin ulaşabileceği maks. ve min. değerleridir.

ΔU_{max}	ΔU_{min}	Açıklama	Birimi
ΔU_{max1}	ΔU_{min1}	Platforma gönderilen, üst platform orta noktası ± $\dot{\phi}$ değişimi sınırları	rad/s
ΔU_{max2}	ΔU_{min2}	Platforma gönderilen, üst platform orta noktası ± $\dot{\theta}$ değişimi sınırları	rad/s
ΔU_{max3}	ΔU_{min3}	Platforma gönderilen, üst platform orta noktası ± $\dot{\Psi}$ değişimi sınırları	rad/s
ΔU_{max4}	ΔU_{min4}	Platforma gönderilen, üst platform orta noktası ± a_x değişimi sınırları	m/s²
ΔU_{max5}	ΔU_{min5}	Platforma gönderilen, üst platform orta noktası ± a_y değişimi sınırları	m/s²
ΔU_{max6}	ΔU_{min6}	Platforma gönderilen, üst platform orta noktası ± a_z değişimi sınırları	m/s²
ΔU_{max7}	ΔU_{min7}	Platforma gönderilen, ± $\dot{\theta}_{Tilt}$ değişimi sınırları	rad/s
ΔU_{max8}	ΔU_{min8}	Platforma gönderilen, ± $\dot{\phi}_{Tilt}$ değişimi sınırları	rad/s

U_{min}, U_{max}: Üst platform orta noktasının açısal hız ve öteleme ivmelerinin ulaşabileceği maks. ve min. değerleridir.

U_{max}	U_{min}	Açıklama	Birimi
U_{max1}	U_{min1}	Platforma gönderilen, üst platform orta noktası ± $\dot{\phi}$ sınırları	rad/s
U_{max2}	U_{min2}	Platforma gönderilen, üst platform orta noktası ± $\dot{\theta}$ sınırları	rad/s
U_{max3}	U_{min3}	Platforma gönderilen, üst platform orta noktası ± $\dot{\Psi}$ sınırları	rad/s
U_{max4}	U_{min4}	Platforma gönderilen, üst platform orta noktası ± a_x sınırları	m/s²
U_{max5}	U_{min5}	Platforma gönderilen, üst platform orta noktası ± a_y sınırları	m/s²
U_{max6}	U_{min6}	Platforma gönderilen, üst platform orta noktası ± a_z sınırları	m/s²
U_{max7}	U_{min7}	Platforma gönderilen, ± $\dot{\theta}_{Tilt}$ sınırları	rad/s
U_{max8}	U_{min8}	Platforma gönderilen, ± $\dot{\phi}_{Tilt}$ sınırları	rad/s

4.2.7 Model Öngörülü Kontrolcünün Parametrelerinin Ayarlanması

Parametrelerin belirlenmesi sırasında bir yol haritası belirlenmiş ve ayarlamalar bu yol haritasına göre yapılmıştır. Simülasyon sırasında insan algısında oluşan hatanın azaltılması ilk hedef olup, kalan parametreler simülatörün hareket kapasitesi ve sürüşü yapan pilotun yorumlarına göre ayarlanmıştır. Bu ayarlama prosedürü ile ilgili maddeler aşağıdaki gibi sıralanabilir.

- Sistem, 100 örnekleme zamanı ile çalışmaktadır.
- N_p sistemin öngörülen durum ve çıkış sayısıdır. Yükseltilmesi durumunda sistemdeki algı hatasını azaltır ve washoutun daha hızlı yapılmasını sağlar.
- N_u sistemin öngörülen kontrol girişi sayısıdır. Platforma gönderilen kontrol girişlerini düzenleyerek sistemdeki hatayı azaltır.
- N_p ve N_u parametreleri simülasyonun başında belirlenir. Sistemin gelecek zamandaki referansları önceden öngörülemediğinden N_p parametresi olabildiği kadar yüksek tutulur.
- Modelde N_p=10, N_u=5 olarak seçilmiştir.
- Model oluşturulduktan sonra yapılan ilk simülasyon çalışmasında insan algı hatasına göre, maliyet fonksiyonunda bulunan **Q** matrisi ayarlanarak pilotun algı hatası düşürülür.
- Platforma gönderilen kontrol girişleri ve kontrol girişleri değişimlerine ait **R** ve **S** ağırlık matrisleri, **Q** matrisi belirlendikten sonra kalan algı hatasını düşürmek için ayarlanmaya başlanır.
- Eğim koordinasyonu için açısal hareket insan algısının altında bulunması gerektiğinden $\Delta \boldsymbol{U_{Tilt}}$ değerleri maksimum ve minimum noktalarda ±0.06 rad/s olarak belirlenmiştir.
- Eğim koordinasyonu hareketi insan algısının altında olması gerektiğinden sınırlı bir düzeyde kalmaktadır. Algı hissiyatı için, eksik kalan hareketin hissettirilmesinde eğim koordinasyonuna bağlı eksen takımı için **Q**, **R** ve **S** matrislerinde ayarlamalar yapılmalıdır. Gerekirse, bu eksen için sınırlandırılan hız değerinde de artışa gidilebilir.

- z eksenindeki harekette hissettirilmesi gereken g yükü, simülatör için oldukça yüksektir ve bu eksende eğim koordinasyonu gibi bir avantaj olmadığından bu hissiyatın pilot üzerinde uygulanması sırasında problemler oluşmaktadır. Belirli bir düzeyde z ekseni için ağırlık matrislerinin ayarlaması yapıldıktan sonra, pilotun hissiyatına göre kalibrasyon işlemlerine devam edilmektedir.
- Platformun fiziksel yapısına bağlı olan sınırlamalar, üst platformun ulaşabileceği maksimum eksen limitlerine göre ayarlanmıştır. Geriye kalan sınırlamalar, helikopter hareket halindeyken hareketin kişiye hissettirilmesi için insan algısı eşik değerlerinin üstünde olacak şekilde belirlenmiştir.
- Bütün temel ayarlamalar tamamlandıktan sonra pilot ile yapılan simülasyon çalışması sırasında pilottan elde edilen hissiyat yorumlarına göre parametre ayarlarına devam edilmektedir.

Deneysel sonuçlara geçmeden önce parametreler sistem koşulları da göz önünde bulundurularak ayarlanmıştır. Ayarlanan parametreler Çizelge 4.12 ile Çizelge 4.17 arasında sıralanmıştır.

Çizelge 4.12 Hata ağırlık matrisinin köşegen parametreleri

	Q Matrisi														
Parametre	$\hat{\dot{\phi}}$	$\hat{\dot{\theta}}$	$\hat{\dot{\Psi}}$	$\hat{a}_x$	$\hat{a}_y$	$\hat{a}_z$	ϕ	θ	Ψ	P_x	V_x	P_y	V_y	P_z	V_z
Parametrenin Ağırlık Değeri	50	50	50	35	30	30	50	50	50	25	30	25	30	35	50

Çizelge 4.13 Kontrol girişine ait ağırlık matrisinin köşegen parametreleri

	S Matrisi							
Parametre	$\dot{\phi}$	$\dot{\theta}$	$\dot{\Psi}$	a_x	a_y	a_z	$\dot{\theta}_{Tilt}$	$\dot{\phi}_{Tilt}$
Parametrenin Ağırlık Değeri	1	1	1	1	1	1	50	50

Çizelge 4.14 Kontrol girişi değişimine ait ağırlık matrisinin köşegen parametreleri

	R Matrisi							
Parametre	$\Delta\dot{\phi}$	$\Delta\dot{\theta}$	$\Delta\dot{\Psi}$	Δa_x	Δa_y	Δa_z	$\Delta\dot{\theta}_{Tilt}$	$\Delta\dot{\phi}_{Tilt}$
Parametrenin Ağırlık Değeri	1	1	1	1	1	1	100	100

Çizelge 4.15 Sistem çıkışlarına ait kontrolcü sınırları

	Sistem çıkışı sınırları (Ymin-Ymax)								
Parametre	ϕ	θ	Ψ	P_x	V_x	P_y	V_y	P_z	V_z
Birimi	deg			m	m/s	m	m/s	m	m/s
$\boldsymbol{Y_{min}}$	-0,33	-0,34	-0,47	-0,45	-0,8	-0,25	-0,8	-0,2	-0,4
$\boldsymbol{Y_{max}}$	0,33	0,34	0,47	0,25	0,8	0,25	0,8	0,3	0,4

Çizelge 4.16 Kontrol girişlerine ait sınırlar

	Kontrol Girişi Sınırları (Umin-Umax)							
Parametre	$\dot{\phi}$	$\dot{\theta}$	$\dot{\Psi}$	a_x	a_y	a_z	$\dot{\theta}_{Tilt}$	$\dot{\phi}_{Tilt}$
Birimi	Deg/s			m/s²			Deg/s	
$\boldsymbol{U_{min}}$	-0,7	-0,6	-0,85	-0,5	-0,5	-0,5	-0,06	-0,06
$\boldsymbol{U_{max}}$	0,7	0,6	0,85	0,5	0,5	0,5	0,06	0,06

Çizelge 4.17 Kontrol girişleri değişimlerine ait sınırlar

	Kontrol Girişi Değişimi Sınırları (dUmin-dUmax)							
Parametre	$\Delta\dot{\phi}$	$\Delta\dot{\theta}$	$\Delta\dot{\Psi}$	Δa_x	Δa_y	Δa_z	$\Delta\dot{\theta}_{Tilt}$	$\Delta\dot{\phi}_{Tilt}$
Birimi	Deg/s			m/s²			Deg/s	
$\Delta\boldsymbol{U_{min}}$	-0,5	-0,5	-0,5	-0,6	-0,6	-0,2	-0,06	-0,06
$\Delta\boldsymbol{U_{max}}$	0,5	0,5	0,5	0,6	0,6	0,2	0,06	0,06

BÖLÜM 5

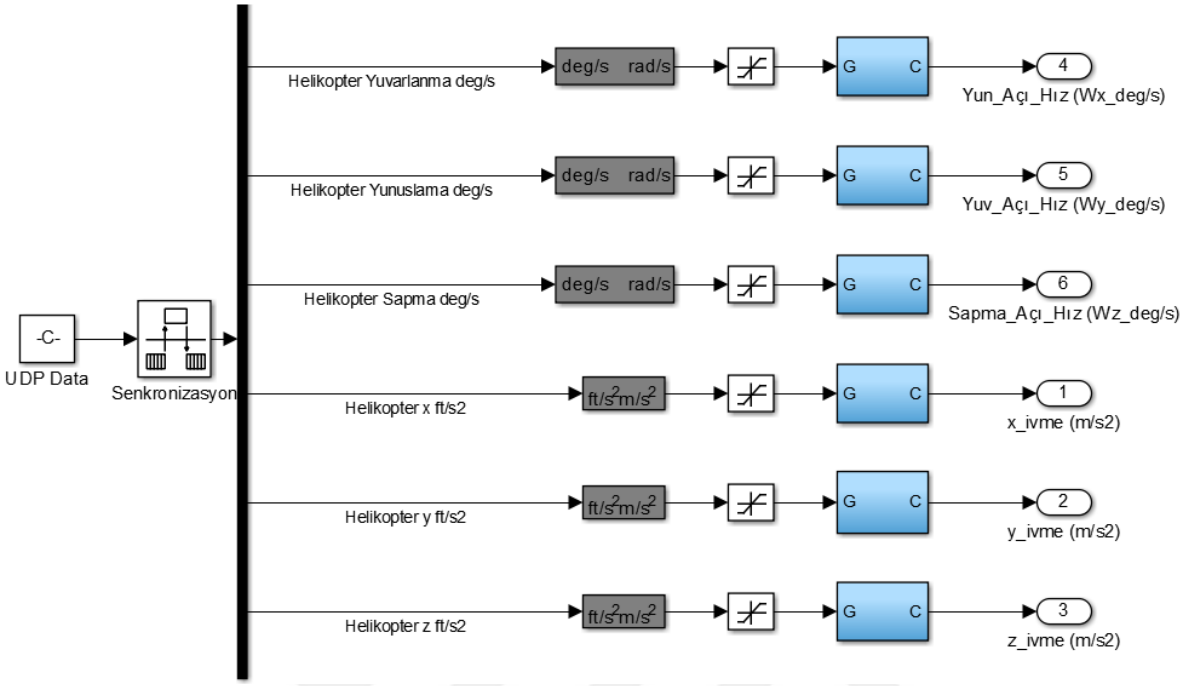
SONUÇLAR

Bir önceki bölümde tasarlanan ve kullanıma hazır hale getirilen hareket algı modelleri üç deneyimli pilot ile beraber denemeye alınmıştır. Pilotlara algoritmalar ile ilgili detay verilmeyip, karşılıklı bilgi alışverişinde bulunmalarına da izin verilmemiştir. Pilotların yaş ve deneyim ile ilgili kişisel bilgiler Çizelge 5.1'de gösterilmiştir. İlk önce klasik algoritma, ardından model öngörülü kontrol algoritması kullanılarak uçuş testi yapılmıştır. D seviye simülatör ile karşılaştırmaları sonucunda 0-5 arasında puanlama yapmaları ve belirledikleri eksikliklerin neler olduğu ile ilgili sorular sorulmuştur. İlerideki çalışmalar için bu eksikliklerin incelenmesi, duruma göre donanım veya yazılım ile ilgili güncellemelerin, değişikliklerin veya yeni tasarımların yapılması için bir fikir oluşmasında katkıda bulunacaktır. Ayrıca, bir sonraki çalışmalarda oluşturulabilecek farklı hareket algı yazılımları için, karşılaştırma yapılabilecek bir sonucun ortaya çıkması hedeflenmiştir. Gösterilen grafik sonuçları ilk iki pilota ait sonuçlardır.

Çizelge 5.1 Pilotlar ile ilgili bilgiler.

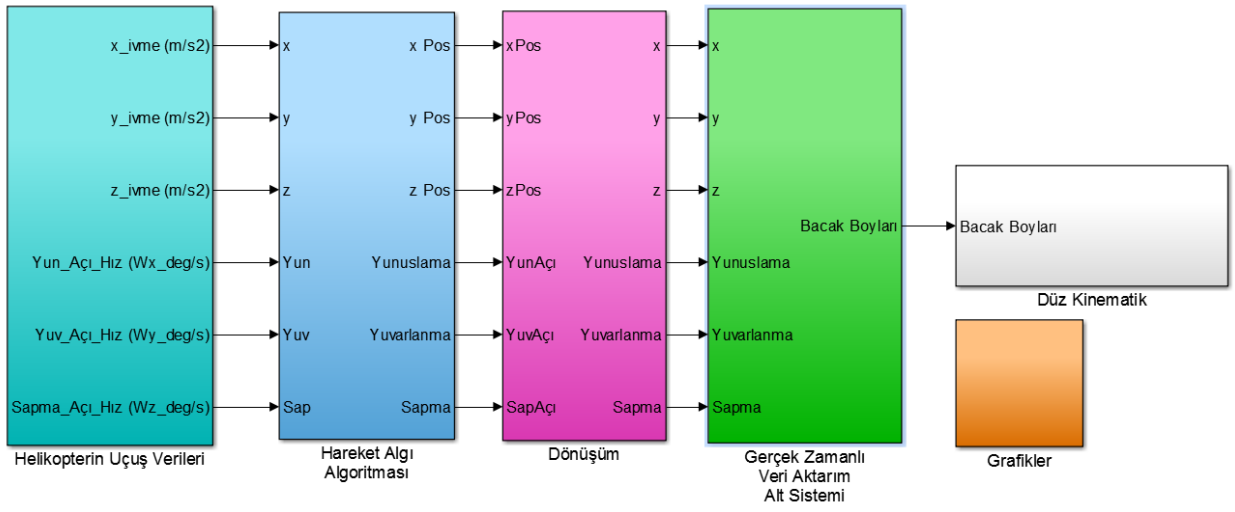
	Adı Soyadı	Yaşı	Tecrübesi
1. Pilot	Oğuzhan Selimoğlu	52	20 Yıl
2. Pilot	Ahmet Özcan Günay	50	19 Yıl
3. Pilot	Faruk Gügünağaoğlu	63	28 Yıl

Dinamik model ve görsel simülasyon beraber olup, denemeler Havelsan'a ait dinamik model ve FlightGear görsel simülasyonuna bağlı EuroCopter hava aracı ile yapılmıştır. Gizlilik gereği yazılımın yapısı hakkında bilgi verilmemiştir. Helikopterden gelen veriler, ilk olarak birim dönüşümü ve sınırlamalardan geçmektedir ve Şekil 5.1'de gösterilmiştir.



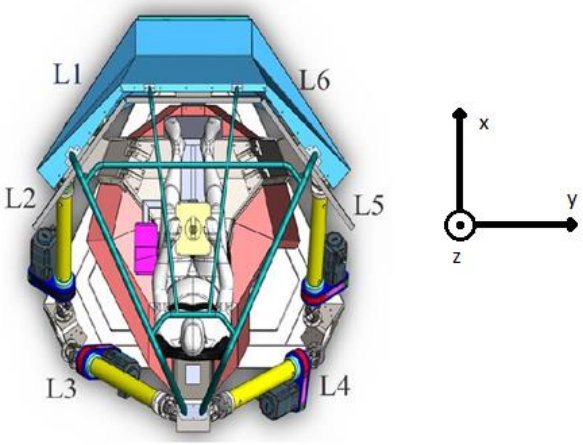
Şekil 5.1 Görsel simülasyondan, kontrolcü bilgisayarına helikopter verilerinin aktarılması ve verilerin birim dönüşümü.

Önceki bölümlerde de anlatılan ve belirtilen simulink görsellerinin birleştirilerek oluşturulan genel sistemin simulink şeması Şekil 5.2'de gösterilmiştir. Pilot denemelerinin sonuçları, tam grafik ve anlatımlarıyla beraber sırasıyla bu bölüm altında anlatılacaktır.



Şekil 5.2 Simulink modelinin genel görüntüsü.

Şekil 5.2'de belirtilen bloklar önceki bölümde anlatılmıştır. "Helikopterin Uçuş Verileri" bloğu Şekil 5.1'de, "Hareket Algı Algoritması" bloğu klasik ve MPC algoritması olduğundan Şekil 4.3 ve Şekil 4.7'de, dönüşüm bloğu Şekil 3.7 ve şekil 3.8'de, "Düz Kinematik" bloğu Şekil 3.2'de gösterilmiştir. "Gerçek Zamanlı Veri Aktarım Alt Sistemi" UDP bloklarından oluşup, simülatöre üst platform merkez noktası konumları gönderip, ardından platformun mevcut bacak boylarını elde etmek için kullanılmaktadır.



Şekil 5.3 Platformun bacakları ve pilotun oturuş pozisyonuna ait görsel

5.1 Klasik Algoritma Yazılımına Ait Pilot Denemeleri Sonuçları

Havelsan tarafından belirlenen test dökümanı ile pilotlara yapılması istenen hareketler simülasyon boyunca iletilmiştir. Uçuşların her biri yaklaşık 10 dakika sürmüştür. Pilotlar, bu test dökümanının maddelerine sadık kalarak uçuş senaryolarını tamamlamışlardır. Uçuş senaryoları ve pilot puanlamaları Çizelge 5.2'de gösterilmiştir.

Çizelge 5.2 Klasik sıfırlamalı algoritma için uçuş senaryoları ve puanları

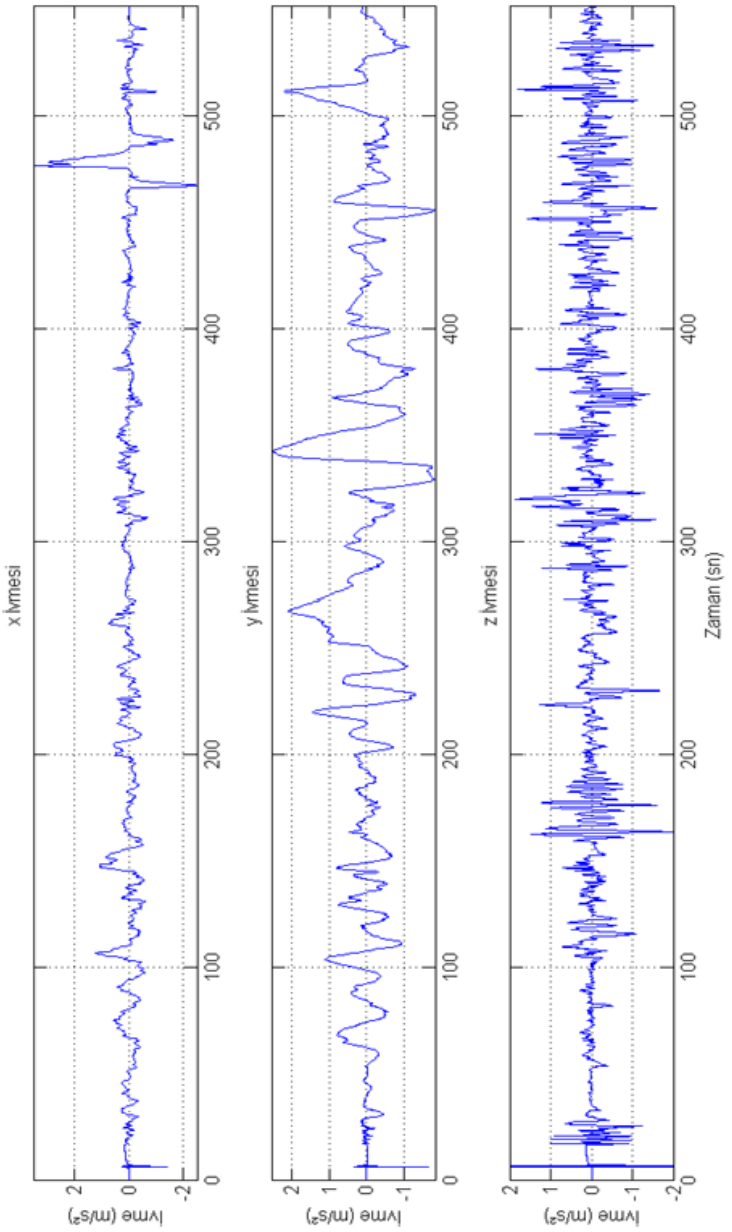
Tarih: 01.04.2016				Yazılım Versiyonu: santezCueing_Classic_v01.slx	PILOT PUANLARI (0-5)		
Test Durumu: HVL Helicopter Model, A/P off							
Test ID	EASA/ JAR grup	Testin Tipi	Bölüm	Açıklama	Pilot 1	Pilot 2	Pilot 3
M_1	Süzülme	Havalanmak	Hangar	Yerden belirlenmiş bir yüksekliketyken, yere çok yakın bir mesafeye alçalınır ve konum korulur.	2	1	3
M_2	Süzülme	Kollektif cevabı	Hangar	Her kollektifin her yöndeki tepki durumları ölçülür.	2	2	2
M_3	Süzülme	Öteleme yönünde kollektif cevabı	Hangar	Havada sabitlenmiş bir yörüngede, ± 15 derece yunuslama hareketi yapılır. Her iki yönde de 3'er saniye hareket yapıldıktan sonra kollektif eski haline geri getirilir.	2	2	4
M_4	Süzülme	Yanal yönde kumanda kolu cevabı	Hangar	Havada sabitlenmiş bir yörüngede, ± 15 derece yuvarlanma hareketi yapılır. Her iki yönde de 3'er saniye hareket yapıldıktan sonar kol eski haline geri getirilir.	2	2	2
M_5	Süzülme	Pedalın cevabı	Hangar	Havada sabitlenmiş bir yörüngede, pedala 2 şer saniye ile her iki bacakla da komut verilir ve eski haline geri getirilir.	1	1	1
M_6	Süzülme	Yükselme ve alçalma	Hangar	Başlangıç yüksekliğinden (yerde olmayacak şekilde) x yönünde ilerleme yapılırken kollektifi kullanarak hızlı bir şekilde tırmanılır. ± 100 feet değişimi yapılır. Yükselme tamamlandıktan sonra tekrar eski haline getirilir. Mesafe sınırları ve helikopterin tork sınırları aşılmamalıdır.	2	2	1
M_7	Süzülme	Sabit dönüşler	Hangar	Yerden yükseklik ve pozisyon korunarak, eksenler etrafında sabit dönüşler yapılır	1	2	2
M_8	Süzülme	Yunuslama hareketi	Hangar	Başlangıçtaki yükseklik konumunda, dikey mesafeyi koruyarak ± yönde dalış hareketi (yunuslama) yapılır.	1	2	1
M_9	Hava taksi / geçişi	Savrulma	Hangar	Hava aracı süzülme esnasındayken, değişkene hızlarla sapma ekseni yönünde döndürülür. Devamında bu hareket eksenin tersi yönünde yapılır.	2	1	3
M_10	Hava taksi / geçişi	Sabit baş yönünde savrulma	Hangar	Pistin hattını referans kullanarak, hem sol hem de sağ yöne kaydırarak helikopteri sabit bir şekilde sürülür.	2	1	3
Tırmanmaya geçiş							
M_11	Tırmanma	Kollektif aracılığıyla tırmanış kontrolü	Havaalanı Üzerinde	Tırmanma süreci devam ederken, helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kolektif ile kontrol sağlanır.	2	2	4
M_12	Tırmanma	Kumanda kolu ile tırmanış kontrolü	Havaalanı Üzerinde	Tırmanma süreci devam ederken, x yönündeki hız ve helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kumanda kolu ile control sağlanır.	3	2	4

Çizelge 5.2 Klasik sıfırlamalı algoritma için uçuş senaryoları ve puanları (devamı)

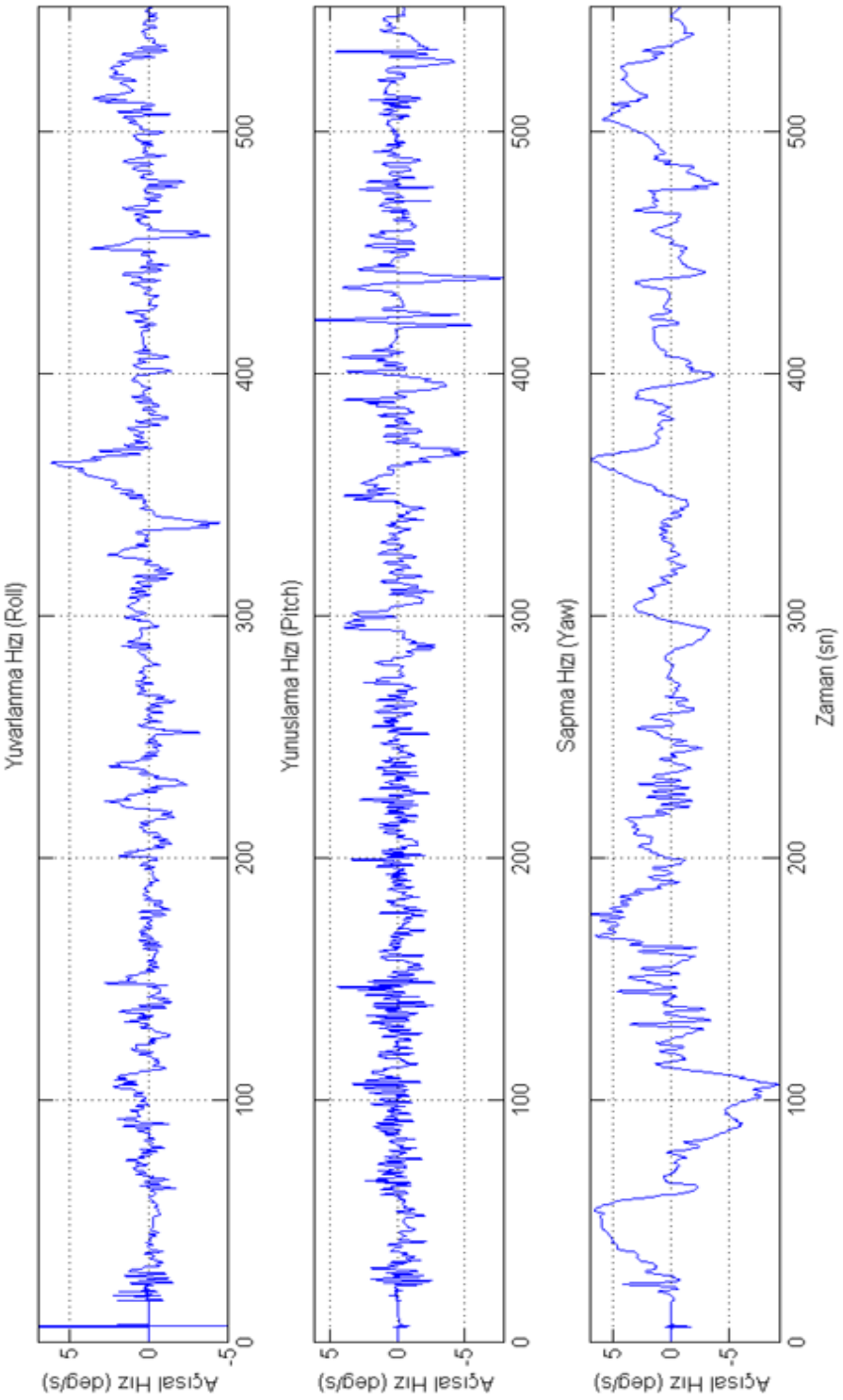
M_13	Tırmanma	Dönüşler ve yuvarlanma	Havaalanı Üzerinde	Tırmanma süreci devam ederken belirlenen bir konumda yuvarlanma hareketi yapılır, ardından ters yönde yuvarlanma hareketi yapılır. Bu süreç devam ederken helikopterin hızı sabit tutulur.	2	2	3
M_14	Tırmanma	Savrulma kontrolü	Havaalanı Üzerinde	Uçağın dengesini bozarak, hem sağ hem de sol yöne doğru savrulur.	1	1	2
Seyire geçiş							
M_15	Seyir	İvmelenme	Havaalanı Üzerinde	Düz seviyede, yükseklik sabit kalacak şekilde x yönündeki hızı +120 npt'a (61.73 m/s) kadar arttırarak ivmelenme etkisi test edilir.	3	2	3
M_16	Seyir	Yavaşlama	Havaalanı Üzerinde	Düz seviyede, yükseklik sabit kalacak şekilde x yönündeki hızı -120 not'a (-61.73 m/s) kadar azaltarak ivmelenme etkisi test edilir.	3	2	4
M_17	Seyir	Dönüş ve yuvarlanma	Havaalanı Üzerinde	Seyir halindeyken yuvarlanma hareketine başlanır. Ardından ters yönde yuvarlanma hareketine devam edilir. Bu hareketler tekrarlanır	2	2	4
M_18	Seyir	Değişen yük faktörü ile seviye değiştirme	Havaalanı Üzerinde	Sabit bir dönüş ayarlanır ve yunuslama hareketi yaparak yük faktörü değiştirilir	2	1	4
M_19	Seyir	Sabit eğimdeki davranış	Havaalanı Üzerinde	Uçağın dengesini bozarak, hem sağ hem de sol yöne doğru savrulma hareketi yapılır.	1	2	2
İnişe geçiş							
M_20	İniş	Kollektife bağlı sabit iniş	Havaalanı Üzerinde	İniş süreci devam ederken, helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kolektif ile kontrol sağlanır.	2	2	3
M_21	İniş	Kumanda koluna bağlı sabit iniş	Havaalanı Üzerinde	İniş süreci devam ederken, x yönündeki hız ve helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kumanda kolu ile kontrol sağlanır.	3	2	4
M_22	İniş	Tersine dönüş ve yuvarlanma	Havaalanı Üzerinde	İniş süreci devam ederken belirlenen bir konuma yuvarlanma hareketi yapılır, ardından ters yönde yuvarlanma hareketi yapılır. Bu süreç devam ederken helikopterin hızı sabit tutulur.	2	1	3
M_23	İniş	Savrulma kontrolü	Havaalanı Üzerinde	Uçağın dengesini bozarak, hem sağ hem de sol yöne doğru savrulma hareketi yapılır.	1	1	3
Normal konuma geçiş							
M_24	Piste yaklaşma ve iniş.	Başlangıç konumuna dönüş	Uçuş Yolu	Helikopterin konumlanacağı bir nokta belirlenir, sabit konuma gelinir ve yaklaşarak inişi gerçekleştirilir.	2	3	4

5.1.1 Birinci Pilota Ait Denemelerin Grafikleri

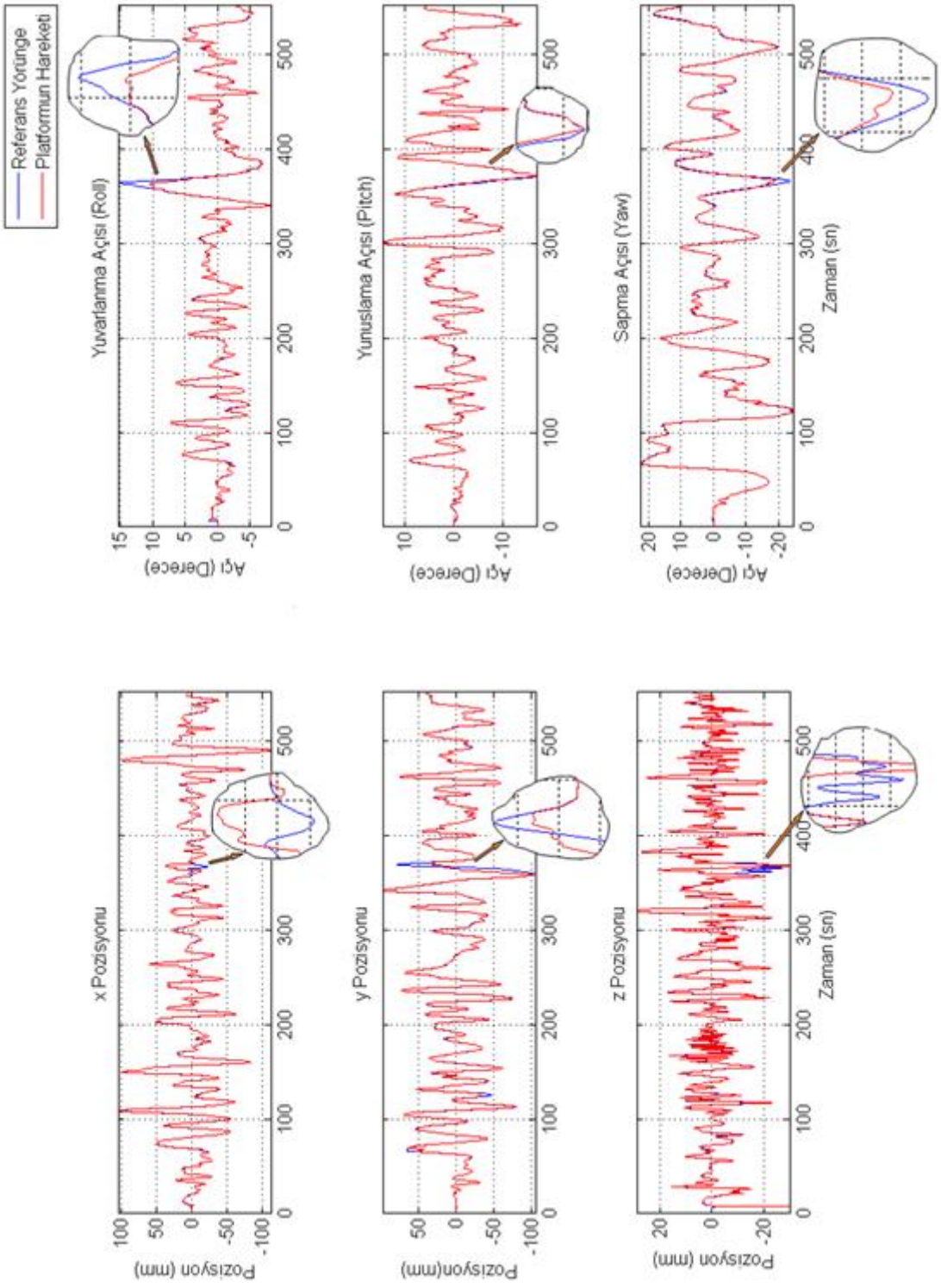
Sonuçlar sırasıyla uçuş verileri, platform hareketleri ve platformun bacak boyu değişim durumları olarak gösterilmiş ve detaylandırılmıştır. Uçuş başlangıcında pilotlar serbest kalkış yapmış ve ardından senaryolara geçiş yapılmıştır.



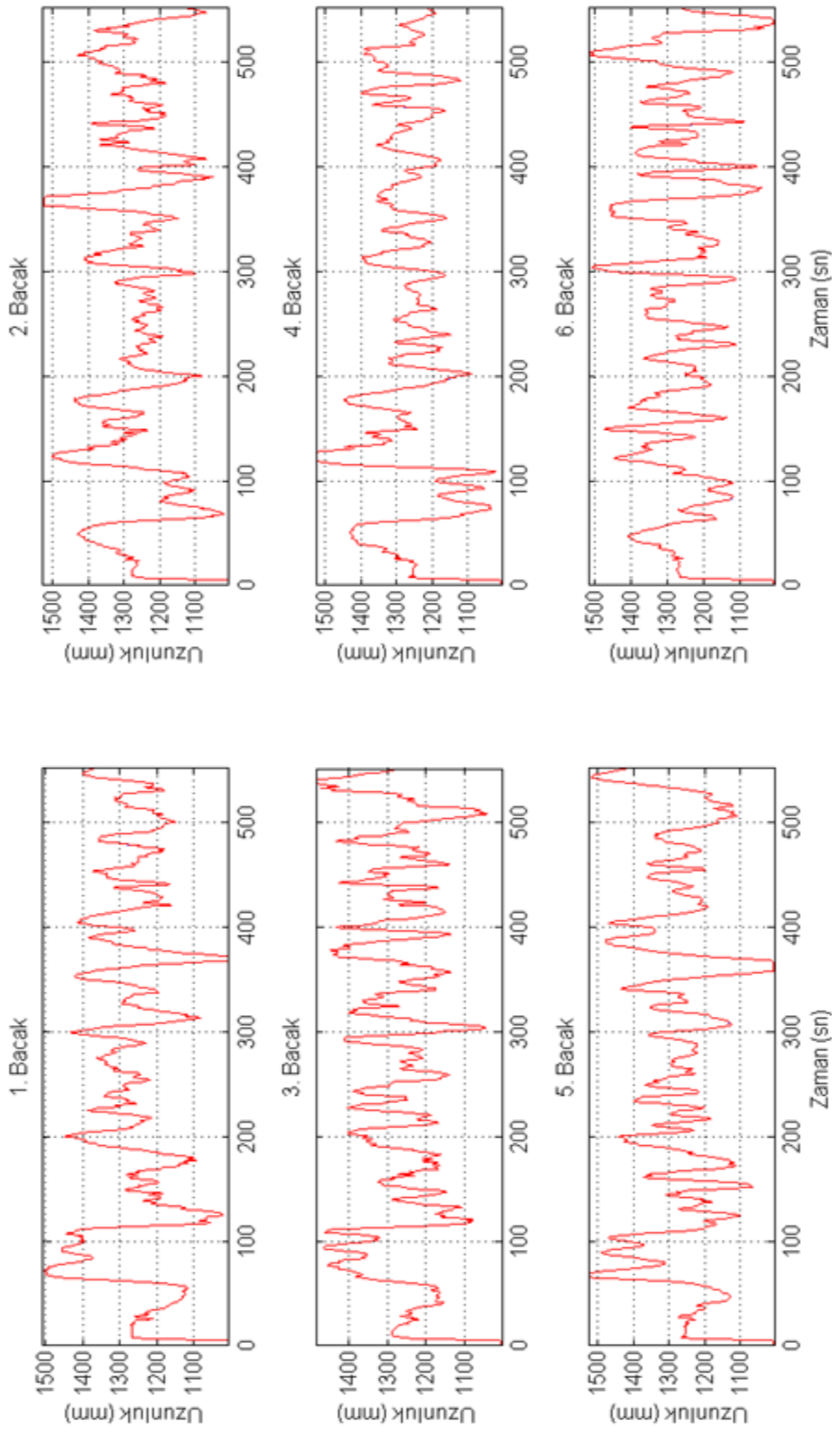
Şekil 5.4 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri



Şekil 5.5 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma açısal hız verileri

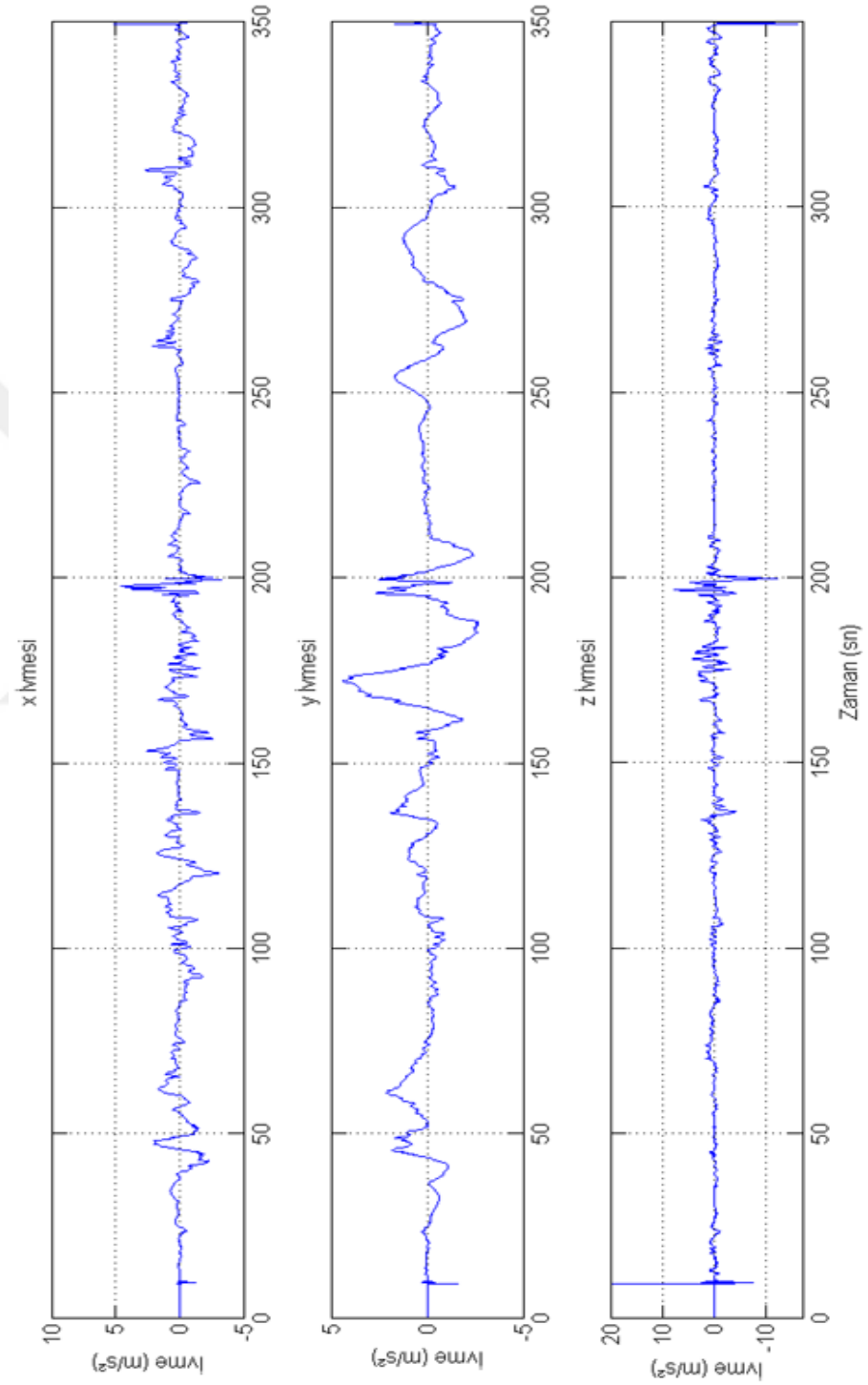


Şekil 5.6 Birinci pilot için, hareketli platformun, uçuş senaryosu boyunca öteleme ve dönme konumları ile oluşan hatalı hareketlerin gösterimi

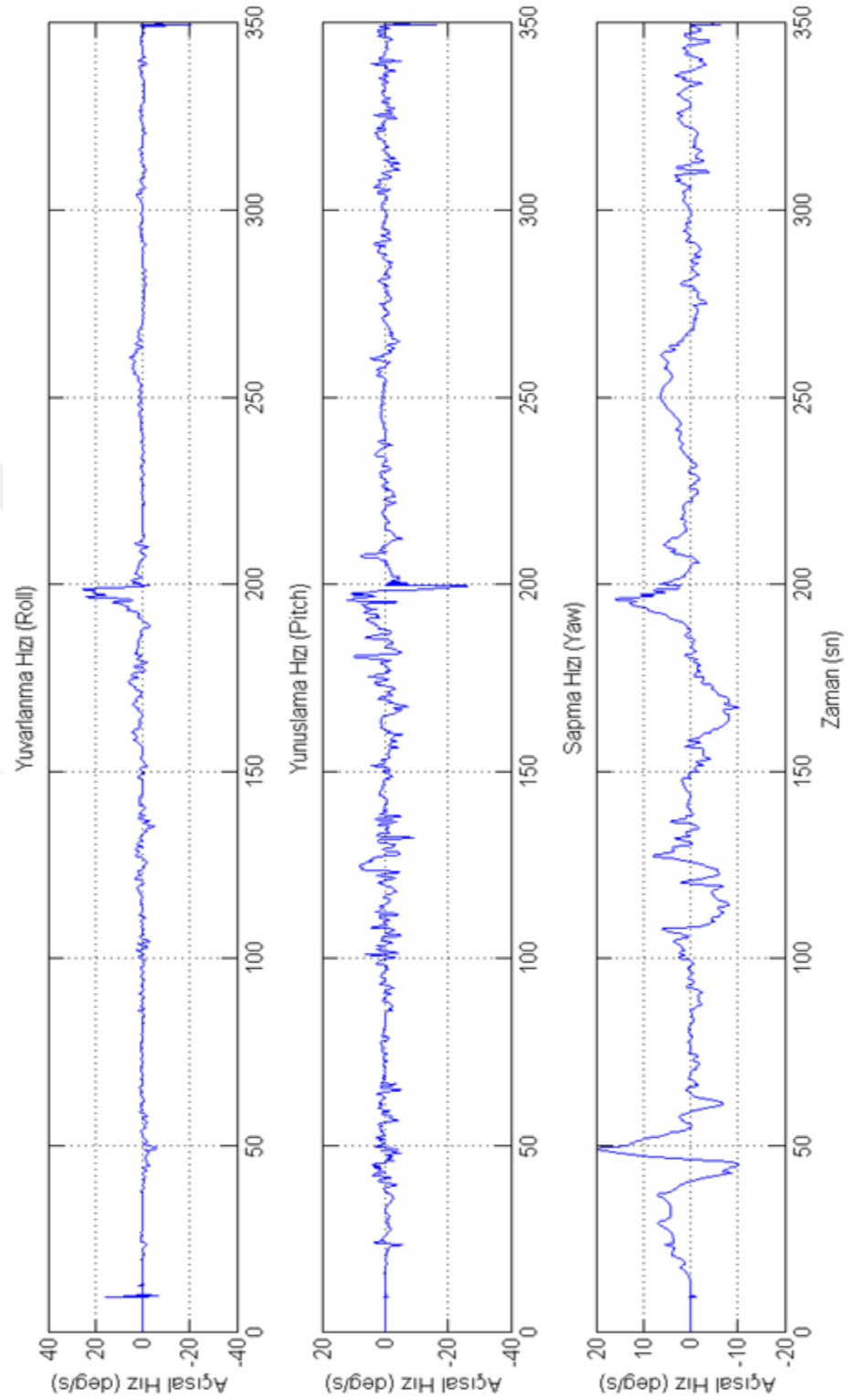


Şekil 5.7 Birinci pilot için uçuş senaryosu boyunca platformun bacak uzunlukları.

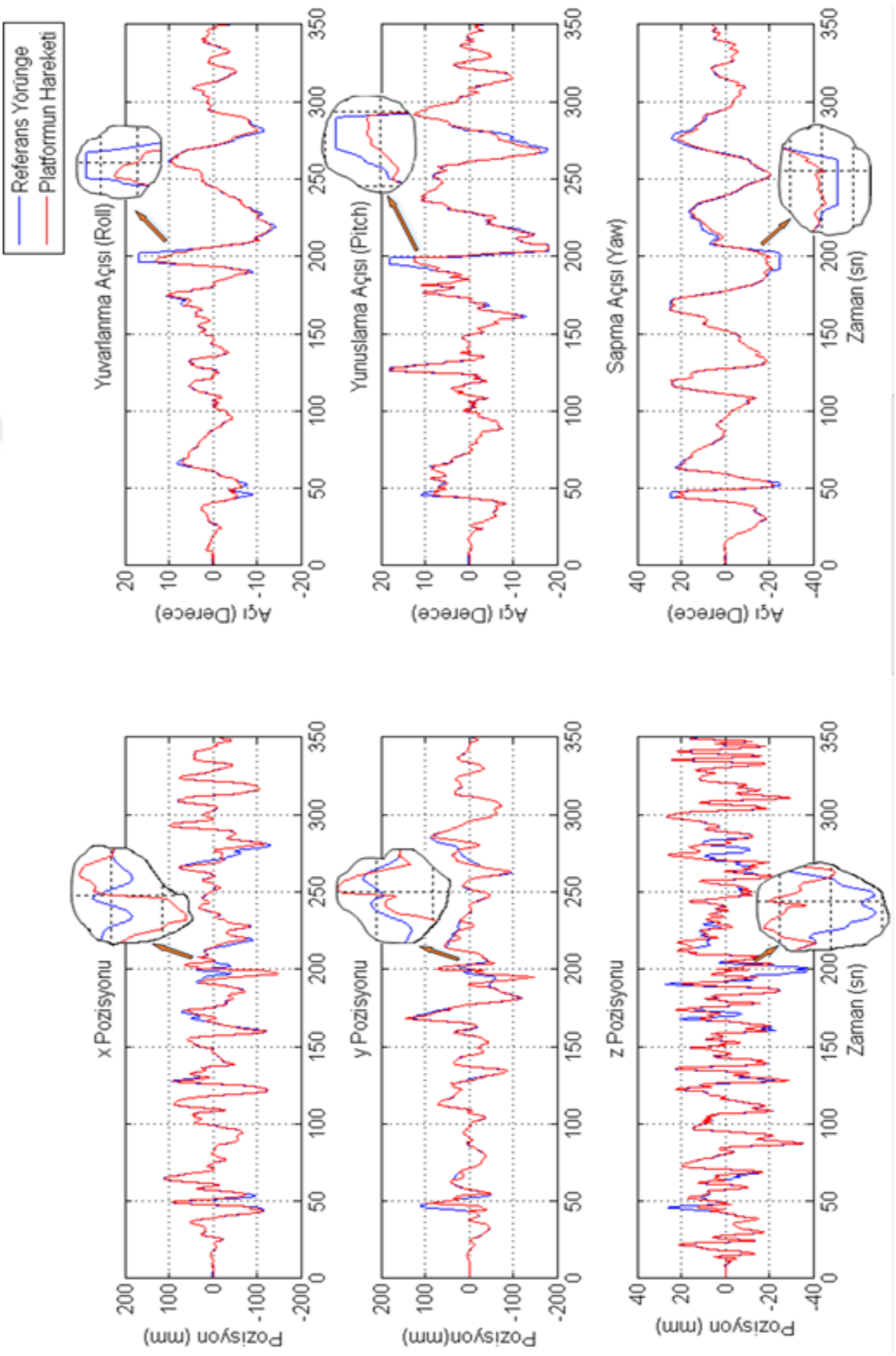
5.1.2 İkinci Pilot Ait Denemelerin Grafikleri



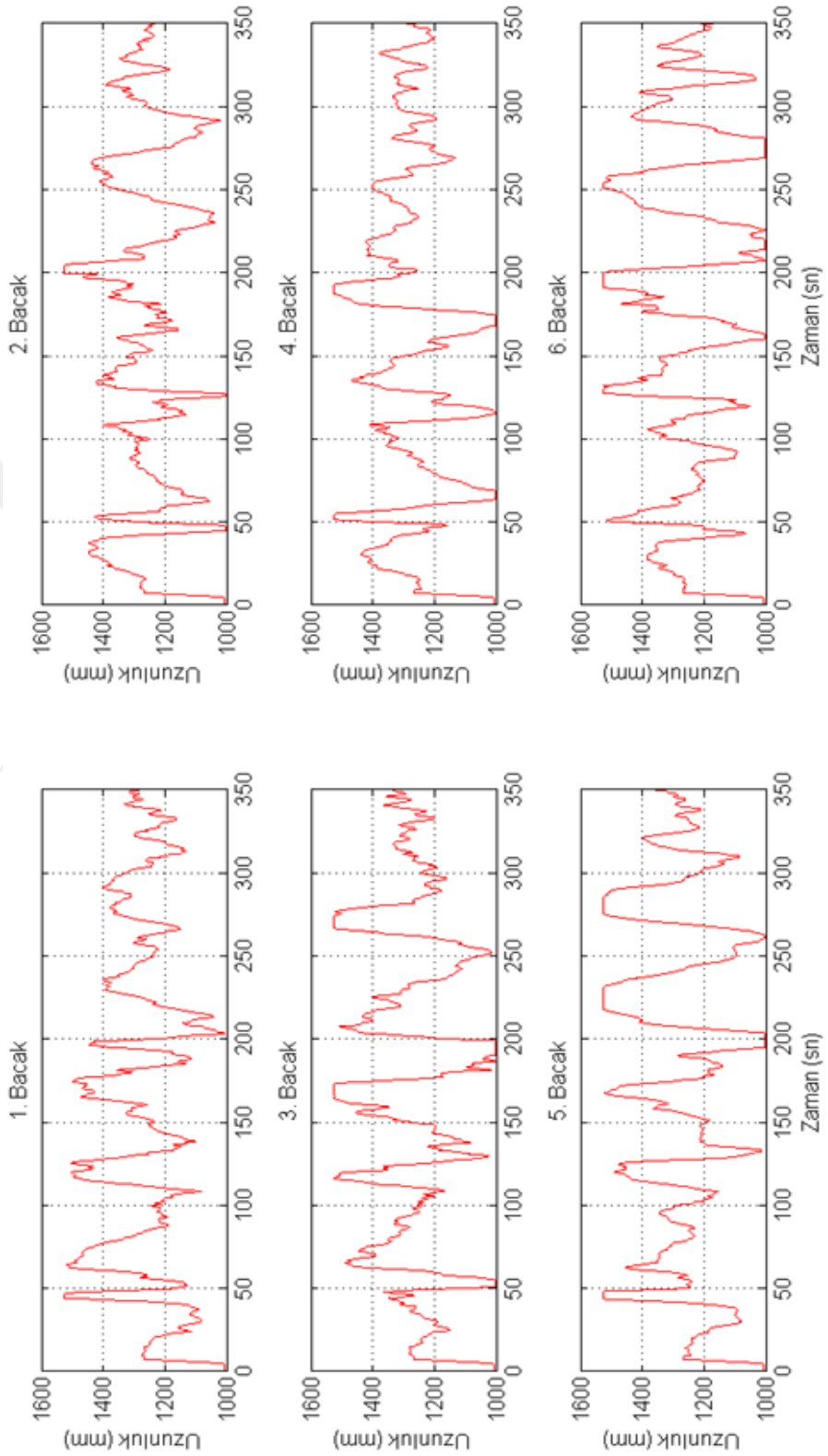
Şekil 5.8 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri



Şekil 5.9 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma açısal hız verileri



Şekil 5.10 İkinci pilot için, hareketli platformun, uçuş senaryosu boyunca öteleme ve dönme konumları ile oluşan hatalı hareketlerin gösterimi



Şekil 5.11 İkinci pilot için uçuş senaryosu boyunca platformun bacak uzunlukları.

5.1.3 Klasik Sıfırlamalı Algoritmaya Ait Genel Sonuçlar

Birinci pilota ait uçuş verileri ve platform davranışları ile ilgili bilgiler Şekil 5.1 ile Şekil 5.7 arasında, ikinci pilota ait uçuş bilgileri ise Şekil 5.8 ile Şekil 5.11 arasında gösterilmiştir. Pilotlar, uçuşa helikopter pistinden başlamaktadır. Senaryolarda da gösterildiği gibi her uçuş fazı için belirlenmiş önemli helikopter hareketlerini tamamladılar.

Her iki pilotta da kalkış yumuşak ve kontrollü olmakla beraber 5-10 saniye aralığında her eksen için veriler üzerinde ani yükselme olmaktadır. Bu başlangıç anı için normal bir durumdur.

Senaryoda ilk faz süzülme fazıdır. Bu aşamada pilotlar yerden belirli bir seviyede uçuşa başlayıp, havalanma, kollektif (simülatördeki gaz kolu), saykılık (kumanda kolu) ve pedal cevaplarını kontrol etmiştir. Süzülme sırasında diğer eksenlerde de senaryoda belirtilen uçuş hareketlerini yapmışlardır.

Pilotlar, tırmanma sırasında kollektif ve saykılık cevabının simülatör hareketiyle beraber doğruluğunu ölçmüştür. Havada iken yuvarlanma, yunuslama ve sapma hareketleri ile beraber ivmelenme kabiliyeti ile ilgili denemeler yapılmıştır. Eksenlerdeki sınırlamalardan kaynaklı olarak platform limitlerine ulaşmış ve pilotlar tarafından zaman zaman ani tepkiler hissedilmiştir. Birinci pilot için Şekil 5.6'da 380. saniyede, ikinci pilot için Şekil 5.10'da 50, 200 ve 280. saniyelerde platformun limitlerine ulaştığı görülmektedir. Klasik sıfırlamalı algoritmada platformun sıfırlama durumuna geçmesi için model öngörülü kontroldeki gibi özel bir şart belirtilememektedir. Dolayısı ile ikinci dereceden filtrenin parametreleri ile bu sıfırlamanın hızı ayarlanmaktadır. Bu durum simülatörün hareket anındaki bütün davranışlarını etkilemektedir.

Süzülme fazında tırmanma işlemi bittikten sonra bir süreliğine uçuş devam etmektedir. Bu fazda da yine önceki hareketlerde olduğu gibi benzer senaryolar uygulanmaktadır. Bu uçuş sırasında ikinci pilot simülatörün donanımdan kaynaklanan eksikler olduğunu belirtmiştir. Özellikle pedal cevaplarındaki çözünürlüğün az olduğunu ve sapma hareketi için yetersiz kaldığını belirtmiştir. Son olarak helikopter inişe geçmekte ve güvenli bir zemine inip simülasyon sonlandırılmaktadır.

Simülasyon sonunda pilotlardan alınan yorumlara göre, her üç pilotta da sıfırlama hareketlerinin hissedildiği belirtilmiştir. Bu durum algıda yanılgılara sebebiyet vermiştir. Bir diğer yandan,

sıfırlama hareketlerindeki hızlı davranıştan dolayı platformda sınırlara ulaşma durumu senaryonun bazı bölgelerinde yaşanmıştır. İkinci pilot, yazılımdan çok donanım olarak yetersizliklerin olduğunu belirtmiştir. Özellikle pedal ve saykılıktaki hareket çözünürlüğü gerçek helikopter uçuşlarına kıyasla yetersiz olduğunu söylemiştir. Üçüncü pilot diğer iki pilota nazaran klasik algoritmanın hissiyat bakımından daha iyi sonuç verdiğini belirtmiştir.

Belirtilen eksik yönler dikkate alınarak, yeniden kalibrasyon işlemine helikopterden alınan verilerin veya platforma gönderilen pozisyon verilerinin ölçeklenmesi ile başlanabilir. Sıfırlama hareketinin şiddetini azaltmak için ikinci dereceden yüksek geçiren filtrelerin frekans değerinde azaltma yoluna gidilebilir. İkinci pilotun yorumundan yola çıkarak, kumanda kolu ve pedal gibi donanımların değiştirilmesi ve çözünürlük bakımından daha güçlü ve helikopter donanımları ile uyumlu ekipmanların kullanılması gerekir.

5.2 Model Öngörülü Kontrol Algoritmasına Ait Pilot Denemeleri Sonuçları

Klasik algoritmaya ait çalışmada yapılan senaryoların benzeri model öngörülü kontrollü hareket algı algoritmasında da denenmiştir. Senaryolar ve pilot puanları Çizelge 5.3'te gösterilmiştir.

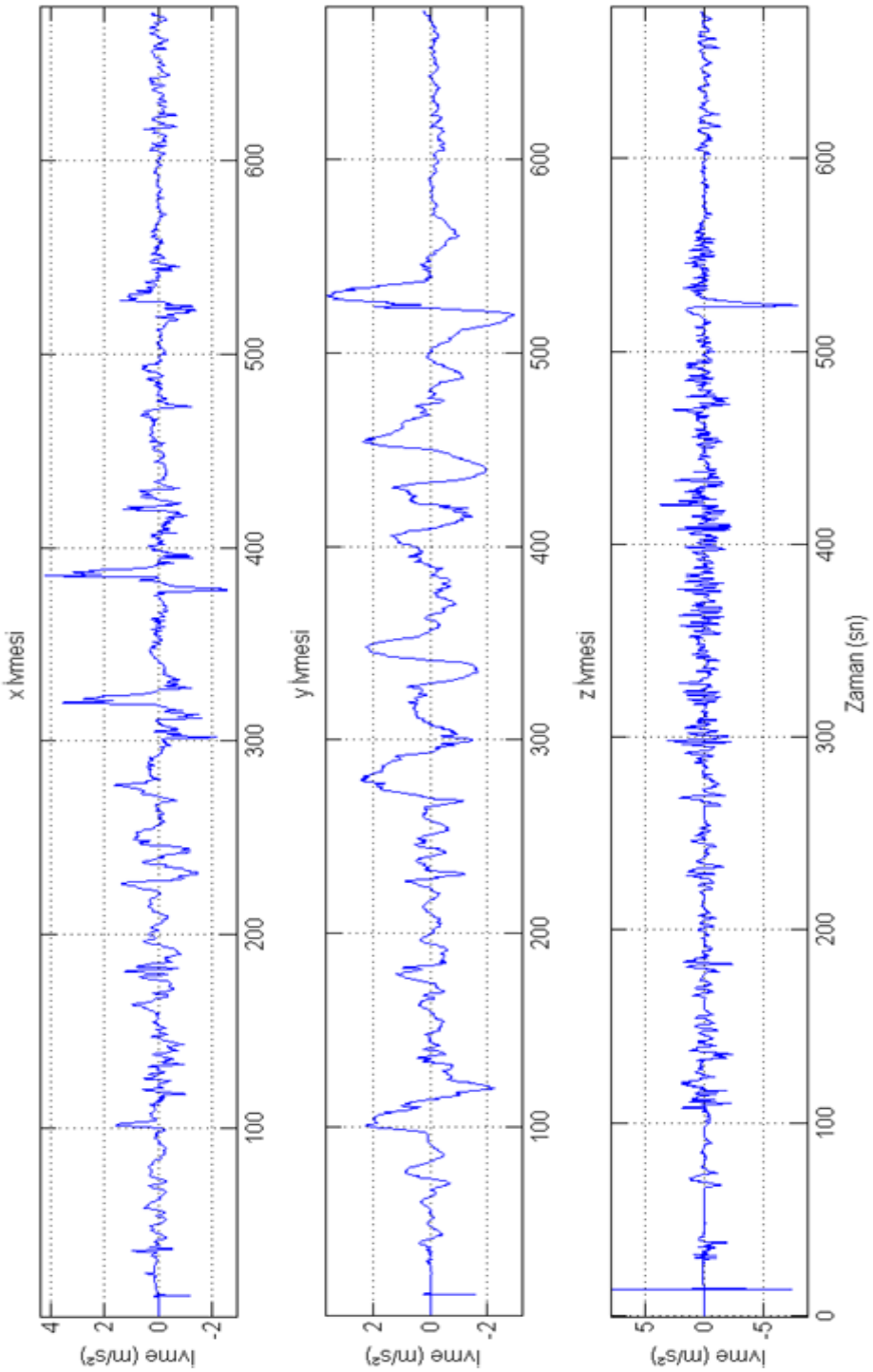
Çizelge 5.3 Model öngörülü kontrol algoritması için uçuş senaryoları ve puanları

Tarih : 01.04.2016				Yazılım Versiyonu: santezCueing_Classic_v01.slx	PILOT PUANLARI (0-5)		
Test Durumu: HVL Helicopter Model, A/P off							
Test ID	EASA/ JAR grup	Testin Tipi	Bölüm	Açıklama	Pilot 1	Pilot 2	Pilot 3
M_1	Süzülme	Havalanmak	Hangar	Yerden belirlenmiş bir yükseklikteyken, yere çok yakın bir mesafeye alçalınır ve konum korulur.	1	1	3
M_2	Süzülme	Kollektif cevabı	Hangar	Her kollektifin her yöndeki tepki durumları ölçülür.	2	2	2
M_3	Süzülme	Öteleme yönünde kollektif cevabı	Hangar	Havada sabitlenmiş bir yörüngede, ± 15 derece yunuslama hareketi yapılır. Her iki yönde de 3'er saniye hareket yapıldıktan sonra kollektif eski haline geri getirilir.	2	2	4
M_4	Süzülme	Yanal yönde kumanda kolu cevabı	Hangar	Havada sabitlenmiş bir yörüngede, ± 15 derece yuvarlanma hareketi yapılır. Her iki yönde de 3'er saniye hareket yapıldıktan sonar kol eski haline geri getirilir.	2	2	2
M_5	Süzülme	Pedalın cevabı	Hangar	Havada sabitlenmiş bir yörüngede, pedala 2 şer saniye ile her iki bacakla da komut verilir ve eski haline geri getirilir.	2	1	1
M_6	Süzülme	Yükselme ve alçalma	Hangar	Başlangıç yüksekliğinden (yerde olmayacak şekilde) x yönünde ilerleme yapılırken kollektifi kullanarak hızlı bir şekilde tırmanılır. ± 100 feet değişimi yapılır. Yükselme tamamlandıktan sonra tekrar eski haline getirilir. Mesafe sınırları ve helikopterin tork sınırları aşılmamalıdır.	2	2	1
M_7	Süzülme	Sabit dönüşler	Hangar	Yerden yükseklik ve pozisyon korunarak, eksenler etrafında sabit dönüşler yapılır	2	2	2
M_8	Süzülme	Yunuslama hareketi	Hangar	Başlangıçtaki yükseklik konumunda, dikey mesafeyi koruyarak ± yönde dalış hareketi (yunuslama) yapılır.	2	2	1
M_9	Hava taksi / geçişi	Savrulma	Hangar	Hava aracı süzülme esnasındayken, değişkene hızlarla sapma ekseni yönünde döndürülür. Devamında bu hareket eksenin tersi yönünde yapılır.	2	1	2
M_10	Hava taksi / geçişi	Sabit baş yönünde savrulma	Hangar	Pistin hattını referans kullanarak, hem sol hem de sağ yöne kaydırarak helikopteri sabit bir şekilde sürülür.	2	1	2

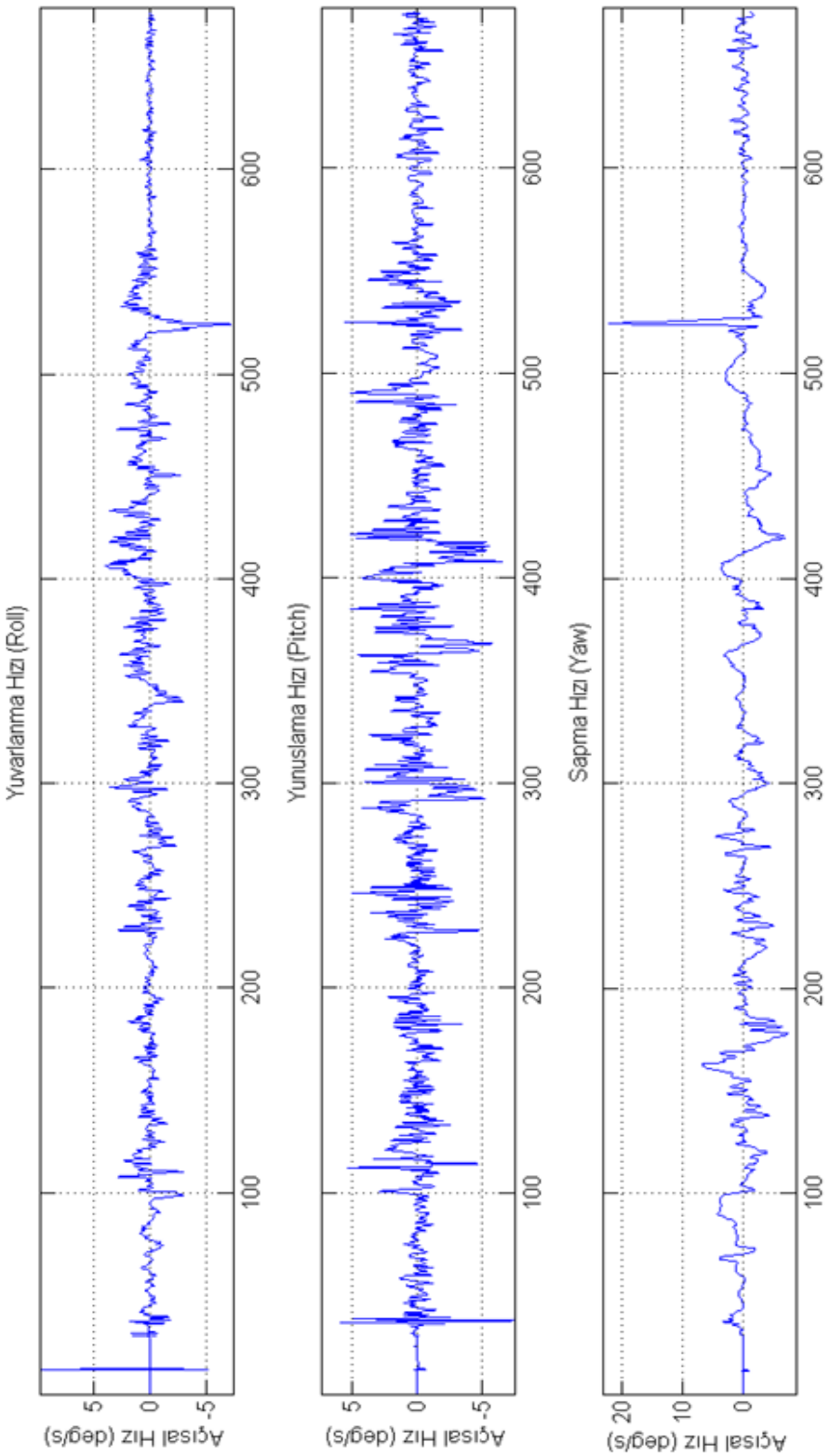
Çizelge 5.3 Model öngörülü kontrol algoritması için uçuş senaryoları ve puanları (devamı)

Tırmanmaya geçiş								
M_11	Tırmanma	Kollektif ile tırmanış kontrolü	Havaalanı Üzerinde	Tırmanma süreci devam ederken, helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kolektif ile kontrol sağlanır.	3	2	3
M_12	Tırmanma	Kumanda kolu ile tırmanış kontrolü	Havaalanı Üzerinde	Tırmanma süreci devam ederken, x yönündeki hız ve helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kumanda kolu ile control sağlanır.	2	2	4
M_13	Tırmanma	Dönüşler ve yuvarlanma	Havaalanı Üzerinde	Tırmanma süreci devam ederken belirlenen bir konumda yuvarlanma hareketi yapılır, ardından ters yönde yuvarlanma hareketi yapılır. Bu süreç devam ederken helikopterin hızı sabit tutulur.	1	2	2
M_14	Tırmanma	Savrulma kontrolü	Havaalanı Üzerinde	Uçağın dengesini bozarak, hem sağ hem de sol yöne doğru savrulur.	1	1	2
Seyire geçiş							
M_15	Seyir	İvmelenme	Havaalanı Üzerinde	Düz seviyede, yükseklik sabit kalacak şekilde x yönündeki hızı +120 not'a (61.73 m/s) kadar arttırarak ivmelenme etkisi test edilir.	3	2	3
M_16	Seyir	Yavaşlama	Havaalanı Üzerinde	Düz seviyede, yükseklik sabit kalacak şekilde x yönündeki hızı -120 not'a (-61.73 m/s) kadar azaltarak ivmelenme etkisi test edilir.	3	2	3
M_17	Seyir	Dönüş ve yuvarlanma	Havaalanı Üzerinde	Seyir halindeyken yuvarlanma hareketine başlanır. Ardından ters yönde yuvarlanma hareketine devam edilir. Bu hareketler tekrarlanır	2	2	3
M_18	Seyir	Değişen yük faktörü ile seviye değiştirme	Havaalanı Üzerinde	Sabit bir dönüş ayarlanır ve yunuslama hareketi yaparak yük faktörü değiştirilir	2	1	3
M_19	Seyir	Sabit eğimdeki davranış	Havaalanı Üzerinde	Uçağın dengesini bozarak, hem sağ hem de sol yöne doğru savrulma hareketi yapılır.	1	2	2
İnişe geçiş							
M_20	İniş	Kollektife bağlı sabit iniş	Havaalanı Üzerinde	İniş süreci devam ederken, helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kolektif ile kontrol sağlanır.	2	2	2
M_21	İniş	Kumanda koluna bağlı sabit iniş	Havaalanı Üzerinde	İniş süreci devam ederken, x yönündeki hız ve helikopterin eğimini koruyarak, dakikada +- 500 feet hız aralığında kumanda kolu ile kontrol sağlanır.	3	2	3
M_22	İniş	Tersine dönüş ve yuvarlanma	Havaalanı Üzerinde	İniş süreci devam ederken belirlenen bir konuma yuvarlanma hareketi yapılır, ardından ters yönde yuvarlanma hareketi yapılır. Bu süreç devam ederken helikopterin hızı sabit tutulur.	2	1	2
M_23	İniş	Savrulma kontrolü	Havaalanı Üzerinde	Uçağın dengesini bozarak, hem sağ hem de sol yöne doğru savrulma hareketi yapılır.	1	1	3
Normal konuma geçiş							
M_24	Piste yaklaşma ve iniş.	Başlangıç konumuna dönüş	Uçuş Yolu	Helikopterin konumlanacağı bir nokta belirlenir, sabit konuma gelinir ve yaklaşarak inişi gerçekleştirilir.	2	3	3

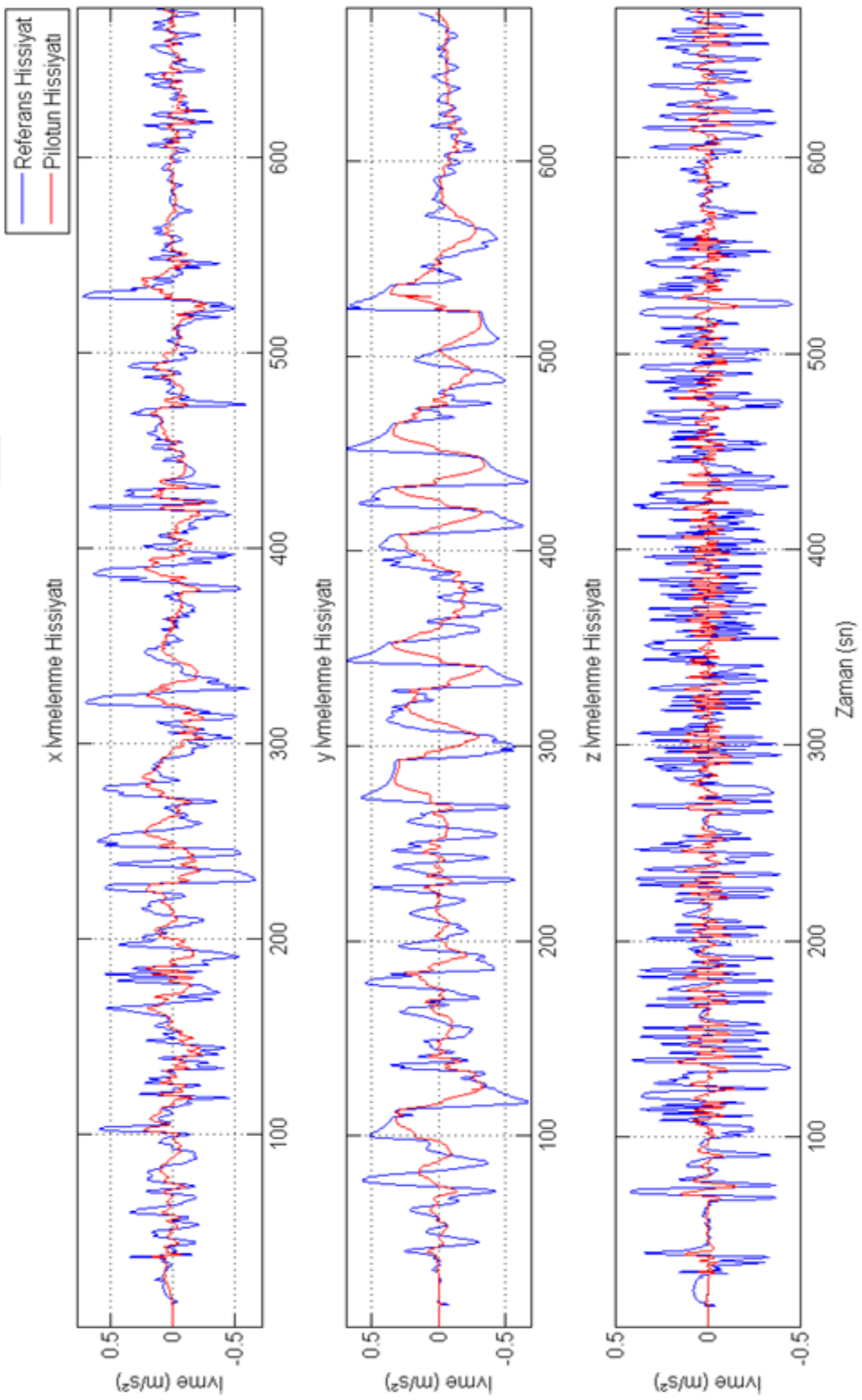
5.2.1 Birinci Pilot Ait Denemelerin Grafikleri



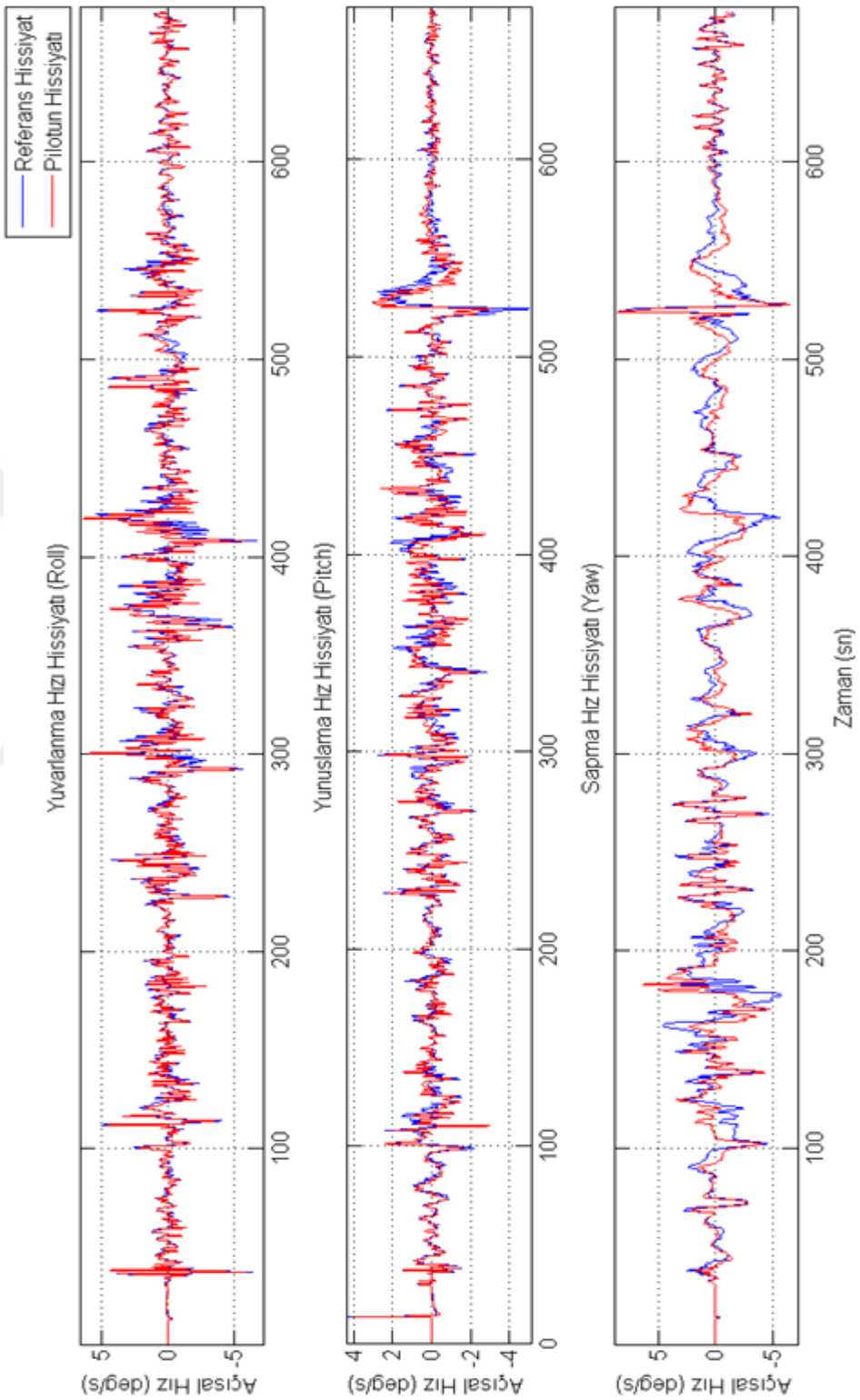
Şekil 5.12 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri



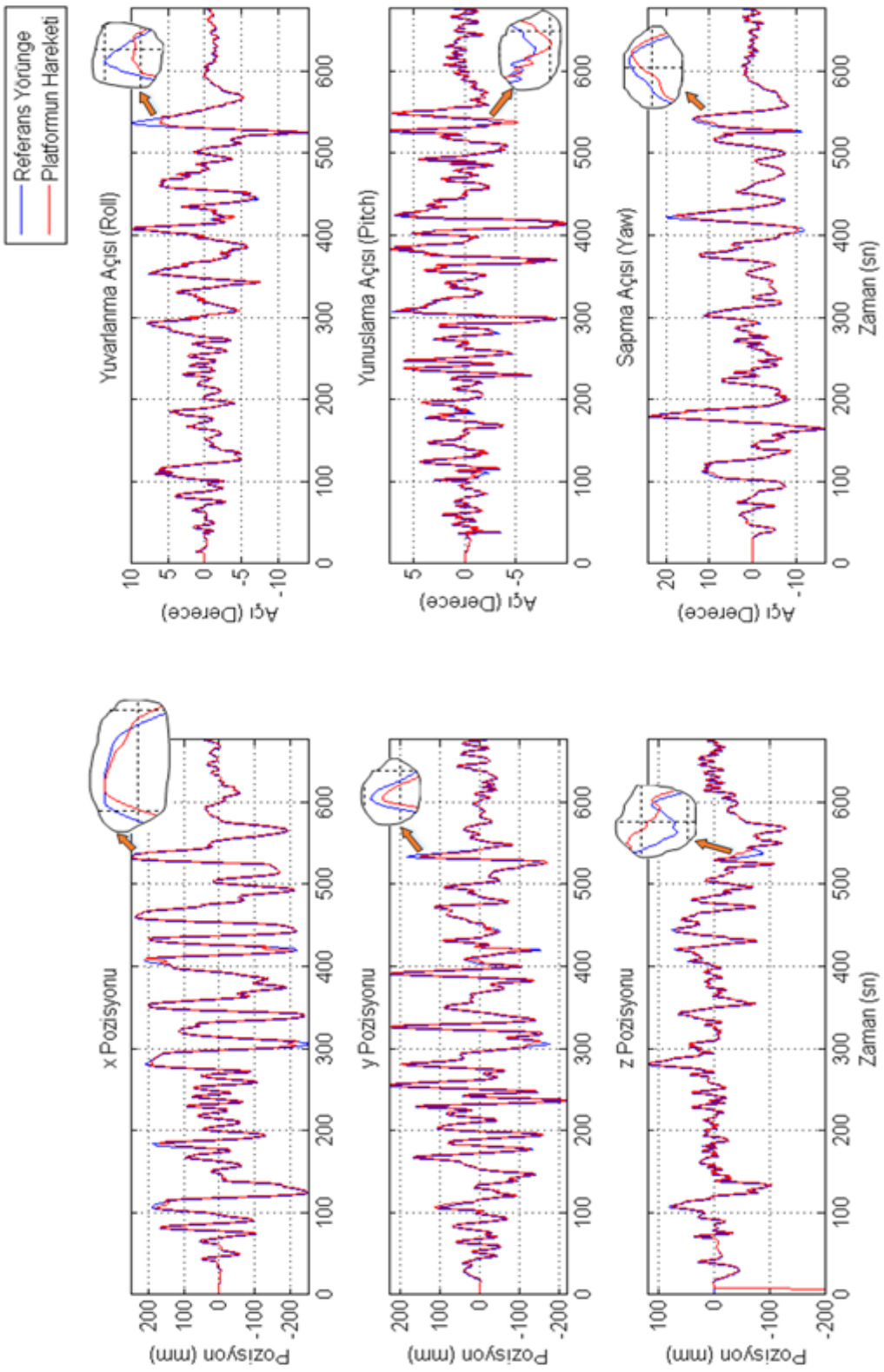
Şekil 5.13 Birinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma yönlerine ait açısal hız verileri



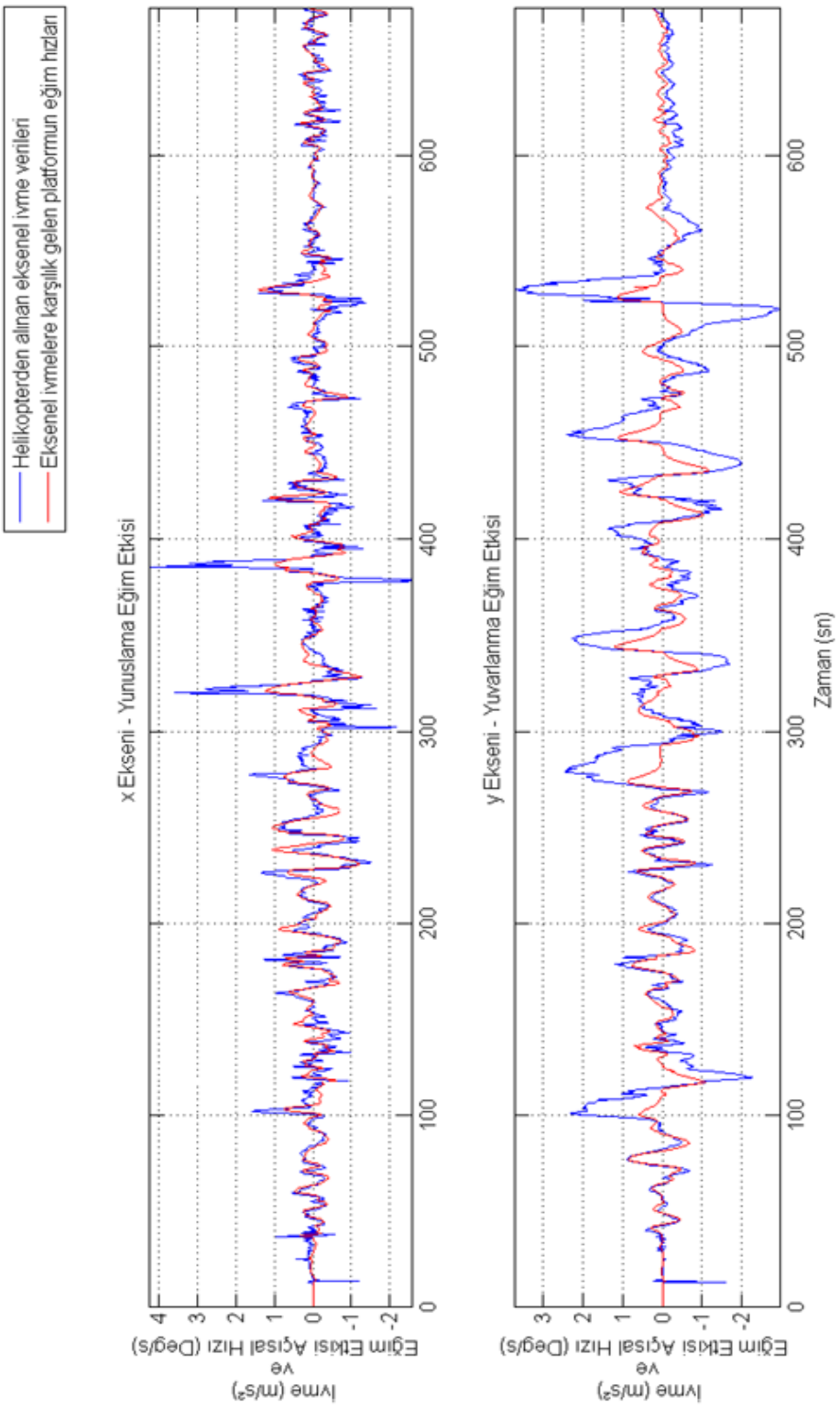
Şekil 5.14 Birinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı x-y-z yönü ivme hissiyatı



Şekil 5.15 Birinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı yuvarlanma – yunuslama – sapma yönlerindeki açısal hız hissiyatı

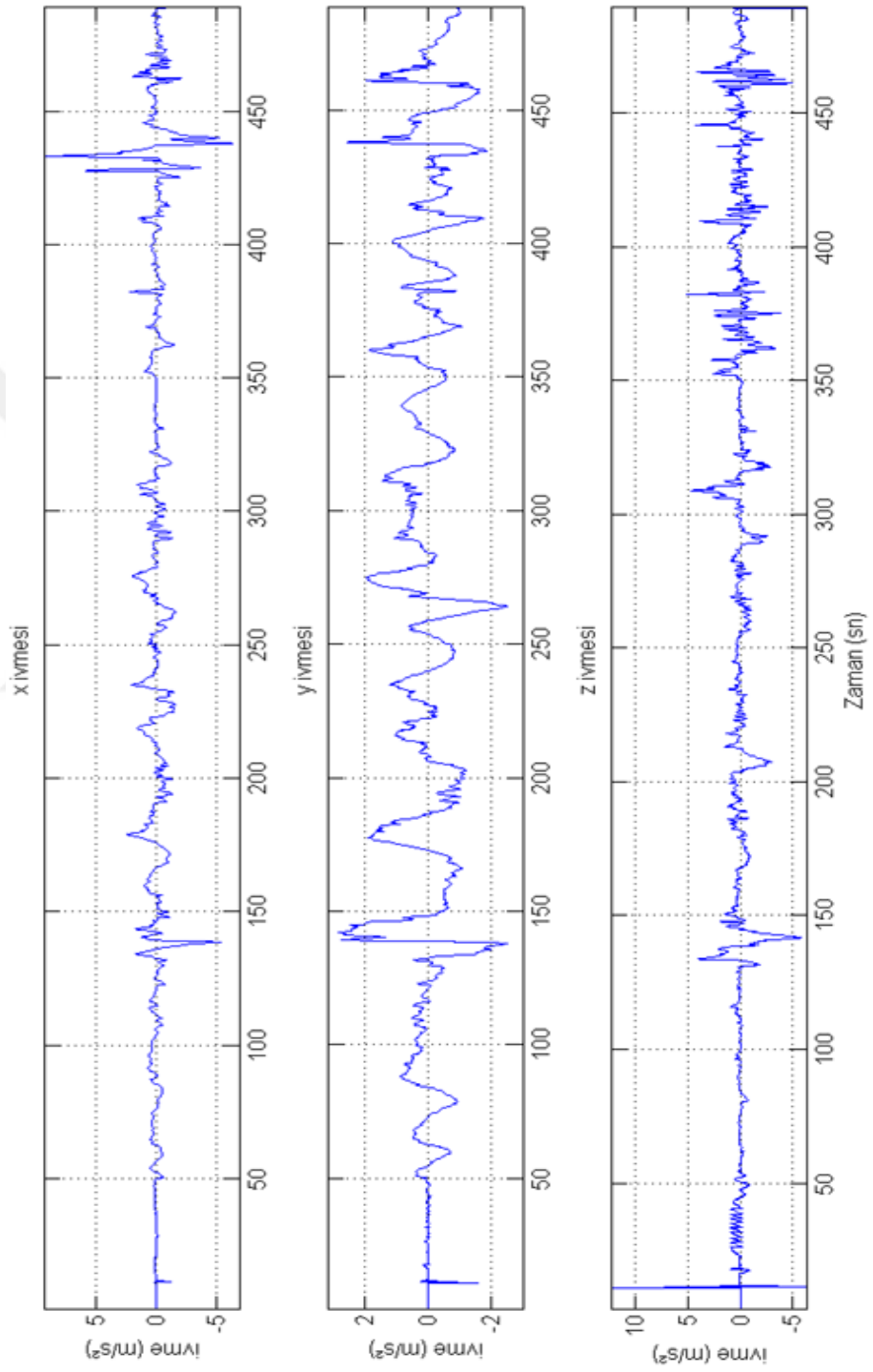


Şekil 5.16 Birinci pilota ait platforma gönderilen ve platformun yapabildiği hareketlerin grafiği.

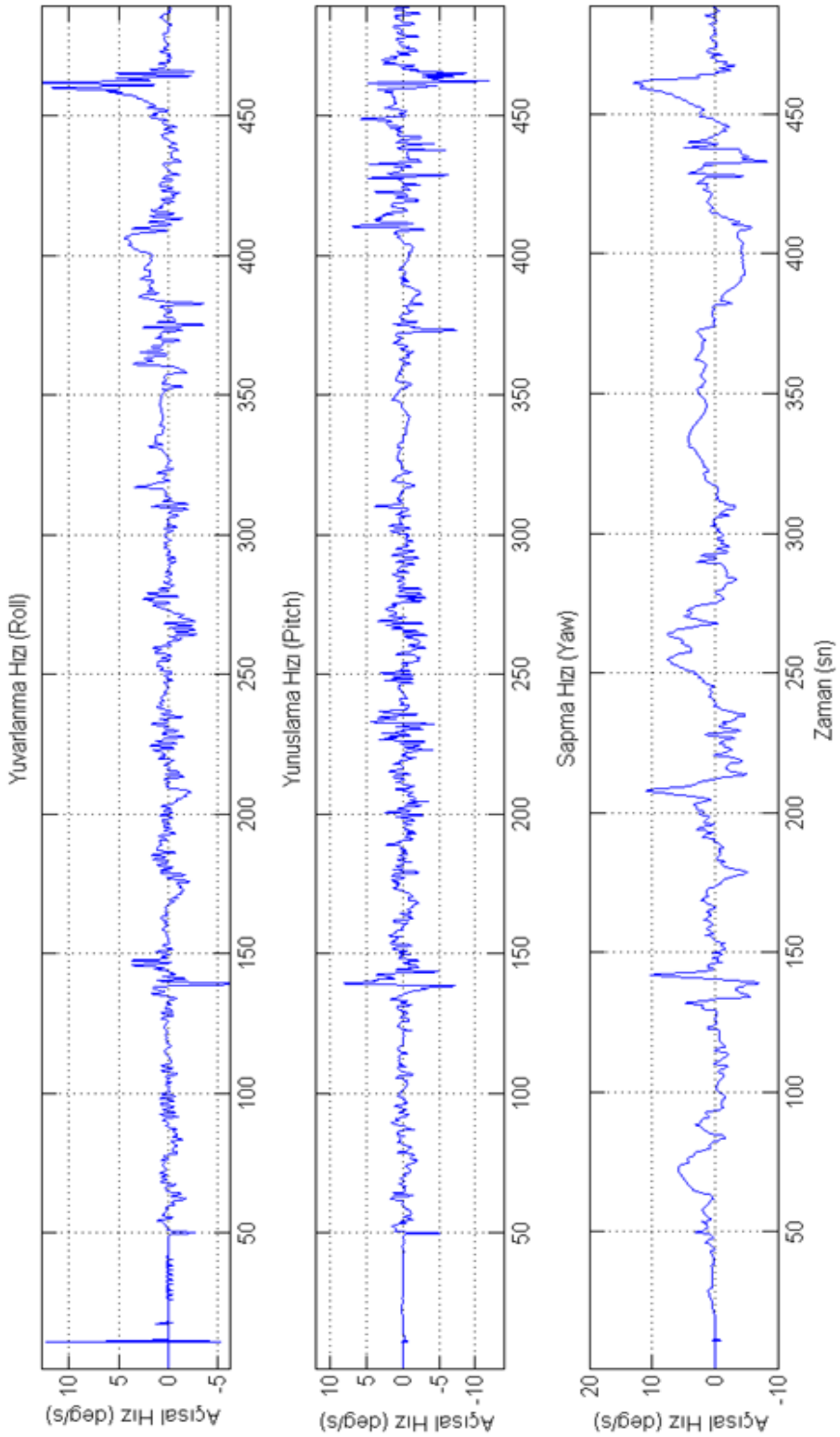


Şekil 5.17 Birinci pilota ait, platformun eğim etkisi (Tilt Coordination) sonuçları

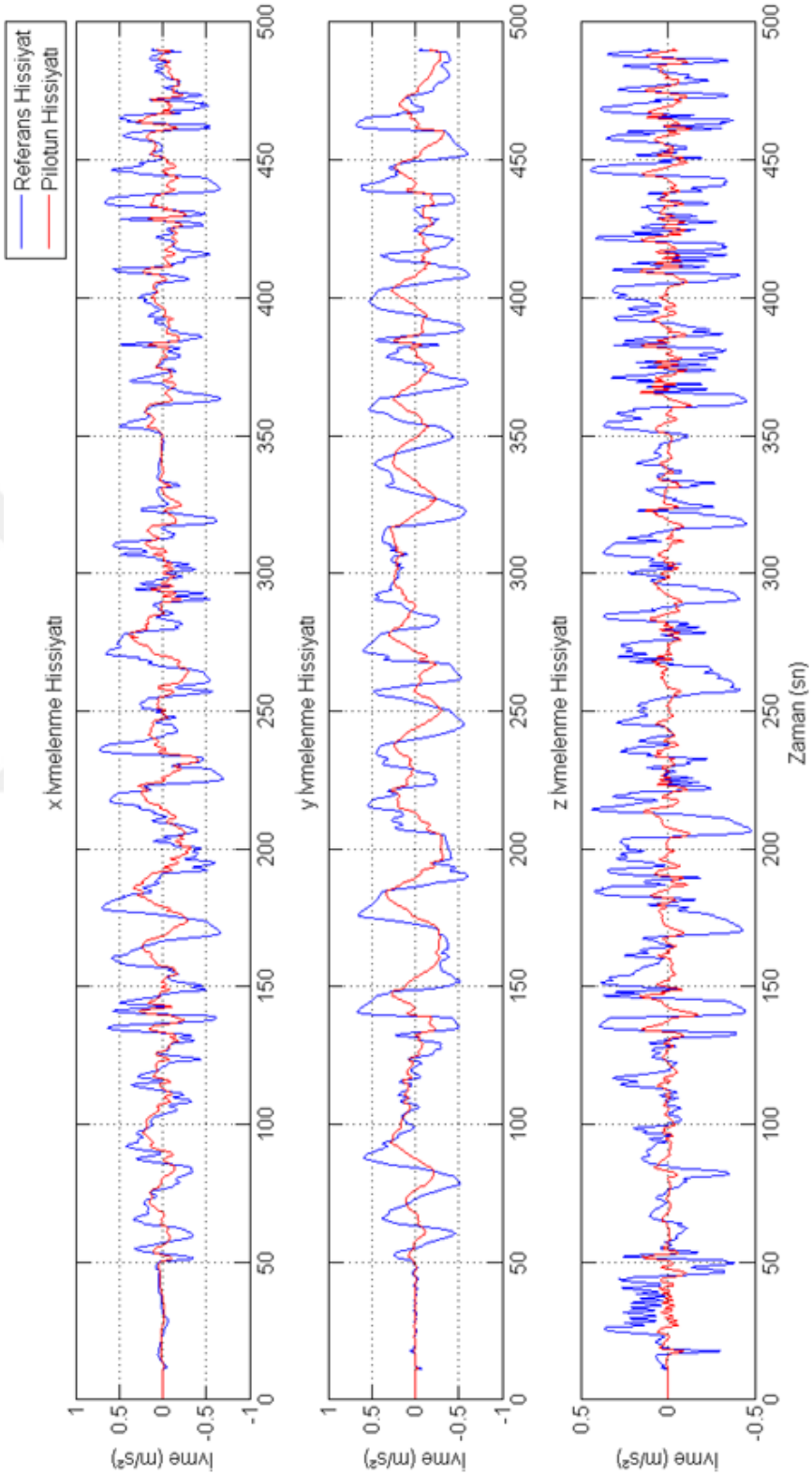
5.2.2 İkinci Pilot Ait Denemelerin Grafikleri



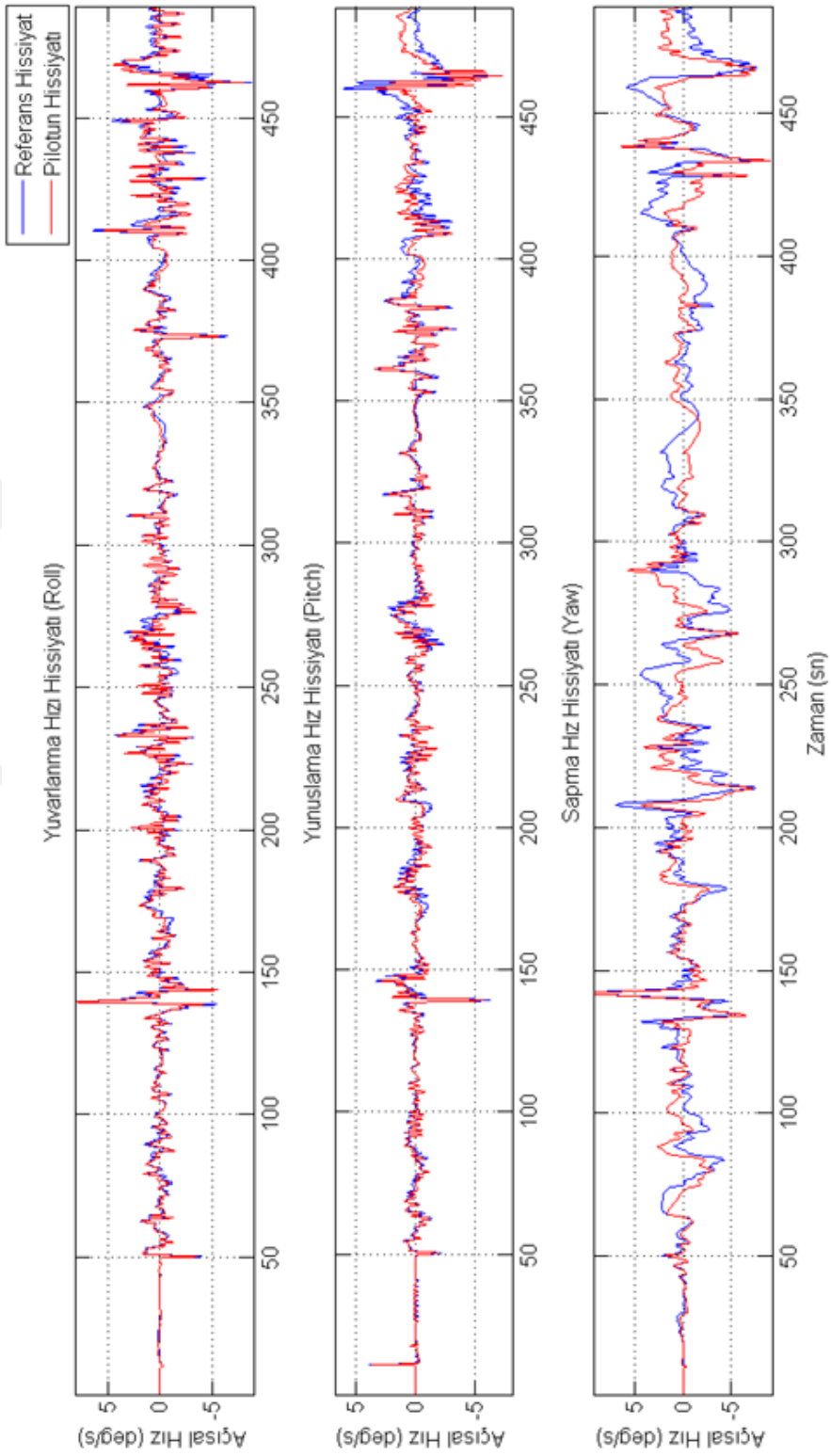
Şekil 5.18 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) x-y-z yönleri öteleme ivmesi verileri



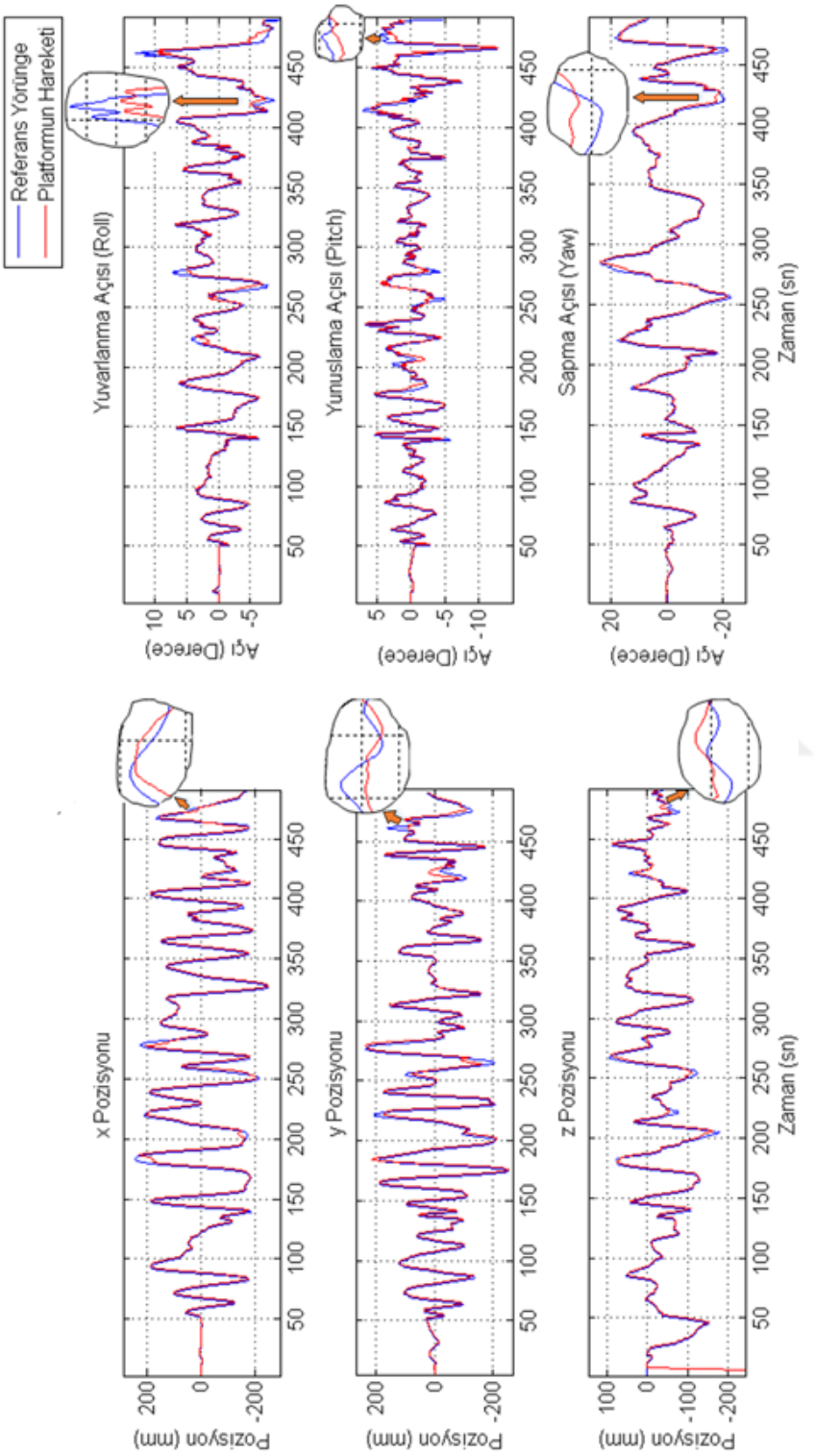
Şekil 5.19 İkinci pilota ait uçuş simülasyonu (Flight Gear – Eurocopter) yuvarlanma – yunuslama – sapma yönlerine ait açısal hız verileri



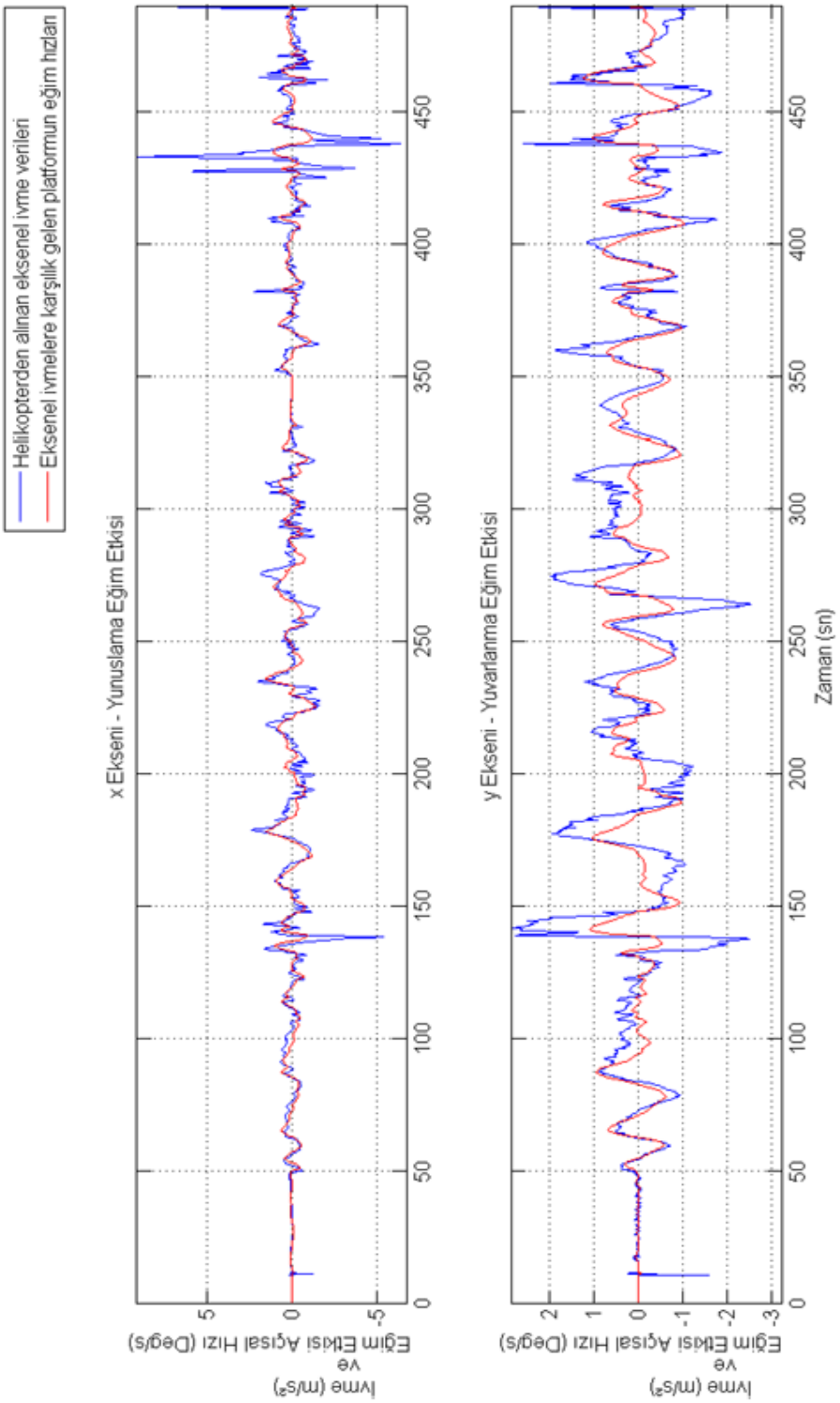
Şekil 5.20 İkinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı x-y-z yönü ivme hissiyatı



Şekil 5.21 İkinci pilotun görsel simülasyon ve hareketli platform üzerinde karşılaştırmalı yuvarlanma – yunuslama – sapma yönlerindeki açısal hız hissiyatı



Şekil 5.22 İkinci pilot için, hareketli platformun, uçuş senaryosu boyunca öteleme ve dönme konumları



Şekil 5.23 İkinci pilota ait, platformun eğim etkisi sonuçları

5.2.3 Model Öngörülü Kontrole Ait Genel Sonuçlar

Klasik algoritmada uygulanan senaryoların aynıları model öngörülü kontrol için de uygulanmıştır. Puanlama tablosu ve senaryolar Çizelge 5.3'te gösterilmiştir.

Platformdaki mekanik kısıtlamaların aynıları bu algoritma için de uygulanmıştır. Çizelge 4.12 ile Çizelge 4.17 arasında yazılım içerisinde kullanılan parametreler ve sistem sınırlamaları gösterilmektedir. İlk sınırlamalar platformun birleşik deplasman hareketi sırasında zarar görmeden hareket edebileceği pozisyon sınırlamaları olmuştur.

Şekil 5.17 ve Şekil 5.23'te eğim koordinasyonun nasıl işlediği gözükmektedir. Platform eksenel yönde hareketini başlattığı sırada eğim koordinasyonu da devreye girmektedir. Eğim koordinasyonu hareketinin hızı insan algısının altında olması için 0.06 rad/s olarak belirlenmiştir.

Şekil 5.14, Şekil 5.15, Şekil 5.20 ve Şekil 5.21'de simülatörün üzerindeki her iki pilotun hissiyatında, olması gereken referans hissiyata göre düşük değerler gözükmektedir. Platformun kısıtlamaları ve mekanik yapısından dolayı hissiyatta azalmalar meydana gelmiştir. Özellikle z ekseni (dikey eksen) yönünde eğim koordinasyonu yönünden bir avantaj olmadığı için platformun hareketi diğer eksenlere nazaran daha az olmuştur.

Şekil 5.16 ve Şekil 5.22'de gösterildiği gibi her iki pilotta da platformun sınırlarına ulaşma durumu meydana gelmiştir. Birinci pilotta 520. saniyede, ikinci pilotta ise 400-450. saniyeler arasında aralıklarla platform sınırlara ulaşmıştır.

Algoritma 100 Hz frekans ile çalıştırılmaktadır. Simülatör uygulamalarında bu hızın 1000 Hz ve yukarısında olması beklenir fakat algoritmanın yoğun içeriği ve donanım yönündeki kısıtlamalardan dolayı frekans bu şekilde belirlenmiştir. Platformun sıfırlama hareketini 1. pilot hissetmemiş, 2. pilot klasik algoritmaya kıyasla daha az hissetmiş ve 3. pilot ise diğer iki pilota kıyasla daha çok hissetmiştir. Algoritma tercihi olarak 1. ve 2. pilot model öngörülü algoritmayı tercih etmiştir.

Platformda oluşan bu kısıtlamalara alternatif çözümler bulunmaktadır. Görsel simülasyondan elde edilen veriler veya platforma gönderilen pozisyon bilgileri ölçeklendirilerek sınırlandırmadan kaynaklanan problem ortadan kaldırılabilir. Ölçeklendirmeden sonra platformun sıfırlama hareketi yeniden gözden geçirilmelidir.

Bunun için **Q** matrisinde bulunan platfomun sıfırlama hızını etkileyen ağırlık katsayıları değiştirilerek istenen seviyeye getirilebilir. Hareket genliği yüksek derecede hissedilen eksenler için **U** ve **ΔU** kontrol girişlerine ait **S** ve **R** matrislerinin ilgili eksen için olan ağırlık sayıları yükseltilmelidir. Kontrolcü bilgisayarı 12 GB ram ve i7 işlemciye sahiptir. Donanım yönünde bir geliştirme yapılması veya derleyici ortamının MATLAB – Simulink ortamı dışında başka bir program aracılığı ile yapılması durumunda örnekleme zamanı 100 Hz'in yukarısına çıkartılabilir. Bu durumda öngörülen sistem çıkışı ve kontrol girişleri miktarında artış olacağından platform daha kararlı hareketler yapacaktır.

KAYNAKLAR

[1] Jamson, A.H.J., (2010). Motion Cueing in Driving Simulators for Research Applications, Doktora Tezi, The University of Leeds, Institue for Transport Studies, United Kingdom.

[2] Fischer, M., Sehammer, H. ve Palmkvist, G., (2010). "Motion Cueing for 3-6-8 DOF Motion System", Driving Simulator Conference, 9-10 Eylül 2010, Linköping.

[3] Aminzadeh, M., Mahmoodi, A., ve Sabzehparvar, M., (2012). "Optimal MC Algorithm Using Motion System Kinematics", European Journal of Control, EUCA, 18:363-375.

[4] Lei, Z., Hongzhou, J. ve Hongren, L., (2007). "PC Based High Quality and Low Cost Flight Simulator", International Conference on Automation and Logistics, 18-21 Ağustos 2007, Jinan.

[5] Tomaske, W., (1999). " A Modular Automobile Road Simulator (MARS) for on and off-road conditions", Driving Simulation Conference, 7-8 Temmuz 1999, Paris.

[6] Suetomi, T., Horiguchi, A., Okamoto, Y. ve Hata, S., (1990). "The Driving Simulator with Large Amplitude Motion System", Journal of Passenger Cars, 103.

[7] Bertollini, G. P., Johnston, C. M., Kuiper, J.W., Kukula, J.C., Kulczycka, M.A. ve Thomas, W.E., (1994). "The General Motors Driving Simuator", Journal of Passenger Cars,103.

[8] Greenberg, J.A. ve Parl, T.J., (1994). " The Ford Driving Simulator", Journal of Passenger Cars, 103.

[9] Reymond, G.i. ve Kemeny, A., (2000). "Motion Cueing In The Renault Driving Simulator", International Journal of Vehicle Mechanics and Mobility, 34:249-259.

[10] Soma, H., Hiramatsu, K., Satoh, K. ve Uno, H., (1996). "System Architecture Of The JARI Driving Simulator And Its Validation", Desing and Validation of Driving Simulators, 21 Mayıs 1996, Valencia.

[11] Freeman, J.S., Watson, G., Papelis, Y.E., Lin, T.C., Tayyab, A., Romano, R.A. ve Kuhl, J.G., (1995). "The Iowa Driving Simulator: an implementation and application overview", Journal of Passenger Cars, 103.

[12] Dagdelen, M., Reymond, G., Kemeny, A., Bordier, M. ve Maizi, N., (2009). “Model-based predictive motion cueing strategy for driving simulators”, Control Engineering Practice, 17:995-1003.

[13] Jamson, A.H., Horrobin, A.J. ve Auckland, R.A., (2007). “ Whatever happened to the LADS? Design and development of the new University of Leeds Driving Simulator”, Driving Simulation Conference, 12-14 Eylül 2007, Iowa.

[14] Chapron, T. ve Colinot, J.P., (2007). “ The New PSA Peugeot-Citroen Advanced Driving Simulator: Overall Design And Motion Cue Algorithm”, Driving Simulation Conferece, 12-14 Eylül 2007, Iowa.

[15] Suikat, R., (2005). “The New Dynamic Driving Simulator At DLR”, Driving Simulation Conference, 30 Kasım – 2 Aralık 2005, Orlando.

[16] Valenta Pais, A.R., Wentink, M., Van Passen, M.M. ve Mulder, M., (2009). “Comparison Of Three Motion Cueing Algorithms For Curve Driving In An Urban Environment”, Teleoperators and Virtual Environments, 18:200-221.

[17] Qaisi, I.A. ve Traechtler, A, (2012). "Constrained Linear Quadratic Optimal Controller For Motion Control of ATMOS Driving Simulator", Driving Simulation Conference, 6-7 Eylül 2012, Paris.

[18] Damveld, H.J., Wentink, M., Van Leeuwen, P.M. ve Happee R., (2012). "Effects of Motion Cueing on Curve Driving", Driving Simulation Conference, 6-7 Eylül 2012, Paris.

[19] Beghi, A., Bruschetta, M., Maran, F. ve Minen, D., (2012). "A Model-based Motion Cueing Strategy For Compact Driving Simulation Platforms", Driving Simulation Conference, 6-7 Eylül 2012, Paris.

[20] Berger, R.D., Schulte-Pelkum, J. ve Bülthoff, H.H., (2007). Simulating Believable Forward Accelerations On A Stewart Motion Platform, Teknik Rapor, Max Planck Institute for Biological Cybernetics, Leipzig.

[21] Advani, S., Hosman, R. ve Haeck, N., (2002). "Integrated Design of a Motion Cueing Algorithm and Motion-base Mechanism for a Wright Flyer Simulator", AIAA Modeling and Simulation Technologies Conference, 5-8 Ağustos 2002, California.

[22] Conrad, B. ve Schmidt, S. F., (1969). The Calculation Of Motion Drive Signals For Piloted Flight Simulators, Teknik Rapor.

[23] Nahon, M.A. ve Reid, L.D., (1990). "Simulator Motion-Drive Algorithms: A Designer's Perspective", Journal of Guidance, Control, and Dynamics, 13:356-362.

[24] Conrad, B., Schmidt, S. ve Douvillier, J., (1973). "Washout Circuit Design For Multi-Degrees-Of-Freedom Moving Base Simulators", Visual And Motion Simulation Conference, 1 Eylül 1973, California.

[25] Parrish, R.V., Dieudonne, J.E. ve Martin Jr., D. J., (1975). "Coordinated Adaptive Washout for Motion Simulators", Journal of Aircraft, 12:44-50.

[26] Yu, Y., Shu-tao, Z. ve Jun-Wei, H., (2011). "Motion Drive Algorithm for Flight Simulator Based on the Stewart Platform Kinematics", Key Engineering Materials, 460-461:642-647.

[27] Chunping, P., Lu, Y., Liang, J. ve Gai, Y., (2012). "A Time Varying Washout Approach for Flight Simulation Hexapod Motion System", Computer Science and Automation Engineering (CSAE), 25-27 Mayıs 2012.

[28] Augusto, B.D.C. ve Loureiro, R.J.L., (2009). Motion Cueing in the Chalmers Driving Simulator: A Model Predictive Control Approach, Yüksek Lisans Tezi, Chalmers University of Technology, Department of Signals and Systems, Göteborg.

[29] Fang, Z. ve Kemeny, A., (2012). "Explicit MPC Motion Cueing Algorithm for Real-Time Driving Simulator", International Power Electronics and Motion Control Conference, 2-5 Temmuz 2012, Harbin.

[30] Wang, L., (2009). Model Predictive Control System Design and Implementation Using MATLAB, Springer, Girona.

[31] Ömürlü, V. E. ve Yıldız, İ., (2012). "Parallel self-tuning fuzzy PD+ PD controller for a Stewart–Gough platform-based spatial joystick", Arabian Journal for Science and Engineering, 37:2089-2102.

[32] Barbagli, F., Ferrazzin, D., Avizzano, C. A. ve Bergamasco, M., (2000). “Washout Filter Design for a Motorcycle Simulator”, Systems, Man, and Cybernetics, 2000 IEEE International Conference, 8-11 Ekim 2000, Nashville.

[33] Gutridge, C. J., Durham, W., Woolsey, C. ve Hovakimyan, N., (2004). Three Degree of Freedom Simulator Motion Cueing Using Classical Washout Filters And Acceleration Feedback, Yüksek Lisans Tezi, Virginina Polytechnic Institute & State University, Aerospace Engineering, Blacksburg.

[34] Baseggio, M., Beghi, A., Bruschetta,M., Maran,F. ve Minen, D., (2011). "An MPC approach to the design of motion cueing algorithms for driving simulators", Conference on Intelligent Transportation Systems, 5-7 Ekim 2011, Washington.

DÜZ KİNEMATİK ALGORİTMASINA AİT YAZILIM DOSYASININ İÇERİĞİ

Düz kinematik algoritmasına ait Matlab/M-File dosya içeriği aşağıdaki gibidir;

```
function [iter,xDf,xD]= forwardKinematic(L,P,B,xDi)
xD=xDi;
Rp = 0.0923; % üst plaka yarıçapı
Rb = 0.0932; % alt plaka yarıçapı
err= 0.000001; % kabul edilebilir hata payı
ind=0;
iter= 0;
R=zeros(3,6);
f=zeros(1,6);
L1=0;
J=zeros(6,6);
%%
 L=L/10000;%birim küçültme dekametre
 B=B/10000;%birim küçültme dekametre
 P=P/10000;%birim küçültme dekametre
while ind~=6
    R11=cos(xD(4))*cos(xD(5));
    R12=cos(xD(4))*sin(xD(5))*sin(xD(6))-
sin(xD(4))*cos(xD(6));

R13=cos(xD(4))*sin(xD(5))*cos(xD(6))+sin(xD(4))*sin(xD(6));
    R21=sin(xD(4))*cos(xD(5));

R22=sin(xD(4))*sin(xD(5))*sin(xD(6))+cos(xD(4))*cos(xD(6));
    R23=sin(xD(4))*sin(xD(5))*cos(xD(6))-
cos(xD(4))*sin(xD(6));
```

```
R31=-sin(xD(5));
R32=cos(xD(5))*sin(xD(6));
R33=cos(xD(5))*cos(xD(6));
R=[R11 R12 R13; R21 R22 R23; R31 R32 R33];
for i=1:6
```

(3.6) denklemindeki hata fonksiyonu kullanılarak platformun belirtilen referans noktası ile arasındaki mesafe hesaplanır.

```
f(i)=xD(1)*xD(1)+ xD(2)*xD(2)+
xD(3)*xD(3)+Rp*Rp+Rb*Rb+2*(R(1,1)*P(i,1)+R(1,2)*P(i,2))*(xD
(1)-B(i,1))+ 2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))*(xD(2)-
B(i,2))+ 2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))*xD(3)-
2*(xD(1)*B(i,1)+xD(2)*B(i,2))-L(i)*L(i);
```

Bu aşamadan sonra iterasyonu yapmadan önce (3.7) denkleminde belirtilen jakobiyen matrisinin hesaplanma işlemi tamamlanır.

```
L1=sqrt(xD(1)*xD(1)+
xD(2)*xD(2)+xD(3)*xD(3)+Rp*Rp+Rb*Rb+2*(R(1,1)*P(i,1)+R(1,2)
*P(i,2))*(xD(1)-
B(i,1))+2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))*(xD(2)-
B(i,2))+2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))*xD(3)-
2*(xD(1)*B(i,1)+xD(2)*B(i,2)));

J(i,1)=[2*xD(1)+2*(R(1,1)*P(i,1)+R(1,2)*P(i,2))-
2*B(i,1)]*(1/(2*L1));
J(i,2)=[2*xD(2)+2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))-
2*B(i,2)]*(1/(2*L1));

J(i,3)=[2*xD(3)+2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))]*(1/(2*L1))
;
J(i,4)=[2*(xD(1)-B(i,1))*(-
sin(xD(4))*cos(xD(5))*P(i,1)+(-
sin(xD(4))*sin(xD(5))*sin(xD(6))-
cos(xD(4))*cos(xD(6)))*P(i,2))+2*(xD(2)-
B(i,2))*(cos(xD(4))*cos(xD(5))*P(i,1)+(cos(xD(4))*sin(xD(5)
)*sin(xD(6))-sin(xD(4))*cos(xD(6)))*P(i,2))]*(1/(2*L1));
J(i,5)=[2*(xD(1)-B(i,1))*(-
cos(xD(4))*sin(xD(5))*P(i,1)+cos(xD(4))*cos(xD(5))*sin(xD(6
```

```
))*P(i,2))+2*(xD(2)-B(i,2))*(-
sin(xD(4))*sin(xD(5))*P(i,1)+sin(xD(4))*cos(xD(5))*sin(xD(6
))*P(i,2))+2*xD(3)*(-cos(xD(5))*P(i,1)-
sin(xD(5))*sin(xD(6))*P(i,2))]*(1/(2*L1));
       J(i,6)=[2*(xD(1)-
B(i,1))*(cos(xD(4))*sin(xD(5))*cos(xD(6))+sin(xD(4))*sin(xD
(6)))*P(i,2)+2*(xD(2)-
B(i,2))*(sin(xD(4))*sin(xD(5))*cos(xD(6))-
cos(xD(4))*sin(xD(6)))*P(i,2)+2*xD(3)*cos(xD(5))*cos(xD(6))
*P(i,2)]*(1/(2*L1));
        %        Lson(:,i)=L1;
    end
    diffJ=inv(J);
    xD(:)=xD(:)-diffJ*f(:);
    for i=1:6

f(i)=xD(1)*xD(1)+xD(2)*xD(2)+xD(3)*xD(3)+Rp*Rp+Rb*Rb+2*(R(1
,1)*P(i,1)+R(1,2)*P(i,2))*(xD(1)-
B(i,1))+2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))*(xD(2)-
B(i,2))+2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))*xD(3)-
2*(xD(1)*B(i,1)+xD(2)*B(i,2))-L(i)*L(i);
    end
    ind=0;
    for k=1:6
        if (abs(f(k))<err)
            ind=ind+1;
        end
    end
    iter=iter+1; %her bir konum için iterasyon sayısı
    if iter>=200 %max iterasyon
        break
    end
end
xDf=[xD(1)*10000;xD(2)*10000;xD(3)*10000;xD(4)*(180/pi)*(-
1);xD(5)*(180/pi)*(-1);xD(6)*(180/pi)*(-1)];
end
```

KLASİK HAREKET ALGI YAZILIMININ PARAMETRE DOSYASI

Klasik hareket algı yazılımı parametre dosyası aşağıdaki gibidir. Çizelge 4.3'te belirtilen parametreler algoritma içerisine yazılmıştır. Model öngörülü kontrolden farklı olup sadece parametre tanımalanmasına gerek duymaktadır.

```
%Yıldız Teknik Üniversitesi
%Mekatronik Mügendisliği Bölümü
%Klasik Algoritma Parametreleri
%2014-2016 SANTEZ Projesi

w0x=1.4;
Qx=0.4;

w0y=1.2;
Qy=0.4;

w0z=1.6;
Qz=0.4;

wlx=0.9;
wly=1.5;

lr=0.8;
lp=1;
ly=0.8;

wroll=0.12;
wpitch=0.12;
wyaw=0.12;

g=9.80665;
```

HİLDRETH'İN KUADRATİK PROGRAMLAMA ÇÖZÜCÜSÜNÜN KODLARI

Hildreth'in kuadratik programlama çözücüsünün matlab m-file dosya kodları aşağıdaki gibidir. Yazılım, (4.71)'den itibaren çözümlenen denklemlerin sıralaması olup sistemde "Matlab Function" bloğu ile dâhil edilmiştir. Giriş parametreleri sabit (A_cons) ve değişken (b) sınır katsayıları ile quadratik denklem matrislerini (H ve f) içermektedir. Algoritma hesaplaması sırasında, sınırlanan değişkenlerin bu sınırlara olabildiği kadar hassas yaklaşmasını hedeflemektedir.

```
function eta=QPhild(H,f,A_cons,b)
% E=H;
% F=f;
% M=A_cons;
% gamma=b;
% eta =x
[n1,m1]=size(A_cons);
eta=-H\f;
kk=0;
for i=1:n1
if (A_cons(i,:)*eta>b(i))
    kk=kk+1;
else
kk=kk+0;
end
end
if (kk==0)
    return;
end

P=A_cons*(H\A_cons');
d=(A_cons*(H\f)+b);
[n,m]=size(d);
x_ini=zeros(n,m);
lambda=x_ini;
al=10;
for km=1:5
```

```
lambda_p=lambda;
for i=1:n
w= P(i,:)*lambda-P(i,i)*lambda(i,1);
w=w+d(i,1);
la=-w/P(i,i);
lambda(i,1)=max(0,la);
end
al=(lambda-lambda_p)'*(lambda-lambda_p);
if (al<10e-3); break; end
end

eta=-H\f -H\A_cons'*lambda;
```

MODEL ÖNGÖRÜLÜ HAREKET ALGI YAZILIMININ KODLARI

Model öngörülü hareket algı yazılımının içeriği aşağıdaki gibidir.

```
%Yıldız Teknik Üniversitesi
%Mekatronik Mühendisliği Bölümü
%Model Öngörülü Kontrol Algoritması
%2014-2016 SANTEZ Projesi

Samp=0.01; %% Örnekleme Zamanı
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Otolit Taşları Transfer Fonksiyonu Parametreleri ve Durum
Uzay Matrisleri
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tl=5; %s
ts=0.66; %s
ta=10; %s
K=0.4;

A1= (tl^-1)+(ts^-1);
A2= (tl^-1)*(ts^-1);
bb1=K*ta*(tl^-1)*(ts^-1);
bb2= K*(tl^-1)*(ts^-1);
B1=((K*ta)*(tl^-1*ts^-1));
B2=bb2-A1*bb1;

Axyz= [0 1 ; -A2 ,-A1];
Bxyz= [B1; B2];
Cxyz= [1 0];

clear bb1 bb2 A1 A2 B1 B2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Yarı Dairesel Kanallar Transfer Fonksiyonu Parametreleri
ve Durum Uzay Matrisleri
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ta=30;      %s hepsi için
Ts=0.1;     %s hepsi için
```

```
TLr=6.1;   %s yuvarlanma
TLp=5.3;   %s yunuslama
TLy=10.2;  %s sapma

%Yuvarlanma için
ar1=1/Ta+1/Ts+1/TLr;
ar2=(1/(Ts*Ta))+(1/(TLr*Ta)+(1/(Ts*Ta)));
ar3=1/(TLr*Ts*Ta);
br1=1/Ts;
br2=-ar1*br1;
br3=(ar1^2*br1-ar2*br1);

Aangr=[0 1 0; 0 0 1; -ar3 , -ar2, -ar1];
Bangr=[br1; br2 ;br3];
Cangr=[1 0 0];

%Yunuslama için
ap1=1/Ta+1/Ts+1/TLp;
ap2=(1/(Ts*Ta))+(1/(TLp*Ta)+(1/(Ts*Ta)));
ap3=1/(TLp*Ts*Ta);
bp1=1/Ts;
bp2=-ap1*bp1;
bp3=(ap1^2*bp1-ap2*bp1);

Aangp=[0 1 0; 0 0 1; -ap3 , -ap2, -ap1];
Bangp=[bp1; bp2 ;bp3];
Cangp=[1 0 0];

%Sapma için
ay1=1/Ta+1/Ts+1/TLy;
ay2=(1/(Ts*Ta))+(1/(TLy*Ta)+(1/(Ts*Ta)));
ay3=1/(TLy*Ts*Ta);
by1=1/Ts;
by2=-ay1*by1;
by3=(ay1^2*by1-ay2*by1);

Aangy=[0 1 0; 0 0 1; -ay3 , -ay2, -ay1];
Bangy=[by1; by2 ;by3];
Cangy=[1 0 0];

clear bp1 bp2 bp3 ap1 ap2 ap3 br1 br2 br3 ar1 ar2 ar3 by1
by2 by3 ay1 ay2 ay3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Model için otolit taşları transfer fonksiyonu pay ve
payda hesaplaması%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num=[K*ta*tl^-1*ts^-1];
den=[1 tl^-1+ts^-1 tl^-1*ts^-1];
```

```
syys=tf(num, den);
syys1=c2d(syys, Samp, 'tustin');
[numd, dend]=tfdata(syys1);
denO=cell2mat(dend);
numO=cell2mat(numd);

num=[12.5+ta^-1 12.5*ta^-1];
den=[1 12.5];
syys=tf(num, den);
syys1=c2d(syys, Samp, 'tustin');
[numd, dend]=tfdata(syys1);
denO1=cell2mat(dend);
numO1=cell2mat(numd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Model için yarı dairesel kanalları transfer fonksiyonu
pay ve payda hesaplaması%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num=[TLr 0];
den=[TLr*Ts TLr+Ts 1];
syys=tf(num, den);
syys1=c2d(syys, Samp, 'tustin');
[numd, dend]=tfdata(syys1);
denrA=cell2mat(dend);
numrA=cell2mat(numd);

%%%%%%%%%%%%%%%%%

num=[TLp 0];
den=[TLp*Ts TLp+Ts 1];
syys=tf(num, den);
syys1=c2d(syys, Samp, 'tustin');
[numd, dend]=tfdata(syys1);
denpA=cell2mat(dend);
numpA=cell2mat(numd);

%%%%%%%%%%%%%%%%%

num=[TLy 0];
den=[TLy*Ts TLy+Ts 1];
syys=tf(num, den);
syys1=c2d(syys, Samp, 'tustin');
[numd, dend]=tfdata(syys1);
denyA=cell2mat(dend);
numyA=cell2mat(numd);

%%%%%%%%%%%%%%%%

num=[Ta 0];
```

```
den=[Ta 1];
syys=tf(num, den);
syys1=c2d(syys, Samp, 'tustin');
[numd, dend]=tfdata(syys1);
denB=cell2mat(dend);
numB=cell2mat(numd);

clear dend numd num den syys syys1;
%%%%%%%%%%%%%%%%%%%%%
%Yarı Dairesel Kanallar Toplam Durum Uzay Modeli
%%%%%%%%%%%%%%%%%%%%%

Ascc=[Aangr, zeros(3,6);zeros(3) Aangp zeros(3);zeros(3,6)
Aangy];
Bscc=[Bangr, zeros(3,2);zeros(3,1) Bangp zeros(3,1);
zeros(3,2) Bangy];
Cscc=[Cangr, 0 0 0 0 0 0;
       0 0 0 Cangp 0 0 0;
       0 0 0 0 0 0 Cangy];

%%%%%%%%%%%%%%%%%
%Otolit Tasları Toplam Durum Uzay Modeli
%%%%%%%%%%%%%%%%%

H=[0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1];
Aoto1=[Axyz, zeros(2,4);zeros(2),Axyz,zeros(2); zeros(2,4),
Axyz];
Bo=[Bxyz, zeros(2,2); zeros(2,1) Bxyz zeros(2,1);
zeros(2,2) Bxyz]*H;
Boto1=Bo(1:6,1:3);
Boto2=Bo(1:6,4:6);
Aoto=[Aoto1,Boto1;zeros(3,9)];
Boto=[zeros(6,3),Boto2;eye(3), zeros(3)];
Coto=[Cxyz 0 0 0 0 0 0 0;
      0 0 Cxyz 0 0 0 0 0;
      0 0 0 0 Cxyz 0 0 0];
clear Boto1 Boto2 Aoto1 Bo;

%%%%%%%%%%%%%%%%%%%%%%%%%
%Vestibuler Sistemin Durum Uzay Modeli
%%%%%%%%%%%%%%%%%%%%%%%%%

Ae=[0 1 0 0 0 0;0 0 0 0 0 0;0 0 0 1 0 0;0 0 0 0 0 0;0 0 0 0
0 1;0 0 0 0 0 0];
Be=[0 0 0 0 0 0;0 0 0 1 0 0;0 0 0 0 0 0;0 0 0 0 1 0;0 0 0 0
0 0;0 0 0 0 0 1];
ia=[zeros(9,2); 0 Bxyz(1)*g; 0 Bxyz(2)*g; Bxyz(1)*g 0;
Bxyz(2)*g 0;zeros(11,2)];
ic=[zeros(6,2);0 0;0 0;zeros(7,2)];
```

```
Aves1=[Ascc,zeros(9,9),zeros(9,6);zeros(9,9), Aoto,
zeros(9,6);zeros(6,18) Ae];
Aves=[Aves1, ia; zeros(2,26)];
Bves1=[Bscc, zeros(9,3); Boto;Be];
Bves=[Bves1 zeros(24,2); zeros(2,6) eye(2)];
Cves1=[Cscc,zeros(3,9),zeros(3,6);zeros(3,9),Coto,zeros(3,6
);zeros(9,15), eye(9)];
Cves=[Cves1, ic];
Dves=zeros(15,8);

clear Aves1 Bves1 Cves1;
%%%%%%%%%%%%%%%%
%Ayrık Zamanlı Durum Uzay Formuna Getirme İşlemi
%%%%%%%%%%%%%%%%

sys=ss(Aves,Bves,Cves,Dves);
dis=c2d(sys,Samp);
[Ad1,Bd1,Cd1,Dd1,T]=ssdata(dis);

clear dis sys;
%%%%%%%%%%%%%%%%%%
%MPC Durum Uzay Matrisleri
%%%%%%%%%%%%%%%%%%

Ampc=[Ad1 zeros(26,15);Cd1*Ad1, eye(15,15)];
Bmpc=[Bd1;Cd1*Bd1];
Cmpc=[zeros(15,26) eye(15)];
Hu=5;  %Kontrol öngörü sayısı
Hp=10; %State ongoru sayısı

for a=1:Hp
    F((1+(a-1)*15):(a*15),1:41)=Cmpc*Ampc^a;
end

sim=1; %köşegen belirlenmesi
for b=sim:Hu
    us=0;
for c=sim:Hp
    t=Cmpc*(Ampc^us)*Bmpc;
    ksi((1+(c-1)*15):(c*15),1+((b-1)*8):(b*8))=t;
    us=us+1;
end
sim=sim+1;
end

%%%%%%%%%%%%%%%%
% U Kazanç Matrisi (S Matrisi)%
%%%%%%%%%%%%%%%%

S1=[1 0 0 0 0 0 0 0;  %Yuvarlanma input gain
```

```
    0 1 0 0 0 0 0 0;  %Yunuslama input gain
    0 0 1 0 0 0 0 0;  %Sapma input gain
    0 0 0 1 0 0 0 0;  %x input gain
    0 0 0 0 1 0 0 0;  %y input gain
    0 0 0 0 0 1 0 0;  %z input gain
    0 0 0 0 0 0 50 0;  %Yuvarlanma Tilt input gain
    0 0 0 0 0 0 0 50]; %Yunuslama Tilt input gain

k=0;
for z=1:Hu
S(1+k:8+k,1+k:8+k)=S1;
k=k+8;
end

%%%%%%%%%%%%%%%%%%%
% Delta u Kazanç Matrisi (R Matrisi)%
%%%%%%%%%%%%%%%%%%%
R1=[1 0 0 0 0 0 0 0;  %Yuvarlanma delta input gain
    0 1 0 0 0 0 0 0;  %Yunuslama delta input gain
    0 0 1 0 0 0 0 0;  %Sapma delta input gain
    0 0 0 1 0 0 0 0;  %x delta input gain
    0 0 0 0 1 0 0 0;  %y delta input gain
    0 0 0 0 0 1 0 0;  %z delta input gain
    0 0 0 0 0 0 100 0;  %Yuvarlanma Tilt delta input gain
    0 0 0 0 0 0 0 100]; %Yunuslama Tilt delta input gain

k=0;
for z=1:Hu
R(1+k:8+k,1+k:8+k)=R1;
k=k+8;
end
%%%%%%%%%%%%%%%%%%%%%%
% Hata Kazanç Matrisi (Q matrisi)%
%%%%%%%%%%%%%%%%%%%%%%
Q1=[50 0 0 0 0 0 0 0 0 0 0 0 0 0 0; %Roll error gain
    0 50 0 0 0 0 0 0 0 0 0 0 0 0 0; %Pitch error gain
    0 0 50 0 0 0 0 0 0 0 0 0 0 0 0; %Yaw error gain
    0 0 0 35 0 0 0 0 0 0 0 0 0 0 0; %x error gain
    0 0 0 0 30 0 0 0 0 0 0 0 0 0 0; %y error gain
    0 0 0 0 0 30 0 0 0 0 0 0 0 0 0; %z error gain
    0 0 0 0 0 0 50 0 0 0 0 0 0 0 0; %Roll Plat. error gain
    0 0 0 0 0 0 0 50 0 0 0 0 0 0 0; %Pitch Plat. error gain
    0 0 0 0 0 0 0 0 50 0 0 0 0 0 0; %Yaw Plat. error gain
    0 0 0 0 0 0 0 0 0 25 0 0 0 0 0; %Px Helikop. error gain
    0 0 0 0 0 0 0 0 0 0 30 0 0 0 0; %Vx Helikop. error gain
    0 0 0 0 0 0 0 0 0 0 0 25 0 0 0; %Py Helikop. error gain
    0 0 0 0 0 0 0 0 0 0 0 0 30 0 0; %Vy Helikop. error gain
    0 0 0 0 0 0 0 0 0 0 0 0 0 35 0; %Pz Helikop. error gain
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 50]; %Vz Helikop. error
gain
```

```
k=0;
for z= 1:Hp
Q(k+1:k+15,k+1:k+15)=Q1;
k=k+15;
end

k=0;
for z= 1:Hp
Rs(k+1:k+15,:)=eye(15);
k=k+15;
end

k=0;
for z= 1:Hp
CostRef(k+1:k+15,:)=[ones(6,1); zeros(9,1)];
k=k+15;
end

Kx=((ksi'*Q*ksi+R)^-1)*(ksi'*Q*F);
Ky=((ksi'*Q*ksi+R)^-1)*(ksi'*Q*Rs);

%%%%%%%%%%%%%%%%
%Sistem Sınırları
%%%%%%%%%%%%%%%%
p=15;
pc=9;
nu=zeros(pc*Hp,p*Hp);
k=0;
n=0;

for z= 1:Hp
nu(k+1:k+pc,n+1:n+p)=[zeros(pc,p-pc) eye(pc)];
k=k+pc;
n=n+p;
end

ksic=nu*ksi;
Fc=nu*F;
KK=kron(tril(ones(Hu),0), eye(8));

M=[KK; -KK; eye(8*Hu); -eye(8*Hu); ksic; -ksic];
E=2*((ksi'*Q*ksi+R+KK'*S*KK));

Umin=kron(ones(Hu,1),[-0.7 -0.6 -0.85 -0.5 -0.5 -0.5 -0.06
-0.06]');
Umax=kron(ones(Hu,1),[ 0.7  0.6  0.85  0.5  0.5  0.5  0.06
0.06]');
DeltaUmin=kron(ones(Hu,1),[-0.5 -0.5 -0.5 -0.6 -0.6 -0.2 -
0.06 -0.06]');
```

```
DeltaUmax=kron(ones(Hu,1),[ 0.5  0.5  0.5  0.6  0.6  0.2
0.06 0.06]');
Ymin=kron(ones(Hp,1),[-0.33 -0.34 -0.47 -0.45 -0.8 -0.25  -
0.8 -0.2 -0.4]');
Ymax=kron(ones(Hp,1),[ 0.33  0.34 0.47  0.25  0.8  0.25
0.8  0.3  0.4]');
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Berkay VOLKANER

Doğum Tarihi ve Yeri : 20.05.1991 – Üsküdar/İstanbul

Yabancı Dili : İngilizce

E-posta : bvolkaner@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektrik Mühendisliği	Yıldız Teknik Üniversitesi	2014
Lisans	Makine mühendisliği	Yıldız Teknik Üniversitesi	2013
Lise	Sayısal	Üsküdar Cumhuriyet Lisesi	2008

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2013	Arçelik A.Ş.	Stajyer
2012	Aselsan A.Ş.	Stajyer
2012	Arçelik A.Ş.	Stajyer
2012	Anel Holding	Stajyer
2011	Arçelik A.Ş.	Stajyer

YAYINLARI

Makale

1. Volkaner, B., Sozen, S.N. ve Omurlu, V.E., "Realization of Desktop Flight Simulation System for Motion Cueing Studies", International journal of Advanced Research (IJAR).

Bildiri

1. Volkaner, B., Sozen, S. N., Altuntas, H., Kaya, E. E. ve Omurlu, V. E, (2015) "Hareket Algı Çalışmaları için Masaüstü Uçuş Simülatörü", Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, 10-12 Eylül, 2015, Denizli.
2. Volkaner, B., Sozen, S.N., Altuntas, H., Kaya, E.E. ve Omurlu, V.E., (2015), "Experimental Motion Cueing Studies Employing Desktop Flight Simulation System". Proceedings of the 14th International Conference on Circuits, Systems, Electronics, Control & Signal Processing (CSECS '15), 20-22 Mayıs, 2015, Konya.

Proje

1. Proje Adı: Delta robotun MATLAB/Simulink ortamında kinematik ve dinamik analizi.

   Danışman: Doç. Dr. Vasfi Emre ÖMÜRLÜ

2. Proje Adı: Stewart platformunun MATLAB/Simulink ortamında ters ve düz kinematik analizi.

   Danışman: Doç. Dr. Vasfi Emre ÖMÜRLÜ