

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI
MAKİNE MÜHENDİSLİĞİ DOKTORA PROGRAMI**

**PARAŞÜTLÜ KARGO İNİŞ SİSTEMİNİN DERİN ÖĞRENME
YÖNTEMLERİYLE MODELLENMESİ VE KONTROLÜ**

HAZIRLAYAN

KEMAL GÜVEN

DOKTORA TEZİ

ANKARA – 2023

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI
MAKİNE MÜHENDİSLİĞİ DOKTORA PROGRAMI**

**PARAŞÜTLÜ KARGO İNİŞ SİSTEMİNİN DERİN ÖĞRENME
YÖNTEMLERİYLE MODELLENMESİ VE KONTROLÜ**

HAZIRLAYAN

KEMAL GÜVEN

DOKTORA TEZİ

TEZ DANIŞMANI

DR. ÖĞR. ÜYESİ ANDAÇ TÖRE ŞAMILOĞLU

ANKARA – 2023

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Makine Mühendisliği Anabilim Dalı Doktora Programı çerçevesinde Kemal GÜVEN tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Doktora Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 09 / 01 / 2023

Tez Adı: Paraşütlü Kargo İniş Sisteminin Derin Öğrenme Yöntemleriyle Modellenmesi ve Kontrolü

Tez Jüri Üyeleri (Unvanı, Adı - Soyadı, Kurumu)	İmza
Prof. Dr. Demir BAYKA (Başkent Üniversitesi)	
Prof. Dr. Ozan TEKİNALP (ODTÜ)	
Prof. Dr. Erhan İlhan KONUKSEVEN (ODTÜ)	
Prof. Dr. Tahir YAVUZ (Başkent Üniversitesi)	
Dr. Öğr. Üye. Andaç Töre ŞAMİLOĞLU (Başkent Üniversitesi)	

ONAY

Prof. Dr. Faruk ELALDI
Fen Bilimleri Enstitüsü Müdürü
Tarih : ... / ... / 2023

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 02 / 02 / 2023

Öğrencinin Adı, Soyadı: Kemal GÜVEN

Öğrencinin Numarası: 21710417

Anabilim Dalı: Makine Mühendisliği Anabilim Dalı

Programı: Makine Mühendisliği Doktora Programı

Danışmanın Unvanı/Adı, Soyadı: Dr. Öğr. Üyesi Andaç Töre ŞAMİLOĞLU

Tez Başlığı: Paraşütlü Kargo İniş Sisteminin Derin Öğrenme Yöntemleriyle Modellenmesi ve Kontrolü

Yukarıda başlığı belirtilen Doktora tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 96 sayfalık kısmına ilişkin, 02 / 02 / 2023 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 8'dir. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY

Tarih: 02 / 02 / 2023

Öğrenci Danışmanı Unvan, Adı, Soyadı, İmza:

Dr. Öğr. Üyesi Andaç Töre ŞAMİLOĞLU



annem ve anneanneme...

TEŞEKKÜR

Öğrenim hayatım boyunca hiçbir fedakârlıktan kaçınmayan aileme, akademik çalışmalarım boyunca her zaman yanımda olan ve beni bu konuda çalışmaya teşvik eden sayın hocam Dr. Öğr. Üyesi Andaç Töre ŞAMİLOĞLU'na, tez çalışması boyunca yardımlarını esirgemeyen değerli meslektaşım Ahmet Cahit GÜNERİ 'ye en içten teşekkürlerimi sunarım.

En büyük teşekkürü bütün çalışmalarında beni yalnız bırakmayan aileme borçluyum.



ÖZET

Kemal GÜVEN

PARAŞÜTLÜ KARGO İNİŞ SİSTEMİNİN DERİN ÖĞRENME YÖNTEMLERİYLE MODELLENMESİ VE KONTROLÜ

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Makine Mühendisliği Anabilim Dalı

2022

Bu tez çalışmasında, günümüzde askeri operasyonlarda kargo teslimatı için kullanılan paraşüt kontrollü bir iniş sisteminin geliştirilmesi ve bilgisayar ortamındaki simülasyonları üzerine çalışılmıştır. İlk olarak sistem matematiksel olarak modellenmiştir. Bu model MATLAB simulink ortamına aktararak yönlenme kontrolleri ele alınmıştır. Geliştirilen kontrolcüler ile sistem istenilen sapma açısı ve sapma açısı hızlarında kontrol edilmiştir. Bir sonraki aşamada sistemin donanımsal ve yazılımsal testlerinin entegrasyonu için Döngüde Yazılım ve Döngüde Donanım çalışmalarına yoğunlaşmıştır. Bu aşamada sistem Gazebo ortamında modellenmiş ve ROS entegrasyonu için gerekli yazılımlar geliştirilmiştir. Sistemde veri toplama, haberleşme ve motor kontrolleri için Pixhawk autopilot kartı kullanılmıştır. Burada PX4 olarak bilinen açık kaynak yazılımlar, sistemin simülasyon ortamına entegre edilerek gerçek uçuş testleri ilk olarak Gazebo’da simülasyonu yapılmıştır. Simülasyon sonuçları, sistem tanımlama ve model öngörülü kontrol çalışmalarında girdi olarak kullanılmıştır. Sistem tanımlama çalışmalarında NARX Sinir ağının parametrik olarak performansları elde edilmiş ve incelenmiştir. Yapılan incelemeler sonucunda tek katmanlı 5 nöronlu oluşan modelin yeterli olduğu görülmüştür. Son olarak, simülasyon verileri ile geliştirilen algoritmalar gerçek uçuş verisi ile test edilmiştir. İki ayrı ortam ve sistem değişkenleri kullanılarak yapılan uçuş testleri gerçekleştirilmiştir. Kontrolcü algoritmasına çevrimiçi eğitim yöntemleri de eklenerek geliştirilen sistem modelinin sürekli güncel olması hedeflenmiştir. Bu doğrultuda, eğitilen model farklı çevresel durumların ve sistem ağırlığına ait uçuş verisi üzerinde hücum açısında ortalama 12 derecelik tahmin ortalaması tutturmuştur. Gerçek zamanlı kontrolcü çalışmaları yapılarak sistemin istenilen hedef konuma inişi rüzgarsız ortamda 3m, rüzgarlı ortamda ise 7m hata ile gerçekleştirilmiştir.

ANAHTAR KELİMELER: paraşüt kontrollü iniş sistemi, döngüde donanım, döngüde yazılım, sistem tanımlama, model öngörülü kontrol, NARX sinir ağı.

ABSTRACT

Kemal GÜVEN

MODELING AND CONTROL OF PARACHUTE CARGO LANDING SYSTEMS USING DEEP LEARNING METHODS

Başkent University Institute of Science and Engineering

Department of Mechanical Engineering

2022

In this thesis, the development of a parachute-controlled descent system for cargo delivery in military operations and the study of its simulation in a computer environment were investigated. The system was first mathematically modeled, and the guidance controls were addressed by transferring the model to the MATLAB Simulink environment. The developed controllers were used to control the system at the desired yaw angle and yaw rates. In the next stage, efforts were focused on hardware and software integration tests for the system. At this stage, the system was modeled in the Gazebo environment and the necessary software for ROS integration was developed. The Pixhawk autopilot card was used for data collection, communication, and motor controls. Here, the open-source software known as PX4 was integrated into the system's simulation environment and the first flight tests were carried out in Gazebo. The simulation results were used as input in system identification and model predictive control studies. In the system identification studies, the performance of the NARX neural network was obtained and analyzed parametrically. As a result of the review, it was seen that a one-layer model consisting of 5 neurons was sufficient. Finally, the developed algorithms were tested with real flight data using simulation data. Flight tests were carried out using two different environments and system variables. By adding online training methods to the controller algorithm, it was aimed to keep the developed system model continuously updated. In this direction, the trained model has been able to achieve an average estimation of 12 degrees in attack angle on flight data for different environmental conditions and system weight. Real-time controller studies were carried out and the system's descent to the desired target position was achieved with an error of 3m in windless environment and 7m in windy environment.

KEYWORDS: parachute controlled descent system, hardware in the loop, software in the loop, Gazebo, system identification, model predictive control, NARX neural network.

İÇİNDEKİLER

TEŞEKKÜR.....	i
ÖZET.....	ii
ABSTRACT	iii
İÇİNDEKİLER	iv
TABLolar LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
SİMGELER VE KISALTMALAR LİSTESİ	ix
1. GİRİŞ	1
1.1. Literatür Araştırması	3
1.2. Tezin Amaç ve Kapsamı	7
2. PARAŞÜT AERODİNAMİĞİ.....	8
3. MATEMATİKSEL MODEL	12
3.1. 3 Serbestlik Dereceli Modeller.....	12
3.2. 4 Serbestlik Dereceli Modeller.....	14
3.3. 6 Serbestlik Dereceli Modeller.....	15
4. SİMÜLASYON ÇALIŞMALARI	23
4.1. MATLAB Uygulaması	23
4.2. Döngüde Yazılım (Software in the Loop).....	29
4.2.1. Oto Pilot Yazılımı.....	29
4.2.2. Robot İşletim Sistemi (ROS)	31
4.2.3. Gazebo Uygulaması	33
5. DENEYSEL ÇALIŞMA	36
6. SİSTEM TANIMLAMA.....	46
6.1. Veri Kümesi Oluşturma	49
6.2. NARX Sinir Ağı	54
6.2.1. Levenberg-Marquardt Algoritması	59
6.2.2. Bayes Düzenleme Algoritması	60
6.2.3. Ölçeklenmiş Konjuge Gradyen Algoritması	61
6.2.4. Bulgular ve Tartışma	62
6.3. Gerçek Uçuş Verisi	75

7. ROTA PLANLAMA	76
7.1. Optimal Rota Teoremi.....	78
7.2. Planlama Algoritması	79
8. KONTROLCÜ ÇALIŞMALARI	84
8.1. Rota Takibi	88
9. SONUÇLAR VE ÖNERİLER.....	94
9.1. SONUÇLAR.....	94
9.2. ÖNERİLER.....	96
10. KAYNAKÇA.....	97
EKLER.....	105

TABLÖLAR LİSTESİ

	Sayfa
Tablo 5.1 Sistem güç bütçesi.....	37
Tablo 5.2 Test parametreleri.....	41
Tablo 6.1 Snowflakes Parametreleri [6].....	50
Tablo 6.2 Sensör Parametreleri	51
Tablo 6.3 Aktivasyon fonksiyonları	58
Tablo 6.4 Performans ölçüm metrikleri	63
Tablo 6.5 Eğitim algoritmalarına göre model performansları.....	63
Tablo 6.6 Mimariye göre model performansları	65
Tablo 6.7 10kg'lık sistem için model performansları	67
Tablo 6.8 ALEX Parametreleri[10]	69
Tablo 6.9 ALEX için model performansları	70
Tablo 6.10 %10 değişimler için model performansları.....	73
Tablo 6.11 Modellerin sınırları.....	74
Tablo 6.12 Gerçek uçuş verisi model performansları.....	75

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1.1 Paraşüt tipleri: a) parasail, b) cloverleaf, c) parawing, d) sailwing, e) parafoil, f) volplane [1].....	3
Şekil 1.2. PKİS düşüş şematığı [2]	4
Şekil 2.1. Kanat dihedral açısı	9
Şekil 2.2. Paraşüt anhedral açısı	10
Şekil 3.1. Paraşüt Kuzey-Doğu-Aşağı Eksenî gösterimi	12
Şekil 3.2. Paraşütlü İniş Sistemi Serbest Cisim Diyagramı [8]	15
Şekil 3.3. Paraşüt geometrisi.	20
Şekil 3.4. Ayırık kütle ve momentlerin hacimsel gösterimi.....	21
Şekil 4.1. Aerodinamik Kuvvet Simulink Bloğu.....	24
Şekil 4.2. Yerçekimi Kuvveti Simulink bloğu	25
Şekil 4.3. Aerodinamik momentler Simulink bloğu	26
Şekil 4.4. 100m Serbest düşüş paraşüt güzergahı (rüzgarsız – rüzgarlı)	28
Şekil 4.5. Gövde bilgileri	30
Şekil 4.6. PX4 Parametreleri	30
Şekil 4.7. Sabit Kanat Yönelme Kontrolü [28]	31
Şekil 4.8. PKİS Launch dosyası 1.....	32
Şekil 4.9. PKİS Launch dosyası 2.....	32
Şekil 4.10. PX4 Döngüde donanım Şeması [31]	33
Şekil 4.11. PKİS sdf dosyası	34
Şekil 4.12. POIS Sdf dosyası MAVROS eklentisi	35
Şekil 5.1 Sistem şematik gösterimi.....	36
Şekil 5.2 Sistem alt bileşenleri	38
Şekil 5.3 Paraşütlü iniş sistemi	38
Şekil 5.4 Sistem blok şeması	39
Şekil 5.5 Sistem şematik gösterimi (Revize model)	40
Şekil 5.6 İtici motorlu sistem gösterimi.....	40
Şekil 5.7 Konum Grafikleri (Test 1).....	41
Şekil 5.8 Euler Açılımları (Test 1).....	42
Şekil 5.9 Hız Grafiği (Test 1).....	42
Şekil 5.10 Kontrol Girdileri (Test 1).....	43

Şekil 5.11 Konum Grafikleri (Test 2)	44
Şekil 5.12 Euler Açılarını (Test 2).....	44
Şekil 5.13 Hız Grafiğı (Test 2)	45
Şekil 5.14 Kontrol Girdileri (Test 2).....	45
Şekil 6.1 4M Tabanlı Sistem Tanımlama	46
Şekil 6.2 Jann tarafından Önerilen Kontrol Girdisi	47
Şekil 6.3 Çıktı-Hata Yöntemi	48
Şekil 6.4 Gazebo Simulasyon Ortamı	50
Şekil 6.5 Kontrol girdileri	52
Şekil 6.6 Konum Grafikleri	53
Şekil 6.7 Euler Açılarını.....	53
Şekil 6.8 Hız Grafiğı	54
Şekil 6.9 Paralel(Sol) ve Seri-Paralel(Sağ) NARX Sinir Ağı	55
Şekil 6.10 Nöron yapısı.....	56
Şekil 6.11 Seri - Paralel Narx Sinir Ağı Mimarisi.....	56
Şekil 6.12 Sistem kestirim hataları	64
Şekil 6.13 10kg'lık sistem için kontrol girdileri	66
Şekil 6.14 10kg'lık ağırlık için model kestirim hataları	68
Şekil 6.15 ALEX için kontrol girdileri	70
Şekil 6.16 ALEX için model kestirim hataları	71
Şekil 6.17 Gerçek uçuş verisi model sonuçları.....	75
Şekil 7.1 Rüzgarsız ortamda planlama evresi [63]	77
Şekil 7.2 Rüzgarlı ortamda planlama evresi [63]	77
Şekil 7.3 PKİS Yörünge Planlama [64]	78
Şekil 7.4 PKİS Rota Şematik Gösterimi	80
Şekil 7.5 PKİS Rota Yatay düzlemde gösterimi.....	81
Şekil 7.6 Örnek rota planı	83
Şekil 8.1. Sapma açısı kontrollü düşüş (90 derece – rüzgarsız - rüzgarlı)	85
Şekil 8.2. Sapma açısı kontrollü düşüş hata analizi (90 derece – rüzgarsız - rüzgarlı)	86
Şekil 8.3 Sapma hızı kontrollü iniş (rüzgarsız-5m/s +x rüzgarlı).....	87
Şekil 8.4. Düşüş hızı kontrolü hata analizi	88

SİMGELER VE KISALTMALAR LİSTESİ

PKİS	Paraşütlü Kargo İniş Sistemi
ROS	Robot Operating System
SITL	Software in the loop
HITL	Hardware in the loop
MÖK	Model Öngörülü Kontrol
NARX	Nonlinear Autoregressive Exogenous Model
SCG	Scaled Conjugate Gradient
LM	Levenberg–Marquardt
BR	Bayesian Regularization



1. GİRİŞ

Gelişen teknoloji ile birlikte akıllı sistemlerin hem endüstride hem de günlük hayattaki kullanım alanları gittikçe artmaktadır. Bu alanlar paraşütlü kargo sistemlerini de içermektedir.

Geleneksel iniş sistemlerinde kargolar belirli bir irtifadan bırakılarak paraşütle yavaşlatılmaktadır. Herhangi bir kontrolü olmayan bu sistemlerde kargoların düşeceği yerler de bilinmemektedir. Bu nedenle yönlendirilebilir paraşüt sistemlerine ihtiyaç duyulmuştur. İlk olarak 1960larda ortaya atılan bu sistemler günümüze kadar geliştirilmiştir.

Paraşütlü Kargo İniş Sistemleri (PKİS), bir uçağın veya helikopterin belirli bir yere sağlam ve hassas bir şekilde kargo teslim etmelerini sağlayan bir teknolojidir. Bu sistemler son yıllarda özellikle askeri ve sivil sektörlerde yaygınlaşmıştır ve uzak bölgelerde malzeme teslim etmek, acil durum yardımı ve tıbbi malzeme teslim etmek ve denizde gemilere kargo teslim etmek gibi çeşitli uygulamalar için kullanılmaktadır.

PKİS'nin temel bileşenleri, kargo konteyneri, paraşüt sistemi, navigasyon ve kontrol sistemleridir. Kargo konteyneri, iniş sürecinde kargoyu koruyan tasarımlar içermektedir. Paraşüt sistemi, kargoyu yavaşlatmak ve yönlendirmek için kullanılır. Genellikle ana paraşüt ve iniş sürecinin farklı aşamalarında uygulanan bir veya daha fazla paraşüt içerir. Navigasyon ve kontrol sistemleri, kargoyu hassas iniş noktasına yönlendirmek için kullanılır ve GPS, ataletsel navigasyon sistemleri ve barometrik altimetre gibi teknolojileri içerebilir.

PKİS geliştirilmesinde en önemli konulardan biri, kargoyu sağlam ve hassas bir şekilde iniş yapmasını sağlamaktır. Bu, rüzgâr, yükseklik ve yük ağırlığı gibi karmaşık bir faktörler kombinasyonunu içermektedir. İniş yerindeki rüzgâr koşulları, kargonun rotasını ve iniş doğruluğunu ciddi şekilde etkileyebilir. Yükseklik ve yük ağırlığı da kargonun iniş hızını belirlemede rol oynar. Bu nedenle, PKİS, çok esnek ve çeşitli koşullarda çalışabilecek şekilde tasarlanmalıdır.

Paraşütlü kargo iniş sistemlerinde kullanılan paraşüt tipleri, sistemin güvenli ve verimli bir şekilde çalışmasını sağlamak için önemlidir. Genel olarak, paraşütlü kargo iniş sistemlerinde kullanılan paraşüt tipleri şunlardır:

- **Rounded Parachutes:** Bu paraşütler, genellikle küçük ve orta boyutlu kargo yükleri için kullanılır. Rounded paraşütler, aerodinamik bir şekilde tasarlanmıştır ve yüksek hızlarda açılabilir. Bu paraşütler, yüksek manevra kabiliyetine sahiptir ve iniş sırasında düşük hızlarda hareket edebilir.
- **Ram-Air Parachutes:** Bu paraşütler, orta ve büyük boyutlu kargo yükleri için kullanılır. Ram-Air paraşütler, hava akımının etkisiyle şekillenen bir yelken şeklindedir. Bu paraşütler, daha yavaş iniş hızlarına sahiptir ve yüksek manevra kabiliyetine sahiptir.
- **Ring Parachutes:** Bu paraşütler, genellikle büyük boyutlu kargo yükleri için kullanılır. Ring paraşütler, birçok küçük paraşütlerin bir araya getirilmesiyle oluşur ve daha yavaş iniş hızlarına sahiptir. Bu paraşütler, daha yüksek yük taşıma kapasitesine sahiptir.
- **Square Parachutes:** Bu paraşütler, genellikle büyük boyutlu kargo yükleri için kullanılır. Square paraşütler, dörtgen şeklindedir ve yüksek hızlarda açılabilir. Bu paraşütler, yüksek manevra kabiliyetine sahiptir ve iniş sırasında düşük hızlarda hareket edebilir.

Bu paraşüt tipleri sadece örneklerdir ve her sistemin ihtiyacına göre farklı paraşüt tipleri de kullanılabilir. Ayrıca, paraşütlerin tasarımı, yapımı ve kullanımı için uzmanlık gerektirir ve paraşütlerin güvenli ve verimli bir şekilde kullanılması için gerekli olan tüm standartlar ve prosedürlerin dikkatli bir şekilde uygulanması gerekir.

Simülasyon çalışmaları, paraşütlü kargo iniş sistemlerinin tasarımı ve kontrolü için önemli bir araçtır. Bu çalışmalar, sistemlerin nasıl çalışacağını ve hangi parametrelerin sistemin performansını etkileyeceğini anlamak için kullanılmaktadır. Bunun yanı sıra, sistemin güvenliğini ve hassasiyetini arttırmak için gerekli olan değişiklikleri belirlemek için önemli bir araçtır. Fiziksel testler, zaman ve maliyet açısından yüksek maliyetli ve zaman alıcı olabilirken, simülasyonlar anında sonuçlar sağlamaktadır.

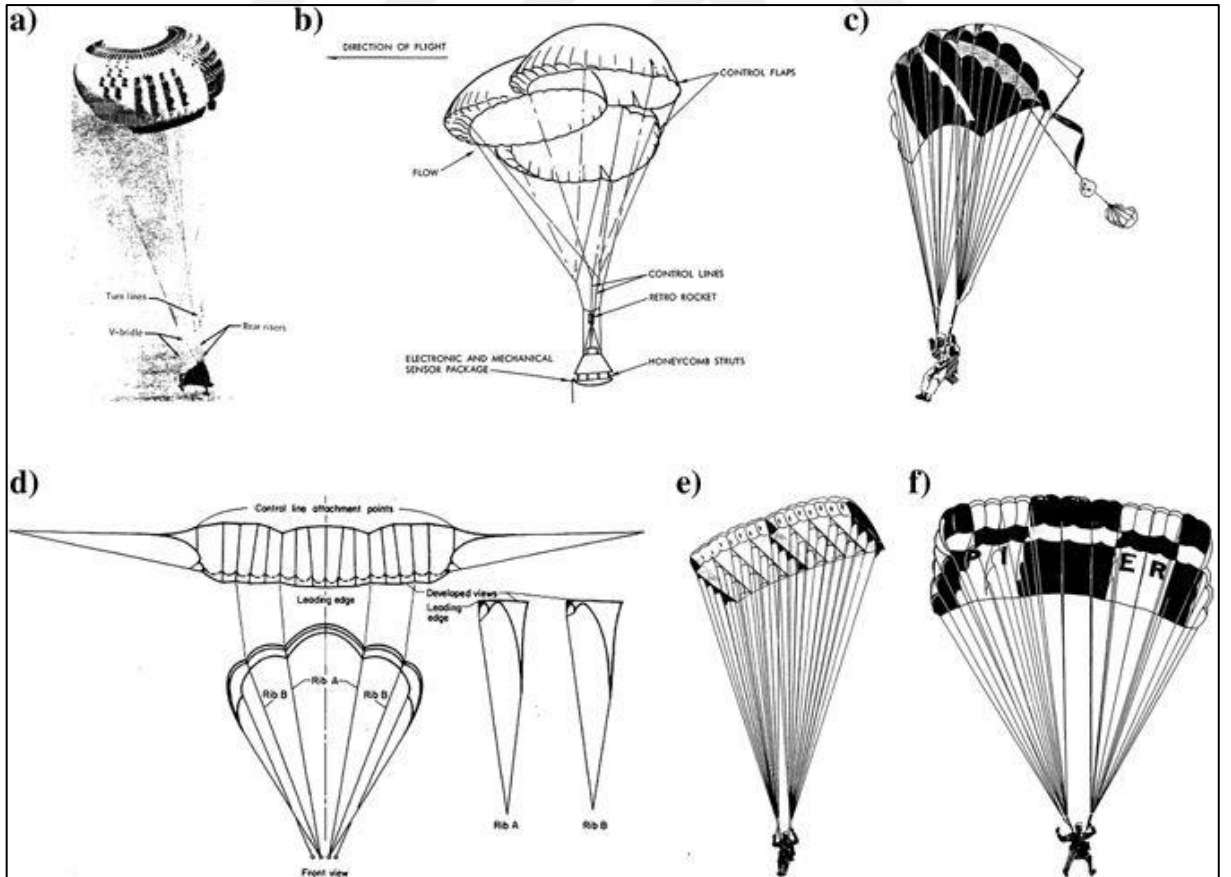
Bu sistemlerin doğruluğu ve verimliliği hava şartları, arazi ve insan hatası gibi birçok faktör tarafından sınırlandırılmıştır. Son yıllarda derin öğrenme teknikleri, paraşütlü kargo teslimatı gibi karmaşık sistemlerin modellenmesi ve kontrolü için güçlü araçlar olarak ortaya çıkmıştır.

Derin öğrenme, makine öğrenmesinin bir alt dalıdır ve büyük veri kümelerinde sanal nöral ağların eğitilmesini içerir. Bu ağlar, insan beyninin bilgi işleme yollarını taklit etmek üzere tasarlanmıştır ve görsel veriler, ses verileri ve sayısal veriler gibi geniş bir veri yelpazesinde eğitilebilir. Paraşütlü kargo teslimatı bağlamında, derin öğrenme, sistemin dinamiklerinin modellenmesine, yükün yöneliminin tahminine ve iniş sürecinin kontrolüne fayda sağlayabilmektedir.

Derin öğrenmenin paraşütlü kargo teslimatı için kullanılmasının avantajlarından biri, büyük miktarda veri işleme ve gerçek zamanlı tahmin yapma kabiliyetidir. Bu, teslimat sürecinin doğruluğunu ve güvenilirliğini arttırmaktadır.

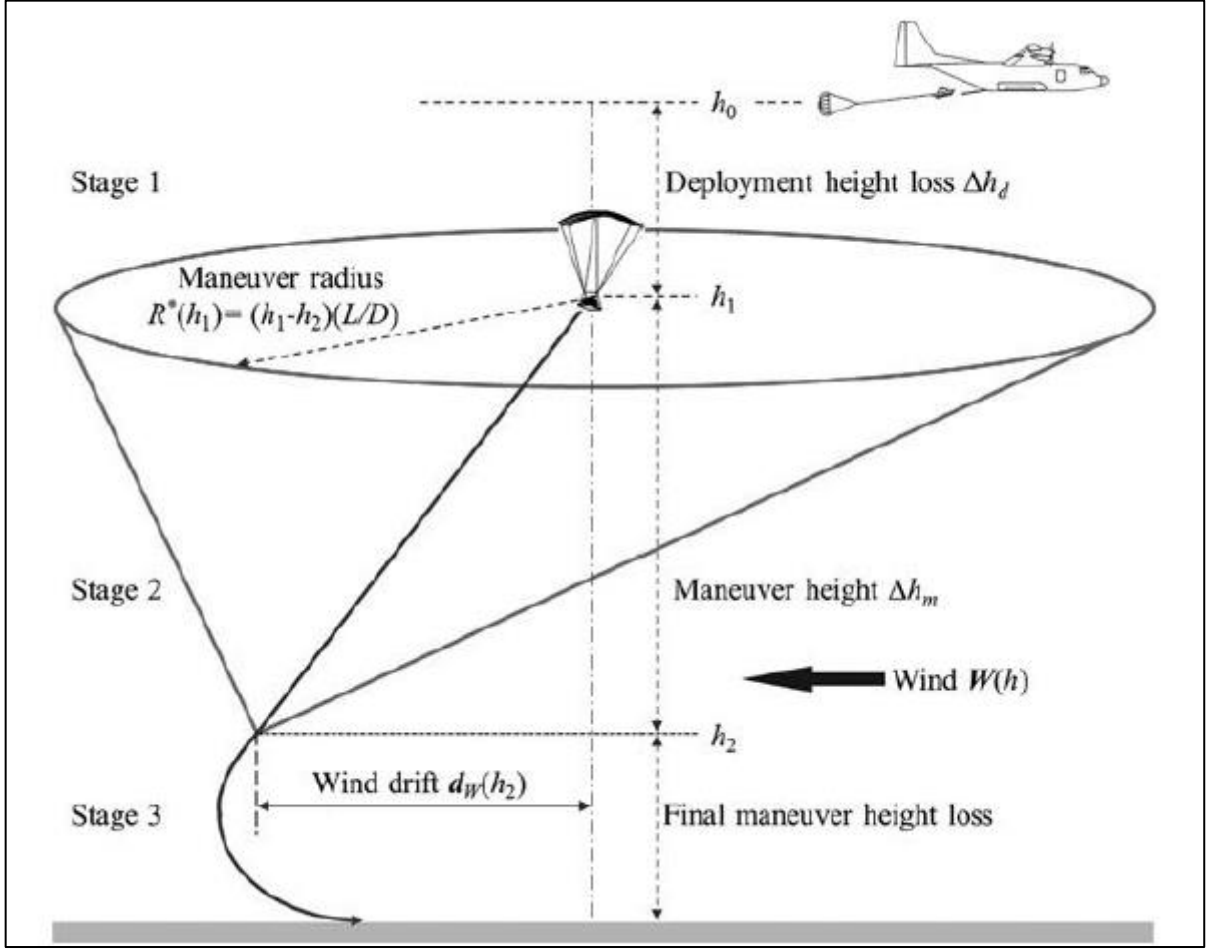
1.1. Literatür Araştırması

Geliştirilen sistemlerde çeşitli paraşüt tipleri üzerine çalışılmıştır. Süzülebilien özelliği paraşütlerin kontrol edilebilirliği diğerlerine kıyasla daha mümkün olmuştur [1]. Şekil 1.1’ de kullanılan paraşüt tipleri görülmektedir.



Şekil 1.1 Paraşüt tipleri: a) parasail, b) cloverleaf, c) parawing, d) sailwing, e) parafoil, f) volplane [1]

Otomatik iniş sistemlerinde en çok tercih edilen paraşüt tipi “Ram-air” tipleridir. Yüksek süzülme ve kontrol edilebilme özelliği bu tipin kullanılmasını yoğunlaştırmıştır.



Şekil 1.2. PKİS düşüş şematığı [2]

Şekil 1.2’ de görüldüğü gibi paraşütlü iniş sistemlerinde düşüş üç bölümde incelenmektedir. Bunlar ayrılma yüksekliği (h_0), manevra başlangıç yüksekliği (h_1) ve istenilen konuma gidebilmek için yapılan son manevra yüksekliğidir (h_2) [2].

İnsansız hava araçlarının testleri oldukça zorlu ortamlarda yapılmaktadır. Testlerde oluşacak hatalar çoğunlukla kırımla sonlandığı için büyük maddi kayıpların yanında zaman problemlerine yol açmaktadır. Bu nedenle bu tip sistemlerde simülasyon çalışmalarının önemi büyüktür. Paraşütlü kargo iniş sistemi simülasyon çalışmaları, paraşütlü kargo iniş sisteminin tasarımı, yapımı ve kullanımı gibi konuların daha iyi anlaşılması ve optimize edilmesi için yapılan bilgisayar tabanlı çalışmalardır. Bu çalışmalar, paraşütlü kargo iniş sisteminin fiziksel davranışlarını ve güvenliğini modelleyerek, gerçek dünya uygulamalarından daha az maliyetli ve güvenli bir şekilde test edilebilir. Geleneksel tipte

sabit kanat ve döner kanatlı sistemlerin kontrolcü çalışmalarında Matlab Simulink sıklıkla kullanılan programlardan biridir. Geliştirilecek sistemin bütünleşik simülasyonu için de Döngüde Yazılım(Software in the loop) ve Döngüde Donanım(Hardware in the Loop) metotları gerekmektedir. Gazebo ve ROS(Robot Operation System) bu tip çalışmalarda en popüler platformlardır. Ardupilot ve PX4 açık kaynaklı oto pilot yazılımları da bu sistemlerle beraber kullanılmaktadır. Geleneksel hava aracı modellerinin yazılım ve dokümanları bu sistemler için sıklıkla çalışılmıştır; fakat paraşütlü otonom iniş sistemi gibi özel bir sistemin SITL ve HITL destekli simülasyonları yeterli değildir.

Literatürde, ilk olarak paraşütlü sistemler için sayısal simülasyon yöntemlerine başvurulmuştur. Bu arada akışkan ve yapısal dinamiklerin üzerinde durulmuştur [3]. Bunun yanı sıra ram-air paraşüt tiplerinde şişme işlemi sırasındaki dinamiklerin gerçekleşmesi de bu tip simülasyon araçlarının önemli hedeflerinden biri olmuştur [4] [5] . Sistemin stabilite ve hareket analizleri içinse Matlab literatürdeki yerini almaktadır. Ginn ve arkadaşları, yaptıkları çalışmada ayırık kuvvet etkileri ile birlikte 6 serbestlik dereceli modelin stabilite analizlerini Matlab programı kullanarak yapmışlardır [6].

Bir başka çalışmada ise Matlab Simulink platformu kullanılmıştır. Dobrokhodov, yaptığı çalışmada kontrol edilebilir dairesel paraşütün 6 serbestlik dereceli modelini simulink ortamına taşımıştır. Döngüde donanım simülasyonlarının mümkün olması simulink tericisinde etkili olmuştur [7]. Mortaloni ve arkadaşları da yaptıkları çalışmada geliştirdikleri paraşütlü sisteminin simülasyonları için aynı ortamı kullanmışlardır [8]. Redkar ve Fiedler ise yaptıkları çalışmada simulink ortamında modellediği otonom paramotorun aerodinamik katsayıları için SimScale programını kullanmıştır [9].

Simülasyon ortamlarının yanında paraşütlü sistemler için geliştirilen kontrolcüler sistemin ağırlığına ve aerodinamik özelliklerine göre performansları etkilenmektedir. Literatürde bu tip sistemler için sistem tanımlama çalışmalar bu nedenle önem kazanmıştır. Sistem tanımlama işlemleri ise genellikle deneysel (rüzgâr tüneli vb.) olarak yapılmaktadır. Bunun sonucu olarak elde edilen modeller ise yine deneyde kullanılan sistem özelinde çalışmaktadır. Yapay zekânın günümüzde artan uygulama alanları sistem tanımlama çalışmalarını da kapsamaktadır. Geleneksel tipte hava araçlarındın bu tip uygulamalarında sistemin uçuş verisi belirlenen bir makine öğrenme modelinin eğitiminde kullanılmaktadır. Danışmanlı öğrenme metodu olarak geçen bu yöntem ile sistemin matematiksel modeli bir yapay sinir ağı modeli ile temsil edilebilmektedir. Bununla birlikte takviyeli öğrenme olarak

bilinen gerçek zamanlı öğrenmeyi de içeren bir başka yöntemde sistem modeli uçuş sırasında da tahmin edilebilmektedir.

Literatürde derin öğrenme yöntemleri, insansız hava araçlarının modellenmesi ve kontrolünde sıklıkla kullanılmaktadır. Sistemin doğası gereği sürekli girdi-çıkı durumlarından oluşan bu problemlerde pekiştirmeli öğrenme yöntemleri ön plana çıkmaktadır. Sinir ağları ve Q öğrenme yöntemlerinin birleşiminden oluşan Derin Q Ağları bu tip dinamiklerin modellenmesinde kullanılabilmektedir [10]. Bu algoritmanın dezavantajı düşük boyutlu uzaydaki problem çözümlerinde yüksek boyutlu gözlemler gerektirmesidir [11]. Bu problemin çözümü içinde Derin Deterministik Politika Gradyent (DDPG) isimli yeni bir algoritma önerilmiştir [12]. Derin Q Ağlara göre daha basit mimarisi olması uygulanabilirliğini kolaylaştırmaktadır.

Zhang ve arkadaşları, geliştirdikleri pekiştirmeli öğrenme mimarisi ile hava aracının rota takibini gerçekleştirmişlerdir. Derin Deterministik Politika Gradyent (DDPG) algoritmasını bu problem için uyarlamışlardır [13]. DDPG kullanılan bir başka çalışmada sabit kanatlı hava araçlarının iniş kontrolü için algoritma geliştirilmiştir [14]. Bu yaklaşımların hava araçları için de bir başka uygulama alanı hedef takip algoritmalarında olmuştur. Liu ve arkadaşları yaptıkları çalışmada görüntü üzerinden direk kontrol komutları üreten model geliştirmişlerdir [15].

Modelden bağımsız yaklaşımların yanı sıra sistemin nonlinear fonksiyonlar ile modellenip bu model üzerinden geliştirilen kontrolcüler sistem karmaşıklığını düşürmekte ve problemin aksiyon uzayını azaltmaktadır. Model öngörülü kontrol yöntemi literatürde sıklıkla kullanılan bir yöntem olarak karşımıza çıkmaktadır. Lucia ve Karg, yaptıkları çalışmada, doğrusal olmayan dinamik sistemler için geliştirdikleri model öngörülü kontrolcü algoritmalarının yüksek hesaplama yüklerinin hızlı çözülmesi üzerine yöntem bulmuşlardır. Geliştirdikleri yöntem ile gömülü bilgisayar uygulamalarında 37 ms hesaplama hızı sağlamışlardır [16]. Yine bir başka çalışmada ise gerçek zamanlı model öngörülü kontrolün hesaplama hızının artırılması için geliştirilen algoritma mevcuttur. Bu çalışmada hava aracının sinir ağı ile modellenmesi ve bu modelin güncellenmesi için geliştirdikleri eğitim algoritması 50Hz hızındaki örnekleme hızlarında çalışabilmektedir [17]. Paraşüt özelinde yapılan kontrolcü çalışmalarda ise kullanılan modellerin aerodinamik katsayılarla dayandırıldığı gözlemlenmiştir [18].

1.2. Tezin Amaç ve Kapsamı

Bu çalışmada istenilen bölgeye otonom olarak iniş yapabilen paraşütlü bir sistemin simülasyon ve deneysel incelemeleri amaçlanmaktadır. Yazılım ve donanımın bir bütün olarak planlandığı bu çalışmada Döngüde Donanım ve Döngüde Yazılım konuları ele alınmaktadır. Bu kapsamda tez çalışmasının amaçları sıralanabilir:

- i. Paraşütlü sistemin matematiksel modellenmesi
- ii. Matematiksel modelin MATLAB Simulink platformuna entegrasyonu
- iii. Matematiksel modelin Gazebo-ROS ortamına entegrasyonu
- iv. Paraşütlü kargo sisteminin mekanik, elektronik tasarımlarının geliştirilmesi ve uçuş verisinin elde edilmesi
- v. Model öngörülü kontrol algoritmasında NARX Sinir Ağı modelinin kullanımı ve PID kontrolcü ile karşılaştırılması

Çalışmanın sistem tanımlama bölümlerinde simülasyon ortamından elde edilen sanal veri ile uçuş testinden elde edilen gerçek uçuş verisi kullanılacaktır. NARX Sinir ağı kullanılarak modellenen sistem, uçuş sırasında model ağırlıklarının değiştirilmesi ile çevrimiçi olarak güncellenecektir. Böylelikle sistem aerodinamik parametrelerinden, ağırlığından ve çevresel durumlardan bağımsız dinamik model elde etmek hedeflenmiştir. Bu da istenilen hedefe iniş için geliştirilecek kontrolcülerin yine belirtilen aerodinamik ve çevresel parametrelerden bağımsız olmasını sağlayacaktır.

Tez toplamda sekiz başlık olarak düzenlenmiştir. İkinci bölümde, paraşütün aerodinamik yapısı kontrol girdilerinin etkisi özelinde incelenmiştir. Üçüncü bölüm sistemin hareket denklemlerinin modellendiği kısımdır. Dördüncü bölümde simualsyon çalışmalarına yer verilmiştir. Beşinci bölümde, mekanik ve elektronik tasarımları da içeren deneysel çalışmalar yer almaktadır. Altıncı bölüm, sistem modeli için derin öğrenme yöntemlerinin çalışıldığı bölümdür. Yedinci bölüm, sistemin hedef konuma gitmesi için gerekli rota planlama çalışmalarını içermektedir. Sekizinci bölüm ise hedef konuma gidiş için gerekli kontrolcü algoritmalarının geliştirildiği yerdir. Son olarak sonuçlar ve öneriler bölümüne yer verilmiştir.

2. PARAŞÜT AERODİNAMİĞİ

İçine hava dolan tipte olan paraşütler (Ram-Air), hava dolumundan sonra düşük açıklık oranına (aspect ratio) sahip kanat olarak düşünülebilir. Bu nedenle aerodinamik özellikler için geleneksel kanat teorisi kullanılmaktadır.

Yüksek açıklık oranına sahip kanatlar için kaldırma eğrisi hesaplarında “Lifting-line” teorisi kullanılmaktadır [19].

$$C_{L\alpha} = \frac{\pi C_{L\alpha}^{\alpha} AR}{\pi AR + C_{L\alpha}^{\alpha} (1 + \tau)} \quad (2.1)$$

2.1 numaralı denklemde $C_{L\alpha}^{\alpha}$, iki boyutlu kaldırma eğrisinin eğimini, AR, açıklık oranını, τ ise hücum açısını arttıran pozitif faktördür.

Düşük açıklık oranına sahip kanatlar içinse bu teori genişletilmiştir. Hoerner ve Borst düşük açıklık oranlı kanatlar için kaldırma eğrisinin eğimini yeniden tanımlamışlardır [20]. 2.2 ve 2.3 numaralı denklemlerde gösterilmiştir.

$$C_{L\alpha}^{\alpha'} = C_{L\alpha}^{\alpha} k \quad (2.2)$$

$$k = \frac{2\pi AR}{C_{L\alpha}^{\alpha}} \tanh\left(\frac{C_{L\alpha}^{\alpha}}{2\pi AR}\right) \quad (2.3)$$

Düşük açıklık oranına sahip kanatlar için kaldırma katsayısının farkı 2.4 ve 2.5 numaralı denklemlerde, Hoerner ve Borst tarafından tanımlanmıştır [20].

$$\Delta C_L = k_1 \sin^2(\alpha - \alpha_0) \cos(\alpha - \alpha_0) \quad (2.4)$$

$$k_1 = \begin{cases} 3.33 - 1.33AR, & 1 < AR < 2.5 \\ 0, & AR \geq 2.5 \end{cases} \quad (2.5)$$

Düzeltilmiş denklemler ve katsayılar kullanılarak paraşüt için toplam kaldırma 2.6 numaralı denklem kullanılarak yazılabilir.

$$C_L = C_{L\alpha}(\alpha - \alpha_0) + k_1 \sin^2(\alpha - \alpha_0) \cos(\alpha - \alpha_0) \quad (2.6)$$

“Lifting-line” teorisine göre dikdörtgen bir kanat için sürüklenme katsayısı;

$$C_D = C_{D0} + \frac{C_L^2(1 + \delta)}{\pi AR} \quad (2.7)$$

olarak yazılmaktadır. Bu denklemde δ , eliptik olmayan yüklemeler için kullanılan düzeltme katsayısıdır. Düşük açıklık oranlı kanatlarda oluşan fark ise Hoerner tarafından tanımlanmıştır [21].

$$\Delta C_D = k_1 \sin^3(\alpha - \alpha_0) \quad (2.8)$$

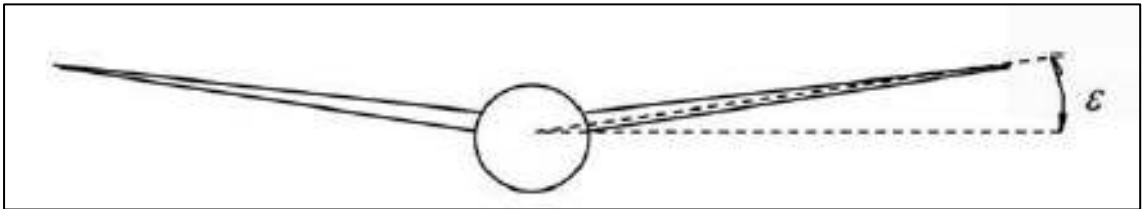
Toplam sürüklenme katsayısı;

$$C_D = C_{D0} + \frac{C_{LC}^2(1 + \delta)}{\pi AR} + k_1 \sin^3(\alpha - \alpha_0) \quad (2.9)$$

$$C_{LC} = C_{L\alpha}(\alpha - \alpha_0) \quad (2.10)$$

olarak yazılabilir.

Kanatlarda kaldırma katsayısı dihedral açısına (ε) göre düzeltilmektedir.

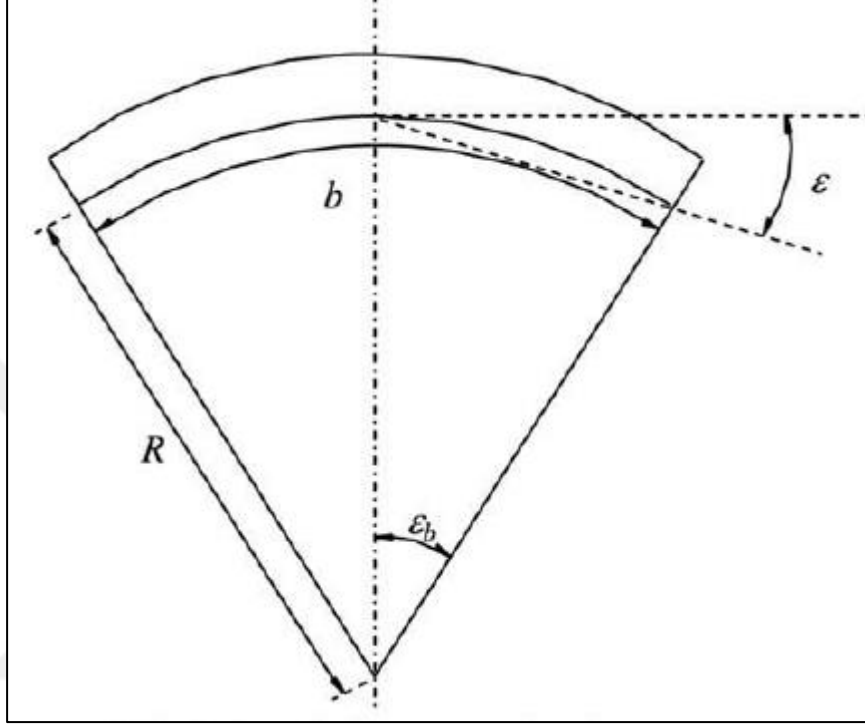


Şekil 2.1. Kanat dihedral açısı

Hoerner ve Borst, dihedral açısının kaldırma katsayısına etkisini;

$$C_L = C_{L, \varepsilon=0} \cos^2(\varepsilon) \quad (2.11)$$

formülü ile tanımlamıştır [20]. 2.11 numaralı denklem anhedral açısına sahip paraşütler içinde yazılabilir.



Şekil 2.2. Paraşüt anhedral açısı

2.12 numaralı denklemde olduğu gibi paraşüt için anhedral açısı genişlik ve ip uzunluğundan elde edilmektedir.

$$\varepsilon = \frac{b}{4R} \text{rad} \quad (2.12)$$

Anhedral etkisi de eklendiğinde, kaldırma katsayısı;

$$C_L = C_{L\alpha}(\alpha - \alpha_0) \cos^2(\varepsilon) + k_1 \sin^2(\alpha - \alpha_0) \cos(\alpha - \alpha_0) \quad (2.13)$$

olarak hesaplanabilir.

Sürüklenme katsayısına ise paraşütün haricinde iplerin ve kutunun etkisi vardır. İplerin sürüklenme katsayısına etkisi;

$$C_D^l = \frac{nRd\cos^3(\alpha)}{S} \quad (2.14)$$

olarak hesaplanır. Burada n, ip adetini, R, ortalama ip uzunluğunu, d ise ipin yarıçapını göstermektedir. Kutunun etkisi ise sürüklenme alanının paraşüt alanına oranına göre hesaplanmaktadır.

$$C_D^s = \frac{(C_D S)_s}{S} \quad (2.15)$$

İplerin ve kutunun etkisi ile toplam sürüklenme katsayısı 2.16 numaralı denklem ile yeniden yazılabilir.

$$C_D = C_{D0} + C_D^s + C_D^l + \frac{C_{L\alpha}^2(\alpha - \alpha_0)^2(1 + \delta)}{\pi AR} + k_1 \sin^3(\alpha - \alpha_0) \quad (2.16)$$

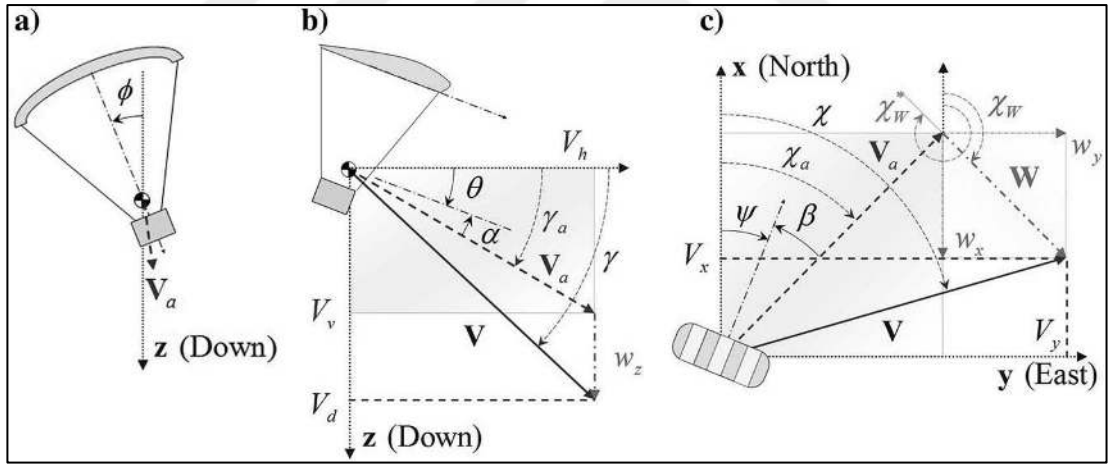
3. MATEMATİKSEL MODEL

Paraşütlü iniş sistemlerinin benzetim ve kontrol çalışmalarının yapılabilmesi için paraşütün matematiksel olarak modellenmesi gerekmektedir. Bu konuda günümüze kadar birçok çalışma yapılmıştır. Çalışmalar iki ana başlıkta toplanmıştır. Bunlardan birinde paraşüt ve kargo tek bir rijit cisim olarak düşünülmüş ve sistem bu şekilde modellenmiştir. 3 ve 4 serbestlikli modeller bu varsayım ile oluşturulmuştur.

Sistemin gerçeğe daha yakın modellenmesi için paraşüt ve kargonun ayrı olarak düşünüldüğü daha karmaşık modeller de oluşturulmuştur.

3.1. 3 Serbestlik Dereceli Modeller

Matematiksel model oluşturmada önce bir rijit cismin hareket denklemlerinin yazılması gerekmektedir. Paraşüt modeli oluşturulurken hava araçlarında sıklıkla kullanılan Kuzey-Doğu-Aşağı eksen takımı kullanılmıştır.



Şekil 3.1. Paraşüt Kuzey-Doğu-Aşağı Eksenini gösterimi

İlk olarak yer hızı vektörü denklem 3.1' de görüldüğü gibi sabit ekseninde yazılabilir. 3.2 nolu denklemde ise yer hızı vektörü gövde eksenine göre düzenlenmiştir.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad \dot{\mathbf{p}} = \mathbf{V} \quad (3.1)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = {}^b_n R^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \dot{p} = {}^b_n R^T V \quad (3.2)$$

Hava hızı vektörünün olduğu rüzgâr ekseninde Euler açıları yardımı ile tekrar düzenlenirse denklem 3.3 ve 3.4 elde edilir.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = V_a \begin{bmatrix} \cos(\gamma_a) \cos(\lambda_a) \\ \cos(\gamma_a) \sin(\lambda_a) \\ -\sin(\lambda_a) \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.3)$$

$$\dot{p} = {}^b_n R^T \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} + W \quad (3.4)$$

Denklem 3.6 ve 3.7' de gösterilen yatay ve dikey hız bileşenleri göz önünde bulundurularak denklem 3.5 yazılmaktadır.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} V_h \cos(\gamma_a) \\ V_h \sin(\gamma_a) \\ V_v \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.5)$$

$$V_h = V_a \cos(\gamma_a) \quad (3.6)$$

$$V_v = -\sin(\lambda_a) \quad (3.7)$$

En basit modellerden biri olan üç serbestlikli modelde dikey ve yatay hareket birbirinden ayrıştırılmıştır. Sapma açısının(ψ) değişimi kullanılarak ve kayma açısı(β) çok küçük olduğundan ihmal edilerek 3 serbestlikli model yazılabilir.

$$\dot{\psi} = u_c \quad (3.8)$$

3 serbestlikli modellerden gerçeğe en yakın olanlarından biri Jann tarafından geliştirilmiştir. Modelinde sapma açısının türevini de kullanmıştır [22].

$$T_\psi \ddot{\psi} + \dot{\psi} = K_\psi \delta_a \quad (3.9)$$

Bu denklemde T_ψ ve K_ψ sabitlerdir. δ_a ise kontrol iplerinin oluşturduğu asimetrik yapıdan meydana gelir.

ParaLander isimli paraşüt sistemi için de Altmann tarafından 3 serbestlik dereceli model geliştirilmiştir [23]. Lift ve sürüklenme katsayıları;

$$C_L = 0.58 + 0.8(\bar{\delta}_s - \delta_0) \quad (3.10)$$

$$C_D = 0.21 + 0.27(\bar{\delta}_s - \delta_0) \quad (3.11)$$

$$\bar{\delta}_s = 0.5(\bar{\delta}_r - \bar{\delta}_l) \quad (3.12)$$

olarak hesaplanmıştır.

3.2. 4 Serbestlik Dereceli Modeller

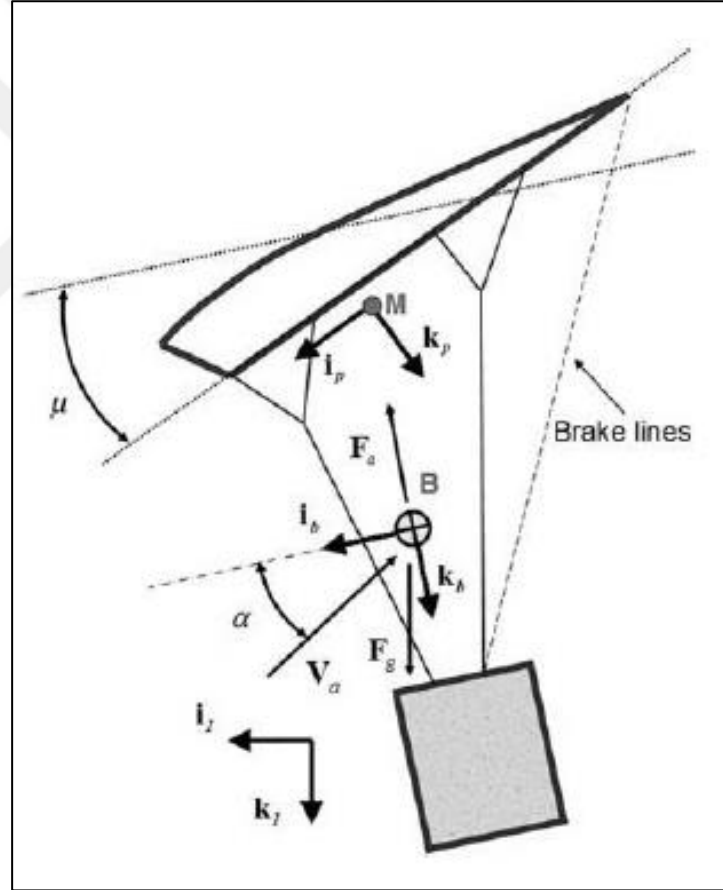
Üç serbestlikli modeller sistemin temel davranışı hakkında bilgi verse de birçok konuda eksikliği görülmektedir. Yalpa açısı uçuş boyunca oldukça değişmektedir. 3 serbestlikli modelde sabit kabul edilmesi modelin en büyük eksikliklerindendir. Bunun yanı sıra dönüşlerde kaldırma kuvvetinin eğimi irtifa kaybını hızlandırmaktadır. Bu sebeplerden ötürü daha gerçekçi bir modele ihtiyaç duyulmaktadır. Jann tarafından geliştirilen 4 serbestlik dereceli sistem aşağıda verilmiştir [22]. Bu model paraşüt üzerindeki dönmeyen eksen kullanılarak geliştirilmiştir. Denklem 3.13' te yalpa açısının etkisi gösterilmiştir. Denklem 3.14 ise kalan serbestlikleri göstermektedir.

$$T_\phi \dot{\phi} + \phi = K_\phi \delta_a \quad (3.13)$$

$$\begin{bmatrix} \dot{u} \\ \dot{\psi} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} L \sin(\alpha) - D \cos(\alpha) \\ 0 \\ -L \sin(\alpha) - D \cos(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ g \tan(\phi) \\ u \\ g \cos(\phi) \end{bmatrix} - \begin{bmatrix} w\dot{\psi} \sin(\phi) \\ \phi w \\ -u\dot{\psi} \sin(\phi) \end{bmatrix} \quad (3.14)$$

3.3. 6 Serbestlik Dereceli Modeller

6 serbestlik dereceli model için ilk olarak hareket denklemlerini yazmak gerekir. Gerekli dinamik denklemleri, denklem 3.15 ve 3.16’da olduğu gibi kütle merkezindeki üzerindeki kuvvet ve momentlerin toplamının açısai ve lineer momentum denklemlerine eklenmesiyle yazılabilir.



Şekil 3.2. Paraşütlü İniş Sistemi Serbest Cisim Diyagramı [8]

$$mI_{3 \times 3} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = F - mS(\omega) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.15)$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = M - S(\omega)I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.16)$$

Bu denklemde m kütle, I eylemsizlik matrisini, $[u \ v \ w]$ gövde eksenindeki doğrusal hızları, $[p \ q \ r]$ gövde eksenindeki açısal hızları, $S(\omega)$ doğrusal hız vektörlerinden oluşan eksi bakışimli (skew-symmetric) matrisi (denklem 3.17), F kuvveti ve M momenti temsil etmektedir.

$$S(\omega) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (3.17)$$

Paraşütlü iniş sistemlerinin x - z simetri düzleminden dolayı eylemsizlik matrisleri 4 farklı bileşenden oluşmakta olup denklem 3.18’de gösterilmiştir.

$$I = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{bmatrix} \quad (3.18)$$

Paraşüte etki eden kuvvet ve momentler yer çekimi ve aerodinamik kuvvetler nedeniyle oluşmaktadır. Şekil 3.2 de gösterilen gövde (b) eksenine göre yerçekimi kuvveti yazılabilir. Denklem 3.19’da θ , yunuslama açısını Φ , ise yalpalama açısını göstermektedir.

$$F_g = mg \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\Phi) \\ \cos(\theta) \cos(\Phi) \end{bmatrix} \quad (3.19)$$

Sisteme etki eden aerodinamik kuvvetler gövde eksenine göre ilgili aerodinamik katsayılar kullanılarak yazılır. 3.20 numaralı denklemde sisteme etki eden aerodinamik kuvvetler gösterilmektedir.

$$F_a = QS_w^b R \begin{bmatrix} C_{D0} + C_{D\alpha^2} \alpha^2 + C_{D\delta_s} \bar{\delta}_s \\ C_{Y\beta} \beta \\ C_{L0} + C_{L\alpha} \alpha + C_{L\delta_s} \bar{\delta}_s \end{bmatrix} \quad (3.20)$$

Bu denklemlerde g yerçekimi ivmesini, C aerodinamik katsayıları, α hücum açısını ve β kayma açısını göstermektedir. $\bar{\delta}_s = \frac{\delta_s}{\delta_{s\max}} \in [0; 1]$ ise simetrik kontrol girdisi bozunmasıdır. Hücum ve kayma açıları gövde eksenindeki hız vektörü yardımıyla hesaplanır. 3.21 ve 3.22 numaralı denklemlerinde sabit eksenden gövde eksenine geçiş matrisleri gösterilmiştir.

$${}^b_w R = R_\alpha R_\beta = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

$${}^b_w R = R_\alpha R_\beta = \begin{bmatrix} \cos(\alpha) \cos(\beta) & \cos(\alpha) \sin(\beta) & -\sin(\alpha) \\ -\sin(\beta) & \cos(\beta) & 0 \\ \sin(\alpha) \cos(\beta) & \sin(\alpha) \sin(\beta) & \cos(\alpha) \end{bmatrix} \quad (3.22)$$

Hücum ve sapma açıları gövde eksenindeki hız vektöründen elde edilir. Bunlar 3.23 ve 3.24 numaralı denklemlerde gösterilmektedir.

$$\alpha = \tan^{-1} \left(\frac{v_z}{v_x} \right) \quad (3.23)$$

$$\beta = \tan^{-1} \left(\frac{v_y}{\sqrt{v_x^2 + v_z^2}} \right) \quad (3.24)$$

Gövde eksenindeki hız vektörü ise 3.25 numaralı denklemde olduğu global hız ve rüzgar etkisinden oluşur.

$$V_a = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} - {}^b_n R \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (3.25)$$

b_nR , merkezi paraşütün kütle merkezi olan Kuzey-Doğu-Aşağı(North-East-Down) eksenindeki koordinat sisteminin, gövde eksenine aktarılmasını sağlayan dönüşüm matrisidir. Yalpalama(roll), yunuslama(pitch) ve sapma(yaw) açıları bu dönüşümde kullanılır. 3.26, 3.27, 3.28 ve 3.29 numaralı denklemler dönüşüm matrislerini göstermektedir.

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.26)$$

$$R_\theta = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.27)$$

$$R_\psi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix} \quad (3.28)$$

$${}^b_nR = R_\phi R_\theta R_\psi \quad (3.29)$$

Paraşüte etki eden aerodinamik momentler de ilgili katsayılar kullanılarak yazılabilir. Bunlar sırasıyla yalpalama, yunuslama ve sapma momentleridir [24]. 3.30 numaralı denklemde aerodinamik momentler gösterilmektedir.

$$M_a = \frac{\rho V_a^2 S}{2} \begin{bmatrix} b(C_{l\beta}\beta + \frac{b}{2V_a}C_{lp}p + \frac{b}{2V_a}C_{lr}r + C_{l\delta_a}\bar{\delta}_a) \\ \bar{c}(C_{m0} + C_{m\alpha}\alpha + \frac{c}{2V_a}C_{mq}q) \\ b(C_{n\beta}\beta + \frac{b}{2V_a}C_{np}p + \frac{b}{2V_a}C_{nr}r + C_{n\delta_a}\bar{\delta}_a) \end{bmatrix} \quad (3.30)$$

ρ hava yoğunluğunu, $\bar{c}$ ortalama aerodinamik kortu, $\bar{\delta}_a = \frac{\delta_a}{\delta_{a\max}} \in [-1; 1]$ asimetrik kontrol kanadı bozulmasını ve S paraşütün referans alanını göstermektedir.

Aerodinamik kuvvetlere ek olarak paraşüt içine dolan havadan kaynaklanan ayrık kütle(apparent mass) etkisi de mevcuttur. Bu etki ağır kanatlı uçaklar da ihmal edilebilir de

olsa paraşütlerde göz ardı edilemeyeceği kanıtlamıştır [25]. Lissaman bu etki için 6 adet terim önermiştir; A, dalga terimi, B, yana kayma terimi, C, batma terimi, I_A , yalpa terimi, I_B , yunuslama terimi ve I_C , sapma terimidir. Bu terimler kullanılarak bu etkinin oluşturduğu kuvvet ve momentler 3.31-3.36 numaralı denklemlerde gösterildiği gibi yazılabilir.

$$F_1 = -A \frac{\partial U}{\partial t} - CWq + BVr \quad (3.31)$$

$$F_2 = -B \frac{\partial V}{\partial t} - AUr + CWp \quad (3.32)$$

$$F_3 = -C \frac{\partial W}{\partial t} - BVp + AUq \quad (3.33)$$

$$M_1 = -I_A \frac{\partial p}{\partial t} - (I_C - I_B)qr - (C - B)WV \quad (3.34)$$

$$M_2 = -I_B \frac{\partial q}{\partial t} - (I_A - I_C)rp - (A - C)UW \quad (3.35)$$

$$M_3 = -I_C \frac{\partial r}{\partial t} - (I_B - I_A)pq - (B - A)VU \quad (3.36)$$

Bu denklemleri basitçe;

$$F_{AM} = [F_1 \quad F_2 \quad F_3]^T \quad (3.37)$$

$$M_{AM} = [M_1 \quad M_2 \quad M_3]^T \quad (3.38)$$

$$K_{ABC} = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix} \quad (3.39)$$

$$K_{I_A I_B I_C} = \begin{bmatrix} I_A & 0 & 0 \\ 0 & I_B & 0 \\ 0 & 0 & I_C \end{bmatrix} \quad (3.40)$$

olarak yazılabilir. 3.41 ve 3.42 numaralı denklemlerde ayrı kütledenin etkisinden oluşan kuvvet ve momentler gösterilmektedir.

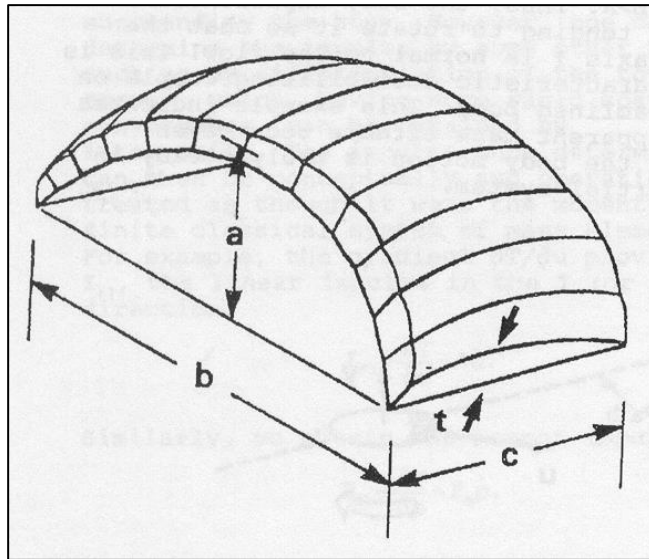
$$F_{AM} = -K_{ABC} \dot{v} - \omega \times (K_{ABC} v) \quad (3.41)$$

$$M_{AM} = -K_{I_A I_B I_C} \dot{\omega} - \omega \times (K_{I_A I_B I_C} \omega) - v \times (K_{ABC} v) \quad (3.42)$$

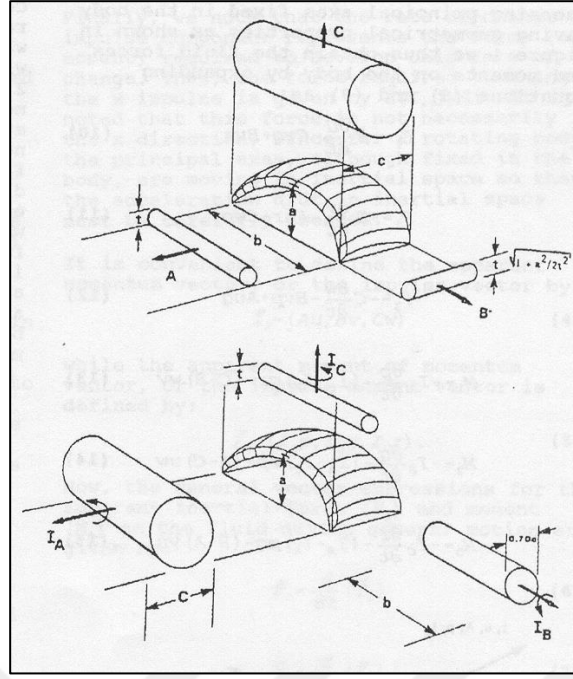
Son olarak aerodinamik ve ayrı kütledenin etkileri birleştirilerek hareket denklemleri yazılabilir. 3.43 ve 3.44 bu etkilerin de bulunduğu modeldir.

$$\dot{v} = \left(I + \frac{1}{m} K_{ABC} \right)^{-1} \left(\frac{1}{m} F_{aero} + G - \omega \times \left(v + \frac{1}{m} (K_{ABC} v) \right) \right) \quad (3.43)$$

$$\dot{\omega} = (J + K_{I_A I_B I_C})^{-1} (M_{aero} - \omega \times (K_{I_A I_B I_C} \omega) - v \times (K_{ABC} v) - \omega \times J \omega) \quad (3.44)$$



Şekil 3.3. Paraşüt geometrisi.



Şekil 3.4. Ayrık kütle ve momentlerin hacimsel gösterimi

Lissaman terimleri paraşüt geometrisi göz önünde bulundurularak üretilmiştir. Bunlar sırasıyla 3.45-3.50 numaralı denklemlerde gösterilmektedir.

$$A = 0.899 \frac{\pi}{4} t^2 b \quad (3.45)$$

$$B = 0.39 \frac{\pi}{4} (t^2 + 2a^2) c \quad (3.46)$$

$$C = 0.783 \frac{\pi}{4} c^2 b \quad (3.47)$$

$$I_A = 0.630 \frac{\pi}{48} c^2 b^3 \quad (3.48)$$

$$I_B = 0.817 \frac{4}{48\pi} c^4 b \quad (3.49)$$

$$I_C = 1.001 \frac{\pi}{48} t^2 b^3 \quad (3.50)$$

Çalışmanın ilerleyen bölümlerinde simülasyon çalışmalarına yer verilmiştir. Bu çalışmalarda 6 serbestlik dereceli model referans alınmış ve ayrık kütle etkileri ihmal edilmiştir.



4. SİMÜLASYON ÇALIŞMALARI

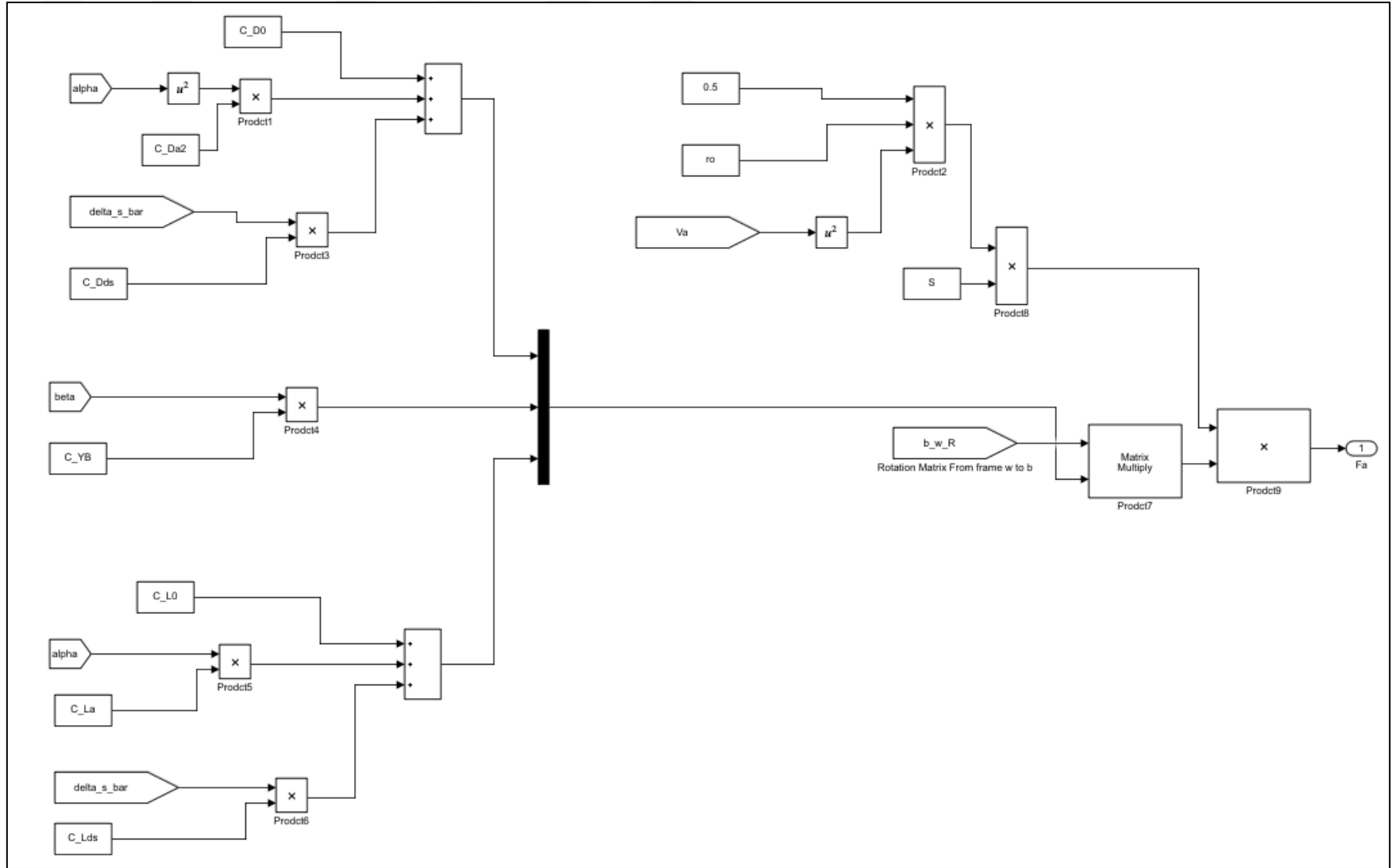
Bu bölümde geliştirilen sistemin kontrolcü testlerinin yapılacağı simülasyon ortamları incelenmiştir. Dinamik model olarak 3. Bölümde anlatılan 6 serbestlik dereceli model kullanılmıştır.

İlk olarak sistem Matlab Simulink kullanılarak modellenmiştir. Hızlı uygulanabilirliği ve basit kontrolcü yöntemlerinin denenebilirliği başlangıç için uygun görülmüştür. Gerçek sistemde kullanılacak olan otopilot kartının (Pixhawk) döngüde yazılım (Software in the loop – SITL) ve döngüde donanım (Hardware in the loop - HITL) çalışmaları için ise Gazebo platformu tercih edilmiştir. ROS (Robot Operating System) ile uyumluluğu tercih sebeplerinden başlıcalarıdır.

4.1. MATLAB Uygulaması

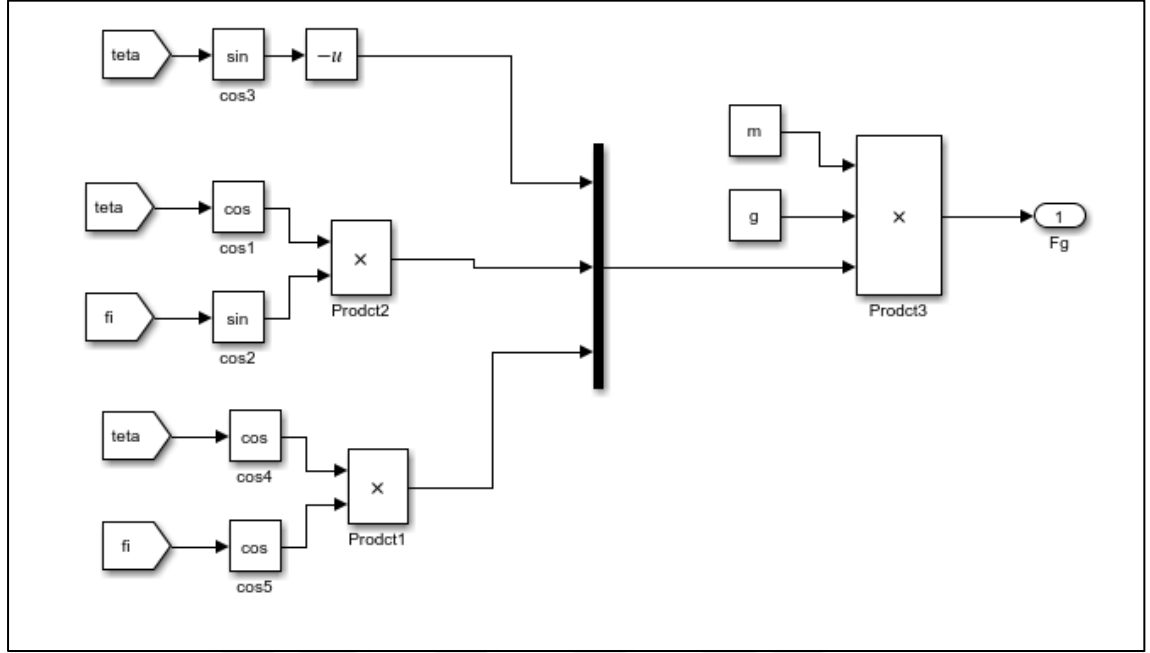
Simülasyon çalışmalarında Matlab Simulink programı kullanılmıştır. Model olarak ise 6 serbestlik dereceli model seçilmiştir. Simülasyon için gerekli parametreler Snowflakes isimli paraşütlü otonom iniş sistemi göz önünde bulundurularak kullanılmıştır [26].

Sistem üzerine etki eden kuvvet ve momentler denklem (3.20) ve denklem (3.30) da verilmiştir. İlk olarak aerodinamik kuvvetler Simulink programına eklenmiştir.



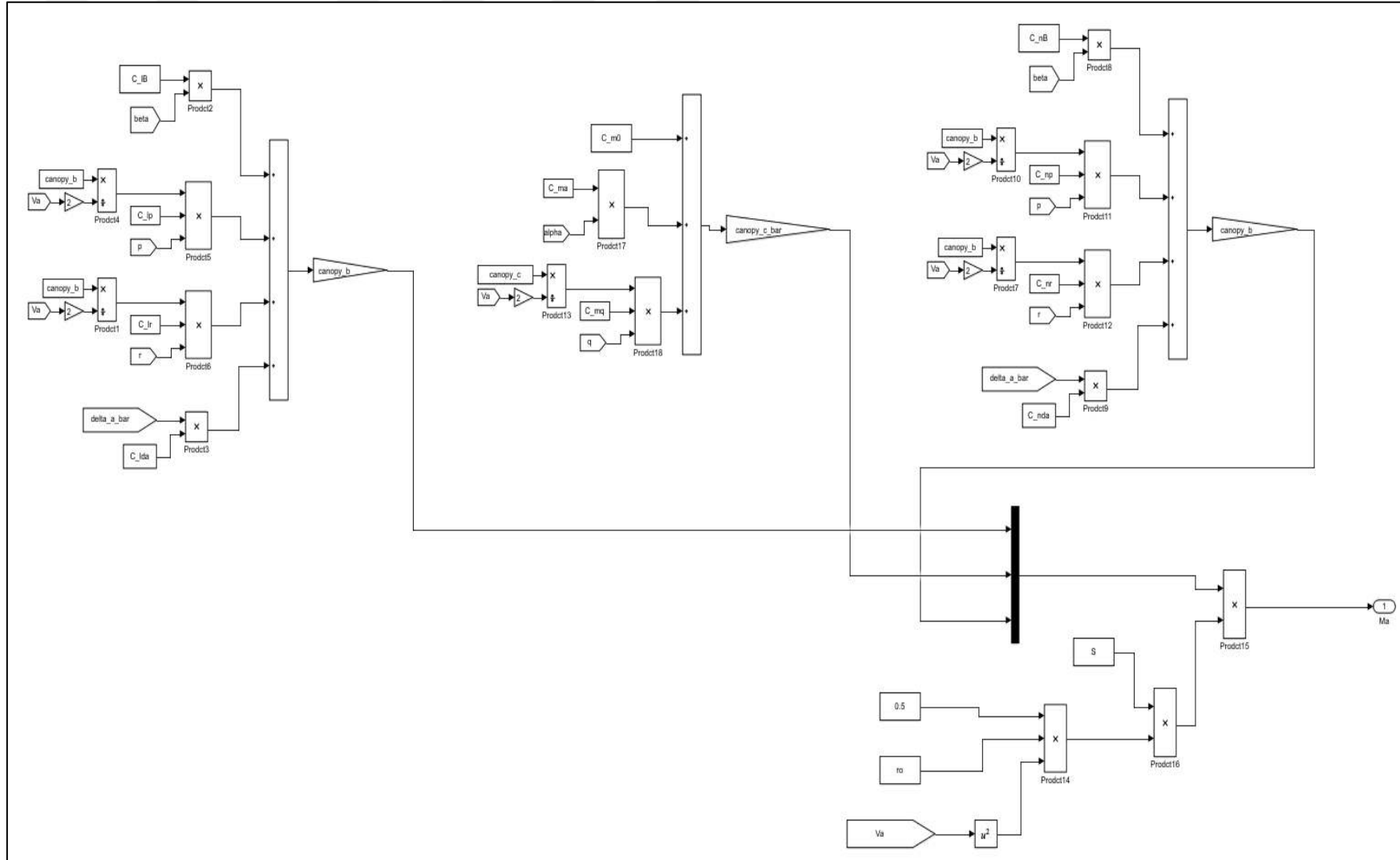
Şekil 4.1. Aerodinamik Kuvvet Simulink Bloğu

Aerodinamik kuvvetlere ek olarak denklem (3.19) da gösterilen yer çekimi kuvvetinin sisteme etkisi simulink bloğuna eklenmiştir.



Şekil 4.2. Yerçekimi Kuvveti Simulink bloğu

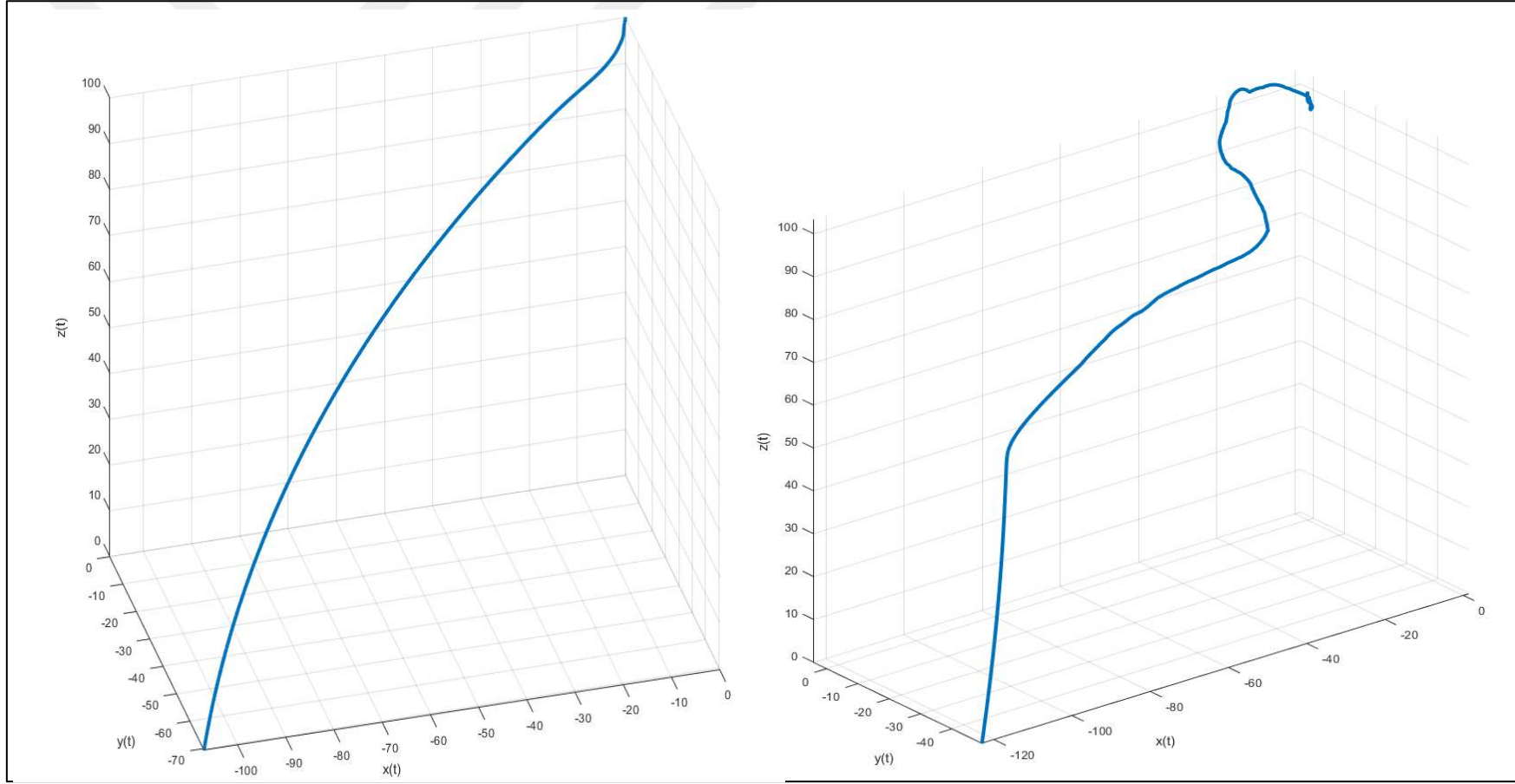
6 serbestlik dereceli model için gerekli moment hesapları da Simulink bloğuna eklenmiştir.



Şekil 4.3. Aerodinamik momentler Simulink bloğu

İlk olarak paraşüt rüzgarsız ortamda 100m yükseklikten serbest bırakılmıştır. Rüzgarsız ve $-x$ yönünde 5 m/s rüzgarlı ortamda paraşütün izlediği yol gösterilmiştir (Şekil 4.4).





Şekil 4.4. 100m Serbest düşüş paraşüt güzergahı (rüzgarsız – rüzgarlı)

4.2. Döngüde Yazılım (Software in the Loop)

Döngüde yazılım çalışmaları için gerekli simülasyon ortamı üç ana bölümden oluşmaktadır. Bunlar aşağıdaki gibi sıralanabilir.

- PX4 (açık kaynak otopilot yazılımı)
- ROS(Robot Operating System)
- Gazebo

Platform olarak Ubuntu 18.04 işletim sistemi ve ROS Melodic versiyonu tercih edilmiştir.

4.2.1. Oto Pilot Yazılımı

PX4, insansız araçlar için geliştirilmiş açık kaynaklı kontrol yazılımıdır. Haberleşme için MAVLink protokolünü kullanır. Burada haberleşme paketleri C dilinde oluşturulmuş stringlerdir. Bilgi paketlerinin yanı sıra, kontrol için gerekli komutlar ya da parametrelerin ayarlanması gibi işlemler MAVLink protokolü ile yapılabilir [27].

PX4 ün desteklediği insansız araçlar arasında uçaklar, döner kanatlı sistemler, kara ve deniz araçları yer almaktadır. Paraşütlü İniş Sistemi için desteği bulunmamaktadır. Bu çalışmada ilk olarak PX4'e PKİS için gerekli eklentiler yapılmıştır.

- PX4 te gövde konfigürasyonları ilgili config dosyalarında tutulmaktadır. Konfigürasyon dosyalarında;
- Gövde hakkında bilgiler
- Araç parametreleri
- Başlatılacak kontrolcüler ve uygulamalar
- Fiziksel konfigürasyon

bilgileri yer almaktadır. İlk bölümde ilgili gövdenin temel özellikleri yer alır.

```
#!/bin/sh
#
# @name POIS
#
# @url https://github.com/kguven/POIS
#
# @type Precision Aerial Delivery System
# @class Parafoil
#
# @output MAIN1 left control
# @output MAIN2 right control
#
# @output AUX1 feed-through of RC AUX1 channel
#
# @maintainer Kemal Güven <kkemalguven@gmail.com>
#
# @board px4_fmu-v2 exclude
```

Şekil 4.5. Gövde bilgileri

İkinci bölüm tasarlanan gövde için spesifik PX4 parametrelerinin ayarlanmasını sağlamaktadır.

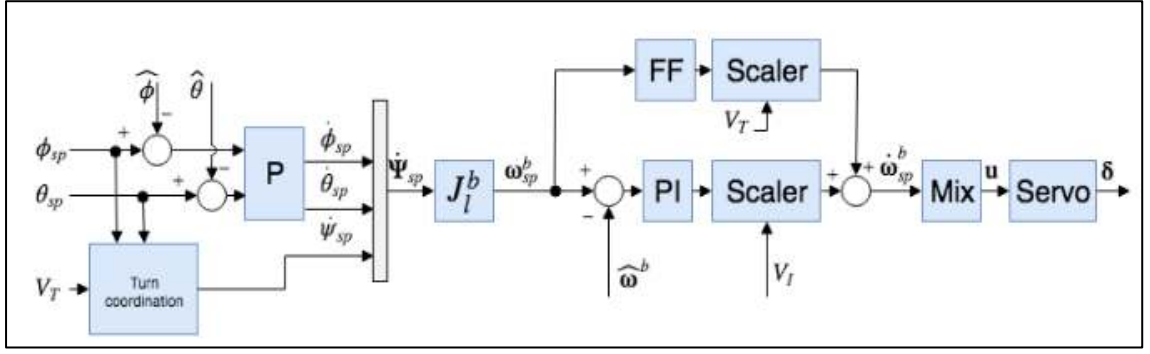
```
. ${R}etc/init.d/rc.pois_defaults

param set-default BAT_N_CELLS 3
param set-default POIS_LND_ANG 15
param set-default POIS_THR_LND_MAX 0
param set-default PWM_MAIN_DISARM 1000

set MAV_TYPE 1
#set MIXER
set PWM_OUT 4
```

Şekil 4.6. PX4 Parametreleri

Mixer dosyaları hava araçlarının kontrolünde gerekli gövde bazlı servo çıktılarının oluşturulması için kullanılır. PX4 yazılımında kullanılan Cascade kontrolcü çıktılar mixer dosyaları ile servo çıktılarına dönüştürülür. Çalışma kapsamında PX4 te bulunan mevcut kontrolü kullanılmayacağı için mixer dosyası eklenmemiştir.



Şekil 4.7. Sabit Kanat Yönelme Kontrolü [28]

4.2.2. Robot İşletim Sistemi (ROS)

ROS, robotik sistem yazılımlarının uygulanmasını kolaylaştırmak amacıyla geliştirilen açık kaynaklı yazılımlar ve kütüphaneler bütünüdür [29]. Tam anlamıyla bir işletim sistemi değildir. Aynı platformdaki sensörlerin ve robotların birlikte çalışabilirliğini sağlayan bir platformdur.

ROS ta sistemlerden gerçek zamanlı veriler düğüm(Node) adı verilen modüller ile gerçekleştirilir. Senkron veya asenkron olarak bu düğümlere bağlanılabilmesi geliştiricilerin harcadıkları eforları oldukça azaltmaktadır. Aynı zamanda sistemlerin modüler olmasını sağlamaktadır.

ROS ortamında fiziksel tanımlamalar özel dosyalarla yapılmaktadır. URDF(Unified Robot Description Format) uzantılı olan bu dosyalarda robotun bütün fiziksel özellikleri temsil edilir. Bu sayede Gazebo ortamından da 3 boyutlu olarak simülasyonu gerçekleştirilir. Mavlink mesajlarının ROS ortamında kullanılabilmesi için MAVROS paketi kullanılmıştır [30].

```

<arg name="x" default="0"/>
<arg name="y" default="0"/>
<arg name="z" default="500"/>
  <arg name="R" default="0"/>
  <arg name="P" default="0"/>
  <arg name="Y" default="0"/>
  <arg name="drop_delay" default="30"/>
  <!-- vehicle model and world -->
  <arg name="est" default="ekf2"/>
  <arg name="vehicle" default="pois"/>
  <arg name="world" default="$(find
mavlink_sitl_gazebo)/worlds/pois.world"/>
  <arg name="sdf" default="$(find mavlink_sitl_gazebo)/models/$(arg
vehicle)/$(arg vehicle).sdf"/>

```

Şekil 4.8. PKİS Launch dosyası 1

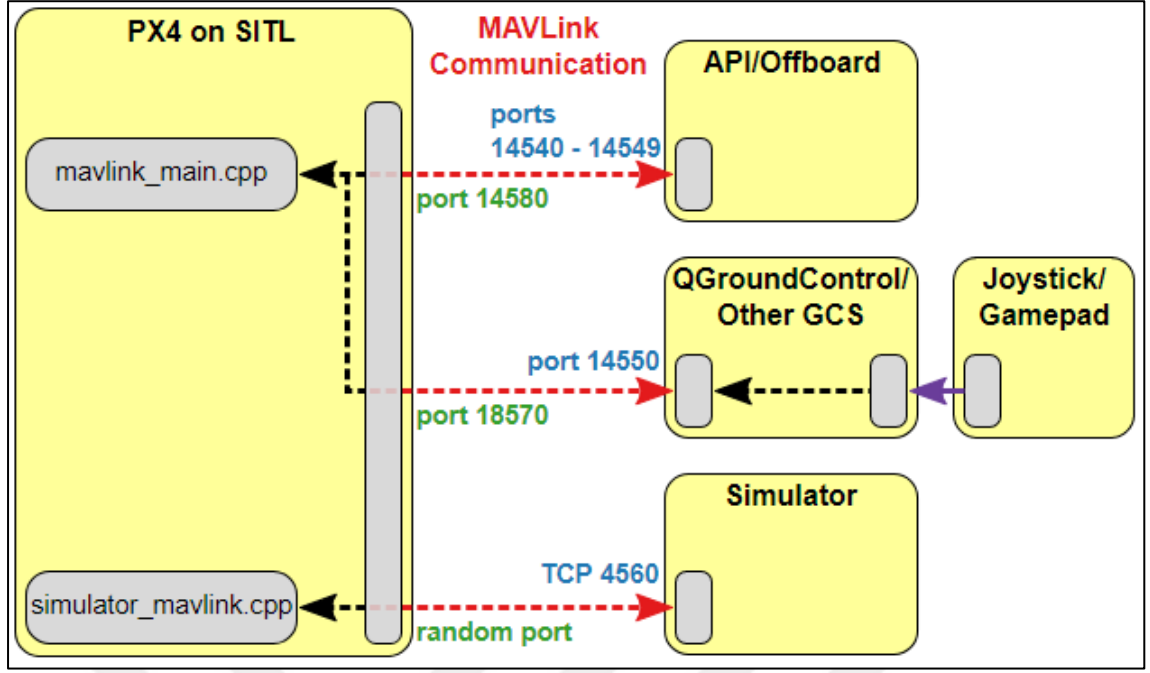
ROS düğümlerini ve bağlı çalışacakları master sistemini başlatılması için roslaunch komutu kullanılarak launch dosyaları koşulur. Simülasyon kapsamında oluşturulan launch dosyasının ilk bölümü sistemin başlangıç koşullarını tanımlar. Diğer komutlar sırasıyla PX4 ve MAVROS düğümlerinin başlatılmasıdır.

```

<arg name="fcu_url" default="udp://:14540@localhost:14557"/>
<arg name="respawn_mavros" default="false"/>
<include file="$(find px4)/launch/posix_sitl.launch">
  <arg name="x" value="$(arg x)"/>
  <arg name="y" value="$(arg y)"/>
  <arg name="z" value="$(arg z)"/>
  <arg name="R" value="$(arg R)"/>
  <arg name="P" value="$(arg P)"/>
  <arg name="Y" value="$(arg Y)"/>
  <arg name="world" value="$(arg world)"/>
  <arg name="vehicle" value="$(arg vehicle)"/>
  <arg name="sdf" value="$(arg sdf)"/>
  <arg name="gui" value="$(arg gui)"/>
  <arg name="interactive" value="$(arg interactive)"/>
  <arg name="debug" value="$(arg debug)"/>
  <arg name="verbose" value="$(arg verbose)"/>
  <arg name="paused" value="$(arg paused)"/>
  <arg name="respawn_gazebo" value="$(arg respawn_gazebo)"/>
</include>
<node pkg="rosservice" type="rosservice" name="apply_body_wrench"
args="call gazebo/delete_model '{model_name: box_target_green}'" launch-
prefix="bash -c 'sleep $(arg drop_delay); $0 $@"'"/>

```

Şekil 4.9. PKİS Launch dosyası 2



Şekil 4.10. PX4 Döngüde donanım Şeması [31]

PKİS'in serbest düşme anının kontrolü için bırakılma yüksekliğinde bir düzlem oluşturulmuştur. İstenilen zamanda bu düzlem kaldırılarak düşme gerçekleşmektedir.

4.2.3. Gazebo Uygulaması

Gerçekleştirilen simülasyonların görselleştirilmesi algoritmaların anlaşılmasında kolaylık sağlayabilir. Gazebo, bu tip sistemler için çok gerçekçi ortamlar sunabilir. Çevresel etkiler dahil olmak üzere dinamik ve kinematik sistemleri modelleyebilen fizik motoruna sahiptir [32]. ROS ve Gazebo arasındaki entegrasyon *plugin* adı verilen yazılımlar ile sağlanabilir. Bu pluginler sayesinde ROS mesajlarına uyumlu düğümler yazılabilir. Böylelikle simülasyon ortamında test edilen sistemlerin gerçek donanımlarına entegrasyonunda çok az değişiklikler gerekir.

PKİS in gazebo ortamında simülasyonu için aerodinamik kuvvet ve momentlerin hesaplanmasını sağlayacak plugin gerekmektedir. Burada geliştirilen pluginde 6DOF model kullanılmıştır.

Pluginde gerekli aerodinamik katsayılar ve fiziksel parametreler model için oluşturulan sdf dosyasından çekilmektedir. Pluginde ilgili linke yine sdf dosyasından eklenmiştir.


```

<plugin name='aerodynamics'
filename='libgazebo_pois_aerodynamics_plugin.so'>
  <robotNamespace/>
  <linkName>base_link</linkName>
  <air_density>1.225</air_density>
  <cd0>0.25</cd0>
  <cdalpha2>0.12</cdalpha2>
  <cddeltas>0.3468</cddeltas>
  <cybeta>-0.23</cybeta>
  <cl0>0.091</cl0>
  <clalpha>0.9</clalpha>
  <cldeltas>0.4138</cldeltas>
  <b>2.4</b>
  <clbeta>-0.036</clbeta>
  <clp>-0.84</clp>
  <clr>-0.082</clr>
  <cldeltaa>-0.0035</cldeltaa>
  <cbar>0.01</cbar>
  <cm0>0.15</cm0>
  <cmalpha>-0.72</cmalpha>
  <c>1</c>
  <cmq>-1.49</cmq>
  <cnbeta>-0.0015</cnbeta>
  <cnp>-0.082</cnp>
  <cnr>-0.27</cnr>
  <cndeltaa>2.215</cndeltaa>
  <left_control_joint_name>
    left_joint
  </left_control_joint_name>
  <right_control_joint_name>
    right_joint
  </right_control_joint_name>
  <brake_control_joint_name>
    brake_joint
  </brake_control_joint_name>
</plugin>

```

Şekil 4.11. PKİS sdf dosyası

Geliştirilen pluginde PKİS'in durum değişkenleri Gazebo'dan alınmaktadır. Kontrol girdileri ise PX4 tarafından üretilip Gazebo'ya MAVROS paketi ile iletilmekte ve modelde bulunan sanal eklemleri kontrol etmektedir. Bu eklemlerin pozisyonlarına göre kontrol girdileri hesaplanıp aerodinamik modelde kullanılmaktadır.

```

<plugin name='mavlink_interface'
filename='libgazebo_mavlink_interface.so'
  <robotNamespace></robotNamespace>
  <imuSubTopic>/imu</imuSubTopic>
  <gpsSubTopic>/gps</gpsSubTopic>
  <magSubTopic>/mag</magSubTopic>
  <baroSubTopic>/baro</baroSubTopic>
  <mavlink_addr>INADDR_ANY</mavlink_addr>
  <mavlink_udp_port>14560</mavlink_udp_port>
  <serialEnabled>>false</serialEnabled>
  <serialDevice>/dev/ttyACM0</serialDevice>
  <baudRate>921600</baudRate>
  <qgc_addr>INADDR_ANY</qgc_addr>
  <qgc_udp_port>14550</qgc_udp_port>
  <sdk_addr>INADDR_ANY</sdk_addr>
  <sdk_udp_port>14540</sdk_udp_port>
  <hil_mode>>false</hil_mode>
  <hil_state_level>>false</hil_state_level>
  <enable_lockstep>>true</enable_lockstep>
  <use_tcp>true</use_tcp>

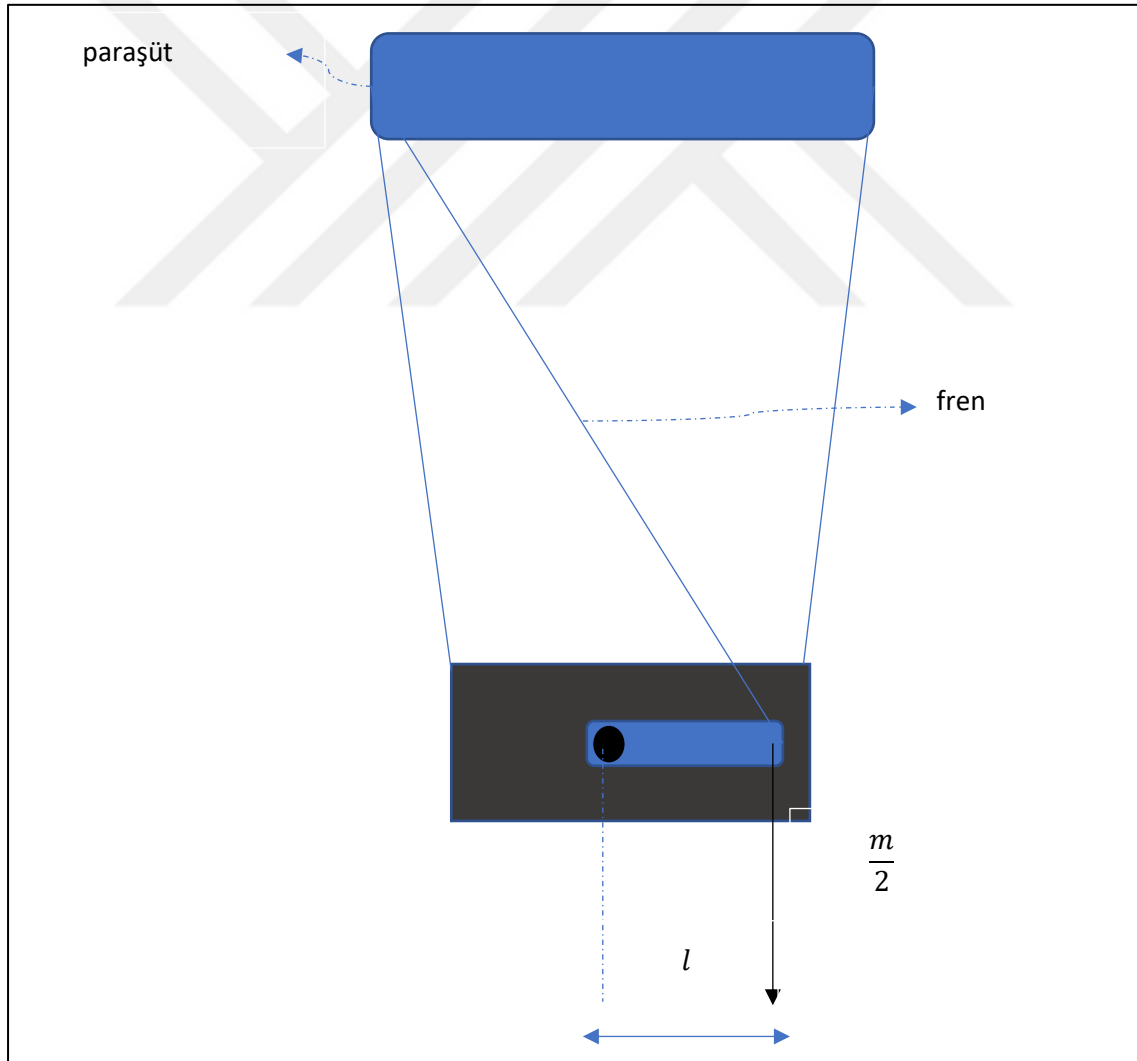
  <motorSpeedCommandPubTopic>/gazebo/command/motor_speed</motorSpeedCommandPubTopic>
  <control_channels>
    <channel name="brake">
      <input_index>4</input_index>
      <input_offset>0</input_offset>
      <input_scaling>1</input_scaling>
      <zero_position_disarmed>0</zero_position_disarmed>
      <zero_position_armed>0</zero_position_armed>
      <joint_control_type>position_kinematic</joint_control_type>
      <joint_name>brake_joint</joint_name>
    </channel>
    <channel name="left">
      <input_index>5</input_index>
      <input_offset>0</input_offset>
      <input_scaling>1</input_scaling>
      <zero_position_disarmed>0</zero_position_disarmed>
      <zero_position_armed>0</zero_position_armed>
      <joint_control_type>position_kinematic</joint_control_type>
      <joint_name>left_joint</joint_name>
    </channel>
    <channel name="right">
      <input_index>6</input_index>
      <input_offset>0</input_offset>
      <input_scaling>1</input_scaling>
      <zero_position_disarmed>0</zero_position_disarmed>
      <zero_position_armed>0</zero_position_armed>
      <joint_control_type>position_kinematic</joint_control_type>
      <joint_name>right_joint</joint_name>
    </channel>
  </control_channels>
</plugin>

```

Şekil 4.12. POIS Sdf dosyası MAVROS eklentisi

5. DENEYSEL ÇALIŞMA

Deneyisel çalışmalar için sistemin mekanik tasarımı yapılmıştır. Azami faydalı yük ağırlığı 3kg olarak belirlenmiştir. Burada iki serbestlik dereceli sağ ve sol fren çekmeli paraşütlü iniş sistemi üretilmiştir. Mekanik tasarımda ana platform olarak karbon fiber plakalar kullanılmıştır. Hafif ve sağlam oluşu bu tercihte ön plana çıkmıştır. Düşüşteki darbenin sönümlenmesi için köşelere TPU filamentle üretilen sönümleyiciler monte edilmiştir. Fren kontrolleri için iki adet servo motor ve karbon fiber çubuklar kullanılmıştır. Üst platformun köşelerine paraşüt bağlantı yerleri servo motorların üstlerini de fren bağlantı yuvaları yerleştirilmiştir. Servo motor seçiminde oluşabilecek azami tork göz önünde bulundurulmuştur. Bir koldaki toplam kuvvet azami bütün ağırlığın yarısı olduğunu düşünerek servo motor için gerekli tork belirlenebilir.



Şekil 5.1 Sistem şematik gösterimi

$m = 3\text{kg}$ ve $l = 10\text{cm}$ için 15 kg-cm tork yeterli olacaktır. PowerHD 20 kg-cm 'lik servo bu sistemde kullanılmıştır.

Otopilot kartı olarak sistemde Pixhawk bulunmaktadır. Ek1 de özellikleri yer almaktadır. Uçuş kontrol komutlarının ve görev çıktıların oluşturulması için ise Raspberry Pi4 görev bilgisayarı kullanılmıştır. Haberleşme modülü olarak 433MHz 3DR alıcı-verici kullanılmıştır. GPS sensörü olarak ise Ublox 6M sisteme entegre edilmiştir. Hava hızının ölçümü için MS4525DO hava hızı sensör kullanılmıştır. Yere iniş sırasında yüksekliğin net ölçünü fren kollarının çekilmesi için gereklidir. Bu nedenle sistemde ultrasonik mesafe sensörü de kullanılmaktadır. Güç ünitesi olarak 3S batarya, güç dağılım modülü ve batarya ölçüm sensörleri sistemdeki yerini almaktadır. Bataryanın kapasitesini belirlemek için güç bütçesi tablosu gösterilmiştir.

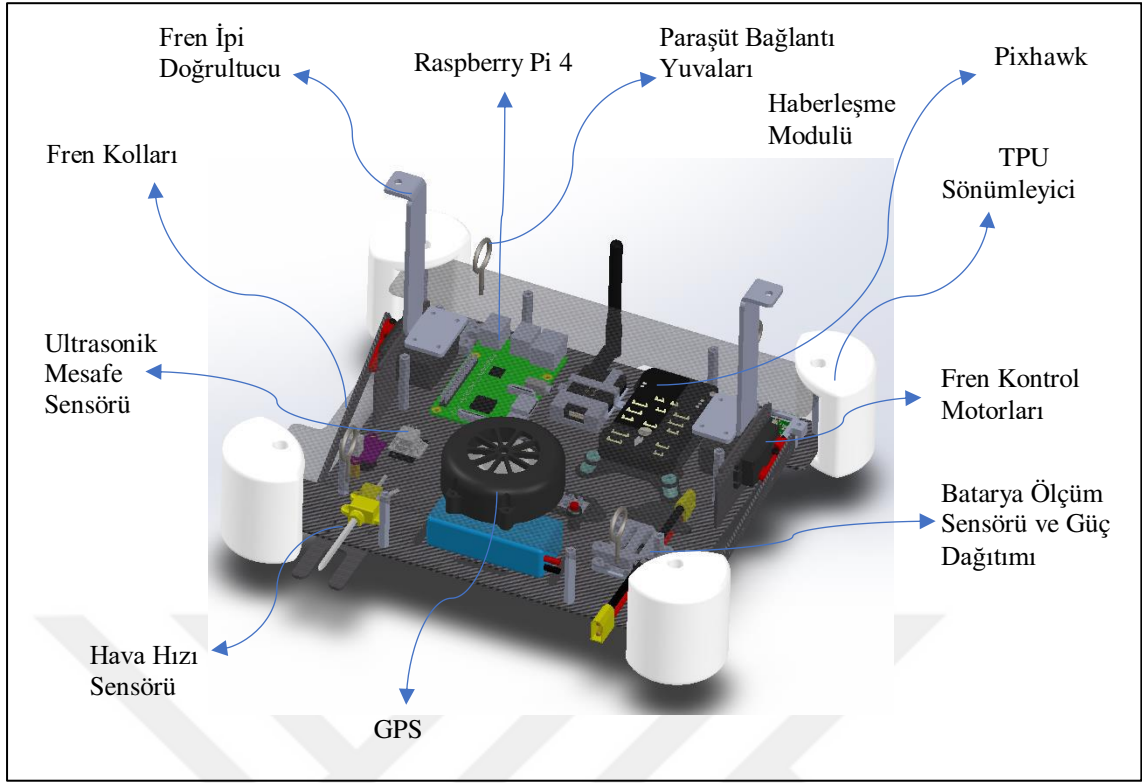
Tablo 5.1 Sistem güç bütçesi

No	Komponent	Azami Güç Tüketimi
1	Raspberry Pi4	5.0 W
2	Pixhawk	2.5 W
3	U-Blox GPS	0.015W
4	3DR Haberleşme Modülü	0.1W
5	Ultrasonik Sensör	0.025W
6	Hava Hızı Sensörü	0.001W
7	Servo Motorlar	23.8 W
Toplam		31.441 W

Azami güç tüketimi için 20dk lık bir görev süresi için gerekli 3s bataryanın kapasitesi;

$$Kapasite (mAh) = \frac{31.441\text{ W}}{11.1\text{ V}} * 1000 * \frac{20\text{ dk}}{60\text{ dk}} = 944.17\text{ mAh}$$

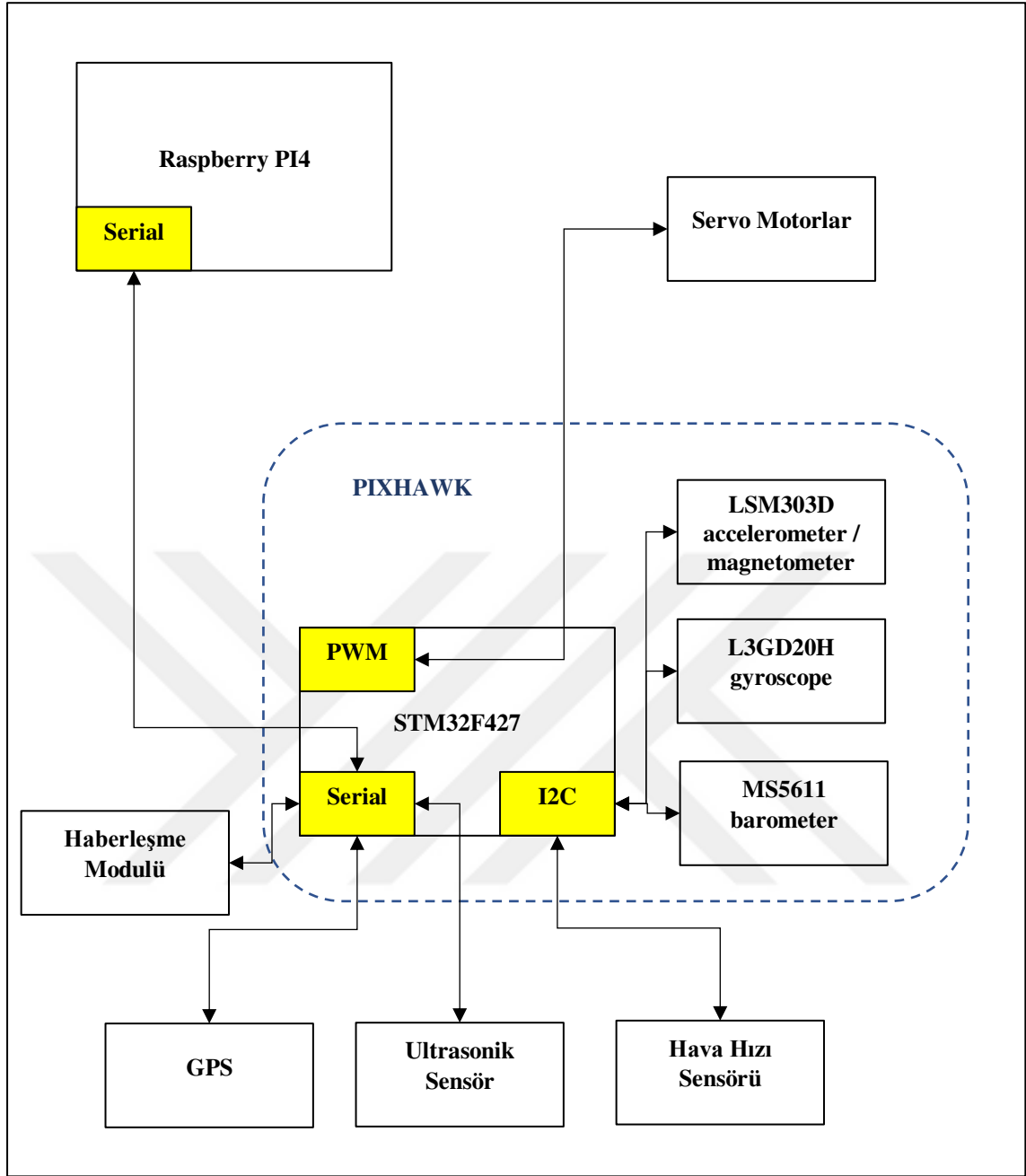
olarak bulunur. Leopard 3S 1000mAh batarya tercih edilmiştir.



Şekil 5.2 Sistem alt bileşenleri

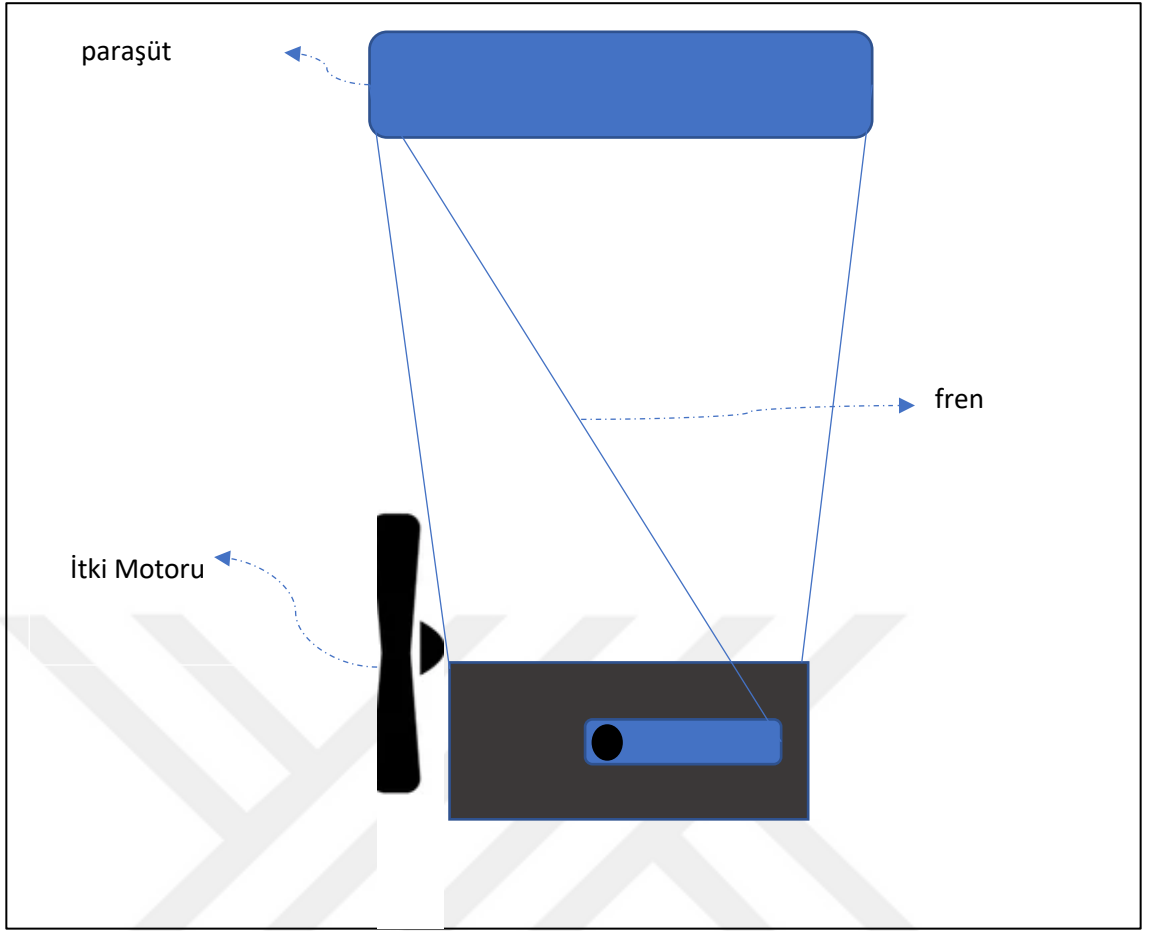


Şekil 5.3 Paraşütlü iniş sistemi



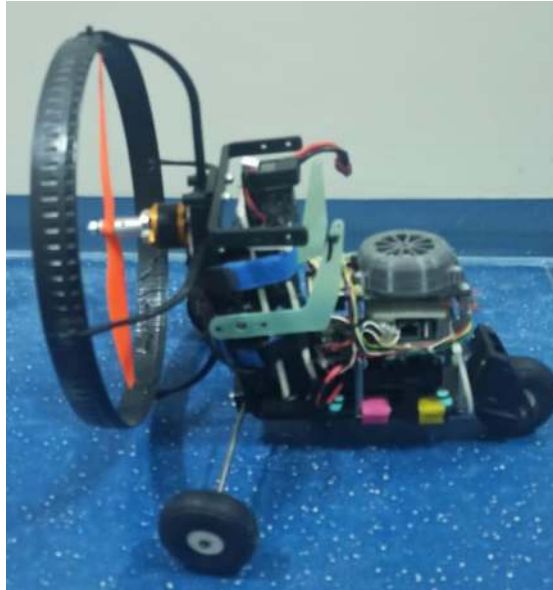
Şekil 5.4 Sistem blok şeması

Yapılan tasarımda testlerin gerçekleşmesi için döner kanatlı bir hava aracına ihtiyaç duyulmaktadır. Sistemin bu bağımlılıktan kurtulması için itki motoru eklenerek paramotor elde edilmiştir. Burada amaç sistemi istenilen yüksekliğe en güvenli şekilde ulaştırmak olacaktır. Hedeflenen yükseklikte itki motoru kapatılarak sistem serbest düşüşe geçmekte ve test tamamlanmaktadır. Bu ekleme göz önünde bulundurularak tasarım güncellenmiştir.



Şekil 5.5 Sistem şematik gösterimi (Revize model)

Sistemin son hali aşağıdaki şekilde gösterilmiştir.



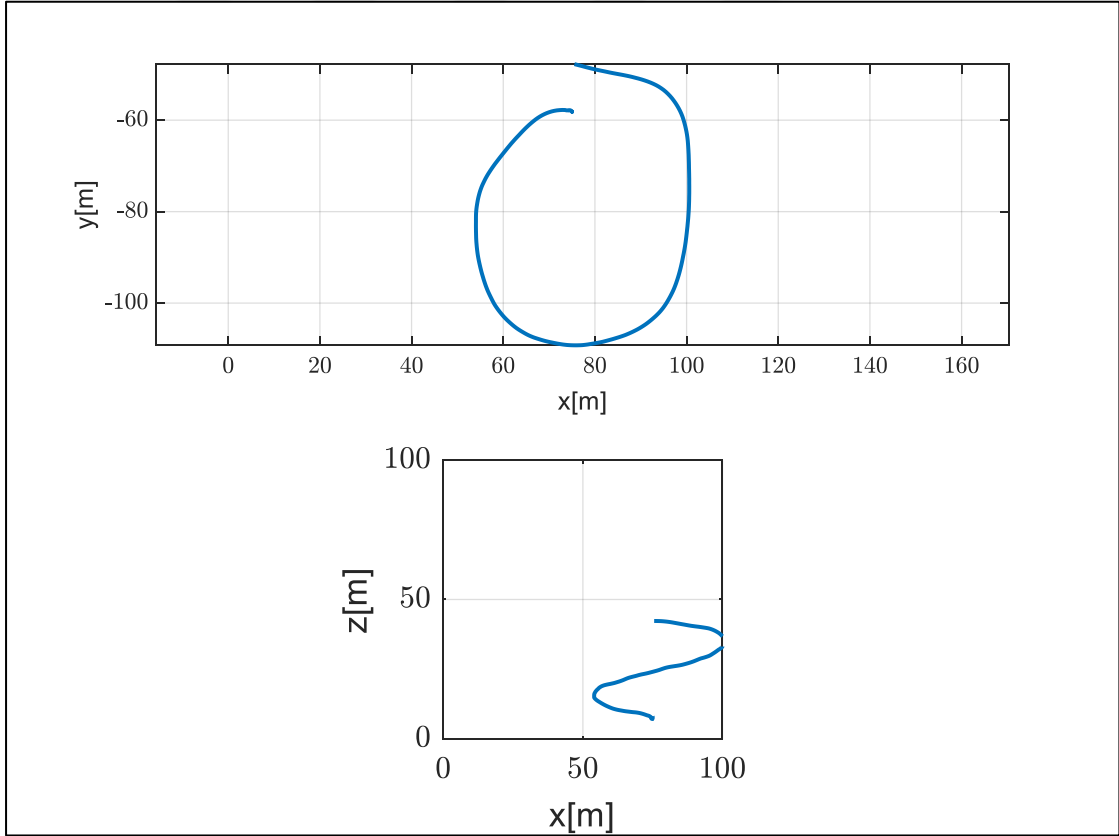
Şekil 5.6 İtki motorlu sistem gösterimi

Yapılan uçuş testinde sistem manuel olarak yükseltilmiş ve 50-60m irtifa aralığında itki motoru kapatılarak serbest düşüşe geçirilmiştir. Serbest düşüş sırasında kontrol girdileri uygulanmaya devam etmiştir. Testlerdeki sistem ve ortam parametreleri aşağıdaki tabloda verilmiştir.

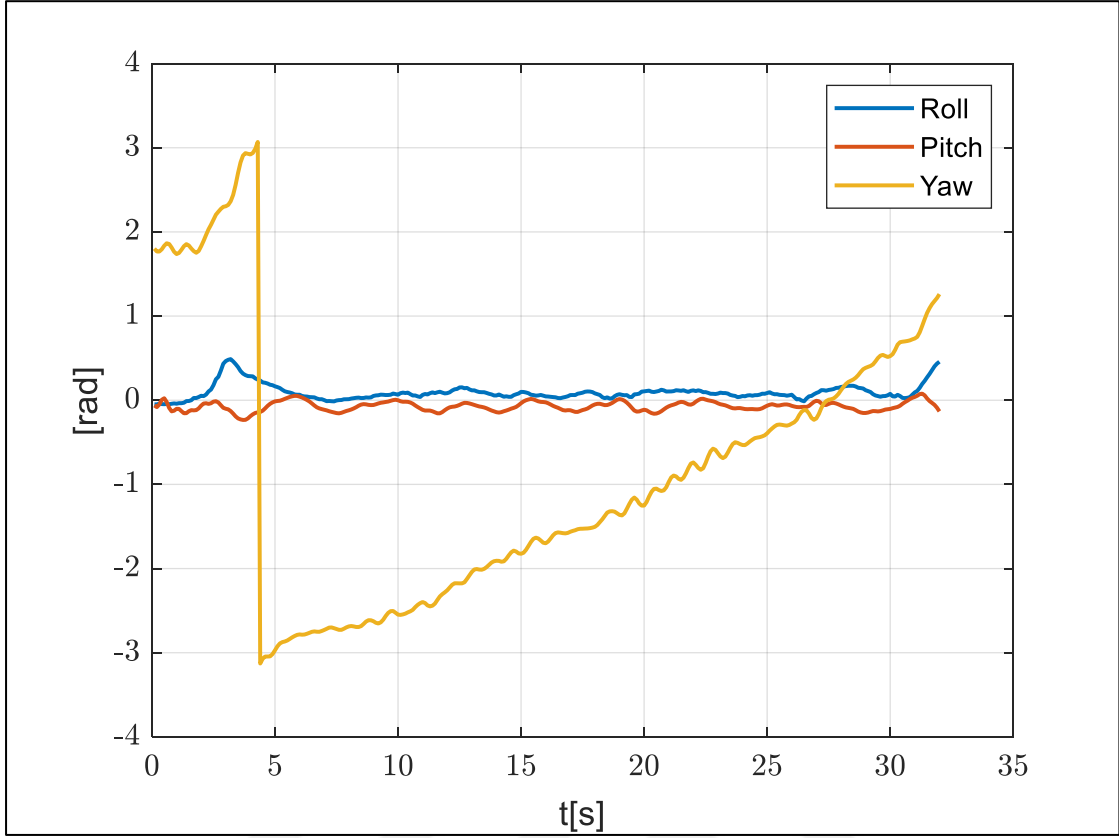
Tablo 5.2 Test parametreleri

	Test 1	Test 2
Rüzgar Durumu	13 km/sa kuzey batı yönü	8 km/sa doğu yönü
Sistem ağırlığı	1700 gr	1858 gr
Paraşüt Genişliği	2400 mm	2400 mm

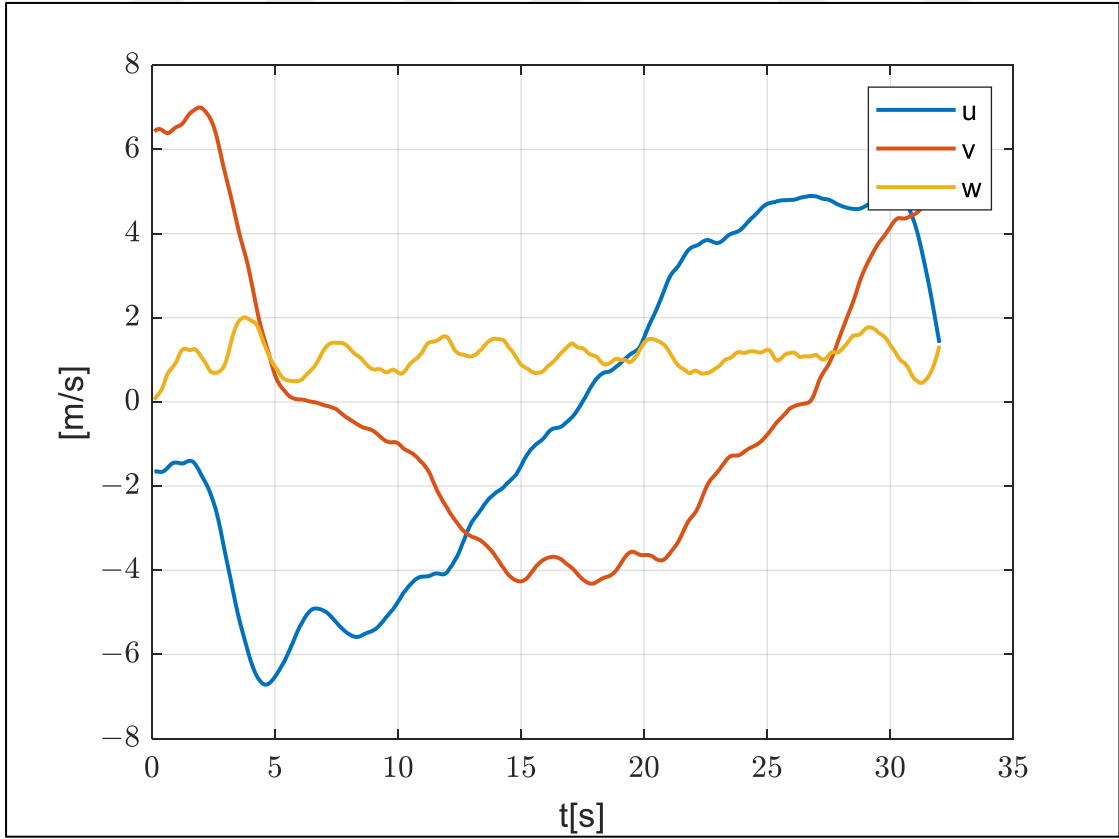
İlk test sırasında sadece sağ fren çekilerek iniş sağlanmış ve veri kaydedilmiştir. İkinci testte ise kontrol girdileri sağ ve sol fren olarak verilmiştir.



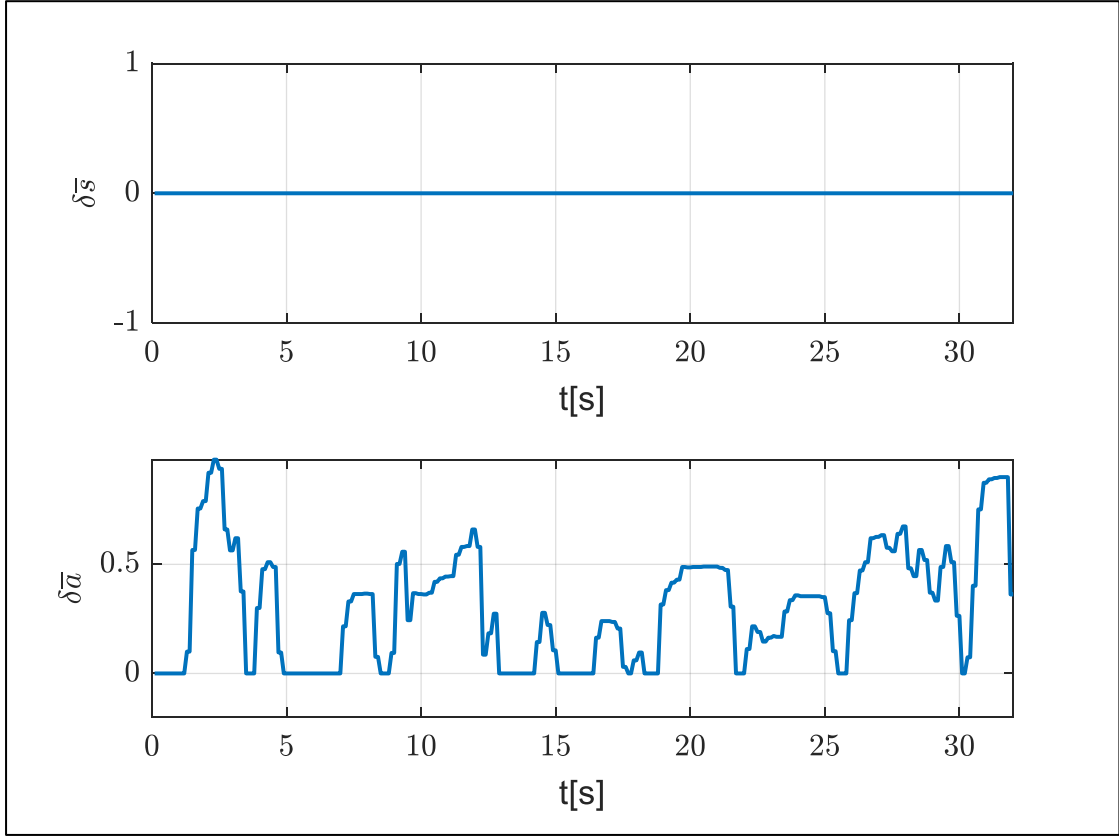
Şekil 5.7 Konum Grafikleri (Test 1)



Şekil 5.8 Euler Açıları (Test 1)

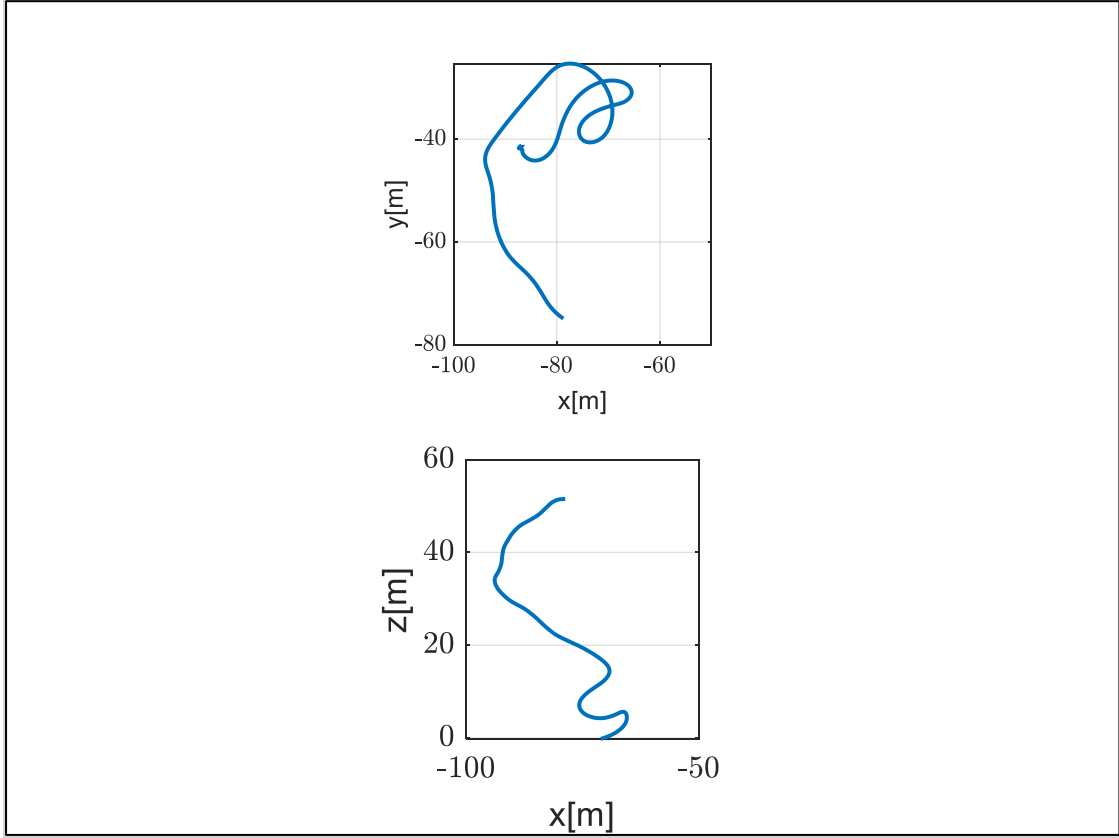


Şekil 5.9 Hız Grafiği (Test 1)

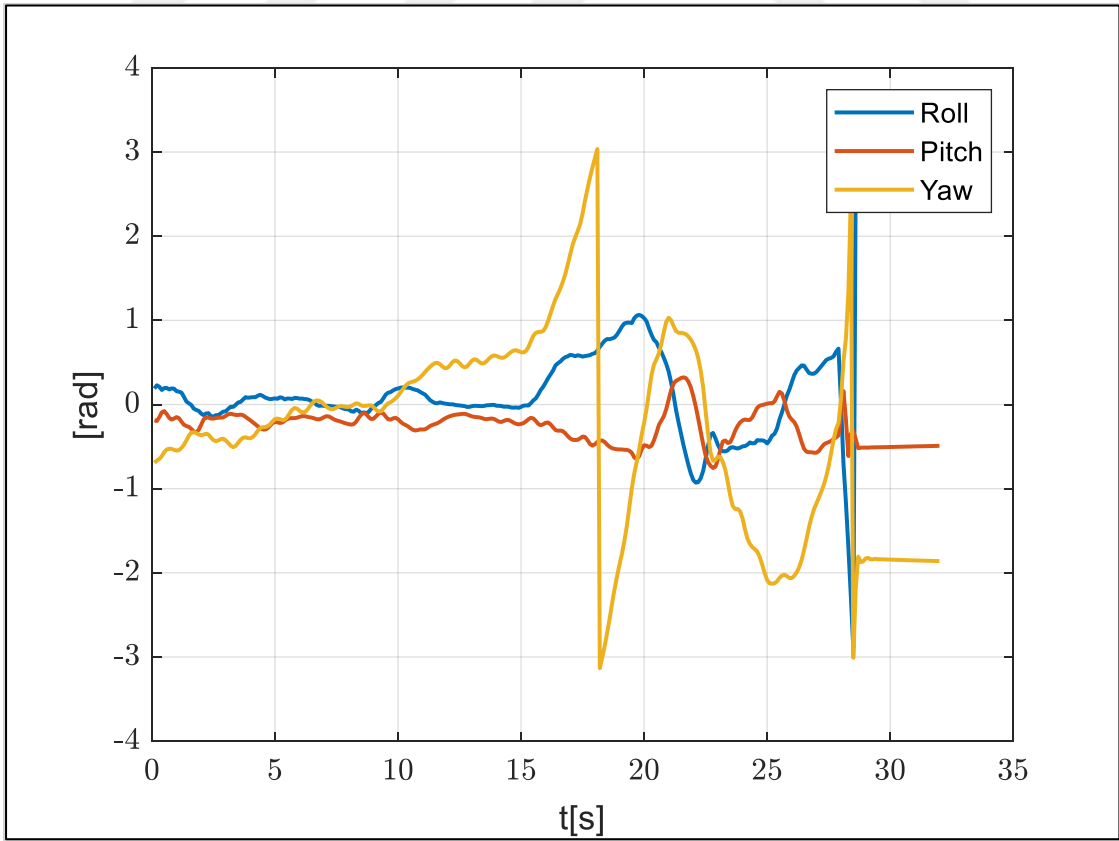


Şekil 5.10 Kontrol Girdileri (Test 1)

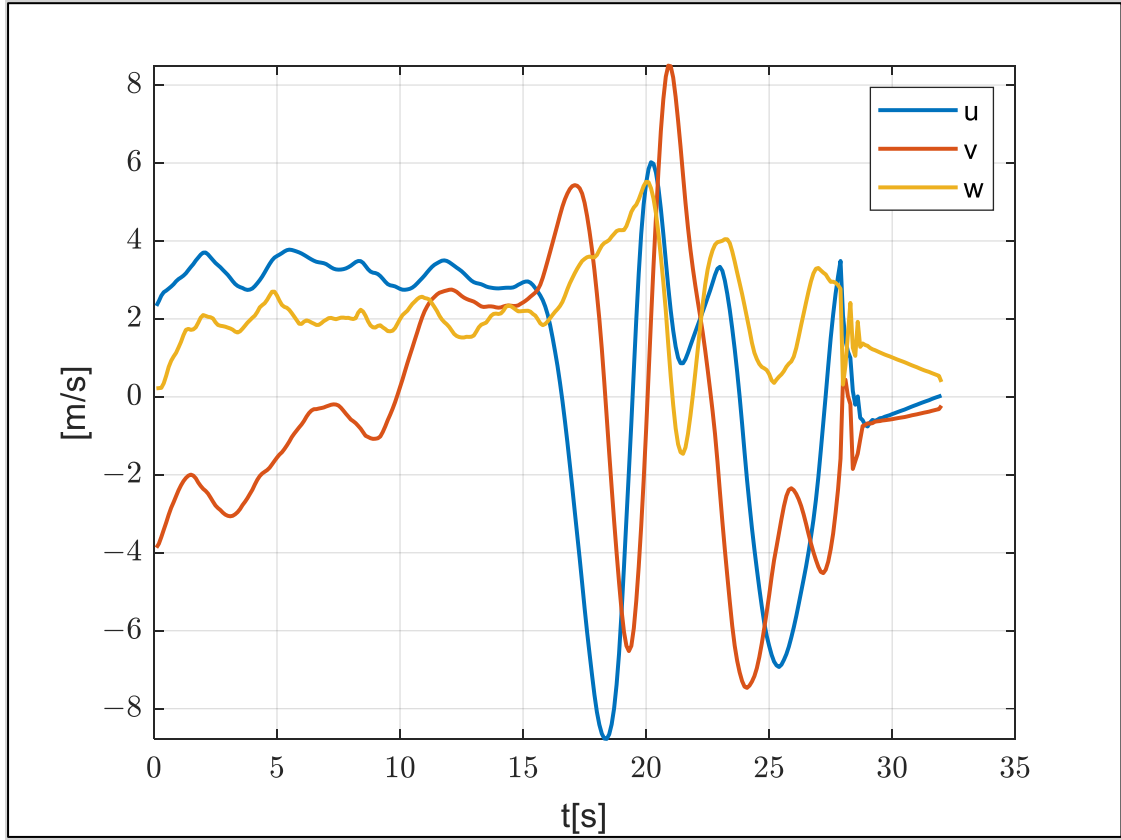
İkinci test te ise ilk olarak isteme ağırlık eklenmiştir. 160gr'lık kamera eklenerek sistem ağırlığı %10 arttırılmıştır. Kontrol girdileri ise sağ ve sol olmak üzere iki yönlü de uygulanmıştır. Simülasyon çalışmalarında kullanılan sistem ile gerçek sistemin ağırlık ve paraşüt boyut parametreleri aynı kullanılmıştır.



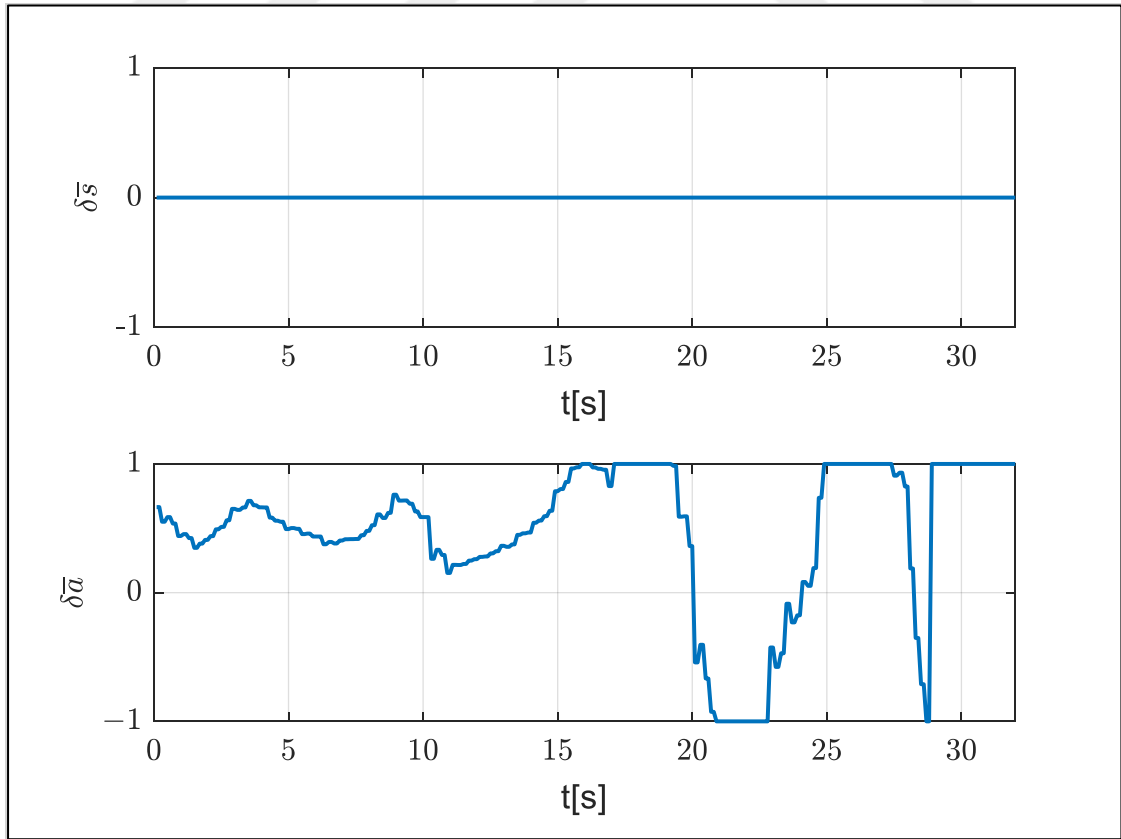
Şekil 5.11 Konum Grafikleri (Test 2)



Şekil 5.12 Euler Açıkları (Test 2)



Şekil 5.13 Hız Grafiği (Test 2)



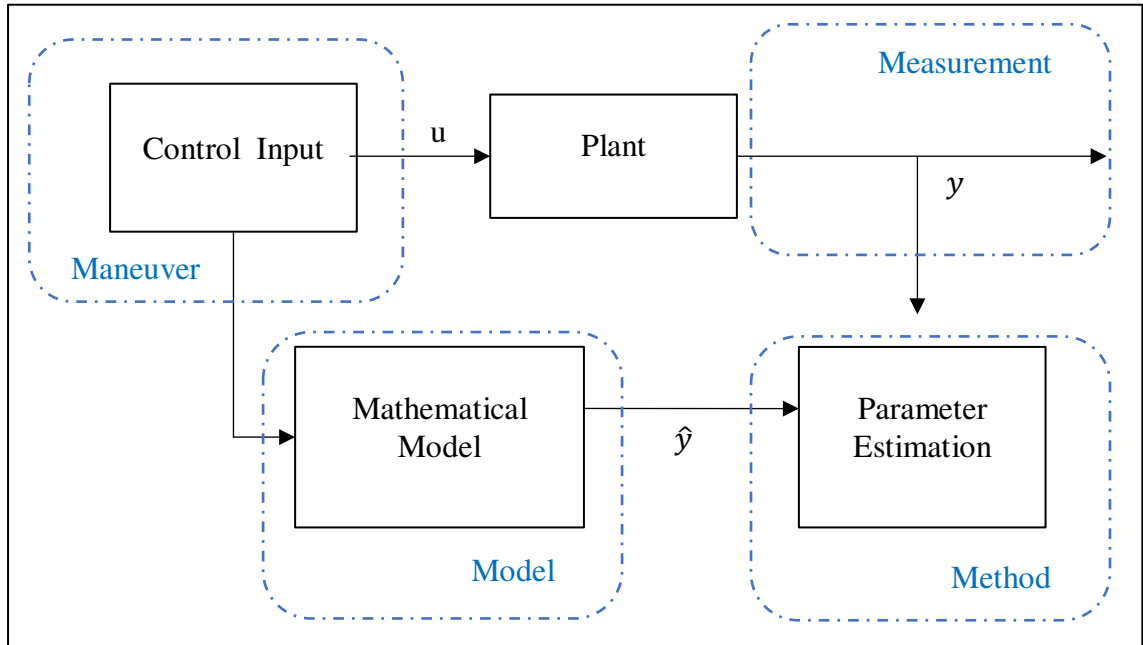
Şekil 5.14 Kontrol Girdileri (Test 2)

6. SİSTEM TANIMLAMA

Hava araçlarında sistem tanımlama aerodinamik parametrelerin tahmini ya da sistemin matematiksel modelinin tanımlanması olarak düşünülebilir. Paraşütlü iniş sistemlerinin aerodinamik parametrelerinin tahmini için literatürde 3 adet metot önerilmiştir [26]. Bunlardan ilki sayısal akışkanlar dinamiğine dayalı analitik yöntemleri kapsamaktadır. Diğerleri ise rüzgâr tüneli testleri ve uçuş testleridir. Bu çalışmada uçuş testlerinde kullanılan yöntemlere odaklanılmıştır.

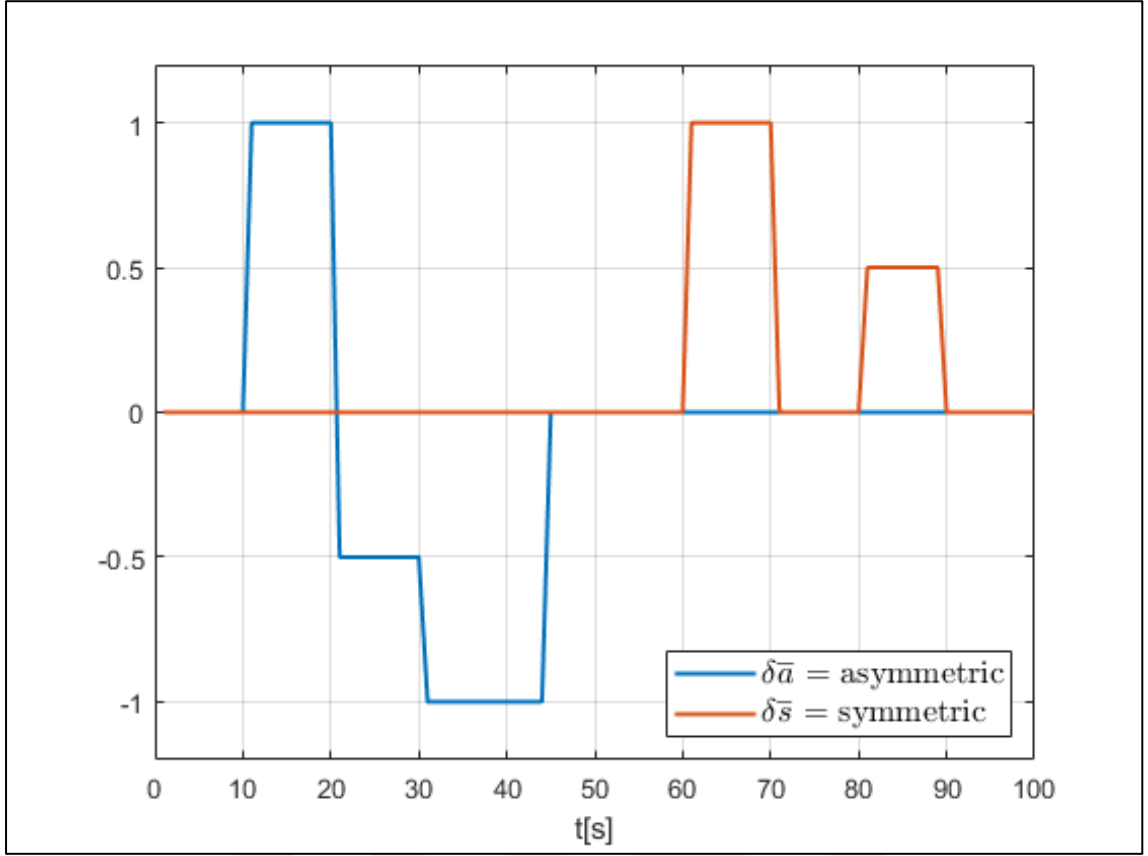
Sistem tanımlamada amaç uçuş testlerinden elde edilen giriş ve çıkışlara göre matematiksel modeli elde etmektir. Hamel ve Jategaonkar, başarılı bir sistem tanımlama için 4M (Manevra, Ölçüm, Metot, Model) gereksinimlerini önermişlerdir [33].

- Kontrol girdileri ekstrem noktaları kapsayacak şekilde oluşturulmalı
- Yüksek çözünürlüklü ölçümler kullanılmalı
- Aracın olası matematiksel modeli tanımlanmalı
- Veriye en uygun yöntem seçilmeli



Şekil 6.1 4M Tabanlı Sistem Tanımlama

Jann ve Strickert, tanımlamada kullanılacak veri oluşumunda yapılması gereken manevralarda simetrik ve asimetric manevraların ayrılmasını önermişlerdir [34].

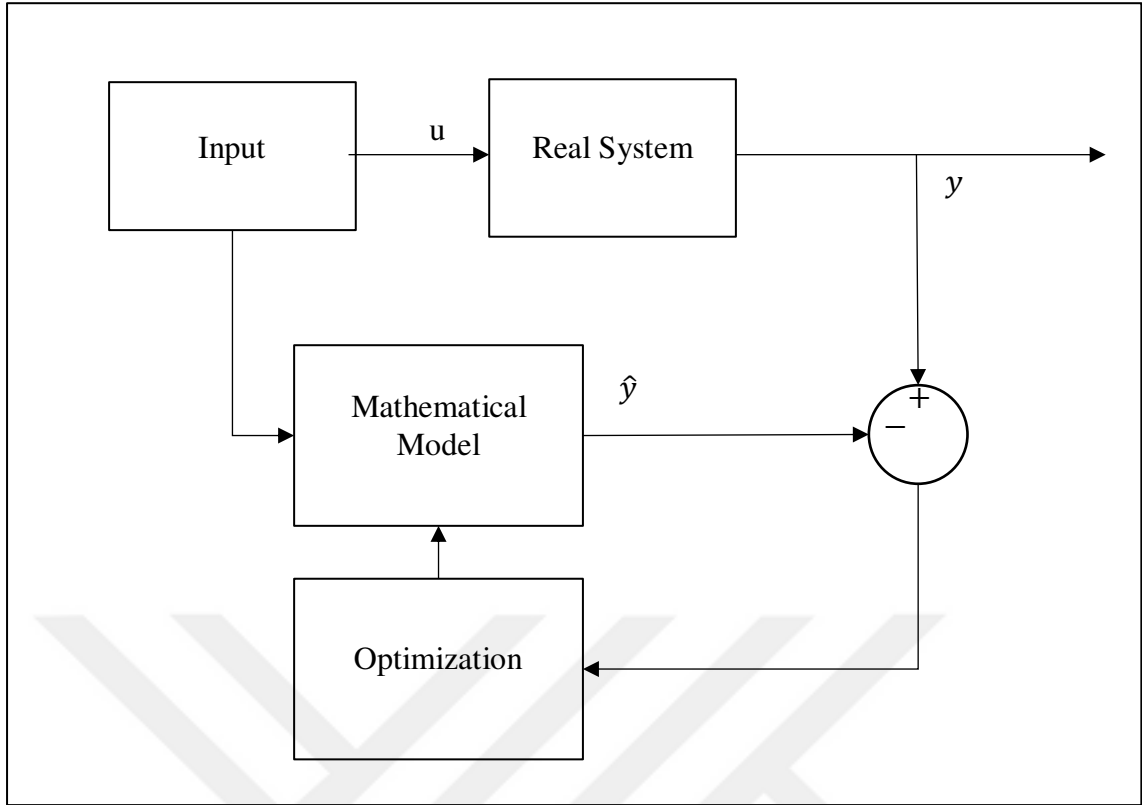


Şekil 6.2 Jann tarafından Önerilen Kontrol Girdisi

Parametre tahmininde kullanılan yöntemler üç başlıkta incelenmiştir. Bunlar

- Eşitlik-hata yöntemi (equation-error)
- Çıktı-hata yöntemi (output-error)
- Filtre-hata yöntemi (filter-error)

olarak sıralanmıştır. Hangi yöntemin seçileceği ise ölçüm ve süreçteki gürültülere göre tercih edilebilir [33]. Eğer her ikisinde de bozucu etmenler göz ardı edilebiliyorsa en hızlı yöntem olan eşitlik-hata yöntemi tercih edilir. Bozucu etmenler sadece ölçümlerde varsayılırsa çıktı-hata yöntemi, her ikisinde de varsa filtre-hata yöntemi önerilmektedir.



Şekil 6.3 Çıktı-Hata Yöntemi

Literatürde parametre tahmini için çıktı-hata yöntemi en çok tercih edilen yöntemdir. Grauer yaptığı çalışmada genellikle uçuş sonrası veri kullanılarak yapılan çıktı-hata yöntemini gerçek zamanlı uçuş verisine uyarlayarak uçağın dinamik modelini uçuş sırasında hesaplamıştır [35]. Çıktı-hata yöntemini kullanan başka bir çalışmada Jann, ALEX adı verilen paraşütlü iniş sisteminin durum değişkenlerini sensör girişleri (GPS, Manyetometreler, Gyros, İvmeölçerler) aracılığıyla tahmin etmiştir [22]. Jaiswal, Prakash ve Chaturvedi ise paraşütlü iniş sisteminin aerodinamik katsayılarını Maksimum Likelihood yöntemi ile çıktı-hata yöntemini kullanarak tahmin etmişlerdir.

İstatiksel yöntemlerin yanında gün geçtikçe popülaritesi arttan Makine Öğrenmesi teknikleri sistem tanımlama probleminin çözümünde de başarılı olmuştur. Literatürde, yapay sinir ağları hava araçlarının dinamiklerinin modellenmesinde [36] [37] [38] [39] [40], aerodinamik kuvvet ve momentlerin tahmininde [41] [42] [43] [44] ve kontrolcü tasarımlarında [45] [46] kullanılmışlardır. Bu çalışmalarda hem İleri Beslemeli Sinir Ağları [43] [47] hem de Tekrarlayan Sinir Ağları yaygın olarak kullanılmıştır [48]. Roudbari ve Saghafi, yüksek manevra kabiliyetli hava araçlarının dinamiklerinin tanımlanması için yeni bir yöntem önermişlerdir. Geliştirdikleri modelde uçuş dinamiklerini Yapay Sinir Ağları ile modellemişlerdir. Geleneksel yöntemlerden farkı ise eğitim sırasında aerodinamik bilgileri

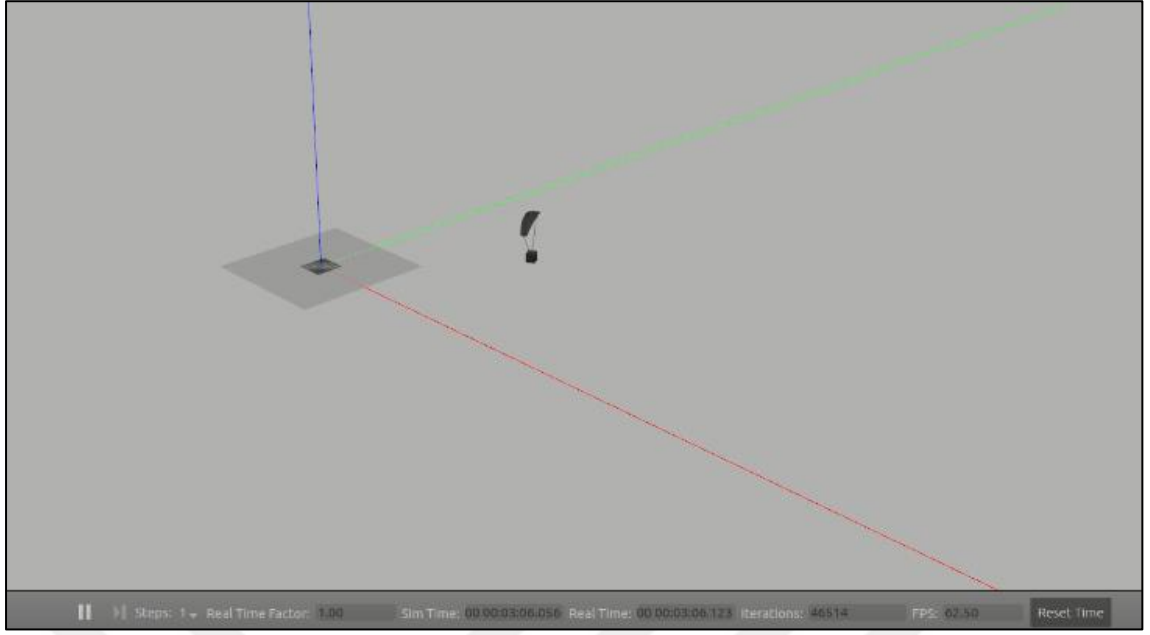
kullanmamalarıdır [49]. Bagherzadeh yaptığı çalışmada, geliştirdiği yapay sinir ağları modelinin performansını arttırmak için uçuş dinamikleriyle modeli desteklemiştir [50].

Derin öğrenme yöntemlerinin gelişmesi, bu yöntemlerin sistem tanımlama problemlerinde sıklıkla kullanılmasını sağlamıştır. Derin Sinir Ağlarının mimarilerinden biri olan Kalıntı Sinir Ağı, bu problemde kullanılan yöntemlerden biridir. Goyal ve Benner dinamik sistemler için LQRes-NET [51] adlı özel bir mimari geliştirdiler. Önerdikleri yöntem, dinamik sistemlerin modellenmesinde kullanılan gözlemin kullanılmasına izin verir. Modelleri, bir değişkenin hızının, değişkenin doğrusal ve ikinci dereceden biçimlerine bağlı olduğu ilkesine dayanmaktadır. Chen ve Xiu, gResNet adlı çerçeveyi önerdi. Kalıntıyı, Önceki modelin tahmin hatası olarak tanımladılar. Ayrıca kalıntıyı modellemek için DNN'yi kullandılar [52] .

Bu çalışmada, bilinmeyen Havadan Sevkiyat Sisteminin Ram-Hava Paraşütü ile tespit edilmesi için seri-paralel yapıya sahip NARX Ağı kullanılmıştır. Veri kümesi, döngü içinde yazılım yöntemi (Döngüdeki Yazılım) kullanılarak oluşturulmuştur. Simülatör olarak Gazebo, otopilot yazılımı olarak PX4 kullanılmıştır. NARX Ağının performansı seçilen eğitim algoritması, giriş ve çıkış gecikmeleri, gizli katman ve nöron sayısı gibi parametrelere göre farklılık göstermektedir. Bu çalışma kapsamında her bir parametre bağımsız olarak incelenmiştir. Modeller MATLAB 2020a kullanılarak eğitilmiştir.

6.1. Veri Kümesi Oluşturma

Veri kümesi döngüde donanım metodu (Software in the Loop) kullanılarak oluşturulmuştur. Simülatör olarak Gazebo, otopilot yazılımı olarak ise PX4 kullanılmıştır. Gazebo ortamında sanal bir uçuş gerçekleştirilmiştir (Şekil 6.4).



Şekil 6.4 Gazebo Simulasyon Ortamı

Paraşütlü İniş sisteminin fiziksel ve aerodinamik katsayıları olarak Snowflakes isim aracın parametreleri kullanılmıştır [22]. Aracın parametreleri Tablo 6.1’de verilmiştir.

Tablo 6.1 Snowflakes Parametreleri [22]

Mass (m)	1.90 kg
Canopy Reference Area (S)	1.0 m ²
Inertia Matrix (I)	$\begin{bmatrix} 0.0420 & 0.000 & 0.0068 \\ 0.0000 & 0.027 & 0.0000 \\ 0.0068 & 0.000 & 0.0540 \end{bmatrix}$
Maximum brake deflection (δ_{smax})	0.25 m
Aerodynamic coefficients	$\begin{bmatrix} C_{D0} = 0.150 & C_{D\alpha^2} = 0.900 \\ C_{Y\beta} = -0.050 & C_{L0} = 0.250 \\ C_{L\alpha} = 0.680 & C_{m0} = 0.00 \\ C_{m\alpha} = 0.000 & C_{mq} = -0.2650 \\ C_{l\beta} = -0.0360 & C_{lp} = -0.3550 \\ C_{lr} = 0.000 & C_{l\delta_a} = 0.1500 \\ C_{n\beta} = -0.0360 & C_{np} = 0.000 \\ C_{D0} = -0.090 & C_{n\delta_a} = 0.0030 \end{bmatrix}$

Simulasyon ortamında aracın uçuş verisi için Gazebo uyumlu sensör modelleri kullanılmıştır. Bunlar jiroskop, manyetometre, ivmeölçer, barometre ve GPS’ ten oluşmaktadır. Araca ait durum değişkenlerinin tahmini PX4 yazılımı tarafından

Geniřletilmiş Kalman Süzgeci(Extended Kalman Filter) yapılmaktadır [53]. Manyetometre oryantasyon deęerlerinin, barometre ise yükseklik deęerinin iyileřtirilmesinde kullanılmıřtır. Kullanılan sensörlerin basitleřtirilmiş modelleri benzer řekilde gösterilebilir.

$$x_m = x + b + n \quad (6.1)$$

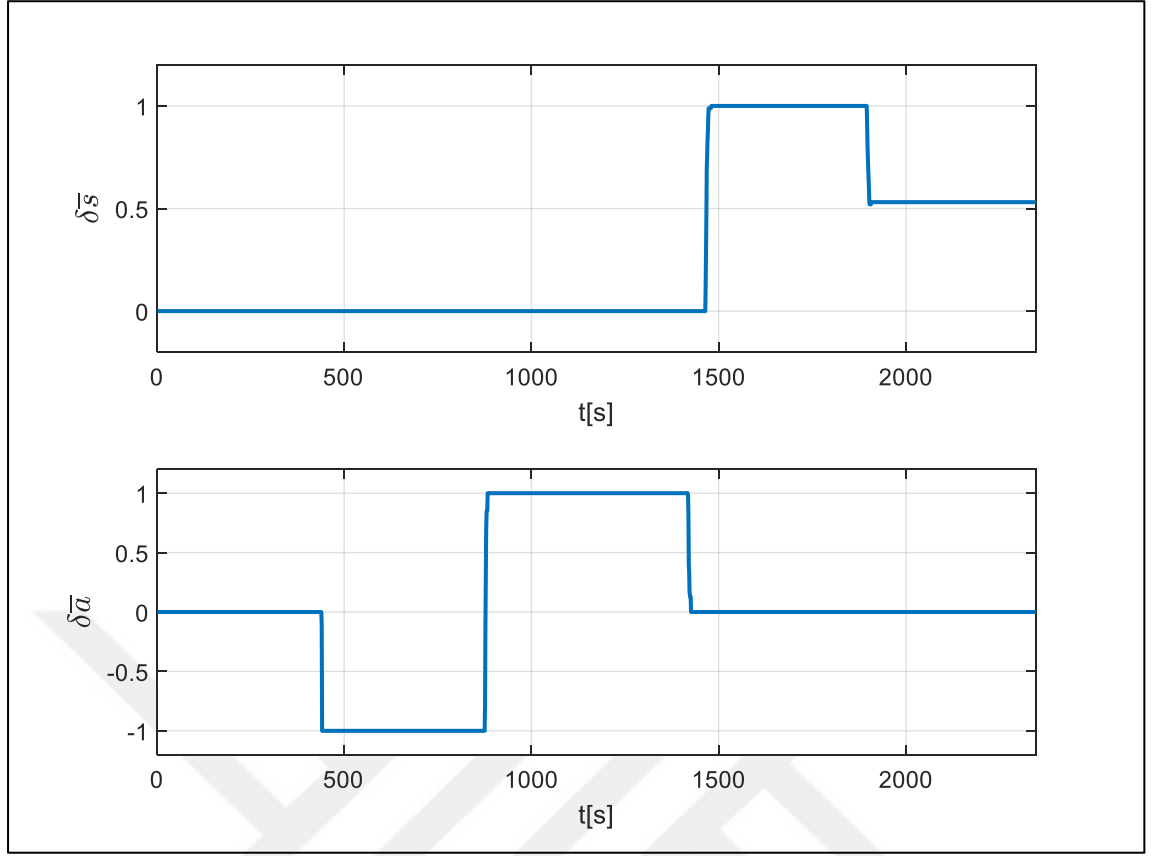
$$\dot{b} = n_b \quad (6.2)$$

x_m ölçülen deęeri, x gerçek deęeri, b bias, n ve n_b gauss gürültüsü olarak ifade edilir. Sensör parametreleri de bu notasyonda ifade edilebilir. Simulasyonda kullanılan sensör parametreleri Tablo 6.2 de verilmiřtir.

Tablo 6.2 Sensör Parametreleri

Sensörler	Gürültü Yoęunluęu (σ_n)	Rastgele Deęiřme (σ_{n_b})	Bias korelasyon süresi (σ_n)	Açılma Bias Sapması (<i>ilk deęer</i> σ_b)
Jiroskop	0.00019	0.000	1000.00	0.00870
İvmeölçer	0.00186	0.006	300.00	0.1960
Manyetometre	0.00040	0.000	600.00	0.0000

Simulasyon rüzgarsız ortamda yapılmıř olup hava yoęunluęu 1.225 kg/m^3 olarak kabul edilmiřtir. Simulasyonda PKİS 500m yükseklikten serbest bırakılmıřtır. 30sn de bırakılma gerçekleřmiřtir. Kontrol girdileri $\bar{\delta}_a$ ve $\bar{\delta}_s$, tam saę ve tam sol olacak řekilde verilmiřtir. (řekil 6.5)



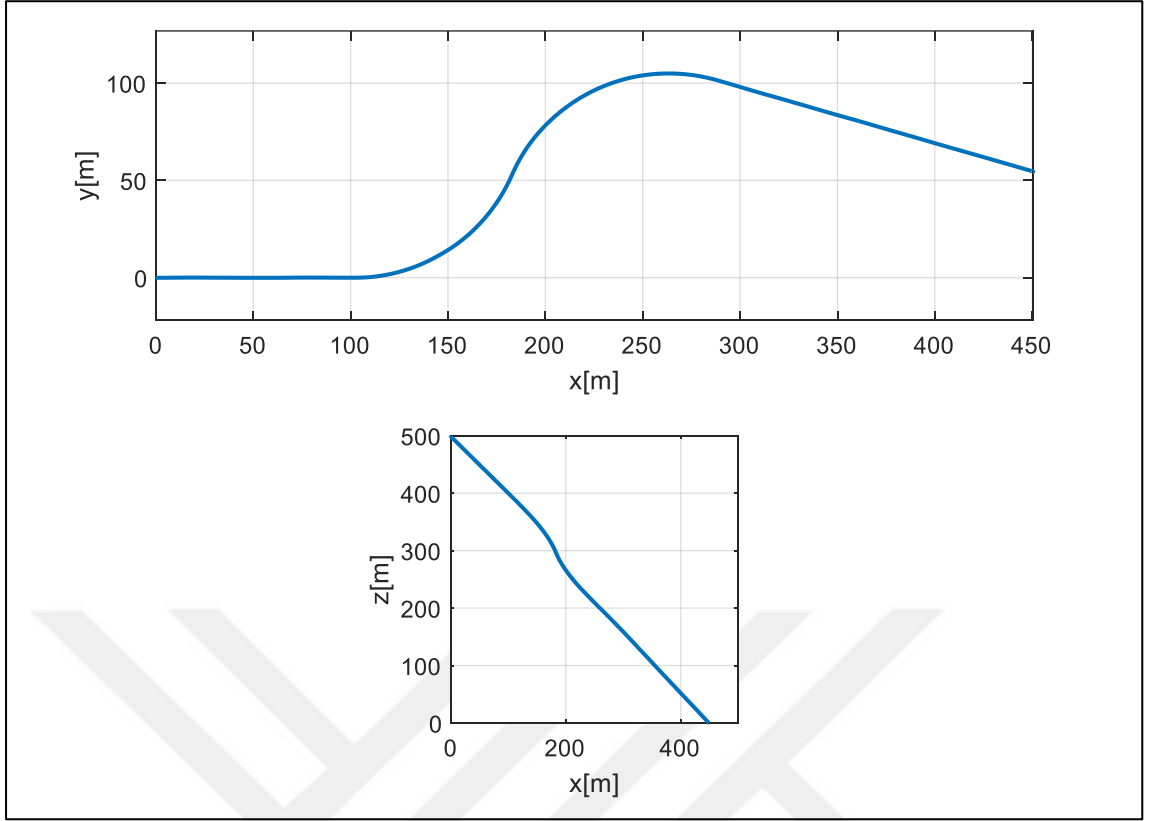
Şekil 6.5 Kontrol girdileri

Sistemden alınan uçuş verisi makine öğrenmesi tekniklerine uygun olacak düzenlenmiştir. Bunun için giriş vektörü x ve y çıkış vektörü oluşturulmuştur.

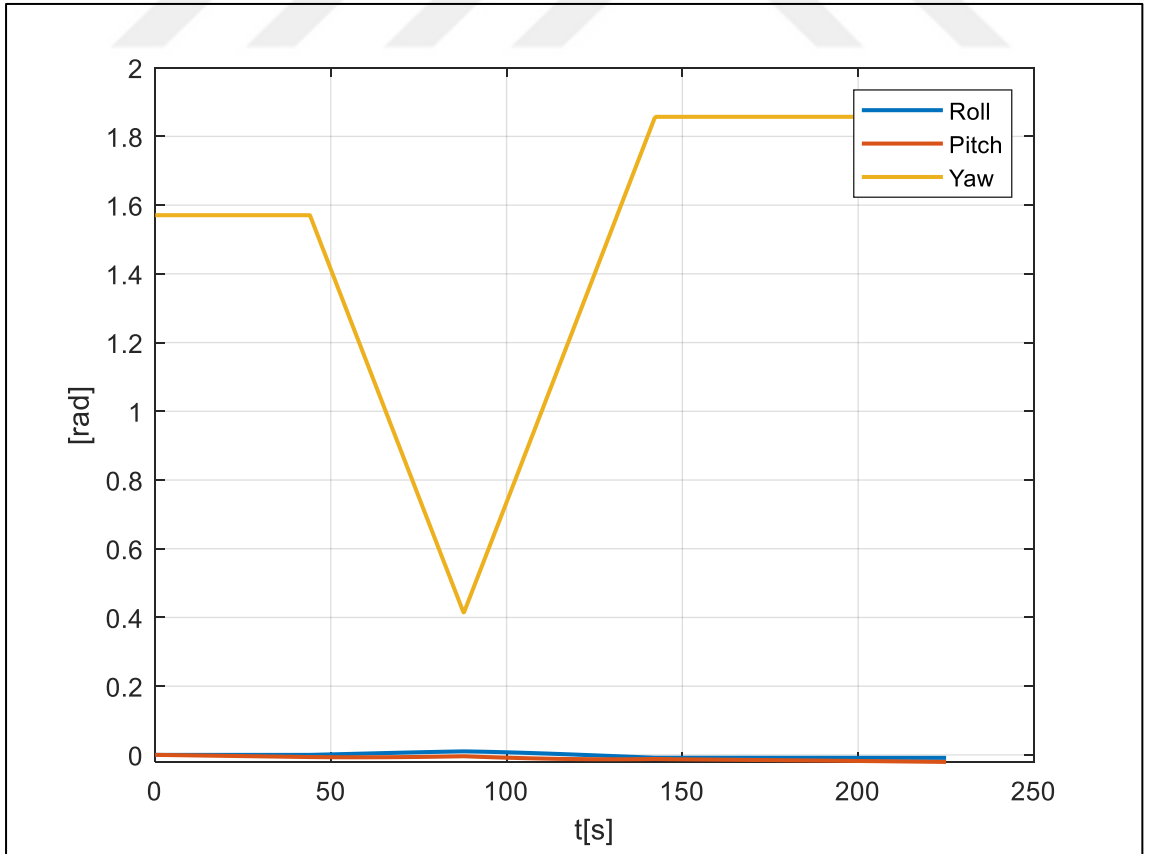
$$x = [\partial \bar{a} \partial \bar{s}] \quad (6.3)$$

$$y = [u \ v \ w \ p \ q \ r] \quad (6.4)$$

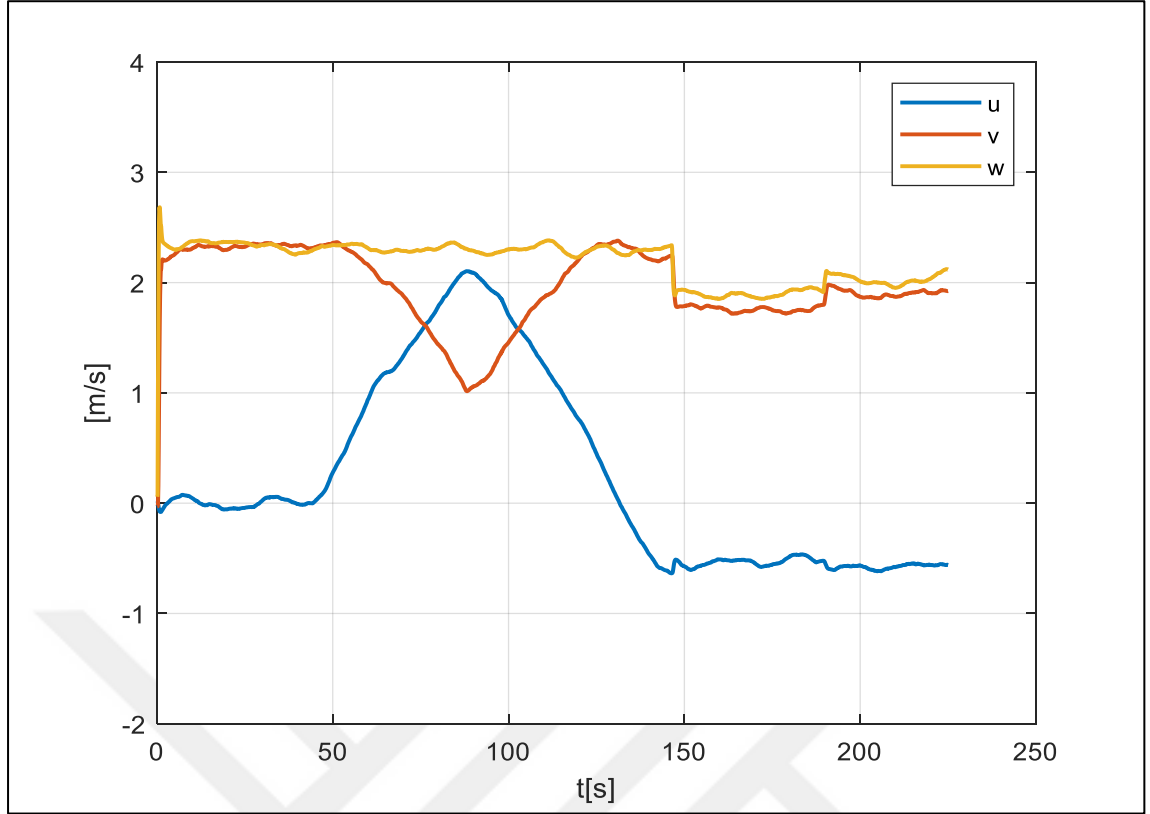
Toplam 270sn lik veri uçuş bölümünü kapsayacak şekilde 225snye indirgenmiş ve 10Hz lik ölçüm kullanılarak 2250 adet veri üretilmiştir. Kullanılan uçuş verisinde aracın konum ve hız bilgileri Şekil 6.6, Şekil 6.7 ve Şekil 6.8’de gösterilmiştir.



Şekil 6.6 Konum Grafikleri



Şekil 6.7 Euler Açıkları



Şekil 6.8 Hız Grafiği

Model performanslarını gözlemlenmesi için uçuşun %70 lik bölümü eğitimde kalan %30 ise test sürecinde kullanılacaktır. İniş sisteminin yere düştüğü konum önemli olduğundan bırakılma anından itibaren ilk bölüm eğitim son bölüm ise test verisi olarak kullanılmıştır.

6.2. NARX Sinir Ağı

NARX (nonlinear autoregressive exogenous), zaman serisi modellerinde kullanılan doğrusal olmayan bir model gösterimidir. Bu gösterimde model çıktıları, geçmiş çıktı değerlerine, girdilere ve girdilerin geçmiş değerlerine bağlı olmaktadır. Matematiksel olarak ifadesi 20 nolu eşitlikte verilmiştir.

$$y(t) = f[y(t-1), y(t-2), \dots, y(t-n_y); u(t), u(t-1), \dots, u(t-n_u)] \quad (6.5)$$

Burada y çıktıları, u girdileri göstermektedir. f ise doğrusal olmayan bir fonksiyonu temsil etmektedir. f 'in sinir ağı olarak modellendiği yapı, NARX Sinir Ağı (NARX Network) olarak isimlendirilmiştir [26]. Bu model, geleneksel sabit kanatlı[27-28] ve döner

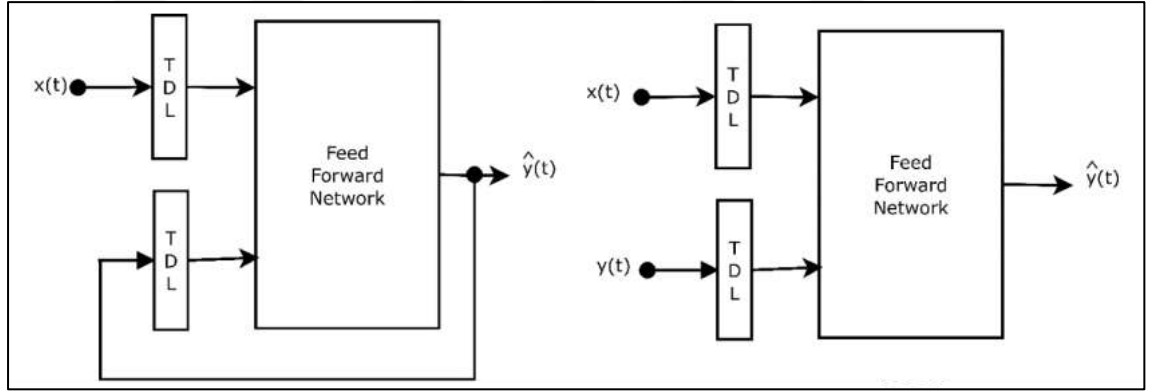
kanatlı[29-30] hava araçlarının modellenmesinde kullanılmıştır. NARX Sinir ağı, Paralel ve Seri-Paralel olmak üzere iki tipte modellenebilir (Şekil 1.8). Paralel yapıdaki modelde, tahmin edilen çıktı değerleri sisteme geri beslenmektedir (Denklem 21).

$$\hat{y}(t) = f[\hat{y}(t-1), \hat{y}(t-2), \dots, \hat{y}(t-n_y); u(t), u(t-1), \dots, u(t-n_u)] \quad (6.6)$$

Seri-paralel yapıdaki modelde ise sadece gerçek sistem çıktıları kullanılmaktadır.

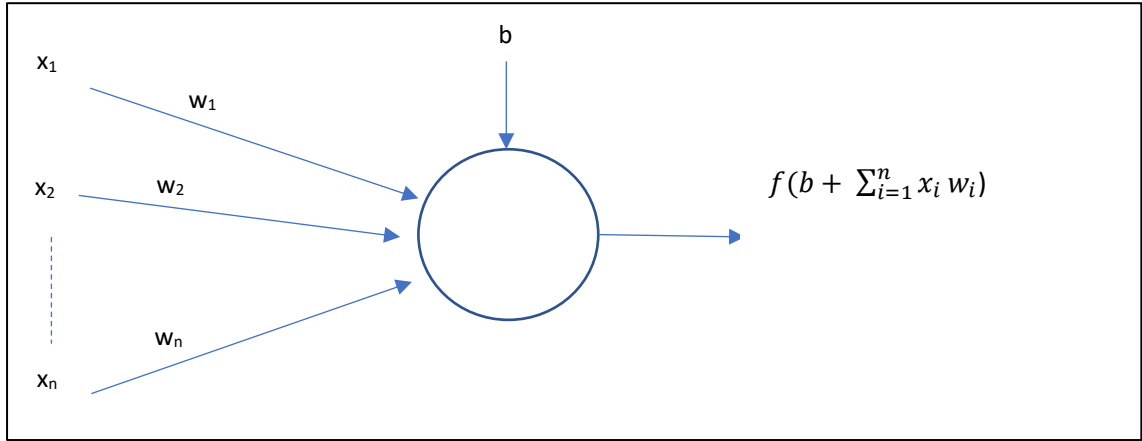
$$\hat{y}(t) = f[y(t-1), y(t-2), \dots, y(t-n_y); u(t), u(t-1), \dots, u(t-n_u)] \quad (6.7)$$

Bu denklemlerde $\hat{y}(t)$, t anındaki tahmin edilen çıktı değerini göstermektedir.

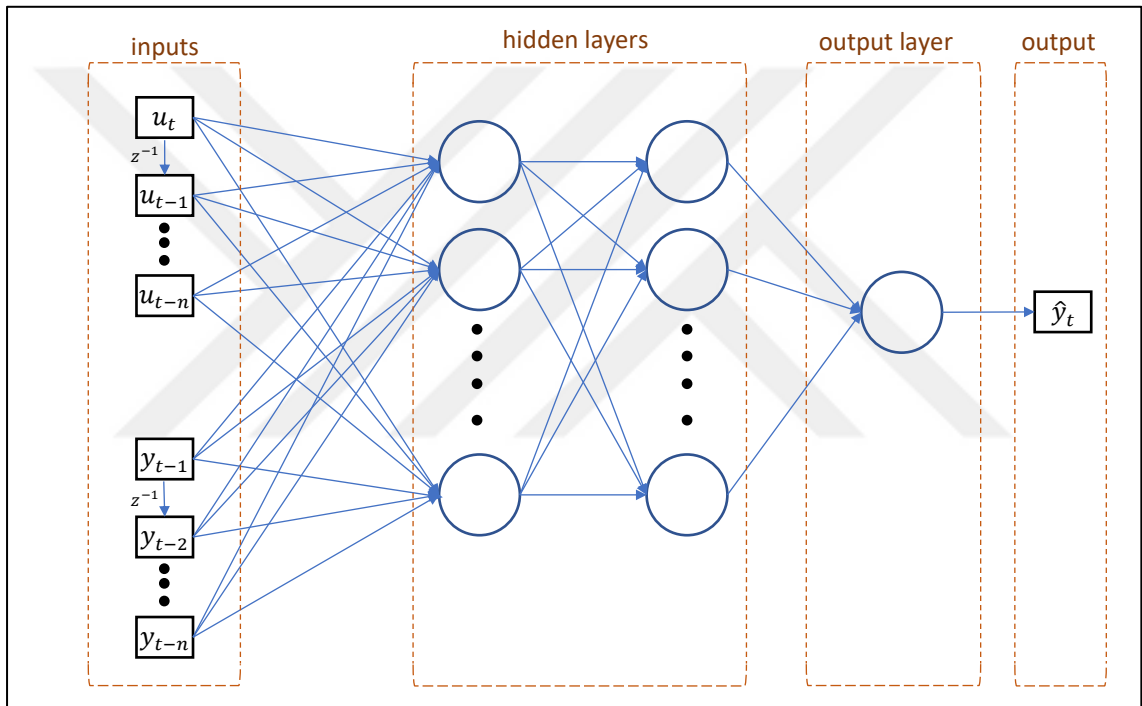


Şekil 6.9 Paralel(Sol) ve Seri-Paralel(Sağ) NARX Sinir Ağı

Bu çalışmada kullanılan veriseti gerçek sistem çıktılarını da bulunduğu için Seri-Paralel model yapısı tercih edilmiştir. Şekil 6.9 'da gösterilen ileri beslemeli ağ(Feed Forward Network) bloğu Çok Katmanlı İleri Beslemeli Sinir Ağları'ndan (Multilayer Feedforward Neural Networks) oluşmaktadır. En az bir gizli katmandan oluşan bu bölümde katmanlar arasındaki geçiş nöronlar vasıtasıyla yapılmaktadır. Her bir nöron, girdileri ve ağırlıklarını kullanarak belirlenen aktivasyon fonksiyonu yardımıyla çıktıları hesaplar (Şekil 6.10). Burada x_n girdileri, w_n ağırlıkları, b bias ve f aktivasyon fonksiyonunu göstermektedir. Seri-Paralel yapıdaki NARX Sinir ağının mimarisi Şekil 6.11 'de gösterilmiştir.



Şekil 6.10 Nöron yapısı



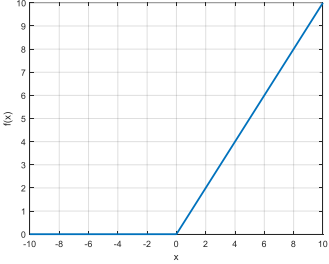
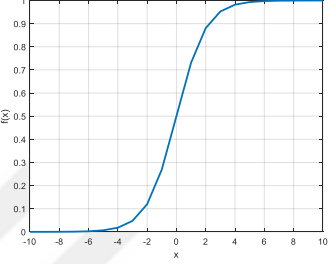
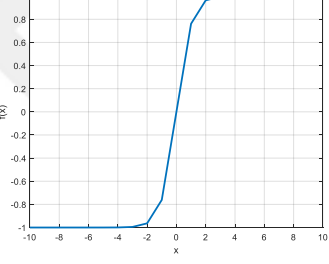
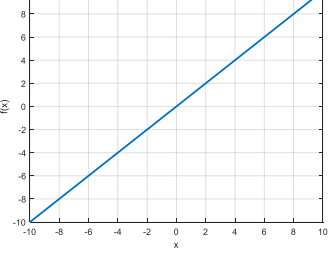
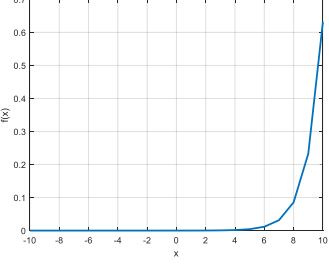
Şekil 6.11 Seri - Paralel Narx Sinir Ağı Mimarisi

Aktivasyon fonksiyonlarının tercihi model tasarımlarında önemli bir rol oynamaktadır. Gizli katmanlarda kullanılan fonksiyonlar ile çıkış katmanında kullanılan fonksiyonlar değişkenlik göstermektedir. Gizli katmanlarda türevlenebilir fonksiyonlar tercih edilmektedir. Eğitim sırasında doğrusal fonksiyonlara karşın tercih edilen bu fonksiyonlar modellerin daha karmaşık problemlere karşı başarılı olmasını sağlamaktadır. Literatürde, gizli katmanlarda sıklıkla kullanılan fonksiyonlar, ReLU(Rectified Linear Activation), Sigmoid(Logistic) ve Tanh(hyperbolic tangent) olarak görülmektedir. Çıkış katmanında kullanılacak olan fonksiyon problemin tipine göre farklılık göstermektedir. Regresyon

problemlerinde doğrusal, sınıflandırma problemlerinde ise softmax ya da sigmoid kullanılmaktadır. Tablo 6.3’ te detaylı olarak gösterilmiştir.



Tablo 6.3 Aktivasyon fonksiyonları

Fonksiyon	Grafik
ReLU $f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$	
Sigmoid $f(x) = \frac{1}{1 + e^{-x}}$	
Hyperbolic tangent $f(x) = \tanh(x)$	
Lineer $f(x) = x$	
Softmax $f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$	

Ağırlıkların hesaplanıp güncellenmesi işlemleri eğitim olarak adlandırılır. Burada amaç model performansı için hedeflenen hata fonksiyonunu minimize etmektir. Sinir ağı modelinde bu fonksiyon hataların kareleri toplamı olarak yazılabilir.

$$E = \sum_{i=1}^n e_i^2 \quad (6.8)$$

Burada e hatayı, n ise veri adetini göstermektedir. İleri beslemeli sinir ağı yöntemlerinde kullanılan eğitim algoritması Geri Yayılım (Back Propagation) algoritması olarak bilinir [54]. Geri yayılım algoritmasında standard olarak kullanılan Steepest Descent Metodu'nun yakınsama hızı yavaş olduğundan sinir ağı eğitimleri için bir çok öğrenme algoritması geliştirilmiştir. Bunlardan başlıcaları Levenberg-Marquardt Algoritması [55], Bayes Düzenleme (Bayesian Regularization) Algoritması [56] ve Ölçeklenmiş Konjuge Gradyen (Scaled Conjugate Gradient) Algoritması [57] olarak söylenebilir.

6.2.1. Levenberg-Marquardt Algoritması

Levenberg-Marquardt algoritması doğrusal olmayan optimizasyon problemlerinin çözümünde kullanılan 2. Dereceden eğitim algoritmasıdır. Güncellenmesi gereken ağırlık değerlerine göre 23 nolu eşitlikte gösterilen hata fonksiyonunun Jakobiye'n'i hesaplanabilir.

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \dots & \frac{\partial e_1}{\partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_n}{\partial w_1} & \dots & \frac{\partial e_n}{\partial w_m} \end{bmatrix} \quad (6.9)$$

Burada m , ağda bulunan ağırlıkların sayısıdır. Jakobiye matrisi bulunduğundan sonra gradyan vektörü (g) ve Hessian matrisi (H) de hesaplanabilir.

$$g = J^T e \quad (6.10)$$

$$H = J^T J \quad (6.11)$$

Ağırlıklar ise Jakobiye matrisine bağı olarak güncellenir.

$$w_{i+1} = w_i - (J_i^T J_i + \alpha_i I)^{-1} (2 J_i^T e_i) \quad (6.12)$$

α_i , öğrenme katsayını, I ise birim matrisi göstermektedir.

6.2.2. Bayes Düzenleme Algoritması

Hata fonksiyonu, düzenleme yöntemi kullanılarak yeniden düzenlenmiştir. Bu işlem, sinir ağıni genelleştirmek için yapılmaktadır [58].

$$F = \mu E_w + \nu E \quad (6.13)$$

Burada, μ ve ν düzenleme parametreleri, E_w ise ağırlıkların karesinin toplamıdır. Burada düzenleme parametrelerinin optimizasyonu için Bayes Düzenleme metodu kullanılmaktadır. Ağırlık değerlerini rastgele değişkenler olarak düşündüğümüzde, verilen veri kümesinde ağırlıkların ardıl olasılık dağılımlarını azami yapacak ağırlık değerlerini hesaplamak amaçlanmaktadır. Ardıl dağılım Bayes kuralına göre ifade edilebilir.

$$P(w|D, \mu, \nu, N) = \frac{P(D|w, \nu, N) P(w|\mu, N)}{P(D|\mu, \nu, N)} \quad (6.14)$$

D , veri kümesini, N ise sinir ağı modelini göstermektedir. $P(D|w, \nu, N)$, yakınlık fonksiyonu, $P(w|\mu, N)$, öncül yoğunluğu ve $P(D|\mu, \nu, N)$ normalizasyon faktörü olarak ifade edilir. Veri kümesindeki ve ağırlıklardaki gürültünün, Gauss dağılımına sahip olduğu söylenebilir. Böylelikle, yakınlık fonksiyonu ve öncül yoğunluk değerleri hesaplanabilir.

$$P(D|w, \nu, N) = \frac{e^{-\nu E}}{Z(\nu)} \quad (6.15)$$

$$P(w|\mu, N) = \frac{e^{-\mu E_w}}{Z_w(\mu)} \quad (6.16)$$

Burada, $Z = \left(\frac{\pi}{\nu}\right)^{n/2}$ ve $Z_w = \left(\frac{\pi}{\mu}\right)^{m/2}$ olarak hesaplanır. Denklem 5.14, 5.15 ve 5.16 düzenlenerek, ağırlıklara ardıl dağılım tekrar yazılabilir.

$$P(w|D, \mu, \nu, N) = \frac{e^{-(\mu E_w + \nu E)}}{Z_w(\mu)Z(\nu)} \quad (6.17)$$

N modelinde düzenleme parametreleri etkilidir. Bu parametrelerin optimizasyonu için de Bayes kuralı uygulanabilir.

$$P(\mu, \nu|D, N) = \frac{P(D|\mu, \nu, N) P(\mu, \nu|N)}{P(D|N)} \quad (6.18)$$

5.18 nolu denkklemden görüldüğü gibi $P(D|\mu, \nu, N)$ fonksiyonu ile $P(\mu, \nu|D, N)$ doğru orantılıdır. Bu nedenle, $P(D|\mu, \nu, N)$ fonksiyonun azami değeri bulunmalıdır. 5.13 nolu denklemin Taylor açılımı kullanılarak düzenleme parametreleri hesaplanabilir.

$$\mu = \frac{\gamma}{2E_w} \quad (6.19)$$

$$\nu = \frac{n - \gamma}{2E} \quad (6.20)$$

$$\gamma = m - \mu \operatorname{tr}(H^{-1}) \quad (6.21)$$

6.2.3. Ölçeklenmiş Konjuge Gradyen Algoritması

Standart geri yayılım algoritmasında uygulanan Steep Descent algoritmasında ağırlıklar güncellenirken gradyan vektörünün tersi yönünde arama yapılır. Bu yönde hata fonksiyonu hızla azalsa da yakınsama hızı için aynı şeyi söylemek mümkün değildir. Konjuge gradyan algoritmaları ise en hızlı yakınsamaya sahip yönü kullanarak arama yapmaktadır. Bu yöne konjuge yönü denir. Bu yöntemde arama ilk olarak Steep Descent algoritması gibi gradyan vektörünün tersinde başlamakla beraber ikinci terasyondan itibaren farklılık göstermektedir.

$$p_0 = -g_0 \quad (6.22)$$

$$x_{k+1} = x_k + \alpha_k g_k \quad (6.23)$$

$$p_k = -g_k + \beta_k p_{k-1} \quad (6.24)$$

Burada β_k katsayısının hesaplanış biçimine göre farklı algoritmalar geliştirilmiştir. Moller ise geliştirdiği algortmada adım sayısının hesabı için LM algoritması ile Konjuge Gradyan algoritmasını birleştirmiştir. Bu algortmaya Ölçeklenmiş Konjuge Gradyan algoritması denilmektedir [59]. Hessian matrisinin yaklaşık değerini hesaplamaya dayalı bu algortmada tasarım parametreleri her iterasyonda değişmekte olup kullanılan bağımsızdır. Bu algortmanın başarısını etkileyen en önemli faktördür.

$$H_k = \frac{E'(w_k + \sigma_k p_k) - E'(w_k)}{\sigma_k} + \lambda_k p_k \quad (6.25)$$

$$\beta_k = \frac{(|g_{k+1}|^2 - g_{k+1}^T g_k)}{g_k^T g_k} \quad (6.26)$$

$$p_{k+1} = -g_{k+1} + \beta_k p_k \quad (6.27)$$

6.2.4. Bulgular ve Tartışma

NARX Ağının performansı seçilen eğitim algoritması, giriş ve çıkış gecikmeleri, gizli katman ve nöron sayısı gibi parametrelere göre farklılık göstermektedir. Bu çalışma kapsamında her bir parametre bağımsız olarak incelenmiştir. Modeller MATLAB 2020a kullanılarak eğitilmiştir. Model performanslarını değerlendirmek için ortalama kare hata (RMSE) ve Ortalama mutlak hata (MAE) değerleri kullanıldı. Tablo 6.4 'te gösterilmiştir.

Tablo 6.4 Performans ölçüm metrikleri

Metrikler	Denklem	Açıklama
Root mean square error	$RMSE = \sqrt{\frac{\sum_{i=1}^n e_i^2}{n}}$	Düşük değerler başarılı olduğunu gösterir.
Mean absolute error	$MAE = \frac{\sum_{i=1}^n e_i }{n}$	

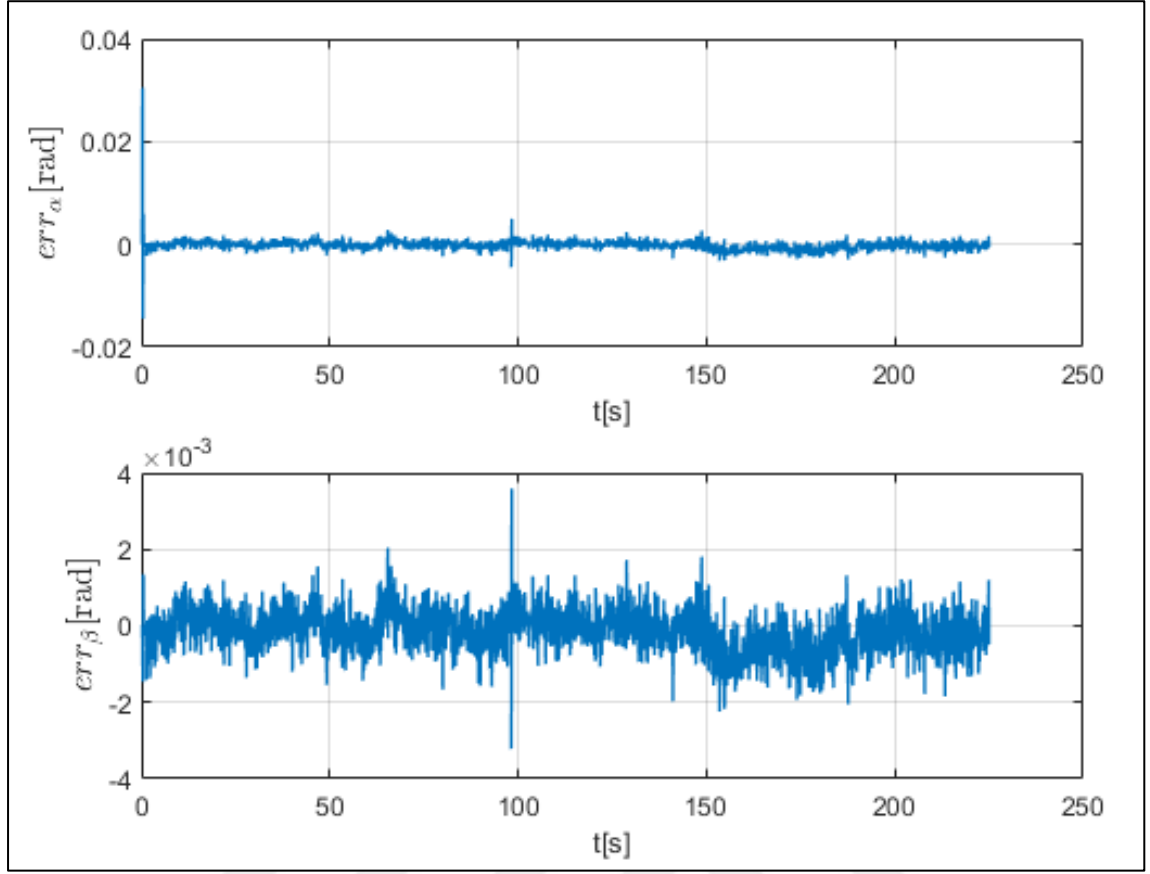
İlk olarak, tek bir gizli katman ve 15 nörondan oluşan modelde eğitim algoritmalarının (Bayes Arrangement, Levenberg-Marquardt, Scaled Conjugate Gradient) performansları karşılaştırılmıştır. Giriş ve çıkış gecikme vektörleri sırasıyla 1 ve 2 olarak belirlenmiştir. Gizli katmanlarda aktivasyon fonksiyonu olarak hiperbolik tanjant, çıkış katmanında ise lineer fonksiyon kullanılmıştır. Eğitim algoritmalarına göre hatalar Tablo 6.5 'te gösterilmiştir.

Tablo 6.5 Eğitim algoritmalarına göre model performansları

Algorithm	RMSE	MAE	RMSE	MAE	RMSE	MAE
LM	0.00070	0.00050	0.00260	0.00230	0.00160	0.00110
BR	0.00070	0.00050	0.00250	0.00210	0.00150	0.00110
SCG	0.02600	0.00810	0.01010	0.00180	0.02180	0.00370

Hızlı eğitim süresine rağmen SCG, LM ve BR'ye göre daha kötü performans göstermiştir. Bu aşamada gizli katman ve içindeki nöron sayısı değiştirilerek sonuçlar incelenmiştir. Eğitim algoritması BR olarak sabit bırakılmıştır.

Hücum açısı ve kayma açısına göre tek gizli katman ve 5 nöronlu model 4'ün en iyisi olduğu görülmektedir. Model sonuçlarının gerçek sistemle karşılaştırılması Şekil 6.12 'de gösterilmiştir.

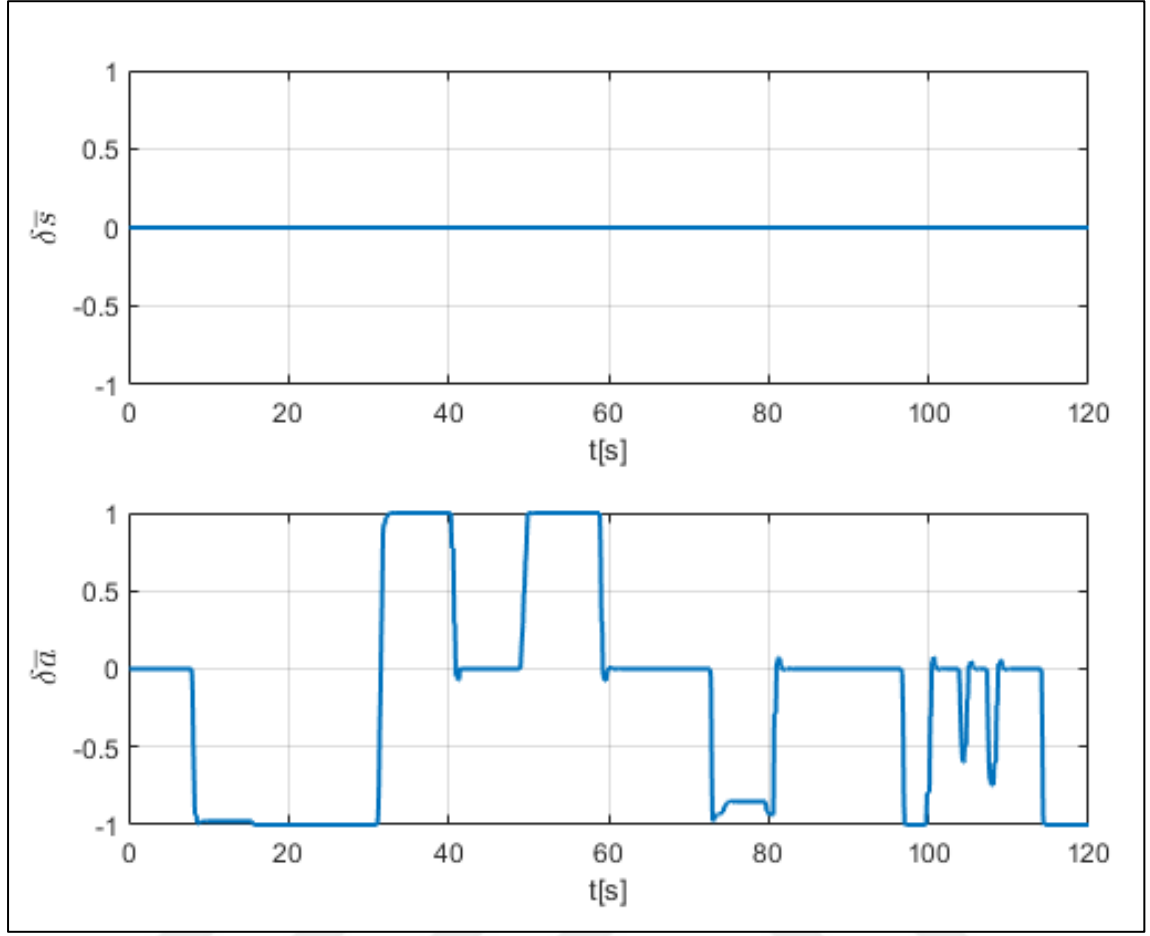


Şekil 6.12 Sistem kestirim hataları

Aynı aerodinamik özelliklere ve farklı ağırlıklara sahip modellerin performansını gözlemleyebilmek için sistem ağırlığı 10 kg'a çıkarılmış ve uçuş 1000m irtifadan gerçekleştirilmiştir. Uçuş sırasında üretilen kontrol girdileri Şekil 6.13'te gösterilmektedir.

Tablo 6.6 Mimariye göre model performansları

No	Gizli Katman				Eğitim		Test		Toplam	
	1	2	3	4	RMSE	MAE	RMSE	MAE	RMSE	MAE
1	10	-	-	-	0.00070	0.00050	0.00250	0.00210	0.00150	0.00110
2	3	-	-	-	0.15380	0.00510	0.00180	0.00160	0.12560	0.00390
3	5	2	-	-	0.02080	0.00720	0.03180	0.03120	0.02500	0.01520
4	5	-	-	-	0.00080	0.00060	0.00120	0.00100	0.00100	0.00070
5	10	5	-	-	0.00070	0.00050	0.00560	0.00440	0.00330	0.00180
6	25	-	-	-	0.00060	0.00050	0.00180	0.00150	0.00120	0.00080
7	50	-	-	-	0.00060	0.00050	0.00230	0.00190	0.00140	0.00100
8	15	12	-	-	0.00070	0.00050	0.00270	0.00200	0.00170	0.00100
9	15	12	12	-	0.00090	0.00050	0.00150	0.00130	0.00120	0.00080
10	15	12	12	6	0.00090	0.00050	0.00160	0.00140	0.00120	0.00080

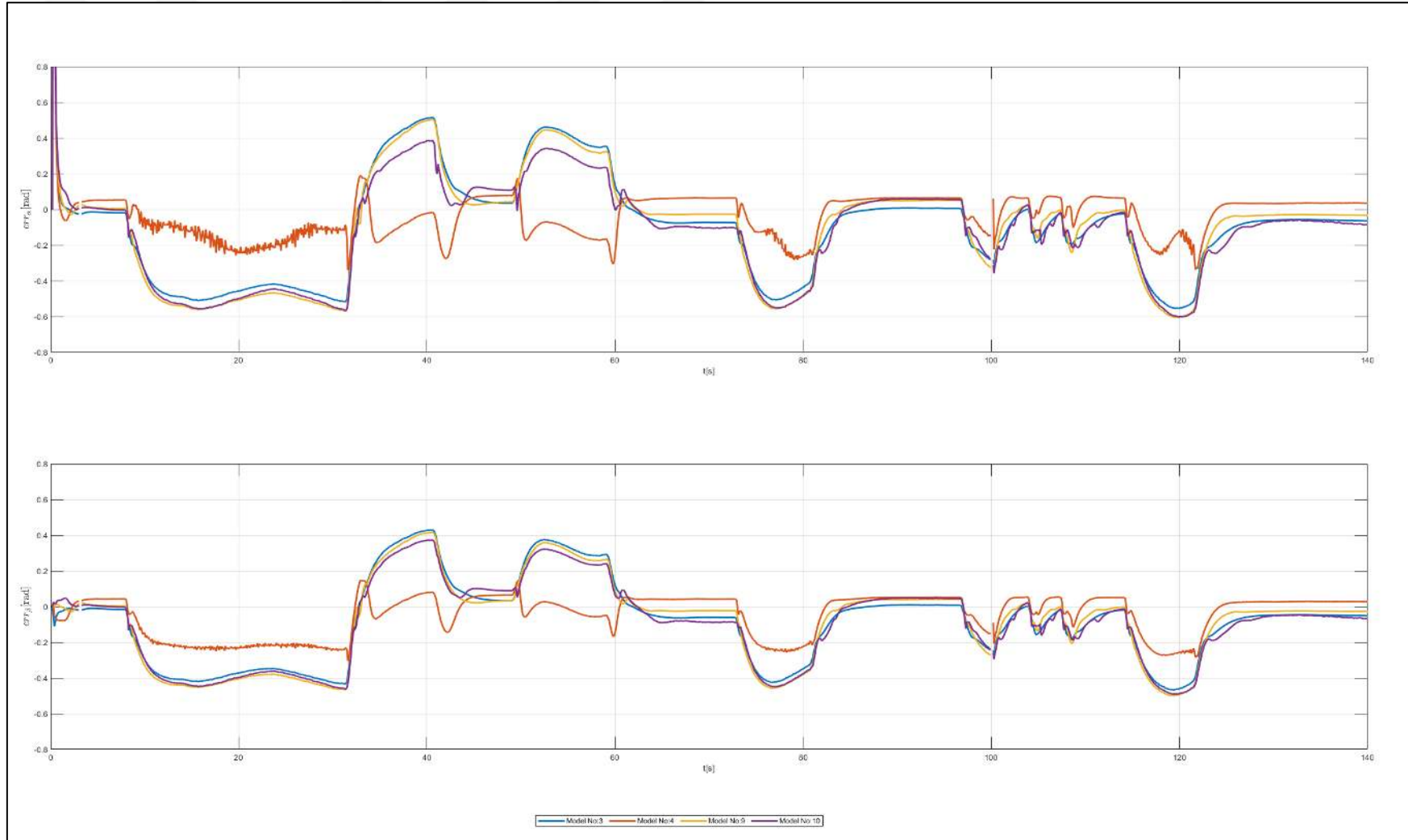


Şekil 6.13 10kg'lık sistem için kontrol girdileri

120 saniyelik uçuş için model performansları, hata ölçümleri ve hesaplama maliyetleri kullanılarak karşılaştırıldı. Tablo 6'da görüldüğü gibi gizli katman ve nöron sayısındaki artış hesaplama süresini uzatmaktadır. Model performanslarına bakıldığında ise 4 numaralı model ön plana çıkıyor.

Tablo 6.7 10kg'lık sistem için model performansları

Model No	RMSE	MAE	İşlem Zamanı (ms)
1	0.23790	0.17820	42.2740
2	0.22350	0.15230	40.9710
3	0.29650	0.22060	49.7470
4	0.13630	0.09730	43.3250
5	0.30290	0.23760	43.7770
6	0.29630	0.20710	43.8720
7	0.26160	0.18020	44.5370
8	0.31000	0.21820	66.2630
9	0.31370	0.22190	66.8510
10	0.30040	0.22440	66.9610



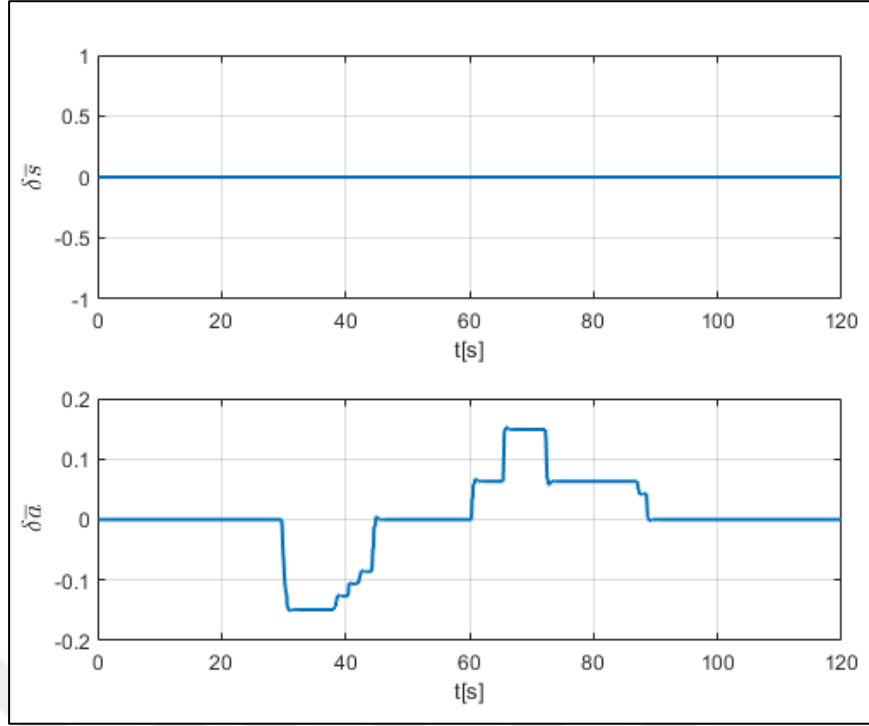
Şekil 6.14 10kg'lık ağırlık için model kestirim hataları

Artan ağırlık için tahmin hataları Şekil 6.14 'te gösterilmiştir. Gizli katman sayısındaki artış, model performansında önemli bir değişiklik yapmasa da aşım değerlerini artırmıştır. Son olarak geliştirilen modellerin performansları farklı aerodinamik özelliklere sahip bir sistemde incelenmiştir. Gerekli parametreler için ALEX adlı bir hava dağıtım sistemi kullanıldı (Tablo 6.8) [26].

Tablo 6.8 ALEX Parametreleri[10]

Parametre	Değer
Ağırlık (m)	97.60 kg
Paraşüt Kesit Alanı (S)	19.720 m2
Aerodinamik Katsayılar	$\begin{bmatrix} C_{D0} = 0.0840 & C_{D\alpha^2} = 0.900 \\ C_{Y\beta} = -0.2160 & C_{L0} = 0.250 \\ C_{L\alpha} = 2.360 & C_{m0} = 0.00 \\ C_{m\alpha} = 0.00 & C_{mq} = -0.1740 \\ C_{l\beta} = 0.1040 & C_{lp} = -0.1490 \\ C_{lr} = 0.0960 & C_{l\delta_a} = -0.0480 \\ C_{n\beta} = 0.0190 & C_{np} = -0.0270 \\ C_{D0} = 0.0840 & C_{n\delta_a} = 0.0390 \end{bmatrix}$

200m yükseklikten başlayan uçuş için sisteme verilen kontrol girdileri Şekil 6.15 'te verilmiştir.

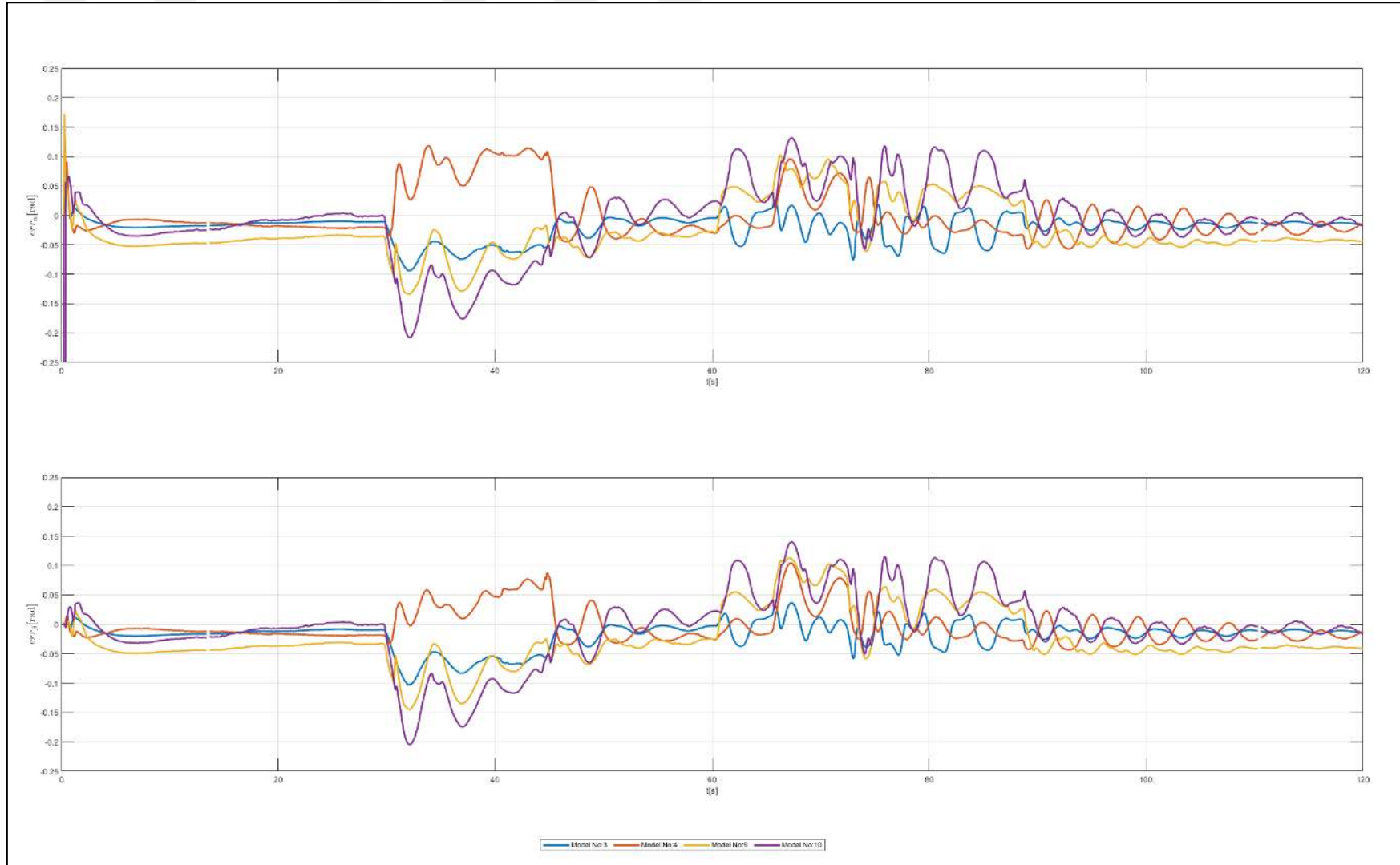


Şekil 6.15 ALEX için kontrol girdileri

Modellerin performansları Tablo 6.9’ da gösterilmiştir. En iyi performansa sahip olduğu görülen Model 1, tek bir gizli katman ve 10 nörondan oluşmaktadır.

Tablo 6.9 ALEX için model performansları

Model No	RMSE	MAE
1	0.02960	0.02250
2	0.04910	0.04570
3	0.07810	0.07210
4	0.12510	0.11950
5	0.04130	0.03080
6	0.05620	0.04710
7	0.03880	0.02730
8	0.07340	0.03750
9	0.05100	0.04620
10	0.10240	0.04350



Şekil 6.16 ALEX için model kestirim hataları

Geliştirilen modellerin sınırlarını belirlemek için ağırlık ve aerodinamik katsayıların modeller üzerindeki etkileri gözlemlenmiştir. Kontrol girdilerinin kuvvet ve moment üzerindeki etkisini belirleyen aerodinamik katsayılar seçilir. Seçilen parametreler +%10 değiştirilerek hata oranları gözlemlendi. Hata ölçüsü olarak RMSE kullanıldı. Tablo 6.10’ da görüldüğü gibi tek gizli katman ve 5 nöronlu oluşan model 4 en iyi performansa sahiptir.

5 derecelik maksimum hata göz önüne alındığında, modellerin sınırları enterpolasyon ile parametrelere göre yaklaşık olarak belirlenmiş ve Tablo 6.11’ de gösterilmiştir.



Tablo 6.10 %10 deęişimler için model performansları

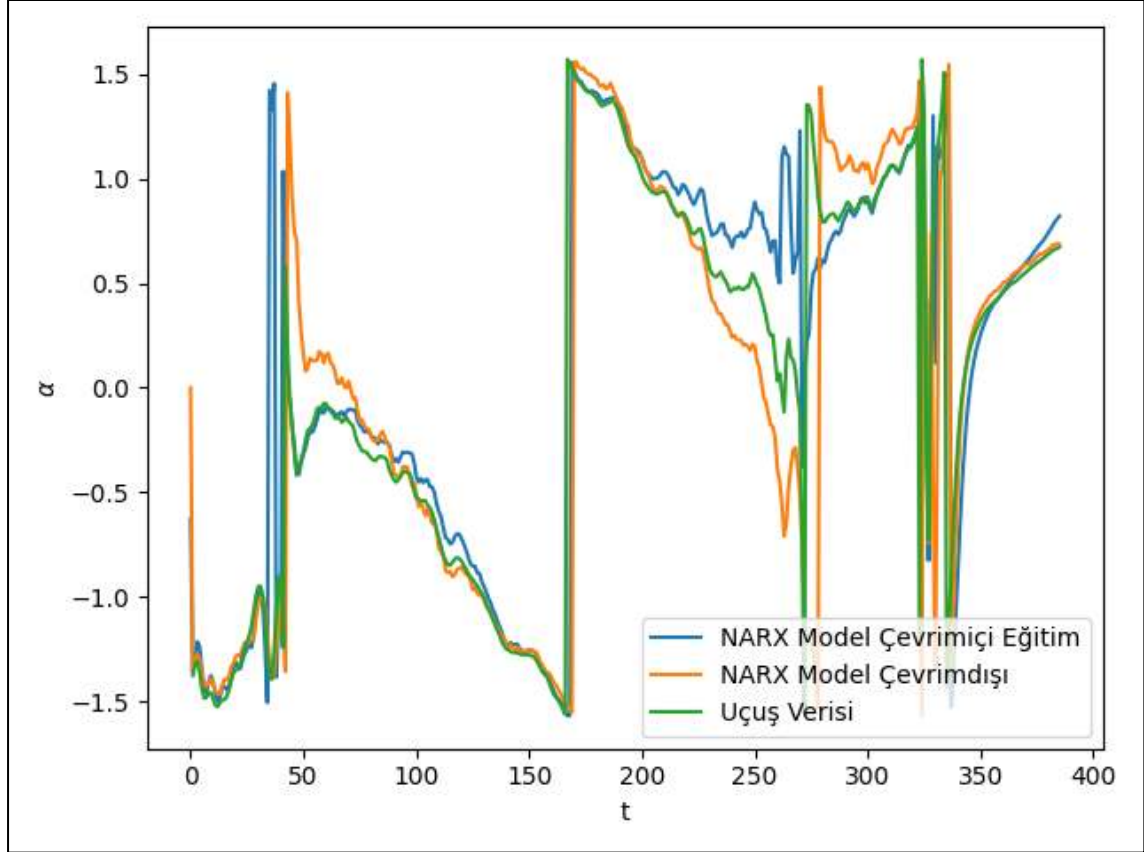
No	m		$C_{d\delta_s}$		$C_{l\delta_a}$		$C_{l\delta_s}$		$C_{n\delta_a}$	
	+10%	-10%	+10%	+10%	-10%	+10%	-10%	-10%	-10%	-10%
1	0.00350	0.00420	0.00360	0.00340	0.00330	0.00320	0.00300	0.00290	0.13450	0.00390
2	0.00700	0.00720	0.00630	0.00450	0.00570	0.00550	0.00520	0.00560	0.03970	0.00650
3	0.01560	0.01570	0.01290	0.02840	0.01660	0.01820	0.01790	0.01830	0.05910	0.01820
4	0.00320	0.00410	0.00320	0.00230	0.00310	0.00310	0.00270	0.00310	0.07390	0.00280
5	0.00550	0.00940	0.00720	0.00540	0.00640	0.00610	0.00530	0.00580	0.20640	0.00630
6	0.01230	0.01160	0.01110	0.00730	0.00990	0.01040	0.00810	0.00860	0.13340	0.00930
7	0.01190	0.01140	0.01220	0.00640	0.00860	0.00870	0.00730	0.00780	0.06230	0.00890
8	0.01230	0.01610	0.00270	0.00840	0.01210	0.01120	0.00980	0.00200	0.20580	0.01100
9	0.01200	0.01040	0.00990	0.00570	0.01100	0.00940	0.00760	0.01130	0.05780	0.00890
10	0.01210	0.01310	0.01020	0.00770	0.01120	0.01020	0.00900	0.01000	0.09500	0.00910

Tablo 6.11 Modellerin sınırları

No	m		$C_{d\delta_s}$		$C_{l\delta_a}$		$C_{l\delta_s}$		$C_{n\delta_a}$	
	+10%	-10%	+10%	+10%	-10%	+10%	-10%	-10%	-10%	-10%
1	6.6230	0.0000	0.3420	-0.1560	0.5450	-0.2580	0.3900	-0.2000	0.0032	-0.0037
2	4.2610	0.0000	0.2380	-0.0930	0.3790	-0.0870	0.2670	-0.0550	0.0037	-0.0010
3	2.9600	0.8470	0.1670	0.0690	0.2290	0.0780	0.1490	0.0520	0.0034	0.0016
4	7.0660	0.0000	0.3720	-0.2780	0.5710	-0.2710	0.4220	-0.1810	0.0034	-0.0063
5	4.9050	0.1410	0.2210	-0.0610	0.3540	-0.0640	0.2640	-0.0500	0.0031	-0.0011
6	3.2440	0.4750	0.1780	-0.0190	0.2820	0.0250	0.2070	-0.0010	0.0032	0.0002
7	3.2890	0.4500	0.1710	-0.0360	0.3020	0.0000	0.2190	-0.0120	0.0034	0.0001
8	3.2440	0.8730	0.4220	-0.0040	0.2580	0.0330	0.1890	-0.3350	0.0031	0.0006
9	3.2780	0.3110	0.1880	-0.0530	0.2690	0.0110	0.2140	0.0230	0.0035	0.0001
10	3.2660	0.6380	0.1850	-0.0130	0.2670	0.0220	0.1970	0.0130	0.0033	0.0001

6.3. Gerçek Uçuş Verisi

Bölüm 5’te elde edilen uçuş verileri modellerin performansında kullanılmıştır. Burada Test 3 ‘te elde edilen veri eğitimde Test 2’de elde edilen veri ise model performansının ölçümünde kullanılmıştır. Sisteme çevrimiçi eğitimde eklenerek model gerçek zamanlı olarak güncellenmiş ve performansları karşılaştırılmıştır. Burada 4 numaralı model kullanılmıştır.



Şekil 6.17 Gerçek uçuş verisi model sonuçları

Çevrimiçi eğitimin sisteme etkisi tabloda gösterilmiştir.

Tablo 6.12 Gerçek uçuş verisi model performansları

Model	MSE	STD
NARX	0.34	0.58
NARX(Çevrimiçi)	0.25	0.51

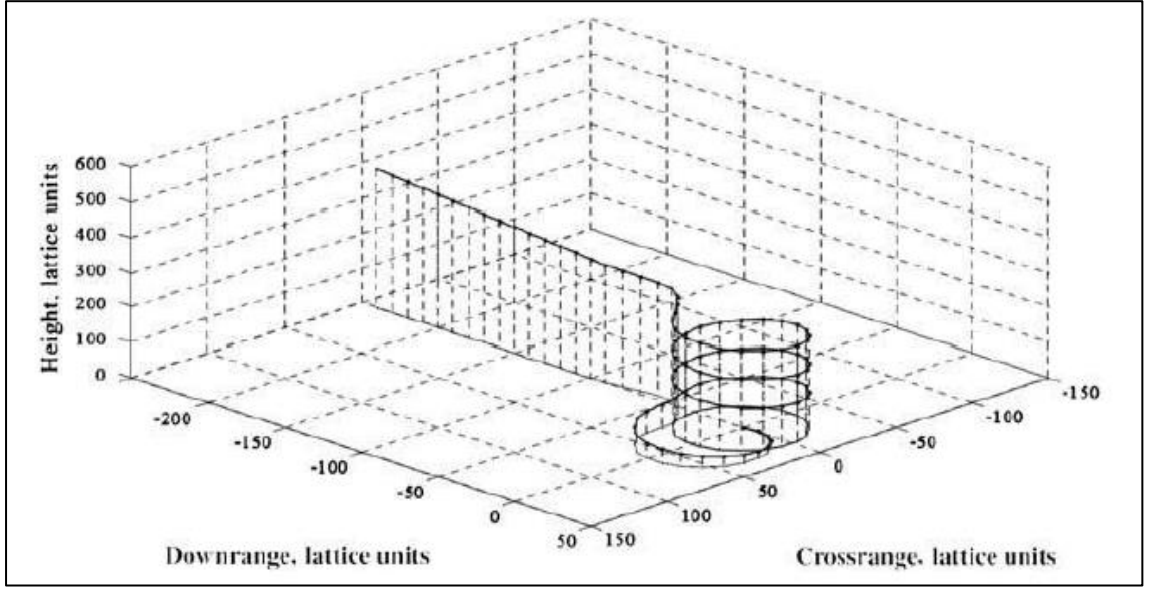
7. ROTA PLANLAMA

Paraşütlü iniş sisteminin hedef noktaya ulaşabilmesi için yörünge planlanması gerekmektedir. Bu tür sistemlerde yörünge planlama problemi sistemin yatay ve dikey hızlarındaki yetersiz kontrolü nedeniyle oldukça zordur. Aynı zamanda rüzgâr etkisi sistem üzerinde bir hayli baskındır.

Literatürde yörünge planlaması için çeşitli yöntemler geliştirilmiştir. Bunlarda bütün rotayı hesaplayanlar olduğu gibi rotayı fazlara bölen yaklaşımlar da mevcuttur. Carter ve arkadaşları geliştirdikleri algoritmada “homing” ve enerji yönetimi aşamaları için farklı yöntemler geliştirmiştir.

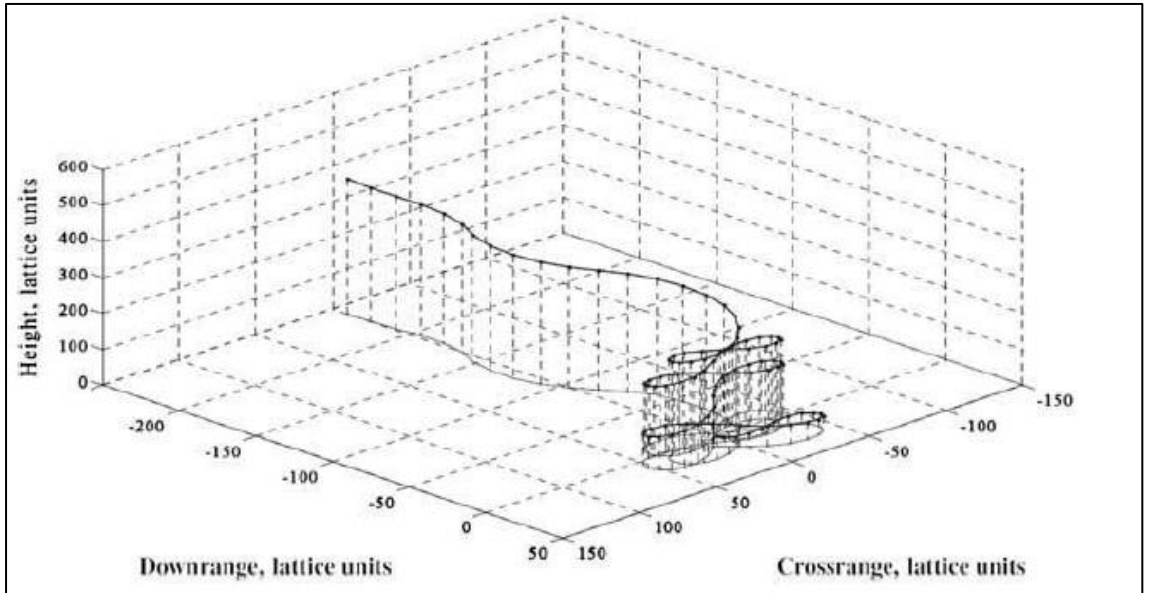
Carter ve arkadaşları, belirtilen problem dinamik programlama tabanlı bir çözüm ürettirler [60] [61] [62]. Burada önceden hesapladıkları en uygun sapma açısı hızlarını 5 boyutlu bir dizide toplamaktadırlar. Bu dizinin elemanları yatay pozisyon, sapma açısı, sapma açısı hızı ve yükseklikten oluşmaktadır. Bu dizi sadece bir kere hesaplanmaktadır. Ve bütün güzergâh boyunca rüzgârı ihmal etmektedir. Gerçek zamanlı olarak uçuş sistemi interpolasyon yardımıyla uygun sapma hızı açısı değerlerini tablodan hesaplamaktadır. Rüzgârın katkısı iste iniş noktasına yaklaştığında sisteme yer değiştirme olarak eklenir. Bu yaklaşımda hareket komutlarını gerçek zamanlı olarak güncellemeye gerek yoktur. Önceden hesaplanan çözümler uçuş sırasında seçilerek kullanılır ve otopilota iletilir.

Dinamik programlama tabanlı bir başka çözüm ise Cleminson tarafından üretilmiştir. Carter ve arkadaşlarından farklı olarak 3 adet durum değişkeni kullanmıştır. Aynı zamanda uçuş komutları doğrudan hesaplanmamaktadır. Kullanılan modelin basitliği ve erişilebilir küme kavramı hesaplama yükünü oldukça azaltmıştır [63].

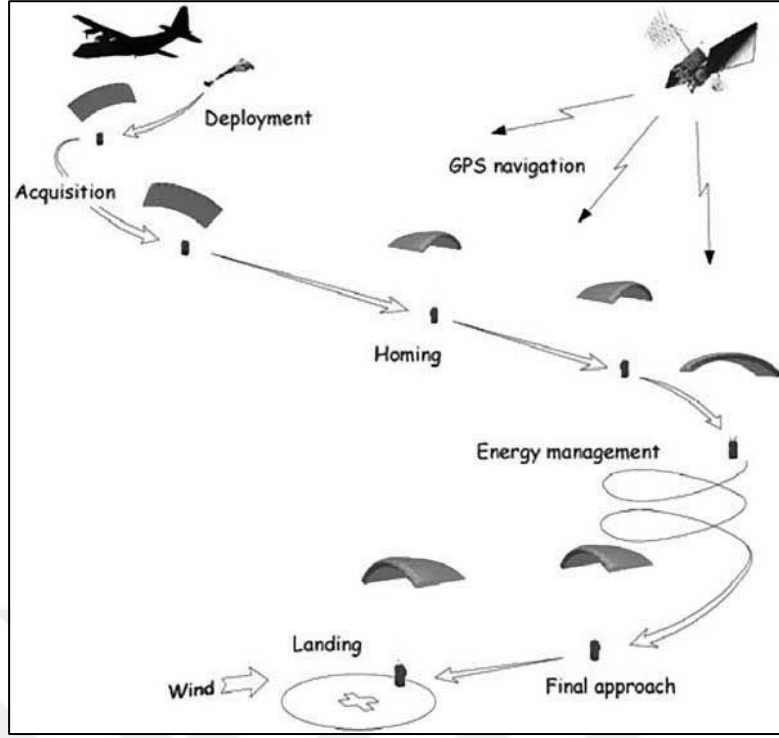


Şekil 7.1 Rüzgarsız ortamda planlama evresi [63]

Bu çalışmalarda farklı yöntemler geliştirilse de ortak sonuç 3 kademeli rota planlamasına varmaktadır. Bunlardan ilki hedef noktaya doğru süzülme olarak başlamaktadır. (homing) Hedef noktaya yaklaşılmca PKİS potansiyel enerji fazlalığını atmaya başlar. Bu evreye enerji yönetimi denir. Belirlenen yüksekliğe alçalma gerçekleşince iniş evresi başlar ve rüzgara karşı iniş gerçekleşir [64].



Şekil 7.2 Rüzgarlı ortamda planlama evresi [63]



Şekil 7.3 PKİS Yörünge Planlama [64]

Bu çalışmada da ilk olarak optimal kontrol teorisi incelenmiş ve hedef konuma varmak için gerekli rota planlama yapılmıştır.

7.1. Optimal Rota Teoremi

Optimal rota hesaplanmasında Optimal Kontrol probleminden faydalanılabilir.

$$\dot{x} = f(x(t), u(t), t) \quad (7.1)$$

$$x(t_0) = x_0, x(t_1) = x_1, t_0 \leq t \leq t_1 \quad (7.2)$$

$$J = \psi(x_1, t_1) + \int_{t_0}^{t_1} L(x(t), u(t), t) dt \quad (7.3)$$

Optimal kontrol problemi denklem 6.3 'te ifade edilen dinamik sistem modeli için J maliyet fonksiyonunu minimize etmek olarak tanımlanabilir.

PKİS' te kontrol kollarının simetrik - asimetrik hareketleri sapma açısı hızında ve iniş hızında etkilidir. Paraşütün aerodinamik özelliklerine bağlı olmakla beraber iniş hızındaki

etki oldukça düşüktür. İniş hızı sabit kabul edildiğinde yatay düzlemde sistem aşağıdaki gibi ifade edilebilir.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} V\cos(\varphi) \\ V\sin(\varphi) \\ u \end{bmatrix} \quad (7.4)$$

$$H = p_{\varphi}u + V[p_x\cos(\varphi) + p_y\sin(\varphi)] - 1 \quad (7.5)$$

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_{\varphi} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ Vp_x\sin(\varphi) - Vp_y\cos(\varphi) \end{bmatrix} \quad (7.6)$$

$$u = K\text{sign}(p_{\varphi}) \quad (7.7)$$

$$\ddot{p}_{\varphi} + K^2 p_{\varphi} \pm K = 0 \quad (7.8)$$

$$p_{\varphi} = C_1 \sin(Kt + C_2) \pm K^{-1} \quad (7.9)$$

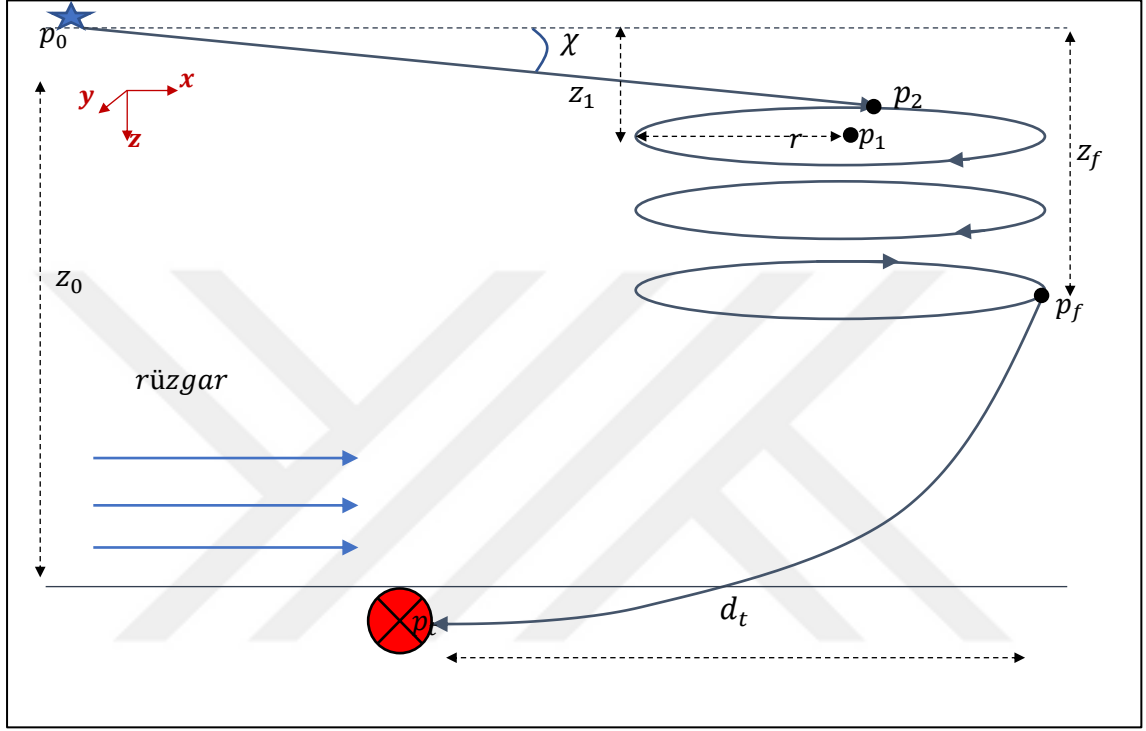
Pontrjagin prensibine göre sistemin Hamiltonian'ı yazılabilir. Hamiltonian'ı ifade eden adjoint değişkenlerin denklemleri de aşağıdaki gibi yazılabilir. Minimum zaman problemine göre J maliyet fonksiyonunu minimize etmek için $H = 0$ eşitliği çözülür. Burada p_{φ} , denkleminin türevi alınıp Hamiltonian'da yerine yazılırsa ikinci dereceden denklem elde edilir. Denklem çözümünde görüldüğü gibi eğer $C_1 \neq K^{-1}$ ise sistem spiral çizerek iniş yapmaktadır. $C_1 = K^{-1}$ olduğunda ise sapma açısı sabit bir şekilde iniş yapmaktadır. Analiz sonucuna göre optimal rota spiral ve düz inişlerden oluşmaktadır.

7.2. Planlama Algoritması

Rota planlaması yapılırken optimal rota göz önüne bulundurulduğunda, sistem fazla potansiyel enerjinin atılması bir başka deyişle istenilen irtifanın elde edilebilmesi için rotanın bir bölümünde spiraller çizmektedir. Bu düzenleme, uçuşun amacı ve iniş yerinin özelliklerine göre değişiklik gösterebilir. Literatürde, genellikle görülen planlama stratejisi,

hedefe yönelme ve hedef bölgesinde alçalma olarak görülmüştür [65]. Aynı zamanda iniş rüzgârı doğru yapılmaktadır. Bu çalışmada da benzer strateji uygulanmıştır.

Rota planlama algoritması ayrılma noktası (p_0), hedef noktası (p_t), alçalma yarıçapı (r) ve rüzgâr yönüne göre hedef noktasından itibaren geriye doğru çalışmaktadır. Süzülme açısı (χ) ve iniş yeri doğrusal yol (d_t) olarak belirtilmiştir (Şekil 8.4).



Şekil 7.4 PKİS Rota Şematik Gösterimi

Rota planlaması yapılırken referans eksen bırakılma noktasına sabitlenmiştir. Yatay düzlemde Şekil 7.5'te gösterilmiştir. Burada $\hat{w}$, rüzgâr vektörünün birim vektörüdür.

$$\xi = \sin^{-1} \left(\frac{r}{d_1} \right) \quad (7.14)$$

$$\hat{n} = \frac{p_1 - p_0}{|p_1 - p_0|} \quad (7.15)$$

$$\hat{m} = \hat{n} \begin{bmatrix} \cos \xi & -\sin \xi \\ \sin \xi & \cos \xi \end{bmatrix} \quad (7.16)$$

$$p_2 = d_2 \hat{m} \quad (7.17)$$

Dönüş noktaları hesaplandıktan sonra kritik irtifa değerleri hesaplanabilir.

$$K = \tan \chi \quad (7.18)$$

$$z_1 = d_2 K \quad (7.19)$$

$$z_f = z_0 - d_t K \quad (7.20)$$

Kritik irtifalar hesaplandıktan sonra, rota irtifaya bağlı olarak gerçek zamanlı hesaplanabilir.

$$p(z) = \begin{cases} \frac{z}{K} \hat{m} & ; 0 < z \leq z_1 \\ p_1 + r \frac{(p_2 - p_1)}{|p_2 - p_1|} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} & ; z_1 < z \leq z_f, \theta = \tan^{-1} \left(\frac{z - z_1}{Kr} \right), \\ p_f - \frac{z - z_f}{K} \hat{w} & ; z_f < z \leq z_0 \end{cases} \quad (7.21)$$

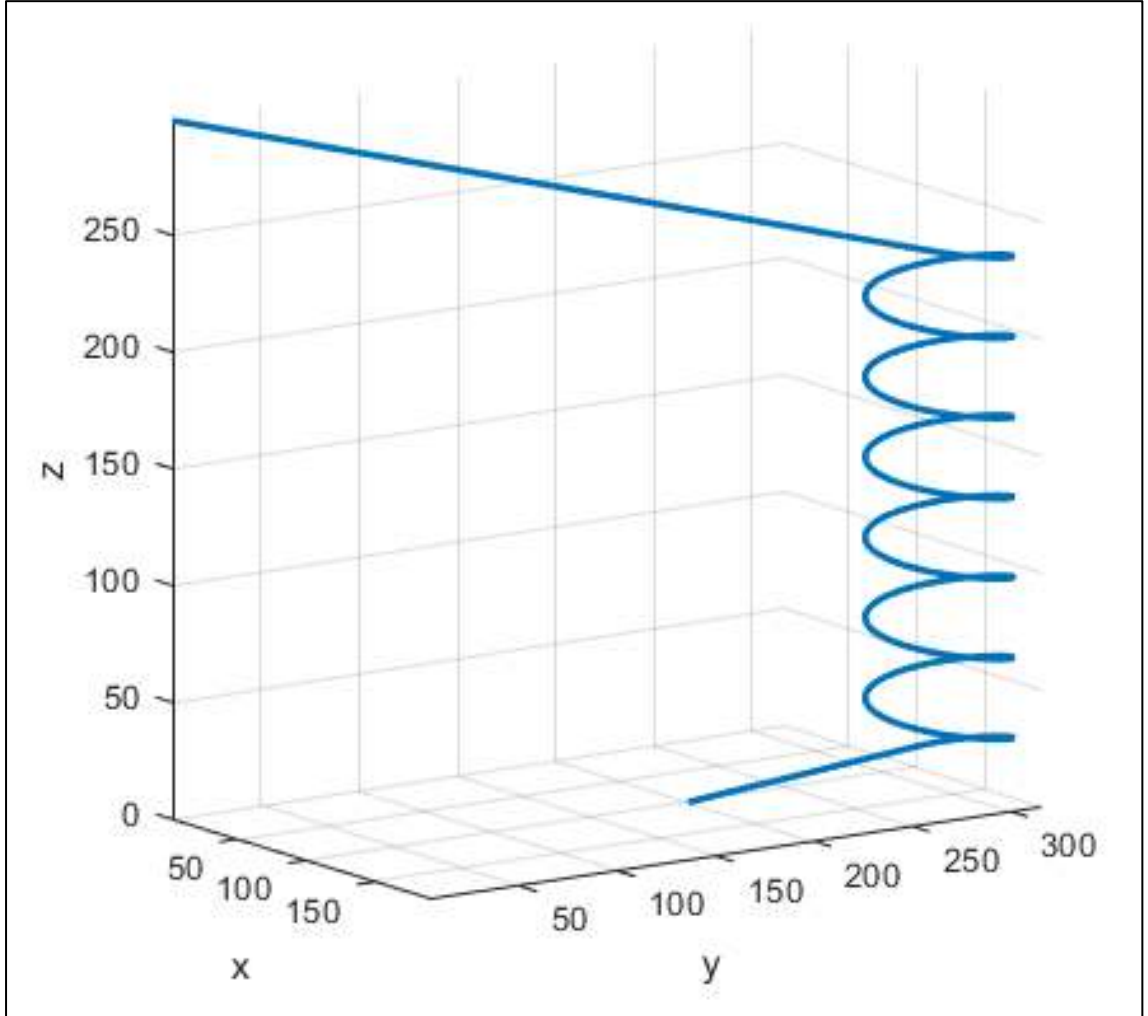
Rotanın sürekli olabilmesi için ;

$$p(z_f) = p_f \quad (7.22)$$

olmalıdır. Bu denklemi sağlayacak r değerini bulmak için optimizasyon probleminin çözülmesi gerekmektedir. r_{min} ve r_{max} değerleri sistemin min ve maksimum dönüş yarıçaplarıdır.

$$r = \operatorname{argmin} |p(z_f) - p_f|_{r_{min} \leq r \leq r_{max}} \quad (7.23)$$

6.21 nolu denklem kullanılarak örnek rota planması Şekil 4.3 teki gibi olmaktadır.



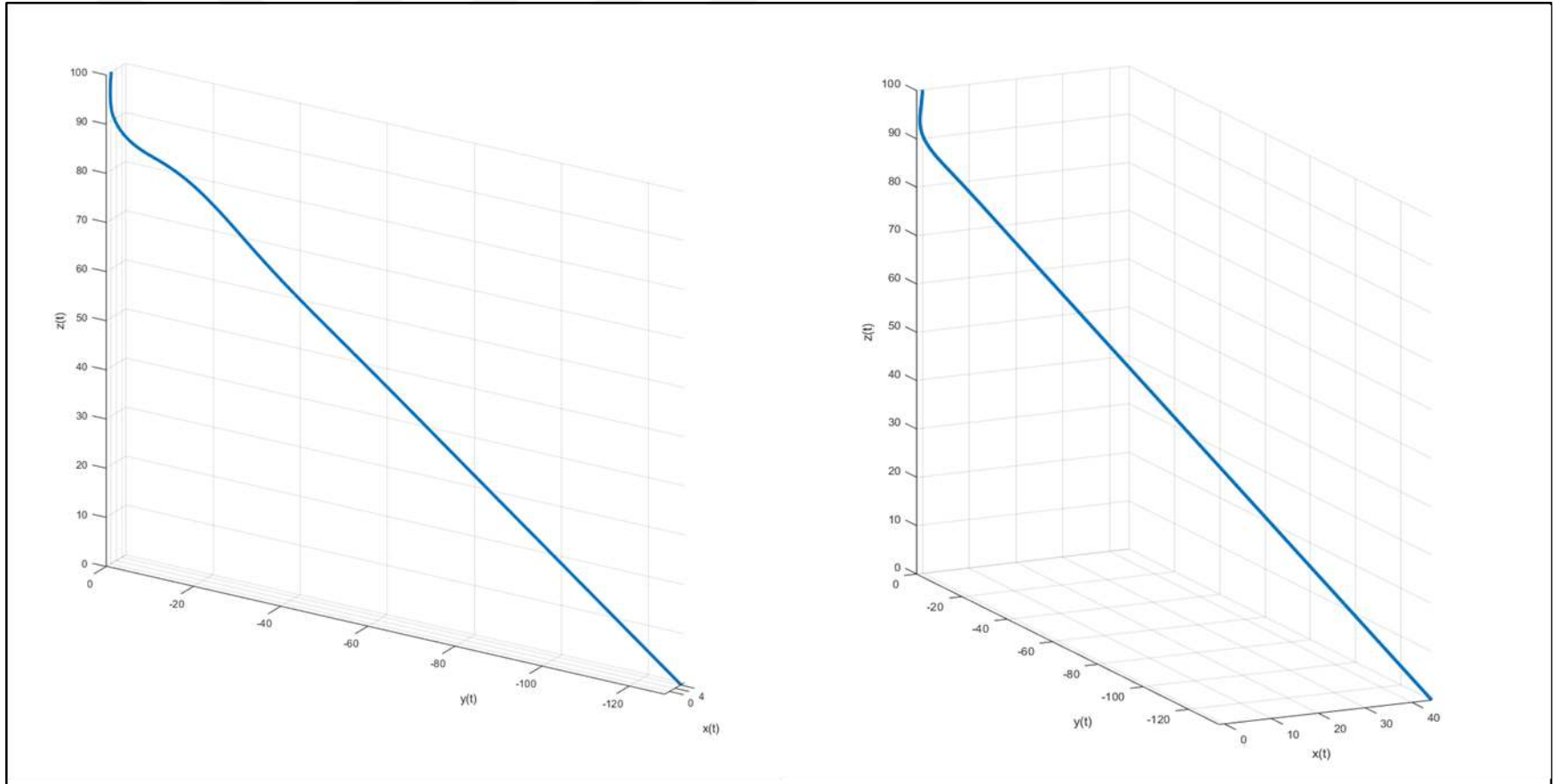
Şekil 7.6 Örnek rota planı

8. KONTROLCÜ ÇALIŞMALARI

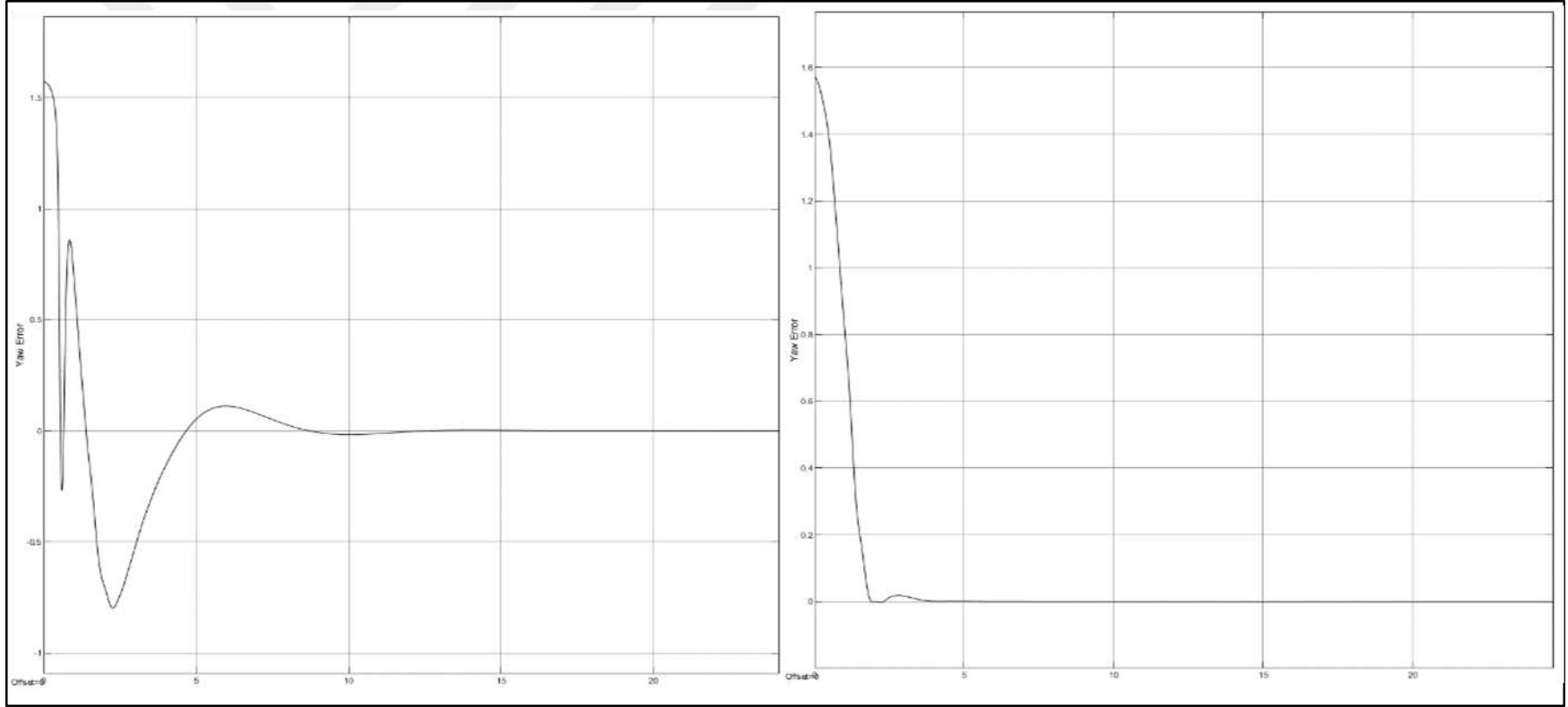
İniş sisteminin istenilen konuma hareketinin sağlanması için belirlenen rotaya göre kontrolünün gerçekleşmesi gerekmektedir. İstenilen rotada hareketin sağlanması için gereken kontrollerden bir tanesi sapma açısı kontrolüdür. Simulasyon ortamında sapma açısı kontrolü için PID kontrolcü geliştirilmiş ve rüzgarlı-rüzgarsız durumlar için kontrol sağlanmıştır.

Bir sonraki çalışma ise model öngörülü kontrol üzerine olmuştur. Burada sistem uçuş verisi ile eğitilen NARX modeli kullanılmıştır. Model çıktısına göre sistemin hedef konuma göre hatası hesaplanarak kontrolcü optimize edilmiştir.



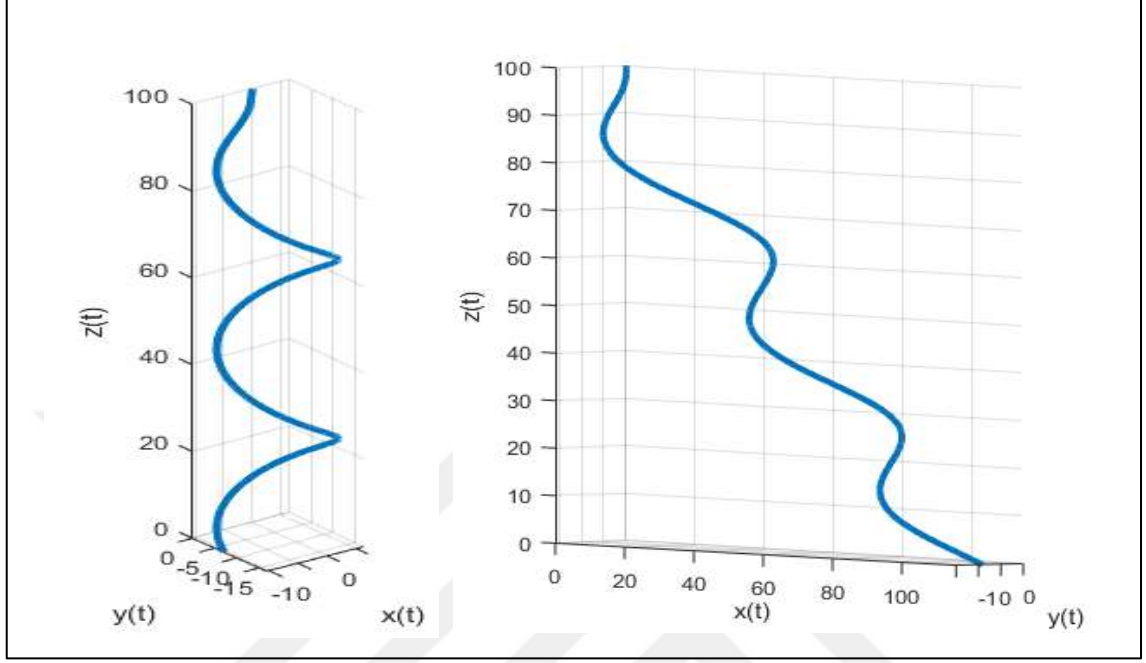


Şekil 8.1. Sapma açısı kontrollü düşüş (90 derece – rüzgarsız - rüzgarlı)



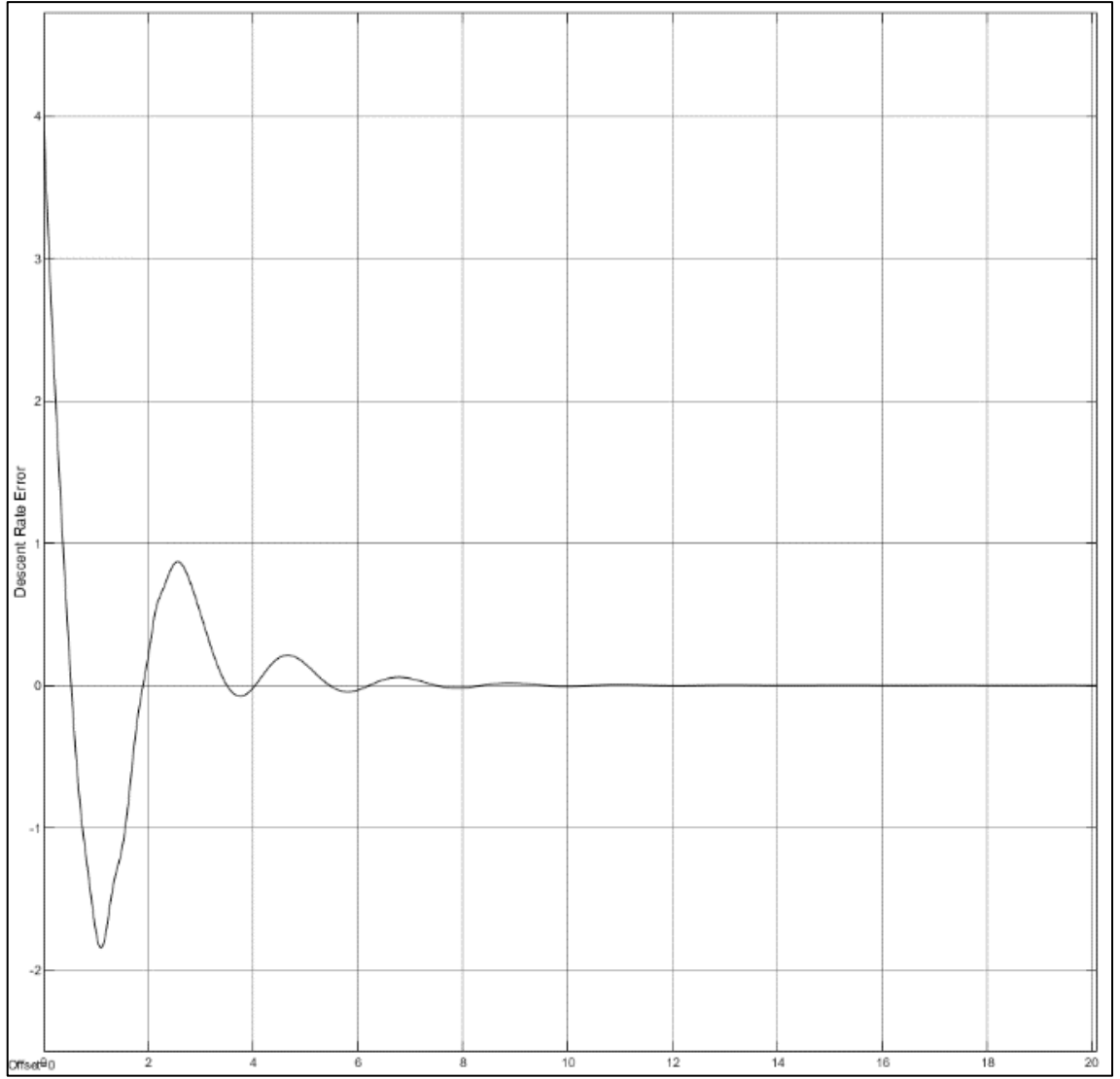
Şekil 8.2. Sapma açısı kontrollü düşüş hata analizi (90 derece – rüzgarsız - rüzgarlı)

İniş sistemi için güzergah planlanması alçalma ve son manevra olarak iki aşamada düşünülmüştür. Alçalma kısmında hedef belirlenen yüksekliğe gelene kadar spiral bir rotayı takip etmesidir. Bunun için sapma açısı hızı kontrolü üzerinde çalışılmıştır.



Şekil 8.3 Sapma hızı kontrollü iniş (rüzgarsız-5m/s +x rüzgarlı)

Son olarak iniş sisteminde düşüş hızı kontrolü güzergah takibinde oldukça önemlidir. Simulasyonda kullanılan parametreler doğrultusunda sistem 4-6 m/s hızlarda iniş sağlamaktadır. Bu nedenle 5 m/s hıza sabitleyecek kontrolcü geliştirilmiş ve sisteme uygulanmıştır.



Şekil 8.4. Düşüş hızı kontrolü hata analizi

8.1. Rota Takibi

Rota planlaması aşamasından sonra PKİS'in oluşturulan rotayı takip etmesi için yörünge takip algoritmalarının geliştirilmesi gerekmektedir. Sabit kanatlı hava araçları için yörünge takibi literatürde sıklıkla çalışılmıştır [66] [67] [68] [69] [70]. Geliştirilen yöntemlerininse doğrusal ve doğrusal olmayan kontrolcüler üzerinde yoğunlaştığı görülmüştür. Doğrusal kontrolcü yöntemlerinden PID [71] [72] ve LQR [73] [74] [75] kontrolcüler en çok kullanılanlar olmuştur. Uygulama kolaylığı ve basitliği PID kontrolcülerini ön plana çıkarsa da PID katsayılarının probleme özgü ayarlanması olası gürültü ve dış etkilere karşı etkisiz bırakmaktadır. LQR tabanlı yöntemlerde ise sistemin lineer modeli kontrolcüyü doğrudan etkilemektedir. Karmaşık sistemlerde lineerize edilmiş model gerçek modelden çok farklı çıktılar verebilir.

Doğrusal olmayan kontrolcülerde ise Model Öngörülü Kontrol büyük bir role sahiptir. Kang ve Hedrick yaptığı çalışmada, sabit kanatlı hava aracının istenilen bir doğruda hareketini doğrusal olmayan model öngörülü kontrol kullanarak yaptılar. Geliştirdikleri online optimizasyon sayesinde, hava aracının dönüş kabiliyeti düşük olmasına rağmen optimal rotaya yakınsıyor [76].

Reinhardt ve Johansen ise geliştirdikleri Doğrusal Olmayan Model Öngörülü kontrolcü ile sabit kanatlı hava aracının yönelim ve hız kontrolünü gerçekleştirmiştir. Elde ettikleri sonuçlar, geliştirdikleri kontrolcünün karmaşık modellerde bile yüksek hesaplama hızına sahip olduğunu göstermektedir. Bu da yöntemi hava araçları için oldukça uygulanabilir kılmıştır [77].

Makine öğrenme tekniklerinin gelişmesi kontrol sistemlerinde de etkili olmuştur. Yapay Sinir Ağı tabanlı modellerden oluşan Model Öngörülü kontrolcüler, dinamik sistem modelinin yapay sinir ağıyla temsiline dayanmaktadır. Zhang ve arkadaşları, geliştirdikleri RNN tabanlı model öngörülü kontrolcü sistemi ile hava aracı sürüsünün formasyon kontrolünü gerçekleştirmişlerdir. Öteleme ve dönme hareketlerini ayırarak geliştirdikleri kontrolcünün stabilite analizlerini göstererek yakınsadığını kanıtlamışlardır [78]. Bir başka uygulamada ise Anavatti ve Garratt, helikopterin yükseklik kontrolünü ileri beslemeli sinir ağı modeline sahip kontrolcü ile sağlamışlardır. Önceden eğitilen modelin ağırlıklarını online olarak ta güncelleyerek modellerinin dinamik olmasını başarmışlardır [79].

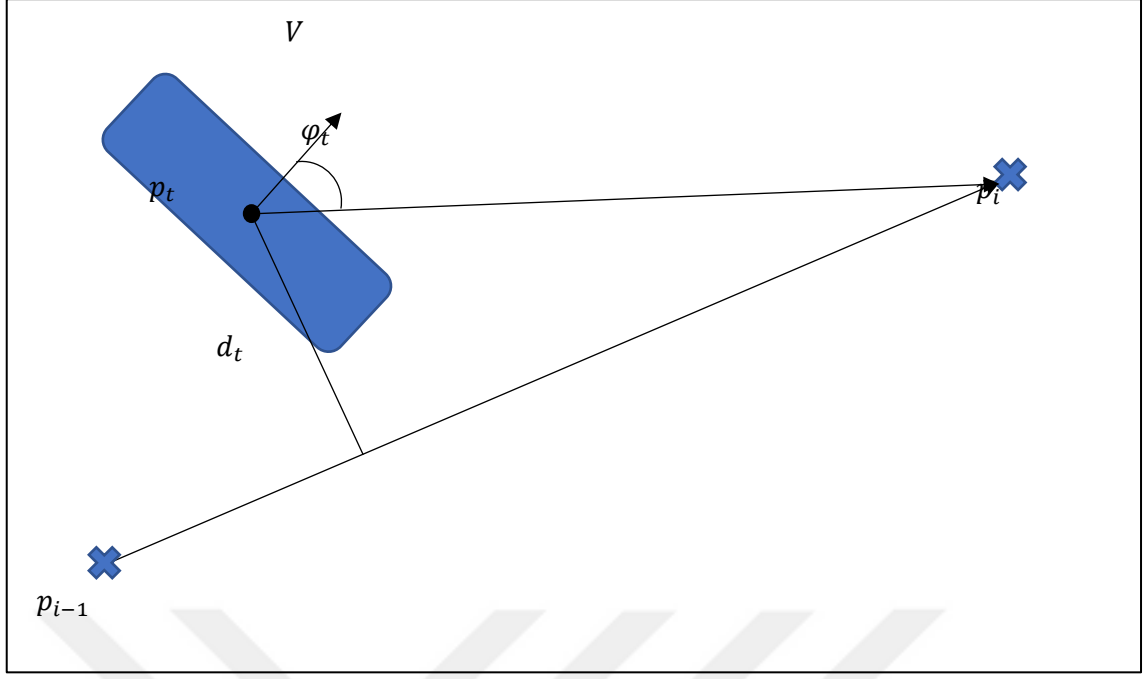
Bu çalışmada sistem NARX Sinir ağı yöntemiyle modellenmiştir. Rota takibinde de bu model kullanılmıştır.

$$u = [\partial \bar{a} \partial \bar{s}] \quad (8.1)$$

$$x = [u \ v \ w \ p \ q \ r] \quad (8.2)$$

$$x(t) = f[x(t-1), x(t-2), \dots, x(t-n_y); u(t), u(t-1), \dots, u(t-n_u)] \quad (8.3)$$

Burada y durum değişkenlerini, x girdileri ve f ise NARX Sinir ağını göstermektedir. Rota takip problemi de doğrusal yörünge üstünde tanımlanmıştır.



Şekil 8.4 Doğrusal Rota

Şekil 4.4 gösterilen rotaya göre takip hatası hesaplanabilir.

$$e_t = \alpha d_t + \beta \varphi_t \quad (8.4)$$

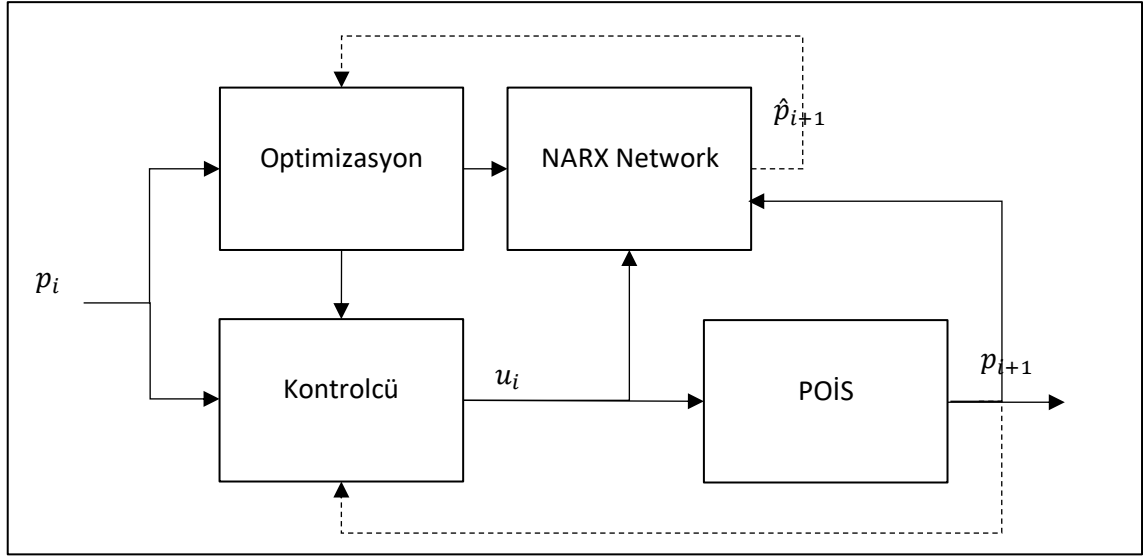
Burada α ve β sapma açısının farkına göre belirlenen parametrelerdir. Problemin çözümü için Model Öngörülü Kontrol yöntemi uygulanmıştır. Probleme göre maliyet fonksiyonu J yazılabilir.

$$J_i = \sum_{j=N_1}^{N_2} e^2(i+j) + \rho \sum_{j=1}^{N_u} \Delta u^2(i+j-1) \quad (8.5)$$

$$\Delta u(i+j-1) = u(i+j-1) - u(i+j-2) \quad (8.6)$$

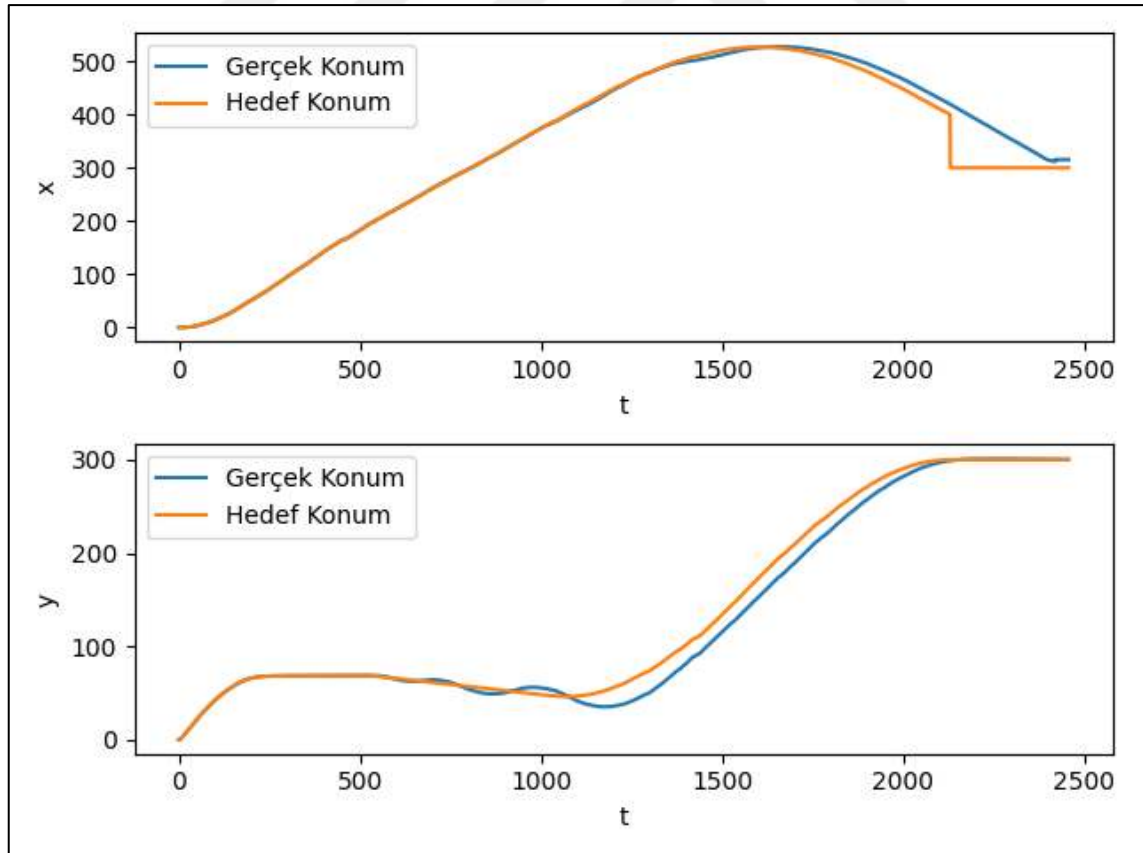
Optimal kontrol girdisinin maliyet fonksiyonu mizinimize etmesi beklenmektedir.

$$u(i) = \text{argmin}(J_i) \quad (8.7)$$

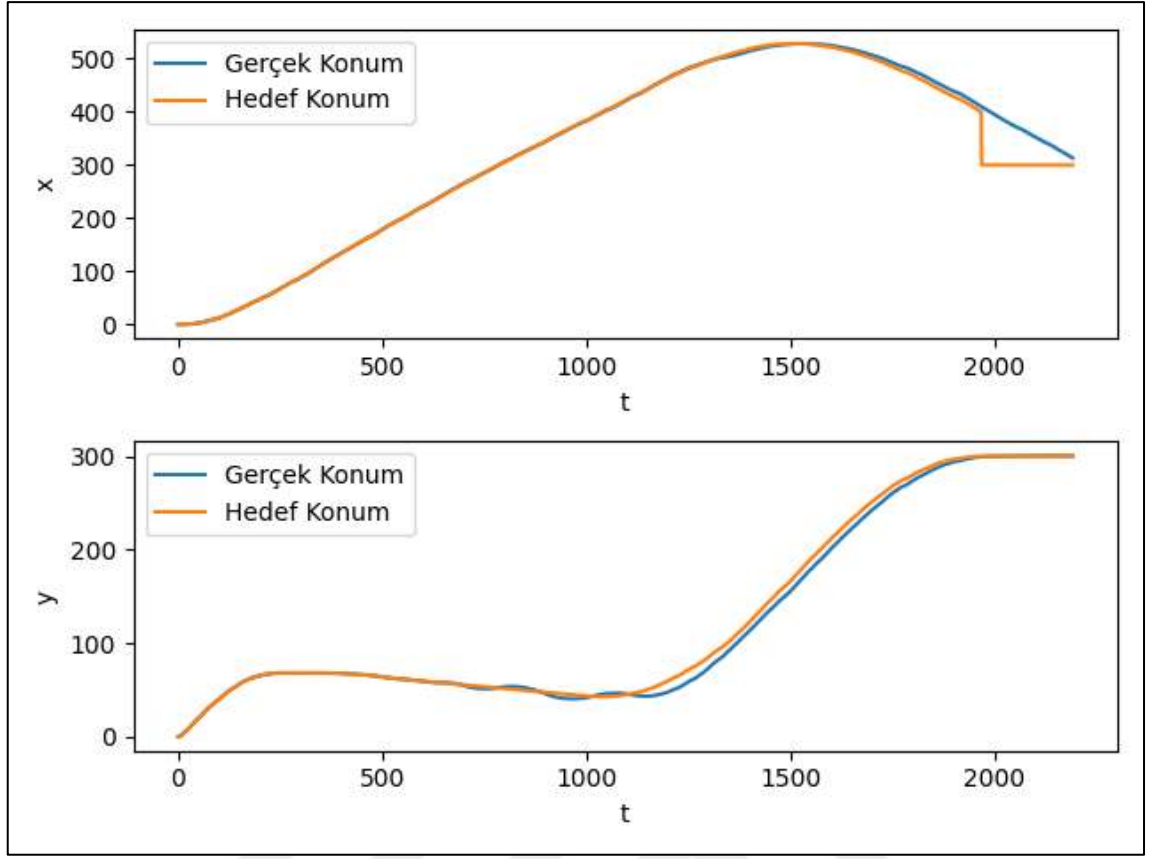


Şekil 8.5 Model Öngörülü Kontrol şematığı

Model eğitiminde kullanılan veri simulasyon üzeridnen rüzgar olmayan ortam kullanılarak sağlanmıştır. Rüzgarsız ortamda PID kontrolcü ve MÖK yakın performanslar göstermiştir. MÖK 8m hata ile 13 m hataya sahip PID kontrolden daha başarılı olmuştur.

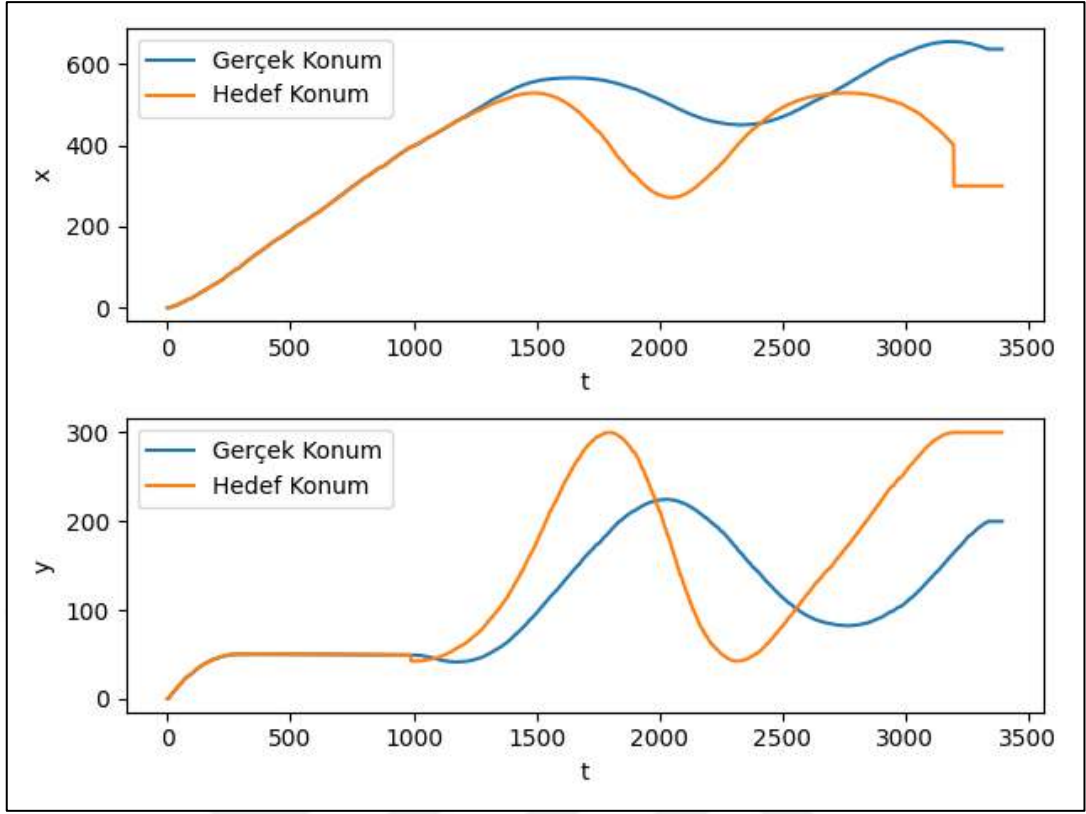


Şekil 8.5 PID Kontrolcü (Rüzgarsız)

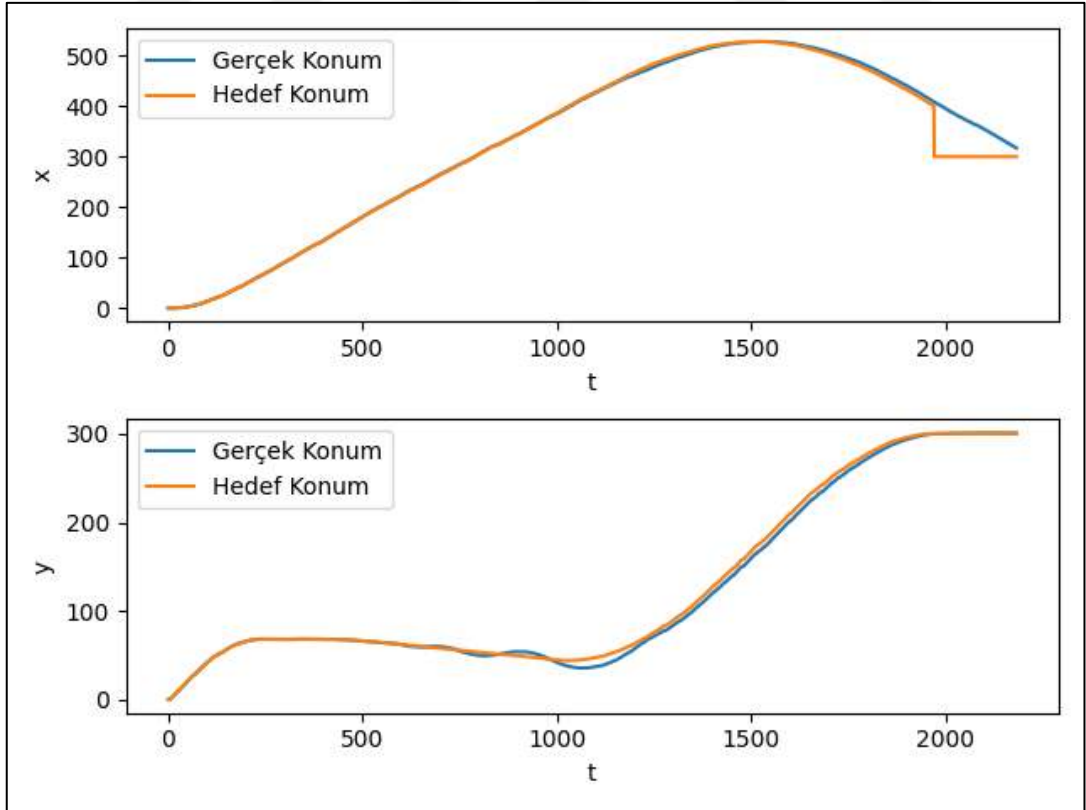


Şekil 8.5 MÖK Kontrolcü (Rüzgarsız)

Rüzgarlı ortamda ise hata oranı oldukça artmış ve hedefe iniş gerçekleştirememiştir. Bu nedenle model öngörülü kontrolcüde kullanılan model çevrimiçi olarak eğitilmiş ve performansı arttırılmıştır.



Şekil 8.5 PID Kontrolcü (Rüzgarlı)



Şekil 8.5 MÖK Çevrimiçi Kontrolcü (Rüzgarlı)

9. SONUÇLAR VE ÖNERİLER

9.1. SONUÇLAR

Bu tez çalışmasında paraşütlü otonom hava aracının deneysel ve simülatif incelemeleri hedeflenmiştir. İlk olarak sistemin aerodinamik özellikleri incelenmiştir. Sistem üzerinde bulunan kontrol frenlerinin aerodinamik olarak etkileşimleri ve yarattığı kuvvet-moment farkları sistem üzerindeki kontrollere alt yapı oluşturmuştur. Simülasyon çalışmalarına geçmeden önce sistemin matematik modeli üzerindeki araştırmalar serbestlik derecelerine göre farklılık göstermiştir. Paraşütün ve görev yükünün ayrı olarak düşünüldüğü ya da bu ikisinin tek bir rijit cisim olarak varsayıldığı modeller bulunmaktadır. Simülasyon çalışmalarında sistemin tek bir rijit cisim olarak düşünüldüğü 6 serbestlik dereceli model kullanılmıştır.

MATLAB Simulink kullanılarak 6 serbestlik dereceli model bilgisayar ortamına taşınmıştır. Araca uygulanan kuvvet ve momentler bu platformda eklenmiştir. Burada aracın belirli bir rotada seyrüsefer edebilmesi için gerekli sapma açısı ve sapma hızı kontrolcileri geliştirilmiştir. Uçuş stabilitesi için de yunuslama ve yuvarlanma açıları takip edilmiştir. Burada elde edilen sonuçlar stabil bir uçuşun ilk adımı olarak ön plana çıkmıştır. Daha sonra sistem simülasyonlarına Döngüde Donanım ve Döngüde yazılım özellikleri eklemek için bu modeller Gazebo-ROS ortamına entegre edilmiştir. Burada oto pilot yazılımı olarak açık kaynaklı PX4 kullanılmıştır. Gazebo ortamı için gerekli sdf dosyası sistem modeli için üretilmiştir. Böylece sistemin katı modeli ortama eklenmiştir. Daha sonra sistem üzerindeki kuvvet ve momentlerin hesaplanıp ilgili linklere aktarıldığı plugin hazırlanmıştır. Gazebo dinamik motorunun da yardımıyla sistem matematiksel olarak tamamlanmış olur. Oto pilot yazılımının entegresi ise ilgili ROS düğümleri kullanılarak yapılmıştır. Burada geliştirilen yazılımlar ile sistem üzerine ilgili sensörler eklenmiş ve motor kontrol sinyalleri üretilerek dinamik modele aktarılmıştır. Ortam ve yer istasyonu ile iletişim MAVLink denilen haberleşme protokolü ile yapılmıştır. Böylece geliştirilen simülasyon ortamında sistem durumları gerçek zamanlı olarak gözlenebilmekte ve araca istenilen kontrol girdisi gerçek zamanlı olarak gönderilebilmektedir. Bu ortamda yapılan sanal uçuşlar ile sanal uçuş verisi üretilmiştir.

Bir sonraki aşamada deneysel çalışmalar için mekanik ve elektronik sistem tasarımları geliştirilmiştir. Mekanik tasarım temel olarak frenlerin, fren motorlarının bulunduğu

paraşütlü bir tasarımıdır. Sistem uçuş testlerinin herhangi bir bırakılma aracından bağımlı olabilmesi içinde itki motoru takılmıştır. Elektronik tasarımın ana merkezinde PX4 yazılımlarının koşulduğu Pixhawk donanımı bulunmaktadır. GPS, barometre, haberleşme birimi ve servo motorlar Pixhawk üzerine entegre edilmiştir. İlgili kontrolcü ve görev yazılımlarının koşulması için de gömülü bilgisayar olarak Raspberry PI4 kullanılmıştır. Güç sistemi olarak LiPo batarya tercih edilmiştir. Sistem ilk olarak manuel uçuşa geçirilmektedir. İstenilen yüksekliğe çıkıldığında itki motoru kapatılmış ve serbest düşüş gerçekleştirilmiştir. Buradan elde edilen gerçek uçuş verisi düzenlenerek hazırlanmıştır.

Üretilen sanal ve gerçek uçuş verisi sistem tanımlama çalışmalarının girdisi olarak kullanılmıştır. İlk olarak sanal uçuş verisi kullanılarak NARX sinir ağı modeli eğitilmiştir. Burada eğitim algoritmalarının, nöron ve katman sayılarının model performanslarına etkisi gözlemlenmiştir. Hızlı eğitim süresine rağmen SCG, LM ve BR'ye göre daha kötü performans göstermiştir. Bu aşamada gizli katman ve içindeki nöron sayısı değiştirilerek sonuçlar incelenmiştir. Eğitim algoritması BR olarak sabit bırakılmıştır. Bunun yanında, tek aktımlı ve 5 nöronlu oluşan modelin en iyi performansı verdiği gözlemlenmiştir. Geliştirilen modellerin sınırlarını belirlemek için ağırlık ve aerodinamik katsayıların modeller üzerindeki etkileri gözlemlenmiştir. Kontrol girdilerinin kuvvet ve moment üzerindeki etkisini belirleyen aerodinamik katsayılar seçilmiştir. Seçilen parametreler $\pm 10\%$ değiştirilerek hata oranları gözlemlenmiş ve tek gizli katman ve 5 nöronlu oluşan model yine en iyi performansa sahiptir. Bu çalışmada son olarak gerçek uçuş verisi üzerindeki model performansları incelenmiştir. Daha önce yapılan iki adet uçuş verisinden bir tanesi model eğitiminde kullanılmıştır. Eğitilen modelin testleri ise eğitimde kullanılmayan uçuş verisi üzerinden yapılmıştır. Sistem modelinin artırılması için de çevrimiçi eğitim yöntemi eklenmiş ve model parametreleri uçuş sırasında güncellenmiştir. Çevrimiçi eğitim yönteminin model performansına olumlu katkıda bulunduğu görülmüştür.

Sistemin istenilen hedef konuma ulaşabilmesi için rota planlama algoritması gerekmektedir. Sistem pozisyonu, hedef konum ve rüzgâr durumu göz önünde bulundurularak geometrik olarak rota planlaması yapılmıştır. Burada literatürde sıklıkla kullanılan optimal kontrol teoremi baz alınmıştır. Belirlenen rotanın takibi için ise PID ve model öngörülü kontrolcü yöntemleri incelenmiştir. PID kontrolcü rüzgarsız ortamlarda iyi performans göstermesine karşın rüzgârlı ortamda oldukça zorlanmıştır. Model öngörülü kontrolde kullanılan model de rüzgarsız ortamda eğitildiği için rüzgarsız ortamda iyi

performans gösterip rüzgar olan ortamda zorlanmıştır. Bunun çözümü olarak ise model uçuş sırasında çevrimiçi olarak eğitime devam ettirilmiş ve başarılı sonuca ulaşılmıştır. . Bu doğrultuda, eğitilen model farklı çevresel durumların ve sistem ağırlığına ait uçuş verisi üzerinde hücum açısında ortalama 12 derecelik tahmin ortalaması tutturmuştur. Gerçek zamanlı kontrolcü çalışmaları yapılarak sistemin istenilen hedef konuma inişi rüzgarsız ortamda 3m, rüzgarlı ortamda ise 7m hata ile gerçekleştirilmiştir.

9.2. ÖNERİLER

Gerçekleştirilen simülasyon çalışmalarında paraşüt ve kargo rijit bir çizim olarak düşünülmüştür. Gerçek sistemde bağımsız olan bu sistemlerin simülasyonda kullanılan 6 serbestlik dereceli modeller yerine daha yüksek serbestlikli modeller kullanılması hesaplama karmaşıklığı getirirse de simülasyon çalışmalarında uygun olacağı değerlendirilmektedir.

İniş sisteminde kullanılan “Ram-air” paraşüt tipi üretimi zor bir paraşüt tipidir. Yüksek kargo ağırlıklarının hedeflenmediği sistemler için “Cruciform” tipi paraşütler kullanılabilir. Bu tip paraşütlerin kolay üretimi sistem testlerinde farklı paraşütlerin kullanımına da olana sağlayacaktır.

Rota planlama algoritması, çevresel ve sistem durumlarına göre değişiklik gösterecek biçimde geliştirilebilir. Çalışmada tek bir derin öğrenme metodu kullanılmıştır. Birden fazla yöntemler kullanılarak performans ve işlem yükü karşılaştırmaları yapılabilir.

10.KAYNAKÇA

- [1] W. H. Eilertson, "Gliding Parachutes for Land Recovery of Space Vehicles", NASA-CR-108990, Washington D.C., 1969.
- [2] J. S. Lingard, "Precision Aerial Delivery/Ram-Air Parachute System Design", Proceedings of the 13th AIAA ADS Conference and Seminar, Washington D. C., 1995.
- [3] K. Stein, T. Tezduyar, S. Sathe, M. Senga, C. Ozcan, T. Soltys, V. Kumar, R. Benney ve R. Charles, "Simulation of Parachute Dynamics During Control Line Input Operations", 17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, California, 2003.
- [4] C. Ibos, C. Lacroix, A. Goy ve P. Bordenave, "Fluid-structure simulation of 3d ram air parachute with sinpa software", 15th Aerodynamic Decelerator Systems Technology Conference, AIAA, Toulouse, France, 1999.
- [5] Y. Mosseev, "Software tools for the paraglider computer-aided design code", 16th AIAA Aerodynamic Decelerator Systems Technology Conference, AIAA, Boston, Massachusetts, 2001.
- [6] J. M. Ginn, I. G. Clark ve R. D. Braun, "Parachute Dynamic Stability and the Effects of Apparent Inertia", AIAA Atmospheric Flight Mechanics Conference, Atlanta, GA, 2014.
- [7] V. Dobrokhodov, O. Yakimenko ve C. Junge, "Simulink implementation of the 6DOF model of controlled circular parachute", AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, California, 2002.
- [8] P. A. Mortaloni, O. A. Yakimenko, V. N. Dobrokhodov ve R. M. Howard, "On the development of a 6DOF model of a low-aspect-ratio parafoil delivery system", 17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Monterey, California, 2003.

- [9] S. Redkar ve B. Fiedler, "Control Design and Simulation Framework for an Autonomous Paramotor UAV", ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, St. Louis, Missouri, USA, 2022.
- [10] M. Volodymyr, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland ve G. Ostrovski, "Human Level Control Through Deep Reinforcement Learning," Nature, cilt 518, no. 7540, pp. 529-533, 2015.
- [11] J. Fan, Z. Wang, Y. Xie ve Z. Yang, "A Theoretical Analysis of Deep Q-Learning", Proceedings of 2020 Learning for Dynamics and Control, 2020.
- [12] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra ve M. Riedmiller, "Deterministic Policy Gradient Algorithms", 31st International Conference on Machine Learning, Beijing, China, 2014.
- [13] Y. Zhang, Y. Zhang ve Z. Yu, "Path following control for UAV Using Deep Reinforcement Learning Approach", Guidance, Navigation and Control, cilt 1, no. 1, p. 2150005, 2021.
- [14] C. Tang ve Y.-C. Lai, "Deep Reinforcement Learning Automatic Landing Control of Fixed Wing Aircraft Using Deep Deterministic Policy Gradient", International Conference on Unmanned Aircraft Systems, Athens, Greece, 2020.
- [15] J. Zhao, H. Liu, J. Sun, K. Wu, Z. Cai, Y. Ma ve Y. Wang, "Deep Reinforcement Learning-Based Ent-to-End Control for UAV Dynamic Target Tracking", Biomimetics, cilt 7, p. 197, 2022.
- [16] S. Lucia ve B. Karg, "A deep learning-based approach to robust nonlinear model predictive control", IFAC-PapersOnLine, cilt 51, no. 20, pp. 511-516, 2018.
- [17] T. Salzmann, E. Kaufmann, J. Arrizabalaga, M. Pavone, D. Scaramuzza ve M. Ryll, "Neural-MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms", 2022. [Online]. Available: arxiv.org/pdf/2203.07747v1.pdf.

- [18] N. Slegers ve M. Costello, “Model Predictive Control of a Parafoil”, *Journal of Guidance, Control and Dynamics*, cilt 28, no. 4, pp. 816-821, 2005.
- [19] Pope, *Basic Wing and Aerofoil Theory*, New York: McGraw-Hill, 1951.
- [20] S. F. Hoerner ve H. V. Borst, *Fluid Dynamic Lift*, Brick Town, NJ: L. A. Hoerner, 1985.
- [21] S. F. Hoerner, *Fluid-Dynamic Drag*, Midland Park, NJ, 1965.
- [22] T. Jann, “Aerodynamic Model Identification and GNC Design for the Parafoil-Load System ALEX”, *Proceedings of the 16th AIAA Aerodynamic Decelerator Systems Technology Conference*, Reston, V.A, 2001.
- [23] H. F. Altmann, “Numerical Simulation of Parafoil Aerodynamics and Dynamic Behavior”, *Proceedings of the 20th AIAA Aerodynamic Decelerator Systems Technology Conference*, Reston, VA. , 2009.
- [24] B. L. Stevens ve F. L. Lewis, *Aircraft Control and Simulation*, John Wiley and Sons, Inc., 2. edition, 2003.
- [25] P. S. Lissaman ve G. J. Brown, “Apparent mass effects on parafoil dynamics”, *American Institute of Aeronautics and Astronautics*, 1993.
- [26] O. A. Yakimenko, *Precision Aerial Delivery Systems: Modeling, Dynamics, and Control*, Monterey, California: American Institute of Aeronautics and Astronautics, 2015.
- [27] Dronecode-Project: Mavlink developer guide (2018), mavlink.io/en/.
- [28] [Online]. Available: docs.px4.io/master/en/flight_stack/controller_diagrams.html.
- [29] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler ve A. Y. NG, “ROS: An open source robot operating system.”, *ICRA workshop on Open Source Software*, Kobe, Japan, 2009.
- [30] [Online]. Available: wiki.ros.org/mavros.
- [31] [Online]. Available: docs.px4.io/master/en/simulation/.

- [32] [Online]. Available: gazebo-sim.org.
- [33] P. Hamel ve R. Jategaonkar, “Evolution of Flight Vehicle System Identification”, *Journal of Aircraft*, cilt 33, no. 1, pp. 9-28, 1996.
- [34] T. Jann ve G. Strickert, “System Identification of a Parafoil-Load Vehicle - Lessons Learned”, 18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Reston, VA, 2005.
- [35] J. A. Grauer, “Real-time parameter estimation using output error”, AIAA Atmospheric Flight Mechanics Conference, 2014.
- [36] F. Heimes, G. Zaleski, W. Land ve M. Oshima, “Traditional and evolved dynamic neural networks for aircraft simulation”, IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 1997 .
- [37] F. Saghafi ve B. Momhed-Heravi , “Identification of Aircraft Dynamics, Using Neural Network Simultaneous Optimization Algorithm”, European Simulation Conference, Porto, Portugal, 2005.
- [38] J. Harris, F. Arthurs, J. V. Henrickson ve J. Valasek, “Aircraft system identification using artificial neural networks with flight test data”, International Conference on Unmanned Aircraft Systems (ICUAS), 2016.
- [39] K. Narendra ve K. Parthasarathy, “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, cilt 1, no. 1, p. 4–27, 1990.
- [40] J. Phan, J. N. Juang ve D. C. Hyland, “On neural networks in identification and control of dynamic systems”, National Aeronautics and Space Administration, Tech. Rep. NASA TM-107702, 1993.
- [41] G. Valmorbida, L. Wen-Chi ve F. Mora-Camino, “A neural approach for fast simulation of flight mechanics”, the 38th Annual Simulation Symposium (ANSS’05), 2005.
- [42] Z. Hu ve S. Balakrishnan, “Parameter estimation in nonlinear systems using hopfield neural networks”, *Journal of Aircraft*, cilt 42, no. 1, p. 41–53, 2005.

- [43] D. J. Linse ve R. F. Stengel, "Identification of aerodynamic coefficients using computational neural networks", *Journal of Guidance, Control, and Dynamics*, cilt 16, no. 6, p. 1018–1025, 1993.
- [44] V. Puttige ve S. Anavatti, "Real-time neural network based online identification technique for a uav platform", 2006 International Conference on Computation Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), 2006.
- [45] S. Kamasaldan ve A. Ghandakly, "A neural network parallel adaptive controller for fighter aircraft pitch-rate tracking", *IEEE Trans Instrum Meas*, cilt 60, no. 1, pp. 258-267, 2011.
- [46] Savran, R. Tasaltin ve Y. Becerikli, "Intelligent adaptive nonlinear flight control for a high performance aircraft with neural networks", *ISA Transaction*, cilt 45, no. 2, pp. 225-247, 2006.
- [47] R. A. Hess, "On the use of back propagation with feed forward neural networks for the aerodynamic estimation problem", *Flight Simulation and Technologies, Guidance, Navigation, and Control and Co-Located Conferences*, Monterey, CA, 1993.
- [48] J. R. Raol ve R. V. Jategaonkar, "Aircraft parameter estimation using recurrent neural networks: a critical appraisal", 20th Atmospheric Flight Mechanics Conference, Guidance, Navigation, and Control and Co-located Conferences, Baltimore, MD, USA, 1995.
- [49] Roudbari ve F. Saghafi, "Intelligent Modeling and Identification of Aircraft Nonlinear Flight Dynamics", *Chinese Journal of Aeronautics*, 2014.
- [50] S. A. Bagherzadeh, "Nonlinear aircraft system identification using artificial neural networks enhanced by empirical mode decomposition", *Aerospace Science and Technology*, pp. 155-171, 2018.
- [51] P. Goyal ve P. Benner, "LQResNet: A Deep Neural Network Architecture for Learning Dynamic Processes", arXiv:2103.02249.

- [52] Z. Chen ve D. Xiu, “On Generalized Residual Network for Deep Learning of Unknown Dynamical Systems”, *Journal of Computational Physics*.
- [53] [Online]. Available: docs.px4.io/master/en/development/development.html.
- [54] P. J. Werbos, “Beyond Regression: New Tools for Prediction and Analysis in The Behavioral Sciences”, Ph.D. Dissertation, Harvard University, Applied Mathematics, Boston, MA, 1974.
- [55] G. Lera ve M. Pinzolas, “Neighborhood based Levenberg-Marquardt algorithm for neural network training”, *IEEE Trans. Neural Networks*, cilt 13, no. 5, p. 1200–1203, 2002.
- [56] F. D. Foresee ve T. H. Martin , “Gauss-Newton approximation to Bayesian learning”, the International Joint Conference on Neural Networks, 1997.
- [57] M. Møller, “A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning”, *Neural Networks*, cilt 6, pp. 525-533, 1993.
- [58] S. Gouravaraju, J. Narayan, R. Sauer ve S. Gautam, 2020.
- [59] J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory”, *Numer. Anal. Lect. Notes Math*, 1978.
- [60] D. Carter, S. George, P. Hattis, M. McConley, S. Rasmussen, L. Singh ve S. Tavan, “Autonomous Large Parafoil Guidance, Navigation and Control System Design Status”, 19th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Reston, VA, 2007.
- [61] D. Carter, S. George, P. Hattis, L. Singh ve S. Tavan, “Autonomous Guidance, Navigation, and Control of Large Parafoils”, 18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Reston, VA, 2005.
- [62] D. Carter, L. Singh, L. Wholey, S. Rasmussen, T. Barrows, S. George, M. McConley, C. Gibson, S. Tavan ve B. Bagdonovich, “Band-Limited Guidance and Control of Large Parafoils”, *Proceedings of the 20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, Reston, VA, 2009.

- [63] R. J. Cleminson, "Path Planning for Guided Parafoils: An Alternative Dynamic Programming Formulation", 22nd AIAA Aerodynamic Decelerator Systems Technology Conference, AIAA, Reston, VA, 2013.
- [64] J. Wegereef ve H. Wegereef, "SPADES: A Parafoil Delivery System for Payloads Until 200 Kg", SPADES: A Parafoil Delivery System for Payloads Until 200 Kg", Proceedings of the 17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, AIAA, Reston, VA, 2003.
- [65] Kaminer ve O. Yakimenko, "Development of Control Algorithm for the Autonomous Gliding Delivery System", 17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, Monterey, California , 2003.
- [66] Kaminer, "Trajectory tracking control for autonomous vehicles: an integrated approach to guidance and control", Journal of Guidance, Control and Dynamics, cilt 21, no. 1, pp. 29-38, 1998.
- [67] W. Ren ve R. W. Beard, "Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints", IEEE Transactions on Control Systems Technology, cilt 12, no. 5, pp. 706-716, 2004.
- [68] W. Ren, "Trajectory tracking control for a miniature fixed-wing unmanned air vehicle", International Journal of Systems Science, cilt 38, no. 4, pp. 361-368, 2007.
- [69] L. Sonneveldt, R. van Oort, P. Q. Chu ve J. A. Mulder, "Nonlinear adaptive trajectory control applied to an F-16 model", Journal of Guidance, Control, and Dynamics, cilt 32, no. 1, pp. 26-39, 2009.
- [70] B. C. Low, "A trajectory tracking control design for fixed-wing unmanned aerial vehicles", IEEE International Conference on Control Applications, 2010.
- [71] M. Gros, A. Schoettl ve W. Fichter, "Spline and OBB-Based Path-Planning for Small UAVs with the Finite Receding-Horizon Incremental-Sampling Tree Algorithm", AIAA Guidance, Navigation, and Control (GNC) Conference, 2013.
- [72] Rhee, S. Park ve C. Ryoo, "A Tight Path Following Algorithm of an UAS Based on PID Control", SICE Annual Conference, Piscataway, NJ, 2010.

- [73] A. Ashari, “Flight Trajectory Control System on Fixed Wing UAV using Linear Quadratic Regulator”, *International Journal of Engineering and Technical Research*, cilt 8, 2019.
- [74] S. Anjali, A. Vivek ve J. L. Nandagopal, “Simulation and Analysis of Integral LQR Controller for Inner Control Loop Design of a Fixed Wing Micro Aerial Vehicle (MAV)”, *Procedia Technology*, cilt 25, p. 76–83, 2016.
- [75] L. Chrif ve Z. M. Kadda, “Aircraft Control System Using LQG and LQR Controller with Optimal Estimation-Kalman Filter Design”, *Procedia Eng*, cilt 80, p. 245–257, 2014.
- [76] Y. Kang ve J. K. Hedrick, “Linear Tracking for a Fixed-Wing UAV Using Nonlinear Model Predictive Control”, *IEEE Transactions on Control Systems Technology*, cilt 17, no. 5, p. 1202–1210, 2009.
- [77] D. Reinhardt ve T. A. Johansen, “Control of Fixed-Wing UAV Attitude and Speed based on Embedded Nonlinear Model Predictive Control”, *IFAC-PapersOnLine*, cilt 54, no. 6, 2021.
- [78] B. Zhang, X. Sun ve S. Liu , “Recurrent Neural Network-Based Model Predictive Control for Multiple Unmanned Quadrotor Formation Flight”, *International Journal of Aerospace Engineering*, 2019.
- [79] M. Samal, S. G. Anavatti ve M. Garratt, “Neural Network Based Model Predictive Controller for Simplified Heave Model of an Unmanned Helicopter”, *3rd International conference on Swarm, Evolutionary, and Memetic Computing*, Bhubaneswar, India, 2012.

EKLER

EK 1: Pixhawk Özellikleri

İşlemci	• FPU ile 32-bit ARM Cortex M4 çekirdeği • 168 Mhz/256 KB RAM/2 MB Flash • 32-bit failsafe co-processor
Sensörler	• Ana ivmeölçer ve jiroskop olarak MPU6000 • ST Micro 16-bit jiroskop • ST Micro 14-bit hızlandırıcı/mıknatıs (manyetometre) • MEAS barometre
Güç	• Otomatik failover ile ideal diyot kontrolcüsü • Yüksek güç (7 V) ve yüksek akım uyumlu servo rail • Tüm dış cihaz çıkışları aşırı akım koruma altında, tüm girişler ESD koruma altında
Arayüzler	• 5 adet UART seri port, 1 adet yüksek güç çıkışı olan, 2 adet HW akış kontrolü olan • Spektrum DSM/DSM2/DSM-X uydu girişi • Futaba S.BUS girişi (çıkış henüz gerçekleştirilmedi) • PPM toplam sinyali • RSSI (PWM veya gerilim) girişi • I2C, SPI, 2x CAN, USB • 3.3V ve 6.6V ADC girişleri
Ölçüler	• Ağırlık 38 g (1.3 oz) • Genişlik 50 mm (2.0") • Yükseklik 15.5 mm (.6") • Uzunluk 81.5 mm (3.2")