

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ALTI ROTORLU BİR HELİKOPTERDE (HEXACOPTER) DENGE VE YÜKSEKLİK
KONTROLÜ İÇİN YENİ BİR KONTROL ALGORİTMASI**

TARIK VELİ MUMCU

**DOKTORA TEZİ
KONTROL VE OTOMASYON MÜHENDİSLİĞİ ANABİLİM DALI
KONTROL VE OTOMASYON MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN
DOÇ. DR. KAYHAN GÜLEZ**

İSTANBUL, 2013

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ALTI ROTORLU BİR HELİKOPTERDE (HEXACOPTER) DENGELİK VE YÜKSEKLİK KONTROLÜ İÇİN YENİ BİR KONTROL ALGORİTMASI

Tarık Veli MUMCU tarafından hazırlanan tez çalışması 11.06.2013 tarihinde aşğıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Kontrol ve Otomasyon Mühendisliğı Anabilim Dalı' nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Kayhan GÜLEZ

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Kayhan GÜLEZ

Yıldız Teknik Üniversitesi

Prof. Dr. Rahmi Güçlü

Yıldız Teknik Üniversitesi

Doç. Dr. Olcay Kurşun

İstanbul Üniversitesi

Prof. Dr. Erkan İmal

Fatih Üniversitesi

Yrd. Doç. Dr. Janset Daşdemir

Yıldız Teknik Üniversitesi

Bu alıřma, Yıldız Teknik Üniversitesi Bilimsel Arařtırma Projeleri Koordinatörlüğü' nün 2010-04-02 KAP05 numaralı projesi ile desteklenmiřtir.

ÖNSÖZ

Katkılarından dolayı danışman hocam Doç. Dr. Kayhan GÜLEZ' e, çalışmalarım süresince desteklerini esirgemeyen tüm çalışma arkadaşlarıma ve aileme teşekkür ederim.

Nisan 2013

Tarık Veli MUMCU

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	viii
KISALTMA LİSTESİ	ix
ŞEKİL LİSTESİ.....	x
ÇİZELGE LİSTESİ	xiii
ÖZET.....	xiv
ABSTRACT	xvi
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	9
1.3 Orjinal Katkı	9
BÖLÜM 2	
SON YILLARDA LİTERATÜRDE YERALAN BAŞLICA VTOL ÇALIŞMALARI	11
2.1 Draganflyer	11
2.2 Avustralya Ulusal Üniversitesi ' nin Tasarımı	12
2.3 Standford Üniversitesi' nin Tasarımı Starmac.....	12
2.4 Ecole Polytechnique Samir Bouabdallah' ın Tasarımı	13
2.5 Markus Kemper' in Tezi.....	14
2.6 Glenn P. Tournier' in Tezi (MIT)	14
2.7 Charles A. Clifton' in Tezi.....	15

BÖLÜM 3

HEXACOPTER DONANIM ve YAZILIM MİMARİSİNİN OLUŞTURULMASI	17
3.1 Hexacofter Gövdesi	17
3.2 Üst Seviye Kontrol	18
3.3 Alt Seviye Kontrol	20
3.4 Genel Donanım Yapısı	20
3.5 Hexacofter Donanım Bileşenleri	21
3.5.1 STM32F4 Discovery Kartı	21
3.5.2 Sparkfun Dokuz Serbest Dereceli Atalet Ölçüm Ünitesi	22
3.5.3 XBEE Kablosuz Haberleşme Modülleri	26
3.5.4 SRF08 Ultrasonik Mesafe Ölçüm Modülü	27
3.5.5 FT232 Seri-USB Çevirici	27
3.5.6 Robbe 2.4 GHz Kumanda Alıcısı ve Futaba RC Kumanda	28
3.5.7 Mikrokofter Elektronik Hız Kontrol Üniteleri (ESC)	29
3.6 Hexacofter Yazılım Mimarisi.....	29
3.6.1 Hexacofter Otopilot Yazılımı	29
3.6.2 Haberleşme Görevi.....	30
3.6.3 Otopilot Görevi.....	31
3.6.4 Sensör Görevi	31
3.6.5 Sistem İzleme Görevi.....	32
3.6.6 Hexapilot Arayüz Yazılımı	32
3.6.7 Hexapilot Yazılım Bileşenleri Yapısı.....	34
3.6.7.1 FreeRTOS Gerçek Zamanlı İşletim Sistemi	34
3.6.7.2 Hexacofter Otopilot Haberleşme Protokolü	35
3.6.7.3 Otopilot Haberleşme Sürücülerini	36
3.6.7.4 Otopilot IMU Sürücüsü.....	37
3.6.7.5 SRF08 Ultrasonik Mesafe Ölçüm Modülü Sürücüsü	37
3.6.7.6 RC Kumanda Alıcısı Sürücüsü	37
3.6.7.7 Hexacofter Arayüz Yazılımı	38
3.6.7.8 Yazılım Mimarisinde Gelecekte Planlanan Çalışmalar	38

BÖLÜM 4

PID ve MODEL ÖNGÖRÜLÜ KONTROL	39
4.1 PID Denetleyicinin Hexacofter Uygulanması.....	39
4.2 Basitleştirilmiş Sistem Dinamikleri	45
4.3 Hexacofter PID Çıktıları	48
4.4 Model Öngörülü Kontrol	50
4.4.1 Çoklu Rotorlu Helikopterlerde Hareketin Modellenmesi	50
4.4.2 Değişken Gecikme Zamanlı MPC İle Kontrol	50
4.4.3 Değişken Gecikme Zamanlı MPC:.....	51
4.4.4 Sınır Şartlı Bir MPC Problemi İçin Matematiksel Yöntemler	52
4.4.5 Gecikmeye Bağlı Suboptimal Çözümün Tek Bir MPC İçin Tanımı:	53
4.4.6 Optimizasyon Problemi İçin Sınır Şartlarının Tanımlanması	56
4.4.7 Problemin Path Following (Yol Takibi) Metodu İle Çözümü	58
4.4.8 Quadratic Programlama' nın Yakınsaması (Convergence).....	63

4.4.9	Gecikme Zamanlı MPC' nin Bir Mikrodenetleyicide Uygulanması.....	66
4.4.10	İkinci Dereceden Bir Sistem İle Yapılan Örnek Çalışma.....	67
4.4.11	Önerilen Denetleyici Modelinin Hexacoptere Uygulanması.....	70

BÖLÜM 5

EŞ ZAMANLI KONUMLANDIRMA VE HARİTALAMA	73
5.1 Eş Zamanlı Konumlandırma ve Haritalama	73
5.2 Başlıca Algoritma Yapıları	75
5.3 Gmapping (Grid Mapping)	82
5.4 Gmapping Geliştirilmiş Öneri Dağılım Fonksiyonu ve Hesaplanması	86
5.5 Yeniden Örneklemeye.....	88
5.6 Bütün Algoritmaya Yeniden Bir Bakış.....	89
5.7 Algoritmanın Hexacoptere Uygulanması ve Saha Çıktıları.....	91

BÖLÜM 6

SONUÇ VE ÖNERİLER	93
KAYNAKLAR.....	97
EK-A.....	102
PMBLDC Motor Parametreleri.....	102
A-1 Motor Parametreleri.....	102
EK-B.....	108
Hexacopter Hareket Denklemleri	108
B-1 Çoklu Rotorlu Helikopter Modelinin Çıkarılması	108
ÖZGEÇMİŞ.....	118

SİMGE LİSTESİ

A	Durum Uzay Sistem Matrisi
A_C	Sürekli Zaman Durum Uzay Sistem Matrisi
B	Durum Uzay Sistem Matrisi
B_C	Sürekli Zaman Durum Uzay Sistem Matrisi
C	Sistem Matrisleri
ϕ, Γ_1	Ayrık Sistem Matrisleri
Γ_0	Ayrık Sistem Matrisi
h	Örnekleme Zamanı
H, J	Ayrık Sistem Matrisleri
H_p	Tahmin ufku
H_u	Kontrol ufku
I_C	Başlangıç değerleri
J_C	Maliyet fonksiyonu
m_i	i. yer işaretinin yerini ifade eden vektör
N	Ağırlık Matrisi Çapraz Terim
Q	Ağırlık Matrisi İlk Terimi
Q_C, N_C	Ağırlık Matrisi Katsayıları
R	Ağırlık Matrisi İkinci Terim
R_C	Ağırlık Matrisi Katsayısı
τ	örnekleme zamanı aralığında değişen zaman gecikmesi
u	Kontrol sinyali
u_k	k-1 zamanında uygulanan, k zamanına ve aracı X_k durumuna getirmeyi amaçlayan kontrol vektörü
x	Durum Uzay Durum Vektörü
X_k	Aracın konumunu ve yönelimi
x_p	Öngörülen Durum Vektörü
z_{ik}	i yer işaretinden k zamanında aracın konumundan alınan gözlem

KISALTMA LİSTESİ

AMR	Anisotropic Magneto Resistance
ARM	Acorn RISC Machine
CRC	Cyclic Redundancy Checksum
DAC	Digital to Analog Converter, Dijital/Analog Çevirici
DoF	Degrees of Freedom, Serbestlik Derecesi
ESC	Electronic Speed Controller, Elektronik Hız Denetleyicisi
GMR	Giant Magneto Resistance
GPS	Global Positioning System, Küresel Konumlama Sistemi
I ² C	(I ² C) Inter-Integrated Circuit, Toprağa Referanslı Çok Denetleyicili Seri Veriyolu
IC	Integrated Circuit, Bütünleşik Devre
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit, Ataletsel Ölçüm Ünitesi
ISM	Industrial Scientific Medical Band, Endüstriyel, Bilimsel, Tıbbi Frekans Aralığı
İA	İnsansız Araçlar
İHA	İnsansız Hava Araçları
Li-Po	Lithium Polymer, Lityum Polimer Pil
LQR	Linear-Quadratic Regulator, Regülatör Problemi
MEMs	Mikro Elektro-Mekanik sistemler
MPC	Model Predictive Control, Model Öngörülü Kontrol
MTJ	Magneto Tunneling Junction
PPM	Pulse Position Modulation, Darbe Durum Modülasyonu
PWM	Pulse-Width Modulation, Darbe Genişlik Modülasyonu
QPC	Quadratic Programming in C
RC	Remote Controller, Uzaktan Kumanda
RISC	Reduced Instruction Set Computing, İndirgenmiş Komut Takımıyla Hesaplama
ROS	Robot Operating Systems, Robot İşletim Sistemi
SAM	Surface to Air Missile, Gdümlü Hava Füzesi
SLAM	Simultaneous Localization and Mapping, Eş Zamanlı Konumlandırma ve Haritalama (EZKH)
SPI	Serial Peripheral Interface, Seri Çevresel Arayüz
UART	Universal Asynchronous Receiver/Transmitter, Evrensel Asenkron Alıcı/Dönüştürücü
VTOL	Vertical Take-off and Landing, Dikey Kalkış İniş Yapabilen Araç
Wi-Fi	Wireless Fidelity, Kablosuz Bağlantı Alanı

ŞEKİL LİSTESİ

	Sayfa
Şekil 1.1 A Shallow Saucer-like Gondola [6].....	2
Şekil 1.2 Lipowski Helikopteri [6]	3
Şekil 1.3 A Brequet-Richet Helikopteri [6]	4
Şekil 1.4 De Bothezat Helikopteri [6]	4
Şekil 1.5 Ömichen Helikopteri [6]	5
Şekil 1.6 Brennan Helikopteri [6]	5
Şekil 1.7 Convertawings Model A Helikopteri [6]	6
Şekil 1.8 Yakovlev VVP-6 Projesi [6]	6
Şekil 1.9 Focke-Wulf Fw.61 Helikopteri [6]	7
Şekil 1.10 Curtis-Wright VZ-7 [6].....	8
Şekil 1.11 1900 lü yıllarda geliştirilen başlıca helikopter modelleri [6]	8
Şekil 2.1 Draganflyer[7].....	11
Şekil 2.2 X4 Flyer[8].....	12
Şekil 2.3 Starmac[9].....	13
Şekil 2.4 Geliştirilen iki farklı OS4 Platformu [10]	13
Şekil 2.5 M. Kemper’ in tezinde ele alınan 4 rotorlu sistem[11].....	14
Şekil 2.6 Glen P. Tournier’ in tezinde ele alınan 4 rotorlu sistem [12]	15
Şekil 2.7 Charles A. Clifton tezinde kullanılan Dragan Flyer [13]	16
Şekil 3.1 Altı rotorlu (hexacopter) helikopter gövdesi	17
Şekil 3.2 Altı rotorlu (hexacopter) helikopter gövdesi	18
Şekil 3.3 Alt Seviye Kontrol Mimarisi	21
Şekil 3.4 STM32F4 Discovery Kartı [15].....	21
Şekil 3.5 Sparkfun Dokuz Serbest Dereceli Atalet Ölçüm Ünitesi [16].....	23
Şekil 3.6 İvme ölçer Çalışma Prensipleri-1 [17]	24
Şekil 3.7 İvme ölçer Çalışma Prensipleri-2 [17]	24
Şekil 3.8 Manyetometre Bozucu Etkileri [19].....	26
Şekil 3.9 XBee Kablosuz Haberleşme Modülü [21]	26
Şekil 3.10 SRF08 Ultrasonik Mesafe Ölçüm Modülü [22]	27
Şekil 3.11 FT232 Seri-USB Çevirici [23]	27
Şekil 3.12 Robbe 2.4 Ghz Kumanda Alıcı [24]	28
Şekil 3.13 Futaba 6 Kanal RC Kumanda [25]	28
Şekil 3.14 Mikrokoopter Elektronik Hız Kontrol Modülü [26].....	29
Şekil 3.15 Mikrokoopter Elektronik Hız Kontrol Modülü	30
Şekil 3.16 Arayüz Bilgisayarı Otopilot Arası İletişim.....	31
Şekil 3.17 PID Çevrimleri	31
Şekil 3.18 Arayüz yazılımı-1.....	32

Şekil 3.19 Arayüz yazılımı-2.....	33
Şekil 3.20 Arayüz yazılımı-3.....	33
Şekil 3.21 Arayüz yazılımı-4.....	34
Şekil 3.22 Otopilot Haberleşme Sistemi.....	36
Şekil 3.23 Kumanda Alıcısından okunan PPM sinyal [25]	38
Şekil 4.1 Fırçasız doğru akım motoruna ait açık çevrim basamak cevabı	40
Şekil 4.2 Fırçasız doğru akım motoruna ait açık-çevrim kök-yer eğrisi cevabı.....	41
Şekil 4.3 Fırçasız doğru akım motoruna ait açık çevrim Nyquist cevabı	41
Şekil 4.4 Fırçasız doğru akım motoruna ait açık çevrim Bode cevabı	42
Şekil 4.5 Fırçasız doğru akım motoruna ait kapalı çevrim kök-yer eğrisi.....	42
Şekil 4.6 Fırçasız doğru akım motoruna ait kapalı çevrim Nyquist cevabı.....	43
Şekil 4.7 PID parametrelerinin Ziegler Nichols ile belirlenmesi	43
Şekil 4.8 Motora ait PID parametrelerinin belirlenmesi	44
Şekil 4.9 Altı rotorlu helikopterin gövde hareketleri ve uygulanan kuvvetler	45
Şekil 4.10 Hexarator PID çevrimleri	48
Şekil 4.11 Orijin takibinde yuvarlanma (roll) açısının zamana göre değişimi	48
Şekil 4.12 Orijin takibinde yunuslama (pitch) açısının zamana göre değişimi.....	49
Şekil 4.13 Orijin takibinde sapma (yaw) açısının zamana göre değişimi	49
Şekil 4.14 Önceki çalışmalarda önerilen durdurma kriteri [31]	51
Şekil 4.15 Gecikme zamanına sahip bir kontrol sinyalinin modeli [38]	53
Şekil 4.16 Önerilen Sistemin Mikrodenetleyiciye Uygulanması	66
Şekil 4.17 $J_D=f(x(\tau))$ Ayrık maliyet fonksiyonunun 'MPC' gecikmeye bağlı değişimi.....	68
Şekil 4.18 Sistem durum çıkışlarının değişimi ($x_1; x_2$) (MPC-LQR).....	68
Şekil 4.19 MPC deki maliyet fonksiyonunun MATLAB ortamında hesaplanması.....	69
Şekil 4.20 Hexacopter sistemine uygulanan kontrol sinyal maliyet değerlerinin gecikme zamanına göre değişimi.....	71
Şekil 4.21 Sistem için belirlenen yükseklik değerinin zamana göre değişimi	71
Şekil 4.22 Ön kesmeli MPC ile sistemim istenilen yüksekliğe oturması	72
Şekil 4.23 Ön kesmeli MPC ile sisteminde hatanın değişimi.....	72
Şekil 5.1 Iterated-Sparse local submap joining filter [51].....	75
Şekil 5.2 Robot Navigation Toolbox ile elde edilen harita kestirimi [53].....	75
Şekil 5.3 CEKF-SLAM uygulaması ile harita kestirimi [54]	76
Şekil 5.4 Örnek bir DP-SLAM uygulaması [55].....	76
Şekil 5.5 Frieberg Üniversitesi kampüsündeki grid mapping uygulaması [57]	77
Şekil 5.6 HOG-Man yöntemiyle oluşturulmuş harita örnekleri [59]	77
Şekil 5.7 Max-mixture yöntemi kullanılarak oluşturulmuş harita örneği [60]......	78
Şekil 5.8 Manifold Tool kit yöntemiyle oluşturulan haritalar [62]	79
Şekil 5.9 Range-only SLAM [63].....	79
Şekil 5.10 Thin Junction Tree Filters for SLAM yöntemi ile haritalama örneği [64]	80
Şekil 5.11 Ağaç haritası uygulaması [66].....	81
Şekil 5.12 Intel Research Lab içerisindeki Gmapping uygulaması [57]	85
Şekil 5.13 Intel Lab. Gmapping uygulamasında robot gezintileri [57]	85
Şekil 5.14 Gmapping ile A009 araştırma laboratuvarının harita kestirimi.....	91
Şekil 5.15 Gmapping ile A009 araştırma laboratuvarında harita kestirimi-2	91
Şekil 5.16 Gmapping ile A Blok Zemin kat asansör çevresi harita kestirimi	92
Şekil 5.17 Gmapping ile A Blok Zemin kat asansör çevresi harita kestirimi-2	92

Şekil 6.1	Hexacopter için otonom uçuş arayüz platformu	96
Şekil 6.2	Hexacopter için internet üzerinden uzaktan kumanda platformu.....	96
Şekil 6.3	Hexacopter için deneme seyrüsefer çalışması	96

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 3.1 Haberleşme paketleri yapısı	35
Çizelge 4.1 Zaman gecikmeli MPC deki ayrık maliyet fonksiyonu değerlerinin iterasyon sayısına göre değişimi.....	69

ALTI ROTORLU BİR HELİKOPTERDE (HEXACOPTER) DENGELİK VE YÜKSEKLİK KONTROLÜ İÇİN YENİ BİR KONTROL ALGORİTMASI

Tarık Veli MUMCU

Kontrol ve Otomasyon Mühendisliđi Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Doç. Dr. Kayhan GÜLEZ

Son yıllarda insansız hava araçları (İHA) ile yapılan çalışmaların ana kısmını, aracın otonom kontrol temelli görevleri yerine getirmesi oluşturur. Kullanıcı müdahalesi olmadan araç bu görevleri yerine getirirken, çalıştığı ortamı tanınması ve algılanması gerekli kararları almasındaki en önemli etkenlerdir.

Bu çalışmada yer alan otonom kontrolün ana teması insansız çevrelerde (ortamlarda) yapılan kontrole işaret etmektedir. Geliştirilen otopilot, sistem yazılımları ile bilinmeyen bir çevrede aracın yetersiz bilgiyle kontrolü hedeflenmiştir. Böylelikle, aracın, algılayıcı ve ölçüm hatalarının olduğu çevrelerde daha iyi bir seyrüsefer (navigasyon) sergileyebilmesi amaçlanmıştır. Bu tezde geliştirilen otopilot, sistem yazılımları, kontrol algoritmaları ve bunlara ek olarak konumlandırma ve haritalama görevleri altı rotorlu bir helikopterde (hexacopter) uygulanmıştır. Hexacopter uygulamasındaki görevler üç ana başlıkta ifade edilebilir:

- Denge ve Yükseklik Kontrolü,
- Haritalama ve Konumlandırma,
- Emniyet Planlaması (Yerel Trafik Akışı).

Bu tezde, geliştirilen otopilot, sistem yazılımları ve kontrol algoritmaları küçük değişiklikler yapılarak diđer çoklu rotorlu sistemlerde (çoklu rotorlu helikopterlerde) de kullanılabilir.

Anahtar Kelimeler: İnsansız Hava Araçları (İHA), Çoklu Rotorlu Sistemler, Denge ve Yükseklik Kontrolü, Konumlandırma ve Haritalama.

**A NEW CONTROL ALGORITHM FOR BALANCE AND ALTITUDE CONTROL
OF A SIX-ROTOR HELICOPTER (HEXACOPTER)**

Tarık Veli MUMCU

Department of Control and Automation

Ph.D. Thesis

Adviser: Assoc. Prof. Dr. Kayhan GÜLEZ

The major part of the research on unmanned aerial vehicles (UAVs) in the last decade was to deploy autonomous control related tasks. Environmental recognition and perception are the main hurdles impeding the decision making ability of the operated UAVs.

The main thrust of autonomous control in this study is to address the impact of the control held in an unmanned environment. Employing, the developed autopilot and framework it is aimed to control the vehicle in an unknown environment with inadequate information, therefore it can perform a better navigation in such environment, where sensor and measurement noise is existing. In this thesis, the developed framework, autopilot, control algorithms, as well as applied localization and mapping tasks are implemented onto a six rotor helicopter (hexacopter). The main objectives we are seeking for the hexacopter application are threefold:

- Balance and Altitude Control,
- Localization and Mapping,
- Safety Planning (Local Traffic Flow).

The developed autopilot and control algorithms can be easily applied onto other types of multirotor systems (multi rotor helicopters) with minor modifications.

Key words: Unmanned Aerial Vehicle (UAV), Multi Rotor Systems, Balance and Altitude Control, Localization and Mapping.

1.1 Literatür Özeti

Kullanıcı kontrolü olmadan robotların istenilen görevleri yerine getirmesi robotik alanındaki en önemli araştırma konularından biridir. İnsan hayatını tehlikeye sokacak durumlarda, robotların insanların yerine kullanılması karşılaşılabilecek can kayıplarını azaltacaktır. Kullanıcı kontrol olmayan robotlardan bir tanesi de bugün askeri ve sivil amaçlarla kullanılan İnsansız Araçlardır (İA). Teknolojideki gelişmeler İA' nın birbirinden farklı pekçok fonksiyonu yerine getirmesini sağlamaktadır. Hava, kara, su ortamında çalışan bu araçlar, aynı fonksiyonları, 'sel, yangın, deprem ... vb. doğal afetler sonrası durum belirleme gibi sivil; mayın arama, askeri malzeme ve yabancı cisim tespiti v.b. askeri' farklı ortamlarda yerine getirmektedirler. Günümüzde İA' nın kullanılabilceği amaç ve ortamlar artık sayılamayacak kadar çoğalmış ve birbirinden farklılaşmıştır [1], [2].

Son yıllarda özellikle askeri alandaki gelişmeler ve ihtiyaçlar doğrultusunda İnsansız Hava Araçları (İHA) kendi alanındaki diğer araçlardan daha fazla önem kazanmıştır. Bugün sadece askeri amaçla değil, film endüstrisinde havadan çekim, radyoaktivite tesbiti konularında artık endüstride sıklıkla kullanılmaktadır.

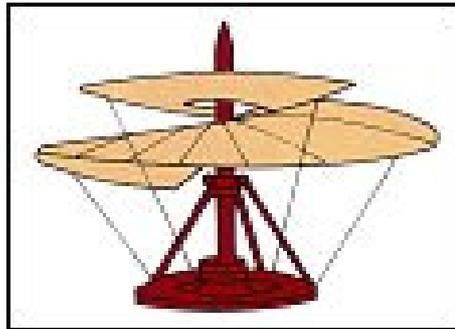
İHA' lar içersinde yer alan çoklu rotorlu helikopterlerle ilgili literatürdeki araştırma ve tarama makalelerinin azlığı bu konuyu çekici kılmaktadır. Çoklu rotorlu helikopterlerin diğer avantajı ise dikey olarak kalkıp alçalabilmeleri, havada askıda kalabilmeleridir. Bu görevleri yerine getirirken çoklu rotorlu helikopter klasik helikopter sistemine göre daha az enerjiye ihtiyaç duyar [3], [4], [5]. Bu tezinde konusu olan altı rotorlu

helikopterlere ait ilk gerekleřtirme alıřmaları, drt rotorlu ‘quadrotor’ helikopterler olarak ortaya ıkmıřtır. Gemiřte bařlıca ilk drt rotorlu hava tařıtı uygulamaları [6]:

- A Shallow Saucer-like Gondola
- Lipowski Helikopteri
- Briquet Richet Helikopteri
- De Bothezat Helikopteri
- michen Helikopteri
- Brennan Helikopteri
- Convertawings Model A Helikopteri
- Yakovlev VVP-6 Projesi
- Focke-Wulf Fw 61
- Curtis-Wright VZ-7

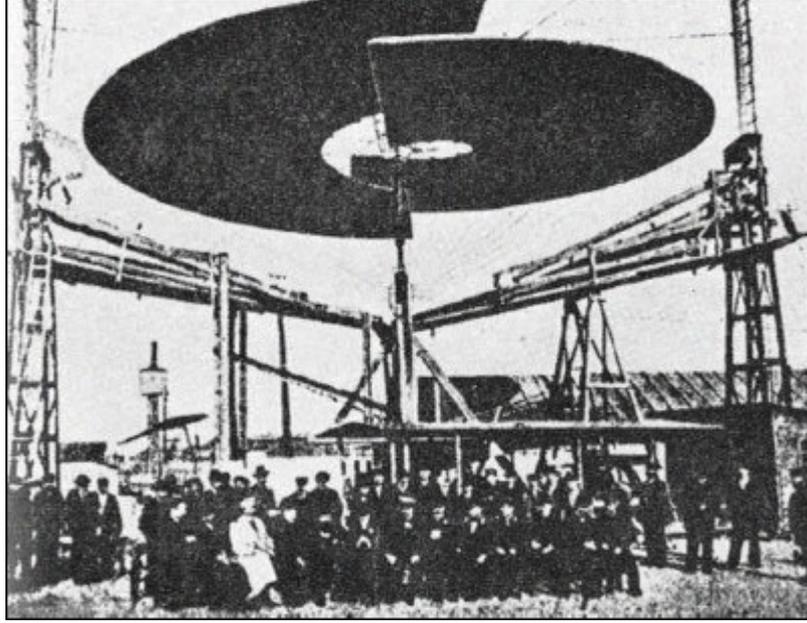
Gemiřteki Bařlıca Uygulama rnekleri

A Shallow Saucer-like Gondola [6]: Leonardo Da Vinci’ nin ilk tasarımlarından biri olan bu helikopter tasarımında iki dayanak noktasına baėlı iki ift kanat bulunmaktaydı. Olduėa karmařık olan bu tasarımda pilot silindirler ayak pedallarını kullanarak hareketi saėlıyordu. Birbirinin zerinden geerek hareket eden kanatlar, yatay hareket etmekteydi. Bu hareket ile kanatlar arasında kalan hava sıkıřtırılarak aracın kalkıřı saėlanmaktaydı. Tasarımda ayrıca merdiven řeklinde iniř takımları aracın iniřini yumuřatmak iin dřnlmřt. řekil 1.1’ de A Shallow Saucer-like Gondola tasarımının resmi, aviastar.org adresinden alınmıřtır.



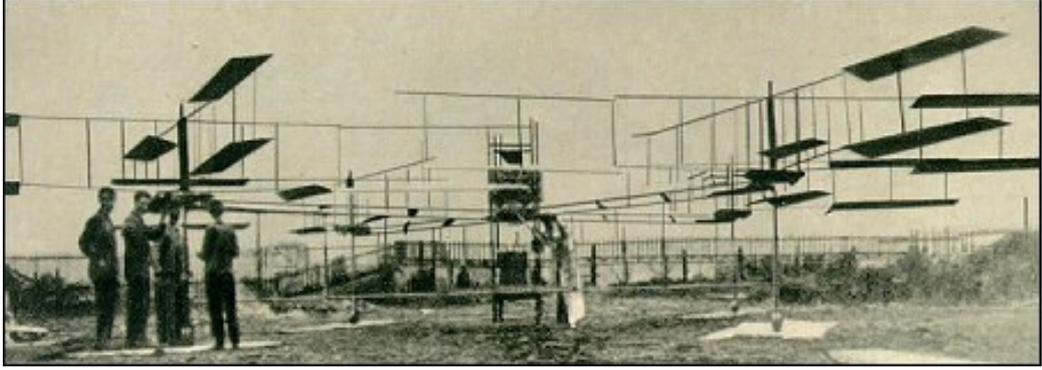
řekil 1.1 A Shallow Saucer-like Gondola [6]

Lipowski Helikopteri [6]: 1905 yılında Polonyalı bir mühendis olan Jozef Lipowski tarafından geliştirilmiştir. Pervane sistemi birbirine zıt dönen iki rotordan oluşur. Pervaneler (kanatlar) yarım daire şeklinde olmakla beraber, uzun bir çelik tüp, ağaç bir jant ve macun ipek ile kaplanmış çelik tekerlek parmağı ile desteklenmiştir. Pervane sisteminin altında pilotu ve motorları taşıyan düz bir düzlem bulunmaktadır. İkili pervane çapı 16 metreye tekabül ederken yaklaşık 200 metrekareyi kaplamaktadır. 35 beygir gücünde bir elektrik motoru ile sürülen sistem yaklaşık 778 kg kaldırırken kaldırma gücü oranı 22.22 kg/hp ile dönemin en başarılı tasarımlarından biriydi. Şekil 1.2 de Lipowski Helikopteri görülmektedir.



Şekil 1.2 Lipowski Helikopteri [6]

Briquet Richet Helikopteri [6]: 1907 yılında Louis ve Jacques Breguet kardeşler, Profesör Charles Richet ile birlikte Gyroplane No.1 adı ile helikopter tasarımı geliştirdiler. Bilinen ilk dikey kalkış yapabilen makine olmasına rağmen, deneylerde her bir rotorun bir insan tarafından desteklenmesinden dolayı ilk uçan helikopter olamamıştır. İlk denemelerde 0,6 metreye yükselirken sonrasında 1,5 metre yükselmeyi başarmıştır. Sistem dikdörtgen merkezi bir çerçeve ve bunun uzantısı olan dört rotordan oluşmaktaydı. Herbir rotora ait dört çift pervane vardı. Pervane çiftleri birbirine ters yönde hareket gerçekleştiriyordu. Sistem toplam 32 küçük kaldırma yüzeyine sahipti. Şekil 1.3' de A Brequet-Richet Helikopteri görülmektedir.



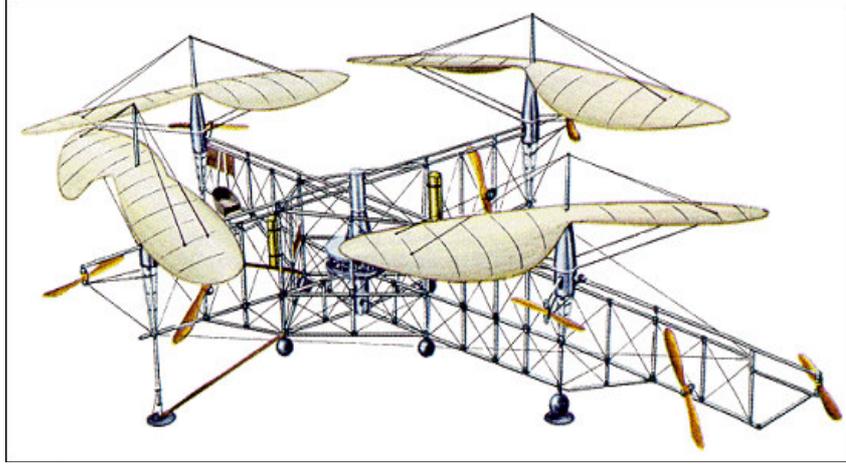
Şekil 1.3 A Brequet-Richet Helikopteri [6]

De Bothezat Helikopteri [6]: 1921 yılında Amerikan ordusu ile Dr. De Bothezat ve Jerome arasında imzalanan sözleşmede düşey kalkış yapabilen bir makine tasarımı vardı. 1678 kg ağırlığına sahip X şekilli 8,1 metre çapında rotorlardan oluşan her bir rotora ait 6 pervane olan sistem merkezden 9 metre uzaklıkta yer almaktaydı. Hava aracı 1700 kg ile ilk uçuşunu 1922’ de gerçekleştirdi. Daha sonra güncellenen 220 beygir gücündeki motoru ile havada 5 metre yüksekliğe erişti. Şekil 1.4’ te De Bothezat Helikopteri görülmektedir.



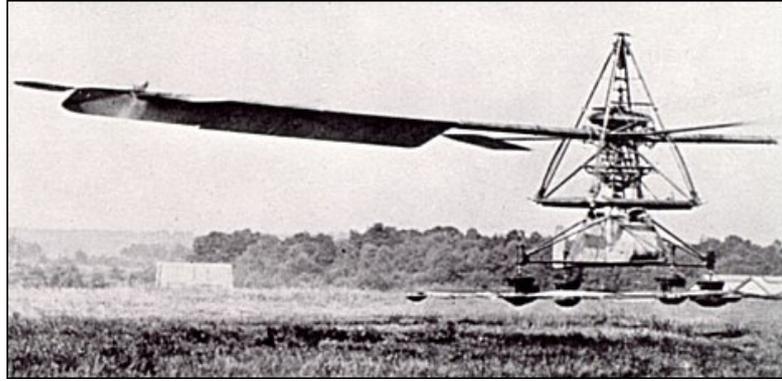
Şekil 1.4 De Bothezat Helikopteri [6]

Ömichen Helikopteri [6]: 1920 yıllarında Fransa’ da Peugeot’ da çalışan genç bir mühendis olan Etienne Ömichen altı farklı araç tasarımı yapmıştır. Bu araçlardan Ömichen No. 2; X şekline sahipti ve iskelette tüpsel çerçeveler kullanılmıştı. Araç 4 rotor 8 pervaneden oluşmaktaydı. Tüm sistem 120 beygir gücündeki Le Rhone motoru ile sürülüyordu. Pervanelerin çoğunluğu yatay olarak dengeyi sağlamakla görevliyken, burun kısmında yer alan bir pervanede ileri gitmek için gerekli olan itiş gücünü sağlıyordu. Şekil 1.5’ de Ömichen Helikopteri görülmektedir.



Şekil 1.5 Ömichen Helikopteri [6]

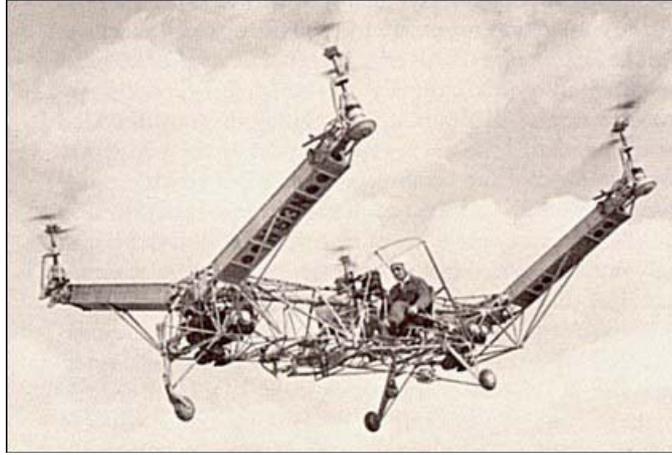
Brennan Helikopteri [6]: İlk tasarımları 1884 de İngiltere’ de Louis Brennan tarafından gerçekleştirilen pervane sürümlü sistem, 1919-1926 yıllarında ilk defa denendi. 1360 kg olan ilk makine 18 metrelik tek bir rotora sahipti. 230 beygir gücündeki Bentley BR-2 motoru ile sürülen sistemde, sıkıştırılmış hava rotor yunuslama açısının kontrolünde kullanılıyordu. Sınırlı yapılan test uçuşlarında araç en fazla 2,4 metre yüksekliğe çıkabilirken, 183 metre yol katetti. 1926 yılında aracın düşmesi sonucu testler sona erdi. Şekil 1.6’ da Brennan Helikopteri görülmektedir.



Şekil 1.6 Brennan Helikopteri [6]

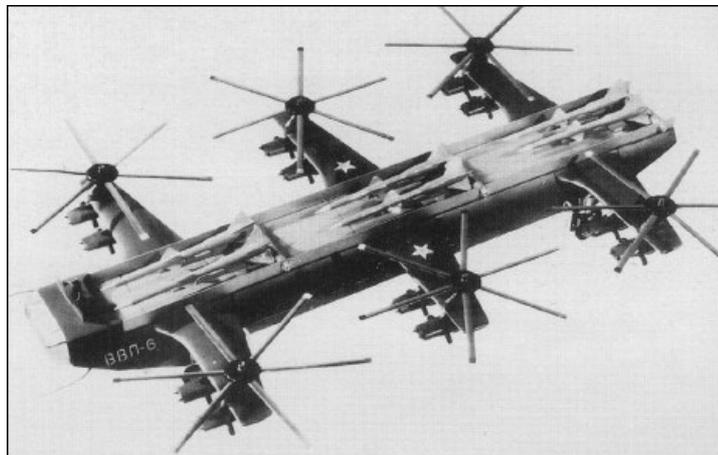
Convertawings Model A [6]: Bu model ile 1922 Fransız Ömichen ve Amerikalı G. de Bothezat tarafından geliştirilen modeller yeniden canlandı. Bu modelin üzerinde yapılan çalışmalar sonucunda ilk prototip 1955 yılında tamamlandı. Tasarımda dört rotor yatay olarak arka arkaya eklenmiş iki çiftten oluşur. Kontrol mekanizması oldukça basitleştirilmiş ve rotorlar arası itici güç kuvvetinin fark değişimi olarak tanımlanmıştır. Gövdesi boru şeklindeki çelikten oluşurken, rotorları destekleyen parçalar alüminyum alaşımdan yapılmıştır. Sistem için gerekli olan güç iki motordan karşılanırken bunların rotorlara iletilmesi çoklu yön değiştirme kayışları ile gerçekleştirilmiştir. Mil ve iletim

kutuları ile ihtiyaç halinde her bir motorun tüm sistemi sürmesi sağlanmıştır. Amerikan ordusu tarafından satın alınan bu sistem ilk uçuşlarını 1956 yılının Mart ayında gerçekleştirmiştir. Şekil 1.7' de Convertawings Model A Helikopteri görülmektedir.



Şekil 1.7 Convertawings Model A Helikopteri [6]

Yakovlev VVP-6 Projesi [6]: Jet kalkışlı V/STOL' un (dikey ve, veya kısa kalkış, iniş yapabilen araç) gelişimi ile birlikte dağınık ve sıklıkla yer değiştiren, yiyecek, mühimmat, yakıt taşıyan operasyon araçları aynı zamanda savunma alanında da ortaya çıkmaya başladı. Altı rotorlu ilk savunma projelerinden biride Zakovlev VVP-6 projesidir. Bu projede tamamiyle uçabilen bir SAM (Güdümlü Hava Füzesi) geliştirilmiştir. 49 metre uzunluğundaki kutu şeklindeki çerçeveler, 6 V-750 (11D), ya da V-755 (20D) güdümlü füzeleri taşımak için tasarlanmıştır. Tasarım herbiri dört turbo motor tarafından sürülen altı pervaneli altı rotordan oluşmaktaydı. Şekil 1.8' de Yakovlev VVP-6 Projesi görülmektedir.



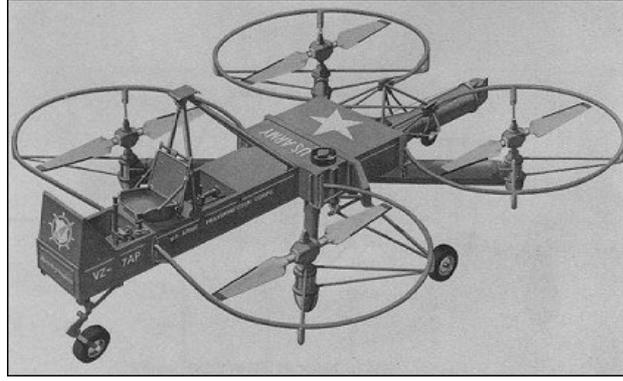
Şekil 1.8 Yakovlev VVP-6 Projesi [6]

Focke-Wulf Fw 61 [6]: Bir araç sınıfı olan helikopterlerde tek bir icat eden olmamakla beraber modern helikoptere en çok katkısı yapan Igor Skorsky' dir. Sonrasında İngiltere, Fransa, Almanya ve Rusya' da oldukça başarılı tasarım çalışmaları yapılmıştır. Avrupada ilk uçan helikopter tasarımı olarak iddia edilen Breguet-Dorand Gyroplane makinesinden sonra yine bir alman tasarımcı Doktor Heinrich Karl Johann Focke tasarımı Fw.61 ile ilk serbest uçuşunu 1936' da gerçekleştirmiştir. Tasarım olarak Breguet-Dorand makinesinden üstün olarak kabul edilmekle beraber, geliştirmeye açık bir tasarım olması sebebiyle de helikopter tarihindeki yerini almıştır. Tasarımda iki rotor modeli kullanılırken rotorlardan her biri üç bıçaklı pervane yapısına sahipti. Araç dikey olarak iniş, kalkış yapabilirken; rotorlar arasındaki kaldırma kuvveti farkını kullanarak yatay hareketi sağlıyordu. Şekil 1.9'da Focke-Wulf Fw.61 Helikopteri görülmektedir.



Şekil 1.9 Focke-Wulf Fw.61 Helikopteri [6]

Curtis-Wright VZ-7 [6]: Diğer bazı yük modellerinde olduğu gibi Curtis-Wright VZ-7 modeli de uçan ve dikey iniş kalkış yapabilen, ordu mühimmatlarını taşıyabilecek arazi aracı fikrinden çıkarak geliştirilmiştir. Dört düşey yerleştirilmiş rotordan oluşan sistemde 425 beygir gücündeki Turbomeca Artouste IIB türbin motoru kullanılmıştır. Geliştirilen tasarımlar başarıyla uçmasına rağmen, ordunun beklentisi olan yükseklik hız değerlerini karşılayamadıkları için 1960 ortalarında proje olarak geri çekilerek üreticiye iade edilmiştir. Geliştirilen sistem Şekil 1.10' da görülebilir. 1900' lü yıllarda dünyada geliştirilen başlıca helikopter modelleri aviaStar.org adresinden alınan bilgilerle ülkeler bazında Şekil 1.11' de derlenmiştir.



Şekil 1.10 Curtiss-Wright VZ-7 [6]

İngiltere	Almanya	Fransa	İtalya
Brennan Cierva Hunting Percival Westland ...	Foeke - Achgelis Dornier Kraus MBB ...	Brequet - Richet Omichen Sud - Quest Sud - Aviaclan Aerospatiale ...	Agusta Lualdi Manzolini Fiat ...

Rusya	ABD	Japonya
Lipowski Bratukhin Kamov Mil Yakovlev ...	Bell Boeing Convertawings Hiller Koman Sikorsky ...	Kayaba Kawasaki Mitsubishi ...

Şekil 1.11 1900 lü yıllarda geliştirilen başlıca helikopter modelleri [6]

1.2 Tezin Amacı

Çoklu rotorlu helikopterlerde yapılan çalışmalar incelendiğinde, çalışmanın ana kısmını:

- Çoklu rotorlu sistemin mekanik olarak inşa edilmesi ve belirli görevleri yerine getirmesi,
- Hazır olan sistemler için daha aktif rol oynayan bir otomatik pilotun geliştirilmesi,
- Literatürde bulunan kontrol yöntemlerinin bu sistemlerde simülasyon ve uygulama bazında denenmesi oluşturmaktadır.

Altı rotorlu bir helikopterde (hexacopter) denge ve yükseklik kontrolü için yeni bir kontrol algoritması başlıklı bu tezin amacı; altı rotorlu bir helikopter platformu olan hexacopter için ROS platformu ve Ubuntu işletim sistemi kullanılarak amaçlanan fonksiyonları yerine getirecek bir uçuş platformu gerçekleştirilmesidir. Gerçeklenen bu platformda klasik ve modern algoritmalar denenebilir. Kamera, lazer ve ultrasonik mesafe ölçücünün sisteme bütünleştirilmesiyle, görüntü, açı-mesafe ve mesafe bilgilerinin platform üzerinde işlenmesi mümkün kılınmış, altı rotorlu sistem birbirinden farklı görevler için hazır hale gelmiştir.

Platformun kablosuz ağ (Wi-Fi noktası) üzerinden haberleşmesi sağlanmıştır. Bu şekilde üzerinde çalıştırılan algoritmaya müdahale edilebileceği gibi aynı zamanda üzerindeki bilgilerin belirli bir merkeze aktarılması mümkündür.

1.3 Orjinal Katkı

Tezin, bu alanda yayınlanan eserlere getirdiği başlıca yenilik, altı rotorlu bir helikopter için otomatik pilot sisteminin geliştirilmesi ve gerçekleştirilmesidir. Yine hazır gövde üzerinde alt kontrol sistemleri için 'motorların sürülmesi, gövdenin pozisyonu, v.b.' bir yazılım geliştirilmiş, daha sonra geliştirilen bu yazılım dokuz serbestlik derecesine sahip bir ataletsel dönüşüm ünitesi üzerinde ve 32 bitlik ARM kartı ile gerçekleştirilmiştir. Gerçeklenen bu sistemi bilgisayar üzerinde görüntüleyebilmek için yine Visual C dilinde bir arayüz oluşturulmuştur. Altı rotorlu helikopterin dış dünya ile haberleşmesini sağlamak için, kamera, ultrasonik mesafe sensörü ve Hokuyo URG 04 tipi lazer seçilerek bu donanımların gövde ile yazılımsal bütünleşmeleri Ubuntu işletim sistemi üzerinde

kurulu Robot İşletim Sistemi (ROS)' nde sağlanmıştır. Altı rotorlu sistem, donanımsal ve yazılımsal olarak herhangi bir algoritmanın ROS-Ubuntu üzerinde denenebileceği sayfaya getirilmiştir.

Sistemde denge için PID ve yumuşak kalkış problemleri için model öngörü kontrol (MPC) stratejisi kullanılmıştır. Geliştirilen bu algoritmaların teorik analizi yapılmış ve gerek klasik PID kontrolör kodları kullanılarak tasarlanan gerekse diğer kontrolcü olan ön kesmeli MPC' nin uygulaması gerçekleştirilmiştir. Klasik MPC stratejisinden farklı olarak burada gecikme zamanını azaltmak için bir ön kesme tekniği tanımlanmış, bu tekniğin performansı ikinci dereceden bir sistem de LQR tipi kontrolcü kullanarak karşılaştırılmıştır. Sonrasında daha iyi performans veren bu teknik yükseklik kontrolüne uyarlanmıştır.

Lazer ve ultrasonik mesafe sensörlerinden gelen bilgiler ile sistemin harita kestirme performansı geliştirilmiştir. Üzerinde yer alan kamera sensörü kullanılarak sistemden elde edilen görüntüler, Bayer filtresinden geçirilerek grinin tonları için kontrast yoluyla çevrilmesi sağlanır, sistemin elde ettiği görüntülerin kablosuz ağ üzerinden ana kontrol bilgisayarına aktarılması sağlanmıştır. Gerçeklenen sistemin alt seviye kontrol kodları C dilinde, üst seviye işletim sistemi ROS-Ubuntu platformunda oluşturulmuştur. Yumuşak kalkış problemi için model öngörülü kontrolün MATLAB ortamında benzetimi yapılmış ve sistem üzerinde gerçekleştirilmiştir. Yine kablosuz ağ üzerinden -uçuş sistemi üzerinden- yapılan kontroller C dili ile Ubuntu-ROS ortamında geliştirilmiştir.

SON YILLARDA LİTERATÜRDE YERALAN BAŞLICA VTOL ÇALIŞMALARI

2000' li yıllarda akademik ve ticari amaçlı geliştirilen başlıca VTOL araçları aşağıdaki gibidir.

2.1 Draganflyer

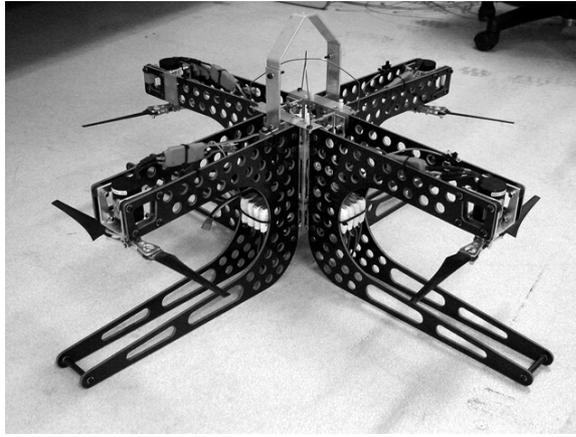
Ticari amaçla geliştirilen Draganflyer şirketi tarafından geliştirilen sistemin ilk örneklerinden biri aşağıdaki şekilde yer almaktadır. Bu sistemde sadece uzaktan kumanda 'Remote Controller' (RC) üzerinden kontrol mümkündür. Yunuslama, yönelme ve yuvarlanma açıları yine kullanıcı tarafından sisteme uzaktan kumanda ile iletilir. Üzerinde bulunan sensörler ise sırası ile kızıl ötesi, piezo jiroskop olup, hesaplama ve motor kontrol amaçlı üzerinde mikrokontrolcüsü bulunmaktadır [7]. Çalışmaya ait platform Şekil 2.1 de görülebilir.



Şekil 2.1 Draganflyer[7]

2.2 Avustralya Ulusal Üniversitesi ' nin Tasarımı

Avustralya Ulusal Üniversitesi tarafından geliştirilen bu tasarımda, X-4 Flyer' in en önemli avantajı olarak dört rotorlu helikopterlerin, klasik tek rotorlu helikoptere göre rotor-pervane çarpması yaşanmadan hedefe daha fazla yaklaşabilme yeteneği vurgulanır. Bu sebeple dört rotorlu helikopterleri iç mekan uygulamaları için adres gösterir. Bununla birlikte X4 Flyer üzerinde farklı tip algoritmaların denenebileceği bir sistem olarak tasarımın yapıldığından bahsedilmiştir [8]. Çalışmaya ait platform Şekil 2.2' de görülebilir.

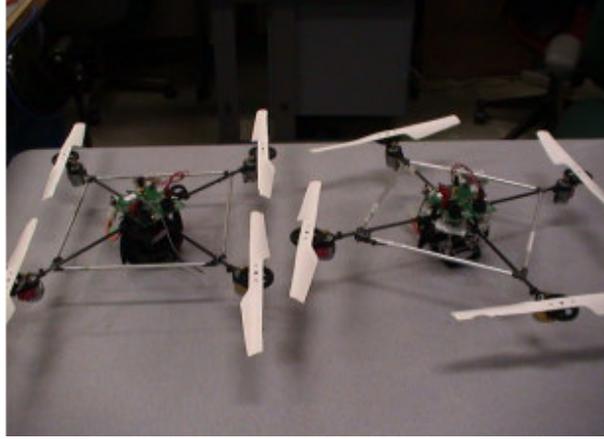


Şekil 2.2 X4 Flyer[8]

2.3 Stanford Üniversitesi' nin Tasarımı Starmac

Starmac tasarımının amacı, işbirlikçi, paylaşımcı ve işbirlikçi olmayan çoklu kontrol algoritmalarını gerçek değişken dış mekanlarda denemek içindir. Engelden kaçınma, otonom çökmenin saptanması, gezinenin belirlenmesi, görev atamaların test edilebilmesi için geliştirilmiştir.

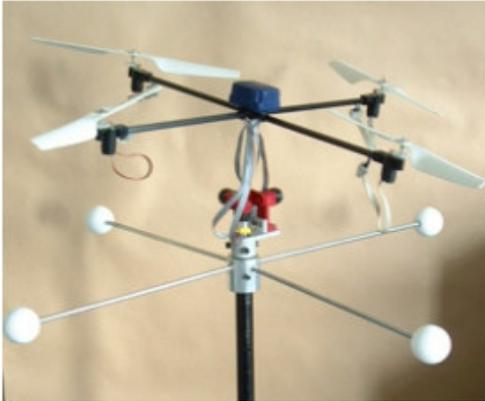
Tasarım Dragan Flyer III modelini baz alarak öncelikle enerji ünitesini LIPO pillerle değiştirmiştir. Üzerinde Stanford da geliştirilen baskı devre kartında kontrolcü ünitesi olarak iki adet PIC16F6520 yer almaktadır. Sensörler ise SRF08 ultrasonik altimetre ve microStrain firmasının 3DM-G modeli 3 eksenli IMU bulunmaktadır. GPS güncellemeler 1 Hz' de gerçekleşirken yatay düzlemde GPS hataları 1-2 metre civarındadır [9]. Çalışmaya ait platform Şekil 2.3' de görülebilir.



Şekil 2.3 Starmac[9]

2.4 Ecole Polytechnique Samir Bouabdallah' in Tasarımı

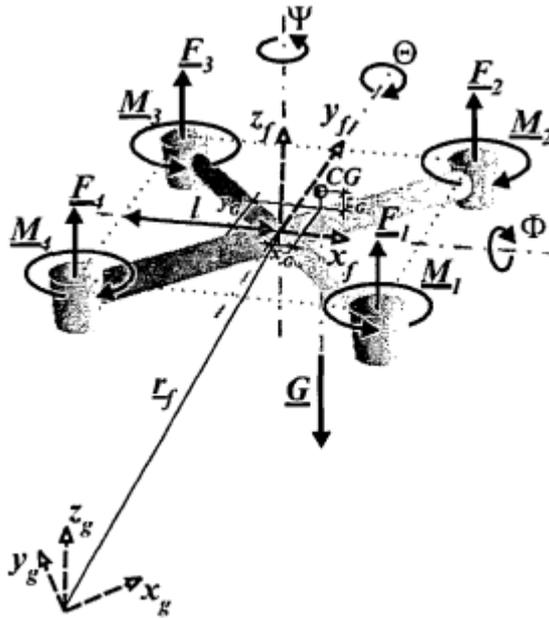
Bu çalışma sırasında iki platform tasarımı yapılmıştır. Bunlardan ilki test platformu olarak kullanılmıştır. Kullanılan OS4 test düzeneğinde RS232' den I²C ye dönüştürücü, Motor sürücü modülleri, 4 adet 29 g' lık gömülü manyetik enkoderleri bulunan motorlar, dişli kutuları ve pervanelerden meydana gelmektedir. Universal eklem üzerine kurulu olan sistemde, micro ölçümsel atalet düzeneğinde bulunmaktadır. Bu test düzeneğinin sonrasında geliştirilen ikinci sistemde ilkinden farklı olarak SRF04 ultrasonik sensörler engel algılamada kullanılmıştır. Geliştirilen sisteme klasik kontrol yöntemlerinden optimal ve geriye adımlama kontrol başarılı bir şekilde uygulanmıştır [10]. Çalışmaya ait platform Şekil 2.4' de görülebilir.



Şekil 2.4 Geliştirilen iki farklı OS4 Platformu [10]

2.5 Markus Kemper' in Tezi

Bu tezde, 4 rotorlu bir hava aracı geliştirilmiş, sistemin kontrolü modern kontrol yöntemleri kullanılarak araştırılmıştır. Tezin ana odak noktası 4 rotorlu bir mikro helikopterin gelecekteki uygulamalar için modellenmesidir. Geleneksel modellemelerde kullanılan merkezdeki yer çekimi etkisi bu tezde değişken olarak ele alınmıştır. Kalman Filtresi kullanılarak atalet sensörleriyle birlikte görüntü tabanlı davranış kontrolü oluşturulmuştur. Bu sistem bir yer işaretini tanır, kameranın davranış açılarını işarete bağlı olarak kestirir. Davranış bazlı seyrüsefer; yol planlama yöntemleri ile (ağaç aramalı v.b.) bilinen ortam bileşenleri ele alınmış, engel haritaları kullanılarak gerçekleştirilmiştir. Kullanılan kontrol yöntemi görüntüsel geri beslemeli kontroldür. Tezde sistem genel olarak yetersiz eyleyicili sistem olarak tanımlanmış ve bu sisteme uygulanan değişken yer çekimi dikkate alınarak modelleme yapılmıştır [11]. Çalışmaya ait platform Şekil 2.5' de görülebilir.



Şekil 2.5 M. Kemper' in tezinde ele alınan 4 rotorlu sistem[11]

2.6 Glenn P. Tournier' in Tezi (MIT)

Bu tezde, Moire örüntüleri kullanarak bir cisim görüntü bazlı pozisyonunu kestirme amaçlanmıştır. Amaç, araçların uçuşları sırasında diğer araçlara olan yakınlık ve iniş platformlarında gerekli olan 6 serbest dereceli kestirim bilgisini elde etmektir. İki farklı frekansta oluşturulmuş iki set ortogonal moire örüntülerini içeren işaretleri aracın

göreceli yönelimini ifade etmede kullanan bir hedef ve hedefin görüş alanında küçük bir araç üzerine monte edilmiş bir kamera bu görüntü ile yönelim kestirimi sistemini meydana getirmektedir. Kullanılan algoritma ile görüntüler işlenerek, kameranın hedefe göre göreceli olan geometri ve moire örüntüleri üzerindeki 4 tekli nokta ayrık Fourier dönüşümünü pozisyon ve davranış bilgilerini elde etmede fayda sağlamak için kullanır. Kullanılan kontrol teknikleri kalkışta ve durağan platformlar üzerine inişte uygulanmıştır [12]. Çalışmaya ait platform Şekil 2.6' de görülebilir.

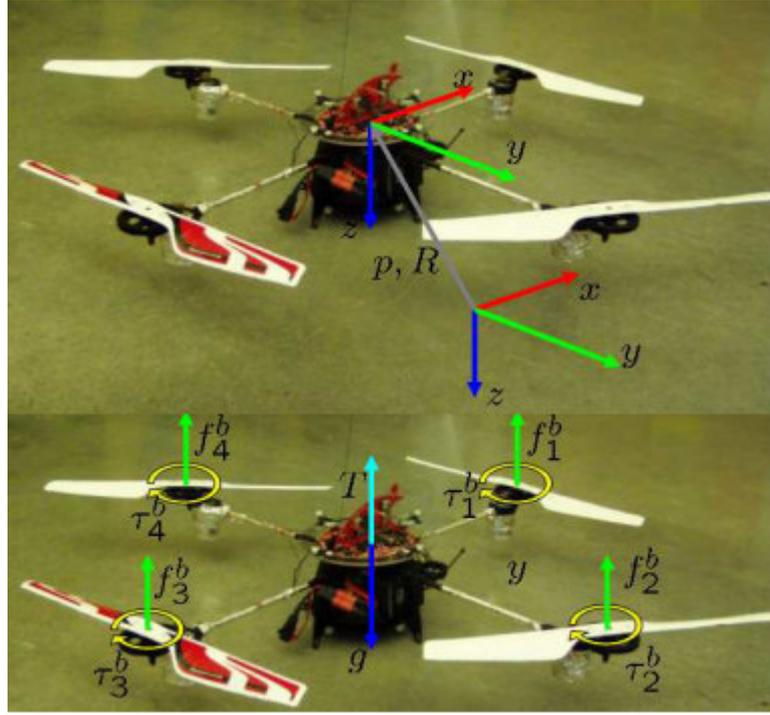


Şekil 2.6 Glen P. Tournier' in tezinde ele alınan 4 rotorlu sistem [12]

2.7 Charles A. Clifton' in Tezi

Vanderbilt Üniversitesinde Gömülü Hesaplama Yazılım Labaratuvarı' nda geliştirilen bu tezde, Vanderbilt Gömülü Hesaplama Platformu otonom araçlar için gerçekleştirilmiştir. Bu geliştirilen arayüz ile otonom araç tasarım aşamalarının daha basit ve hızlı olduğu belirtilmiştir. Oluşturulan bu arayüz düşük seviyeli kontrol (tasarım) ve yüksek seviyeli kontrol olmak üzere iki aşamada ele alınmıştır. Ayrıca, her bir tasarım aşaması yine birbirinden farklı görevleri bir araya gelerek icra eden alt sistemlerden oluşmaktadır. Düşük seviyeli kontrolün odak noktası, uzaktan kontrollü dört rotorlu helikopterin sürekli zamanda kontrolü oluştururken, hava aracının doğası gereği var olan kararlılık problemi çeşitli kontrol teknikleri kullanarak incelenmiştir. Yüksek seviyeli kontrolde ise, hava aracı için tanımlanan görevler ifade edilmiştir. Aracın seyrüseferi sırasında yüksek seviyeli hareket komutları hem kulanıcının hemde bilgisayarın anlayacağı bir

algoritma üzerinden gereklenmiřtir [13]. alıřmaya ait platform řekil 2.7' de grlebilir.

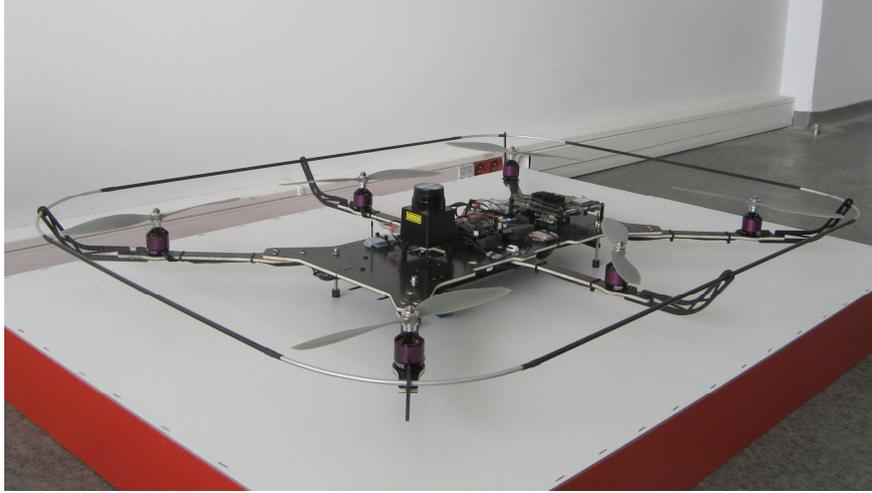


řekil 2.7 Charles A. Clifton tezinde kullanılan Drogan Flyer [13]

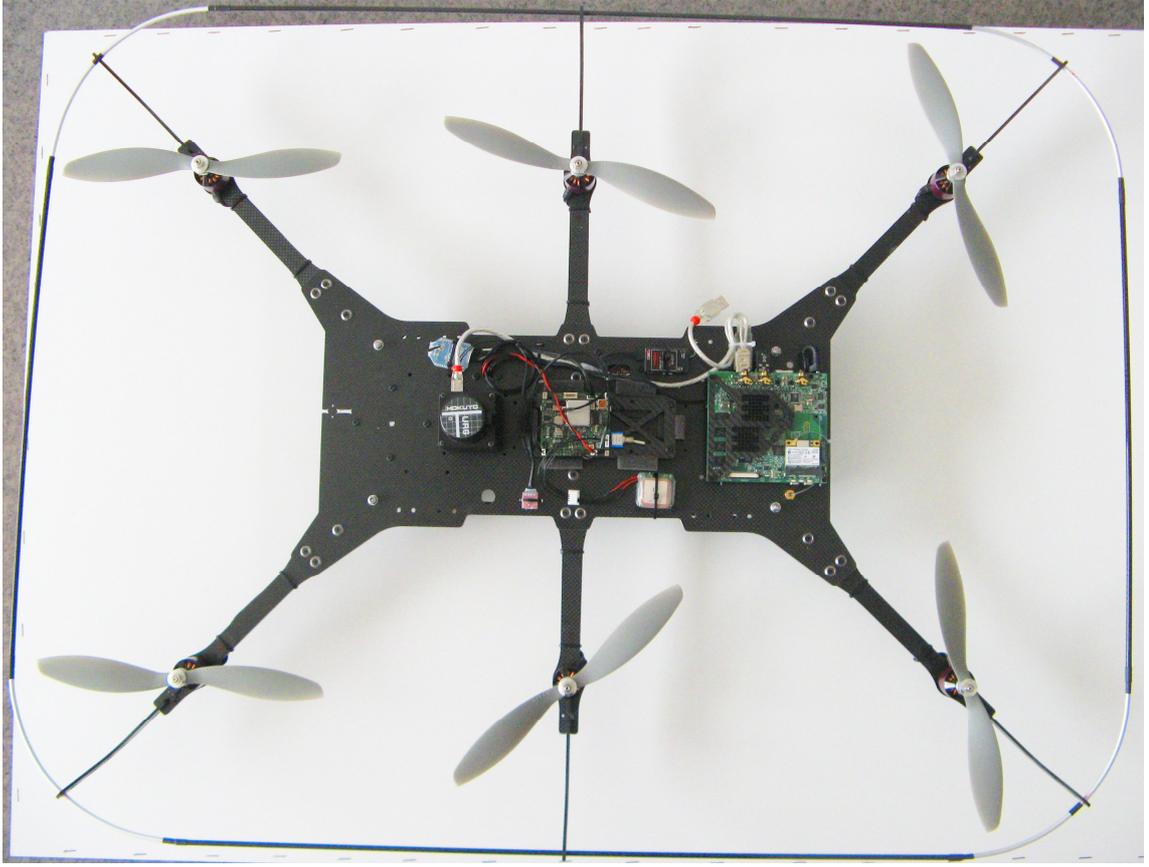
HEXACOPTER DONANIM ve YAZILIM MİMARİSİNİN OLUŞTURULMASI

3.1 Hexacopter Gövdesi

Şekil 3.1 ve Şekil 3.2' de gövdesi Skybotix firması tarafından tasarlananan altı rotorlu helikopter (hexacopter) görünmektedir.



Şekil 3.1 Altı rotorlu helikopter (hexacopter) gövdesi



Şekil 3.2 Altı rotorlu helikopter (hexacopter) gövdesi

3.2 Üst Seviye Kontrol

Üst seviye kritik algoritmalar için bir mini PC kullanılmıştır. Kullanılan bu mini PC 1.6 GHz hızında işlemciye sahip hareketli soğutma aksamı olmayan bir sistemdir. Oldukça düşük olan 5 watt seviyesinde bir güç tüketimine sahiptir. Hem işletim sistemi barındırması hemde düşük güç tüketimi ile gerçekleştirilen uygulama için optimum özelliklere sahiptir. Mini PC herhangi bir kaydedilebilir hafıza birimi barındırmamaktadır. Bu sebeple, işletim sistemi harici bir usb disk üzerinden koşulmaktadır.

Koşulan kompleks algoritmalar için mini PC üzerinde bir işletim sistemi bulunmalıdır. Uygulamada bu işlem için Linux tabanlı Ubuntu işletim sistemi tercih edilmiştir. Ubuntu gerek açık kaynak ve ücretsiz olması, gerek sürekli güncellenmesi, geliştirilmesi sebebiyle akademik çalışmalar için uygun bir platformdur.

Ubuntu üzerinde koşturulmak istenilen algoritmalar için çeşitli yöntemler mevcuttur. Ancak, bu iş için dünya çapında çeşitli akademik araştırmacılar tarafından geliştirilen Robot Operating System ROS [14], yani robot işletim sistemi tercih edilmiştir. Kontrol

ve robotik alanında, akademik çalışmaların uygulamaları tüm dünyada yaygın bir şekilde bu sistem üzerinde yürütülmektedir. ROS bir konsept olarak işletim sistemidir, ancak uygulama kısmı Ubuntu işletim sistemi üzerindedir.

ROS'un bir çok özelliğinin yanında en önemli özelliği bir bağlantı arayüzü sağlandığı sürece bir robot platformuyla ilgili tüm bileşenlerin tek bir sistem gibi yürütülebilmesine imkan tanımasıdır. Bu sayede örneğin iki robot ve bir kontrol platformu tek bir ağ üzerinde kontrol edilebilir. Robottan alınan veriler anlık olarak kontrol platformunda kullanılabilir. Gerçeklenen uygulamada kullanılan araç ile merkezi kontrol bilgisayarı arasında iletişim bu yöntemle kurulmuştur. ROS'un kullanacağı iletişim arayüzü 2.4 GHz ISM bandından çalışan kablosuz modem ile haberleşmektedir. Bu modem 54 Mbps hızındadır. Platform üzerindeki mini PC ve merkezi bilgisayar üzerindeki aynı bantta çalışan alıcı verici bileşenleri ile bu ağa bağlanmaktadır.

Sistem üzerindeki gerçek zamanlı sistem ile mini PC arasında seri iletişim ile sensör verileri alınmaktadır. Bunun yanında platform üzerindeki lazer sensör ve kamera verileri direk mini PC ile alınmaktadır. Alınan tüm bu veriler mini PC ile merkezi kontrol bilgisayarı arasında ROS ağı ile iletilmektedir. Bu yöntem ile uygulanan merkezi kontrol bilgisayarı veriler kendi üzerindeki gibi işlem yapabilmektedir. Ayrıca, bu ağ yapısı TCP/IP tabanlı olduğu için uygun kurulum gerçekleştirildiğinde, platformu internet üzerinden de yönetmek mümkün olabilecektir.

Platform üzerinde bulunan kamera PointGrey marka 1.3 MP Sony Carl-Zeiss lense sahip bir kameradır ve genellikle yüksek çözünürlüğü sebebiyle bilimsel araştırmalarda kullanılmaktadır. Kamera çıkışında siyah beyaz görüntü elde edilmektedir. Ancak Bayes filtresi kullanılarak bu görüntüden renkli görüntü elde edilir. Uygulamada bu işlem ROS ortamında mevcut sürücü kütüphaneleri ile gerçekleştirilmiştir. Kamera bilgisinin aktarımı ve yönetimi için bir ROS kütüphanesi ayrıca uygulanmıştır.

Aynı şekilde, platform üzerinde aksenel taramalı mesafe bilgisi elde etmek için Hokuyo marka lazer sensör kullanılmıştır. Lazer sensörler 240 derecelik tarama mesafesine sahiptir. Yani 0-240 derece açıyı her bir açı değeri için tarayarak; her bir açı değerinde en yakın objeye olan mesafeyi ölçer. Ölçebileceği maksimum mesafe 5.6 metredir. Elde edilen bu çoklu mesafe bilgisi, otonom davranışı sağlamaya yönelik algoritmalarda en

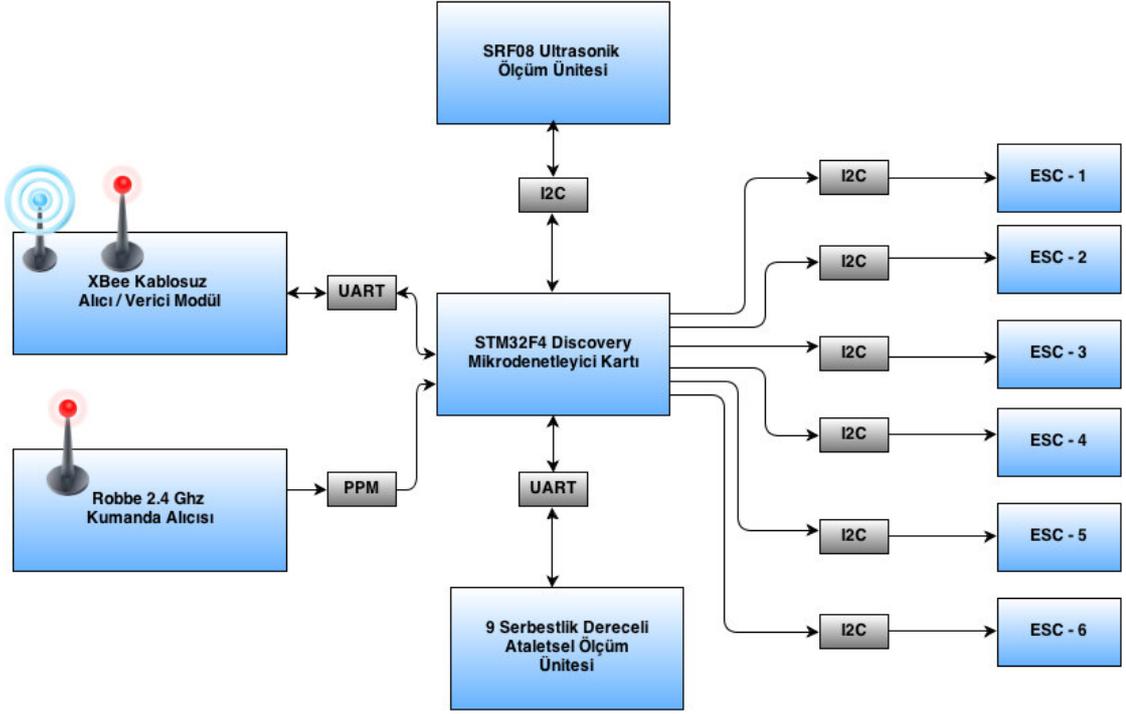
önemli bilgiyi oluşturur. Lazer sensörün okunması için aynı şekilde ROS platformunun sağlamış olduğu kütüphanelerden faydalanılmıştır. Lazer bilgisinin platformumuz için uygulanması adına sürücü kütüphanenin yanında, geliştirilen uygulamaya yönelik veri yönetimi için bir ROS kütüphanesi uygulanmıştır.

Uçuş sırasında kontrol ve uçuş bilgilerinin izlenmesi için bir de kokpit arayüzü tasarlanmıştır. Kokpit arayüzünün tasarımı C++ dili ile arayüz geliştirmeleri için kullanılan ve Nokia firması tarafından geliştirilen Qt platformu ile gerçekleştirilmiştir. Qt platformu ROS ile entegre edilerek Arayüz ile ROS çekirdeği arasında veri iletişimi sağlanmıştır. Ayrıca, arayüze çeşitli kontrol komut butonları yerleştirilerek, otomatik pilot ve manuel sürüş arasında geçiş yapılması sağlanmıştır. Manual sürüş sırasında kontrol için, uçuş simülatörlerine yönelik olarak üretilen gerçek askeri uçaklarda kullanılan joystick ve gaz kontrolün birebir modeli olan Saitek marka X-65f model joystick kullanılmıştır. Bu joystick, hareketli bir joystick olmayıp sabittir ve üzerine uygulanan basıncı algılayarak 12 Bit eksen verisine çevirir. Avantajı ise, hızlı manevra kabiliyeti kazandırmasıdır.

3.3 Alt Seviye Kontrol

3.4 Genel Donanım Yapısı

Hexapilot donanımı STM32F4 Discovery mikrodenetleyici kartı, Sparkfun IMU, SRF08 Ultrasonic Mesafe Ölçücü, XBEE haberleşme modüllerinden oluşmaktadır. Sisteme ait genel diyagram aşağıda görülmektedir. Hava aracının kontrolü, I2C protokolü üzerinden kontrol edilebilen elektronik hız kontrol üniteleriyle ESC (Electronic Speed Controller) sağlanmaktadır. Hız kontrol üniteleri, sabit mıknatıslı fırçasız motorları sürmektedir. Sistem geliştirilecek kompakt tasarımla genel amaçlı bir otonom kontrol kartı haline getirilecektir. Böylelikle hem hava aracının kontrolünde, hem de diğer çalışmalarda çeşitli araçların kontrolünde kullanılacaktır. Şekil 3.3' de alt seviye kontrol sistemin genel mimarisi resmedilmiştir.

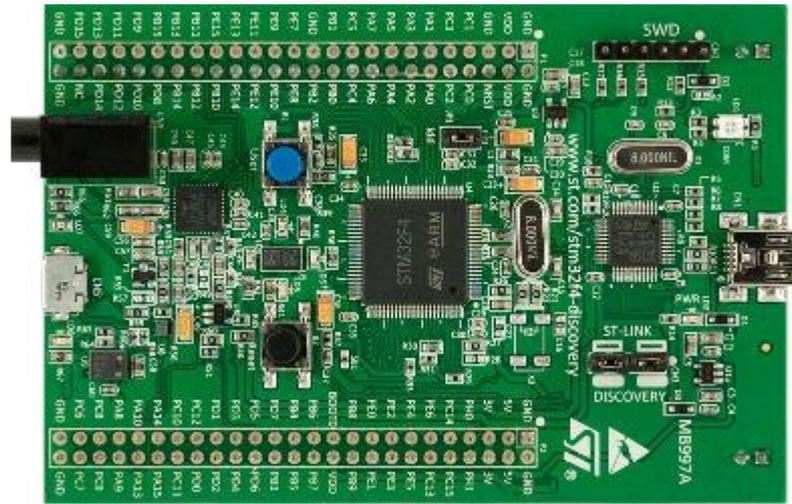


Şekil 3.3 Alt Seviye Kontrol Mimarisi

3.5 Hexacopter Donanım Bileşenleri

3.5.1 STM32F4 Discovery Kartı

STM32F4 Discovery Mikrodenetleyici kartı, üzerinde ST firmasının üretmiş olduğu ARM Cortex-M4 Mimarisine sahip STM32F407 mikrodenetleyicisini ve programlama ünitesini içeren bir geliştirme kartıdır[15]. Şekil 3.4' de kartın fiziki görünümü ve üzerindeki bileşenleri görülmektedir.



Şekil 3.4 STM32F4 Discovery Kartı [15]

Bu kart ile kolay bir şekilde uygulamalar geliřtirmek m¼mk¼nd¼r. zerinde bulunan mikrodenetleyici gerek zamanlı uygulamalar geliřtirmek iin olduka uygun olup, 168 MHz hızında alıřmaktadır. Ayrıca, yoęun matematik iřlemlerde y¼ksek performans g¼steren bir kayar-noktalı sayı iřlem nitesi (Floating Point Processor) de iermektedir. B¼ylelikle kayar noktalı matematik hesaplamalar ve filtre iřlemleri kolaylıkla gerekleřtirilebilmektedir. Kartın b¼t¼n giriř/ıkıř pinlerine kart kenarındaki baęlantı ularından ulařılabilmektedir.

STM32F4 Discovery kartı otopilotun alt seviye kontrol iřlemlerinde merkez g¼revi g¼rmektedir. zerinde alıřan gerek zamanlı iřletim sistemiyle ařaęıdaki sens¼rler ve mod¼llerden verileri almaktadır:

- Ultrasonik mesafe sens¼r¼
- Ataletsel l¼m nitesi
- El Kumandası Alıcısı
- Mini PC nitesi

Hava aracının alt seviye kontrol iřlemleri STM32F4 Discovery kartındaki mikrodenetleyici ierisindeki yazılımla gerekleřtirilmektedir. st seviye kontrol ve algoritmik iřlemler mini PC mod¼l¼nde gerekleřtirilip ilgili komutlar STM32F4 kartına g¼nderilmektedir. Ayrıca alt seviye kontrol iřlemleriyle alakalı veriler hem kablosuz baęlantı zerinden aray¼z bilgisayarına hem de , mini PC mod¼l¼ne g¼nderilmektedir.

STM32F4 Discovery kartı alt seviye kontrol iřlemlerinin yanında sens¼rlerden verilerin alınması, el kumandası komutlarının okunması, mini PC ve aray¼z bilgisayarıyla iletiřim iřlemlerinin y¼r¼t¼lmesinden sorumludur. Kart zerinde ayrıca 3 eksen ivmeler, ses ıkıř DAC'ı ve MEMS mikrofon gibi bileřenler de bulunmakta ancak bunlar projede kullanılmamaktadır.

3.5.2 Sparkfun Dokuz Serbest Dereceli Atalet l¼m nitesi

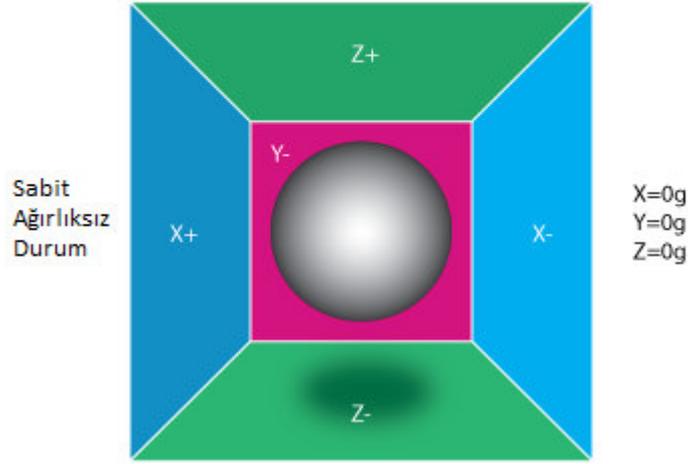
Sparkfun 9 DOF IMU zerinde 3 eksen ivme ler, 3 eksen aısal hız ler ve 3 eksen manyetik alan ler sens¼r mod¼lleri ve 8 bitlik bir mikrodenetleyici barındırmaktadır. Kendi ierisindeki yazılımla 3 eksen iin aısal duruř deęerleri elde edilebilmektedir. Bu aısal deęerler hava aracının  eksen durum kontrol¼nde kullanılmaktadır. Őekil 2' de

9 serbest dereceli ataletsel ölçüm ünitesinin fiziki görünümü ve üzerindeki bileşenleri görülmektedir. Hava aracının otopilot modülü ile Inertial Measurement Unit (IMU) arasındaki haberleşme seri haberleşme üniteleri ile (UART) sağlanmaktadır. Haberleşmede ileriki bölümde anlatılacak olan projeye özel tasarlanmış haberleşme protokolü kullanılmaktadır [16]. Şekil 3.5' de kartın fiziki görünümü ve üzerindeki bileşenleri görülmektedir.

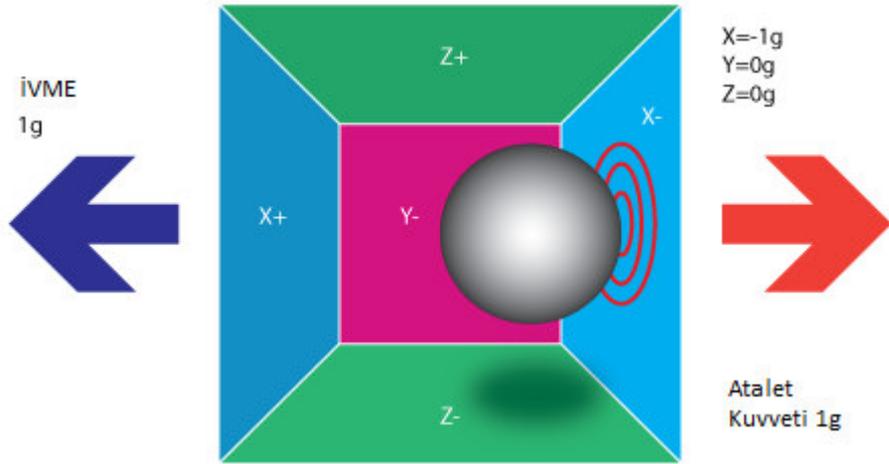


Şekil 3.5 Sparkfun Dokuz Serbest Dereceli Atalet Ölçüm Ünitesi [16]

İvme ölçerler aracın bize 3 eksendeki hızlanması hakkında bilgi verirler. Statik olarak ivmenin büyüklüğü, dinamik olarak ta ivmenin 3 eksendeki yönü anlaşılır. İvme ölçerler dijital olabileceği gibi analog da olabilir. Dijital olanlar PWM çıkışına sahiptir. PWM sinyali sabit bir frekansa sahipken birim zamandaki frekansın değeri gerilim arttıkça artar. Analog olanlar ise artırımlı gerilimle ivmeyi ifade eder. İvmelendirme arttıkça ivme ölçer çıkışındaki gerilimde doğru orantı ile artar. Genel kullanımlarda 2 eksenli ivme ölçerler yeterli iken, daha hassas 3 boyutlu konumlandırmalarda ya 2 adet 2 eksenli veya bir adet 3 eksenli ivme ölçerler kullanılır. Genel uygulamalarda 1,5 gramlık ivme ölçerler kullanılırken; araçlar, uçaklar ve robotlarda 2 gramlık, anlık durma ve anlık başlangıçlar için 5 gram ve üzeri ivme ölçerler kullanılır. Bir ivme ölçerin hassaslığı veri çıkışındaki bilginin değerini artırır. Güvenilir bir ivme okuması yapılabilmesi için, yavaş hareketli tilt açısı durumlarında 50 Hz yeterli olurken, hızlı hareket eden bir aracın ivme okumasında veya titreşim kontrolünde yüzlerce Hz' lik bantgenişliğine sahip olan ivme ölçerler gerekir. Analog ivme ölçerler 32 k Ω ' luk bir dirence sahip iken istenilen ve beklenen 10 k Ω ' luk çıkış direnci op-amp buffer devreleriyle elde edilir [17]. Şekil 3.6' da ivme ölçerin çalışma prensibi denge hali için örneklendirilmiştir. Şekil 3.7' de ileri yön için ivme değişimi örneklendirilmiştir.



Şekil 3.6 İvme ölçer Çalışma Prensipleri-1 [17]



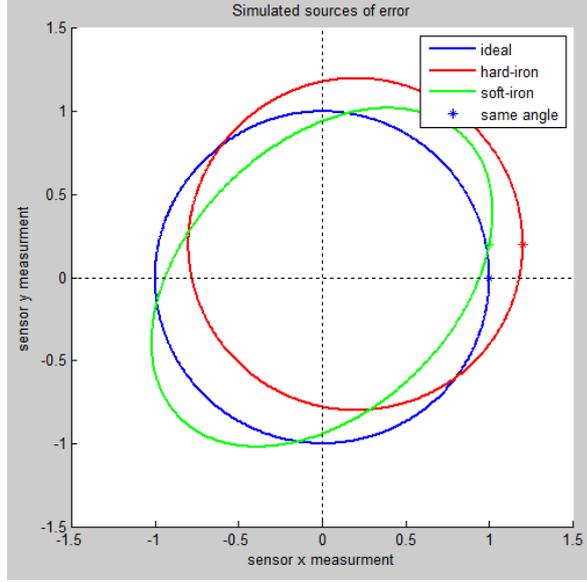
Şekil 3.7 İvme ölçer Çalışma Prensipleri-2 [17]

Jiroskoplar açısal hızları ölçerler. Bir cismin bir eksen etrafında ne kadar hızla kendi etrafında döndüğü sorusunun cevabını verir. Hareket halindeki bir cismin konumunu gözlemlerken ivme ölçerler uzaydaki konum hakkında yeterince bilgiyi veremeyebilirler. İvme ölçerlerden farklı olarak jiroskoplar yer çekiminden etkilenmedikleri için konumlandırmada ivme ölçerlerle birlikte çalışırlar. Çıktılarında açısal hız veri değerleri; ya derece/saniye ya da devir/dakika ile belirtilir. Jiroskoplarda dönme eksenini x, y, z veya yuvarlanma, yunus, kafa (sapma) açısı olarak da ifade edilir.

Geçmişte jiroskoplar füze kontrolünde, su altında seyrüseferde, uçuş kontrolünde, uzay seyrüseferinde kullanılırken bugün hareket yakalamada ve araç seyrüseferinde de kullanılmaktadır. İvme ölçerler de kullanılan seçim kriterleri jiroskoplarda

uygulanmaktadır. Ölçüm aralığı olarak seçilen jiroskop beklenen maksimum ölçme değerini geçmemelidir. Jiroskoplar' ın %95'i analog çıkışlıdır. Sadece I2C ve SPI olanlar olmak üzere bugün dijital çıkışlı olanları da vardır. I2C 400 kHz mertebesinde çalışırken SPI lar 20 MHz ve üzerinde çalışmaktadırlar. Jiroskopların çoğu iki eksenlidir. Bazıları roll (yuvarlanma) ve yaw (sapma) bilgilerini verirken bazıları roll ve pitch (yükseltme) verilerini vermektedir. Bugün artık 3 eksenli jiroskoplarda orta maliyetlerde üretilmeye başlamıştır. Güç tüketimleri birkaç yüz μA ' ler mertebesinde dir. Günümüzde kullanılan jiroskopların aynı zamanda sıcaklık çıkışıda bulunmaktadır. Bu çıkış sinyali sıcaklıkla beraber drift (kayma) etkisini telafi etmede kullanılır [18].

Akı ölçerler (Magnetometre) başlangıçta askeri, uzay, endüstri amaçlı üretilirken bugün ses ten görüntüye, seyrüseferden cep telefonlarına birbirinden farklı bir çok tüketici uygulamalarında kullanılmaktadır. MEMs ve IC teknolojisinin ilerlemesiyle de bugün pek çok ürün içerisinde yer almaktadırlar. Bu projede kullanılan 9 DoF (serbest dereceli) Razor ataletsel ölçüm ünitesi (IMU) içerisinde yer alan manyetometreler Anisotropic Magneto Resistance (AMR)' dir. Bununla birlikte, Hall Effect, Giant Magneto Resistance (GMR), Magneto Tunneling Junction (MTJ), Lorentz Force olmak üzere başka çeşitleride bulunmaktadır. AMR yapılı manyetometrelerde nikel demir alaşımı silikon temel üzerinde kullanılırken, gerekli direnç değişimi ölçümleri wheatstone köprüsü üzerinden yapılmaktadır. Manyetometrelerde adı geçen hard iron etkisi etraftaki sert cisimler manyetik cisimlere dikkat çekerken, soft iron etkisi ise görünmeyen akı akışlarını işaret eder. Şekil 3.8' de manyetometre bozucu etkileri örneklendirilmiştir. Manyetometre seçiminde temel faktörlerden birisi maliyettir. Bugün son kullanıcı manyetometrelerinde bir Dolar (\$)’ dan birkaç yüz Dolar (\$) mertebesine kadar manyetometre bulmak mümkündür. Diğer bir seçim kriteri ise hassaslıktır. Daha hassas manyetometreler beraberinde daha fazla gürültüyü meydana getirmektedir. Bugün manyetometre tasarımında hassasiyet, gürültü, güç tüketimi iç içe geçmiştir [19].



Şekil 3.8 Manyetometre Bozucu Etkileri [19]

3.5.3 XBEE Kablosuz Haberleşme Modülleri

Digi firması tarafından üretilmiş IEEE 802.15.4 protokolü üzerinden kablosuz iletişim gerçekleştiren modüllerdir. Özellikle robotik uygulamalarda sıkça kullanılan modüllerle maksimum 115200 bps hızında iletişim gerçekleştirilebilmektedir. İki nokta arasında kablosuz iletişim imkanı veren modüller projede hava aracı ile arayüz bilgisayarı arasındaki iletişimde kullanılmıştır. Proje için tasarlanmış iletişim protokolü ile arayüz ve hava aracı arasında iki yönlü iletişim kurulmuştur. Böylelikle arayüzden verilen komutlar hava aracına iletilmekte, hava aracı verileri de arayüze gönderilebilmektedir. Mevcut durumda iletişim hızı 57600 bps' tir. Alıcı kısımda Xbee modülü bir seri Universal Serial Bus (USB) modül ile birlikte çalışmaktadır. Böylelikle arayüz bilgisayarında sanal bir COM iletişim portu üzerinden iletişim gerçekleştirilebilmektedir. Şekil 3.9' de kullanılan Xbee modülünün fiziki görünümü görülmektedir [20].



Şekil 3.9 XBee Kablosuz Haberleşme Modülü [21]

3.5.4 SRF08 Ultrasonik Mesafe Ölçüm Modülü

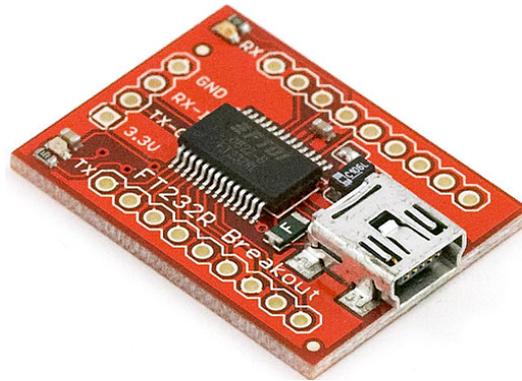
Devantech adlı firma tarafından üretilen SRF08 ultrasonik ölçüm modülü ile 3 cm ile 6 m arasında mesafe ölçümü yapılabilmektedir. Modül ile I2C protokolü üzerinden iletişim kurulabilmektedir. Şekil 3.10' de kullanılan ultrasonik modülün fiziki görünümü görülmektedir.



Şekil 3.10 SRF08 Ultrasonik Mesafe Ölçüm Modülü [22]

3.5.5 FT232 Seri-USB Çevirici

FT232 Çevirici modül ile mikrodenetleyicide 3.3 Volt seviyesinde olan UART haberleşme birimi USB haberleşmeye çevrilebilmektedir. Bilgisayara yüklenen sürücü ile sanal bir COM port oluşturulmaktadır. Böylelikle üzerinde seri port bulunmayan bilgisayarlarla USB portu üzerinden seri haberleşme sağlanabilmektedir. Şekil 3.11' de kullanılan seri/USB çevirici modülünün fiziki görünümü görülmektedir.



Şekil 3.11 FT232 Seri-USB Çevirici [23]

3.5.6 Robbe 2.4 GHz Kumanda Alıcısı ve Futaba RC Kumanda

Hava aracının elle kontrolü için Futaba marka model araç kumandası ve onunla uyumlu olan Robbe 2.4 GHz kumanda alıcısı kullanılmaktadır. Alınan kumanda sinyalleri otopilot ünitesinde sayısal değerlere çevrilerek hava aracının kontrolünde kullanılmaktadır. Kumanda sinyalleri darbe pozisyon modülasyonu (Pulse Position Modulation, PPM) ile iletilmektedir. PPM' den sayısal verilere çevrim işlemi yazılım tasarımı bölümünde anlatılacaktır. Şekil 3.12 ve Şekil 3.13' de sırasıyla kullanılan kumanda alıcı ve verici modüllerinin fiziki görünümü görülmektedir.



Şekil 3.12 Robbe 2.4 Ghz Kumanda Alıcı [24]



Şekil 3.13 Futaba 6 Kanal RC Kumanda [25]

3.5.7 Mikrokopter Elektronik Hız Kontrol Üniteleri (ESC)

Hava aracının hareketini sağlayan 6 adet fırçasız doğru akım motoru (Brushless DC Motor) bulunmaktadır. Fırçasız DC motorlar çalışmak için ayrı bir elektronik hız kontrol ünitesine (ESC) ihtiyaç duymaktadır.

Hava aracı üzerinde motorların kontrolü için 6 adet ESC bulunmaktadır. Bu ESC' lere motorlara verilecek olan komutlar I2C protokolü üzerinden iletilmektedir. Bütün ESC'ler aynı I2C veriyoluna bağlı olarak haberleşmektedir. Şekil 3.14' de kullanılan elektronik hız kontrol modülünün fiziki görünümü görülmektedir.



Şekil 3.14 Mikrokopter Elektronik Hız Kontrol Modülü [26]

3.6 Hexacopter Yazılım Mimarisi

3.6.1 Hexacopter Otopilot Yazılımı

Hexacopter otopilot yazılımı C programlama dilinde gerçekleştirilmiştir ve genel olarak aşağıdaki işlerden sorumludur:

- Haberleşme fonksiyonlarının yürütülmesi: Arayüz bilgisayarından gelen komutların çözümlenmesi ve ilgili işlemlerin yerine getirilmesi
- İlgili sensörlerden verilerin toplanması ve kullanıma hazır hale getirilmesi
- Hava aracının stabilizasyonunun sağlanması: Kontrol döngülerinin çalıştırılması



Şekil 3.15 Mikrokopter Elektronik Hız Kontrol Modülü

Bütün bu işlemlerin gerçek zamanlı olarak yürütülmesi için FreeRTOS adlı gerçek zamanlı işletim sistemi kullanılmaktadır. Açık kaynaklı olan bu işletim sistemi <http://www.freertos.org> [27] adresinden ücretsiz olarak sağlanabilir. Açık zamanlı işletim sistemi içerisinde oluşturulan görevlerle (task) farklı işlemlerin aynı anda çalışıyormuş gibi yürütülmesi sağlanmaktadır. Ayrıca, görevler arasındaki veri senkronizasyonu, kesmelerin yönetimi gibi işlemler için de işletim sistemi fonksiyonlar sunmaktadır.

Hexacopter Otopilot Yazılımında temel olarak 4 görev çalışmaktadır:

- Haberleşme Görevi
- Otopilot Görevi
- Sensör Verileri İzleme Görevi
- Sistem İzleme Görevi

3.6.2 Haberleşme Görevi

Bu görev içerisinde arayüzden gönderilen haberleşme paketlerinin çözülmesi ve ilgili verilerin güncellenmesi, otopilottan arayüze veri paketlerinin gönderilmesi işlemleri yapılmaktadır. Şekil 3.16' da arayüz bilgisayarı ile STM32F4 Discovery kartı arasındaki iletişim resmedilmiştir.

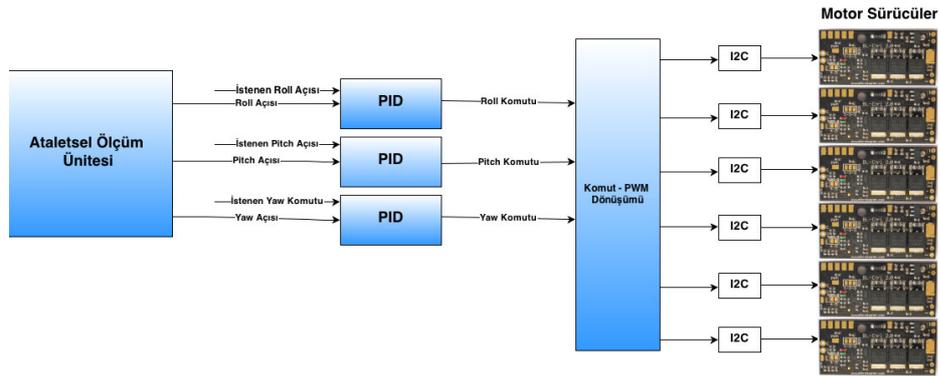


Şekil 3.16 Arayüz Bilgisayarı Otopilot Arası İletişim

Hexacopter otopilot ünitesi ile arayüz bilgisayarı ve mini PC ünitesi arasındaki iletişimde Çizelge 3.1' de anlatılan haberleşme protokolü kullanılmıştır. Bu protokole ait haberleşme paketlerinin oluşturulması, gelen haberleşme paketlerinin çözülmesi gibi işlemlerden sorumlu fonksiyonlar haberleşme görevi içerisinde çalışmaktadırlar. Haberleşme görevinin periyodu 100 ms' dir.

3.6.3 Otopilot Görevi

Bu görev içerisinde sensörlerden alınan verilerin kullanılarak hava aracının üç eksen de duruş kontrolünü sağlayan otopilot kontrol çevrimlerinin (PID) işletilmesinden sorumludur. Ataletsel ölçüm ünitesinden gelen roll, pitch ve yaw açısı bilgileri ve mini PC ya da arayüz bilgisayarından gönderilen duruş komutları PID kontrol fonksiyonlarına uygulanarak motorlara iletilecek olan hız komutları üretilmektedir. Kontrol döngülerinden her bir motor için elde edilen hız değerleri I2C protokolü ile elektronik hız kontrol ünitelerine iletilmektedir. Şekil 3.17' de otopilot görevinde kullanılan PID çevrimlerine ait çalışma şeması görülmektedir.



Şekil 3.17 PID Çevrimleri

3.6.4 Sensör Görevi

Bu görev içerisinde ultrasonik mesafe ölçüm sensörü, ataletsel ölçüm ünitesi ve kumanda alıcısından alınan verilerin alınarak diğer görevlerin kullanımı için

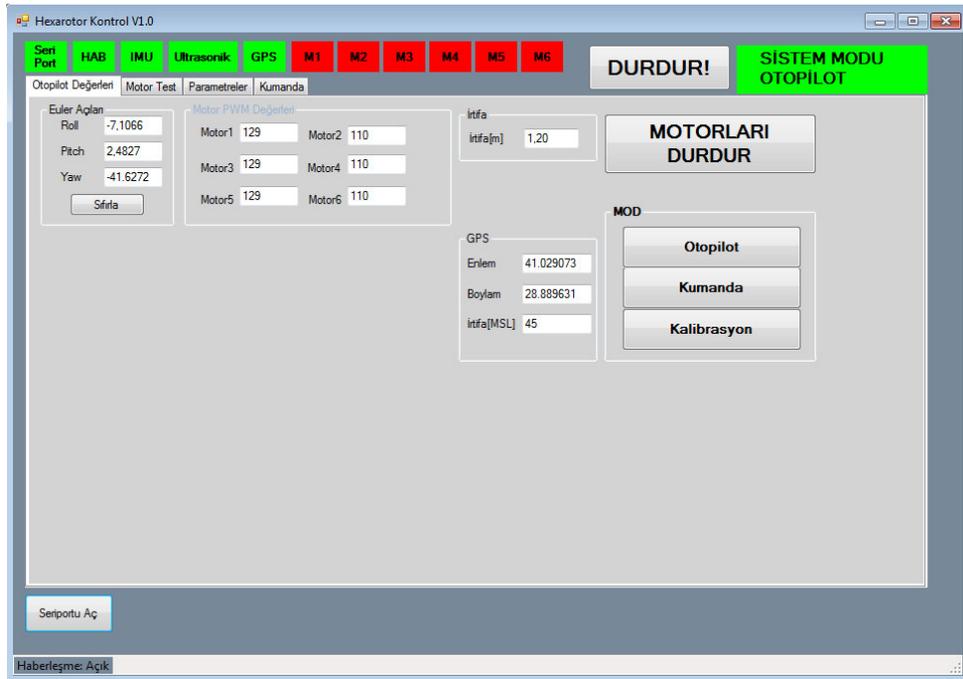
hazırlanması sağlanmaktadır. Sensör verilerinin sınır kontrolleri ve filtreleme işlemleri de bu görev içerisinde gerçekleştirilmektedir.

3.6.5 Sistem İzleme Görevi

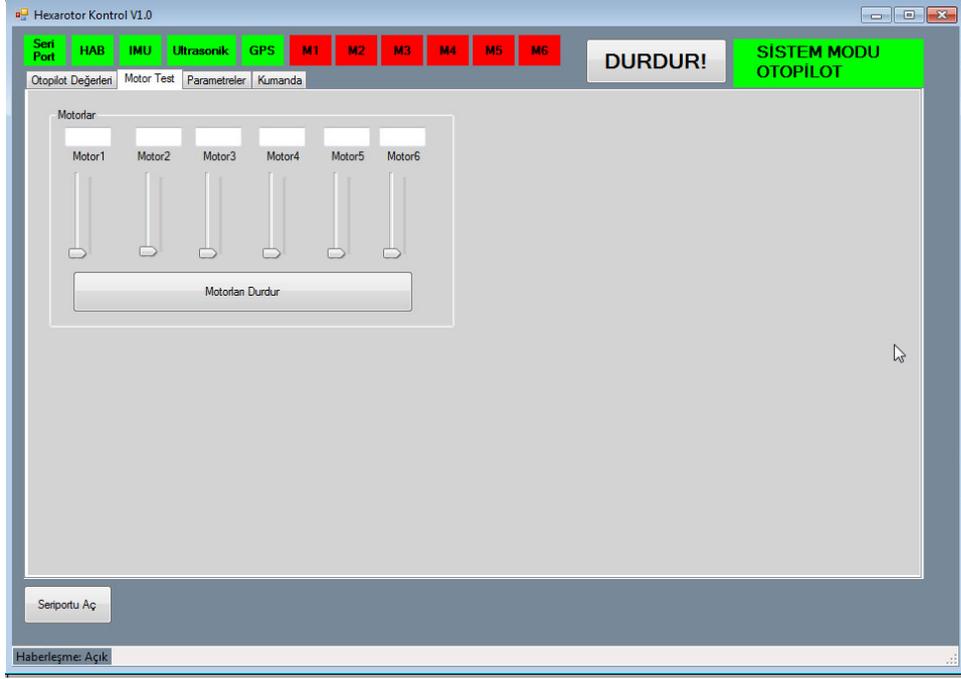
Bu görev içerisinde sistem içerisindeki diğer görevlerin periyotlarındaki düzensizlikler takip edilmektedir. Eğer görevlerin çalışma periyotlarında bozulmalar oluşmaya başlarsa bu görev sistemin yeniden başlatılmasını sağlayarak sistemin çalışmasını düzenlemektedir. Ayrıca sensörlerden gelen veri akışı da kontrol edilip eğer sensörlerde herhangi bir problem tespit edilirse yine sistem arıza durumuna geçirilip acil durum prosedürlerinin çalıştırılmasını sağlamak bu görevin sorumlulukları arasındadır.

3.6.6 Hexapilot Arayüz Yazılımı

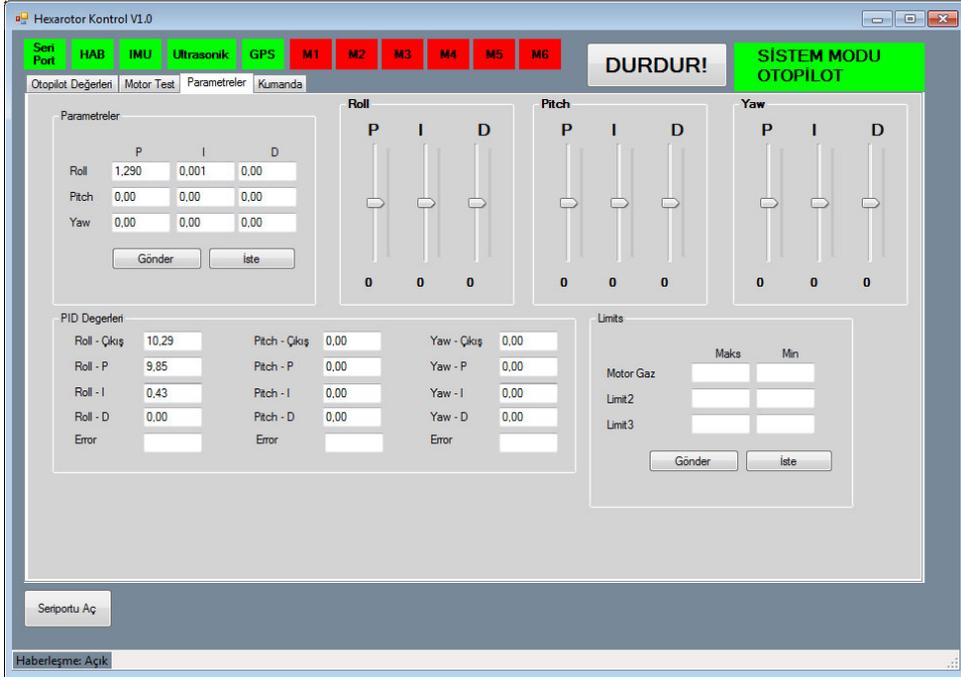
Hexacopter arayüz yazılımı kullanıcının hava aracına verdiği komutların gönderilmesi ve hava aracından gelen verilerin görüntülenmesinden sorumludur. C# programlama diliyle Microsoft Visual Studio Express geliştirme ortamında gerçekleştirilmiştir. Şekil 3.18' de arayüz yazılımı gösterge sekmesi, Şekil 3.19' da autopilot sekmesi, Şekil 3.20' de parametre sekmesi, Şekil 3.21' de uzaktan kumanda sekmesi görülmektedir.



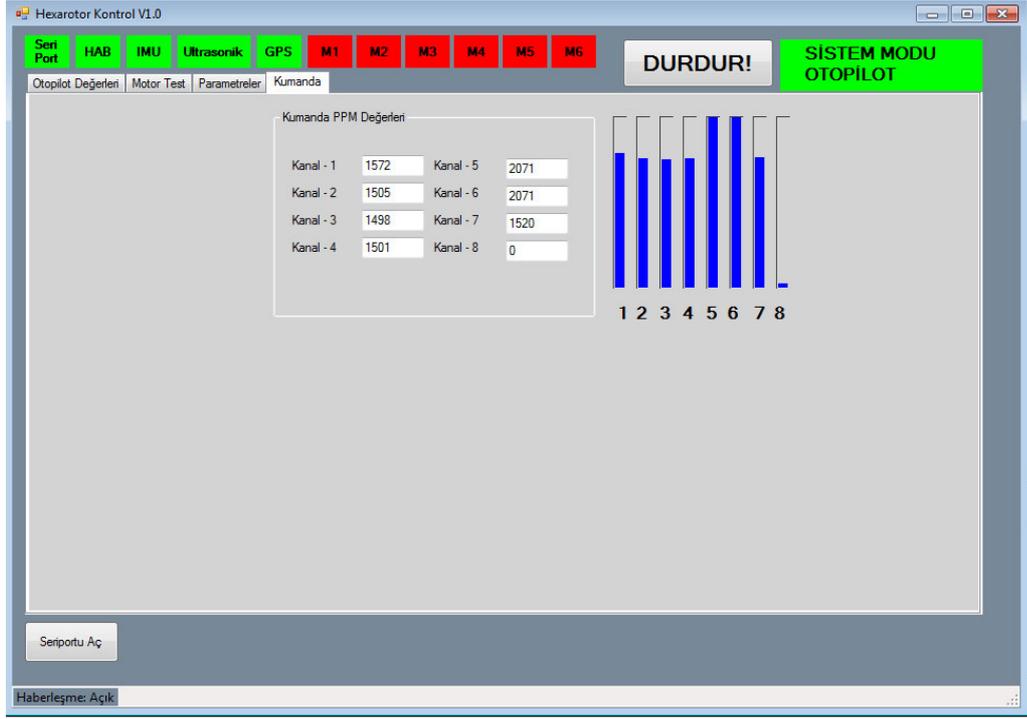
Şekil 3.18 Arayüz yazılımı-1



Şekil 3.19 Arayüz yazılımı-2



Şekil 3.20 Arayüz yazılımı-3



Şekil 3.21 Arayüz yazılımı-4

Arayüz üzerinde hava aracına ait duruş açıları, sensör çıkış değerleri, irtifa değeri, motorlara uygulanan gaz komutu değerleri, kumanda giriş değerleri gibi veriler görüntülenebilmektedir. Ayrıca modüllerle ilgili bağlantı durumları da arayüzün üst tarafındaki kutularda kırmızı ve yeşil renklerle ifade edilmektedir. Bir modülle iletişim kesildiğinde kutular kırmızı renge dönmektedir.

3.6.7 Hexapilot Yazılım Bileşenleri Yapısı

3.6.7.1 FreeRTOS Gerçek Zamanlı İşletim Sistemi

Açık kaynaklı gerçek zamanlı işletim sistemleri arasında en çok kullanılan FreeRTOS ücretsiz olarak kullanıma sunulmuştur. 30'dan farklı işlemci mimarisinde çalışan işletim sistemi oldukça düşük sistem kaynağı tüketmektedir. Açık kaynaklı ve ücretsiz oluşu nedeniyle projede tercih edilmiştir [26].

FreeRTOS'un temel özellikleri şunlardır:

- Pre-emptive ve Co-operative zamanlama desteği
- Konfigure edilebilir ve ölçeklendirilebilir
- Öncelik devralmalı mutex yapılarını destekleme

- İkili ve semaphore mutex yapılarını destekleme
- Derleyici bağımsız
- 6K ile 10K ROM alanı ihtiyacı
- Kullanışlı mesaj geçirme mekanizmaları

FreeRTOS işletim sistemi ile kontrol, haberleşme, sensörlerden veri alınması gibi gerçek zaman kısıtlamaları içeren işlemler senkronize bir biçimde kolayca yürütülebilmektedir. İşletim sistemi görevleri arasındaki iletişim global değişkenlerle sağlanmaktadır.

3.6.7.2 Hexacopter Otopilot Haberleşme Protokolü

Hexacopter Otopilotu ile kullanıcı arayüz yazılımı arasındaki haberleşme için özel bir haberleşme protokolü tasarlanmıştır. Bu protokol, seri haberleşme üzerinden çalışmaktadır. Oluşturulan genel paket yapısı bütün veri iletiminde kullanılabilir. Çizelge 3.1’ de protokolda kullanılan haberleşme paketi yapısı görülmektedir.

Çizelge 3. 1 Haberleşme paketleri yapısı

Bayt 0	Başlık 1 - 0x55
Bayt 1	Başlık 2 - 0xAA
Bayt 2	Boyut
Bayt 3	Paket Tipi
Bayt 4	Veri -0
Bayt
Bayt ..	Veri n-1
Bayt ...	Veri n
Bayt n -1	CRC 1. Bayt
Bayt n	CRC 2. Bayt

3.6.7.4 Otopilot IMU Sürücüsü

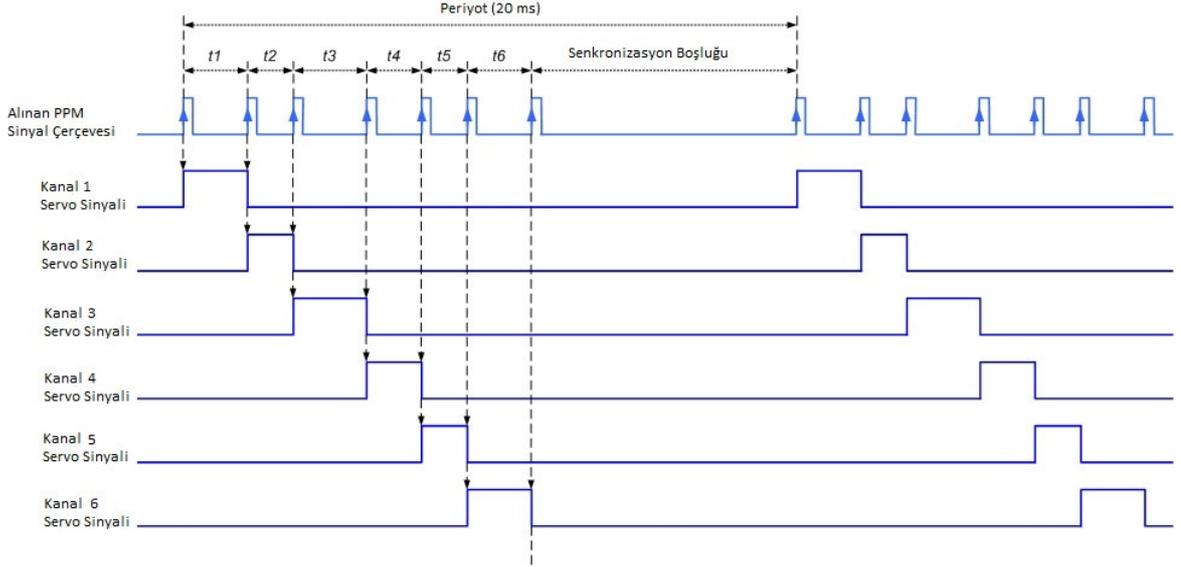
Otopilot IMU sürücüsü de yukarıda bahsedilen haberleşme sürücülerini kullanmaktadır. IMU'den gelen açı ve sensör değerleri çözülerek ilgili veri yapıları güncellenmektedir.

3.6.7.5 SRF08 Ultrasonik Mesafe Ölçüm Modülü Sürücüsü

SRF08 Ultrasonik modülü I2C protokolü üzerinden iletişim kurmaktadır. Sürücü de I2C fonksiyonlarını içeren alt seviye fonksiyonlardan oluşmaktadır. Öncelikle SRF08'in başlangıç ayarları yapmakta ardından periyodik olarak mesafe ölçümü gerçekleştirilmektedir. SRF08 ile iletişim ayrı bir görev içerisinde yürütülmektedir. Mesafe verisi okunduktan sonra ilgili veri yapısı güncellenmektedir.

3.6.7.6 RC Kumanda Alıcısı Sürücüsü

RC Kumanda alıcısı 6 Kanallı Futaba kumandadan gelen komut değerlerini PPM ile tek bir hat üzerinden iletmektedir. Bu değerlerin çözülmesi için darbelerin zaman değerlerinin tespit edilmesi gerekmektedir. Bunun için STM32F407 mikrodenetleyisinin zamanlayıcı modüllerinden bir tanesi giriş darbe ölçümü (input capture) modunda çalıştırılarak darbelerin uzunlukları ölçülmektedir. Bunun için PBO pini kullanılmaktadır. Her bir yükselen kenar sinyalde kesme oluşmakta ve iki kesme arasındaki süre kaydedilmektedir. Böylelikle 6 kanala ait darbe süreleri ölçülmekte ve ilgili veri yapılarına kaydedilmektedir. RC kumanda sinyalleri hava aracının yarı-otonom kontrolünde kullanılacaktır. Şekil 3.23' de okunan kumanda sinyalleri görülmektedir.



Şekil 3.23 Kumanda Alıcısından okunan PPM sinyal [25]

3.6.7.7 Hexacopter Arayüz Yazılımı

Hexacopter arayüz yazılımı hava aracı ile kullanıcılar arasındaki iletişimi sağlayan bir makina-insan arayüz yazılımıdır. Visual Studio geliştirme ortamında C# programlama diliyle geliştirilmektedir.

Hava aracından haberleşme protolü ile RF alıcı-verici üzerinden gelen veriler çözülerek kullanıcıya görüntülenmesi, arayüz üzerinden kullanıcılar tarafından verilen komutların yine aynı yolla hava aracına iletilmesi sağlanmaktadır.

Arayüz üzerinde IMU verileri, kumanda kanallarının değerleri, motorlara gönderilen PWM değerleri ve PID parametreleri de görüntülenmektedir.

3.6.7.8 Yazılım Mimarisinde Gelecekte Planlanan Çalışmalar

Tasarlanan sistem prototip aşamasında olup bütün komponentleri içeren tek bir kart tasarımı gerçekleştirilecektir. Sistem üzerinde içerdiği komponentler itibariyle otonom araç çalışmaları için genel ve çok donanımlı bir platform olma potansiyeli taşımaktadır. Bundan sonra insanız hava ve kara aracı çalışmaları kullanılması planlanmaktadır.

PID ve MODEL ÖNGÖRÜLÜ KONTROL**4.1 PID Denetleyicinin Hexacoptere Uygulanması**

Bu bölümde öncelikle çoklu rotorlu helikopterde kullanılan Hacker marka A20-22L model helikopter motoru için PID katsayıları bulunmuş daha sonra sistemde motor katsayıları olarak uygulanmıştır.

Motor Parametreleri ve Matematiksel Modeli [28]:

$$\tau_m = \frac{3R\phi J}{K_e \cdot K_t} \quad \tau_e = \frac{L}{3R} \quad (4.1)$$

$$G(s) = \frac{\frac{1}{K_e}}{\tau_m \cdot \tau_e \cdot s^2 + \tau_m \cdot s + 1} \quad (4.2)$$

Burada,

τ_m : mekanik zaman sabiti

τ_e : elektrik zaman sabiti

K_e (v-sec/rad): Elektriksel moment

K_t (N-m/A): Mekanik moment

J : Motor ataleti

R : Fazlar arası direnç

L : Fazlar arası endüktans

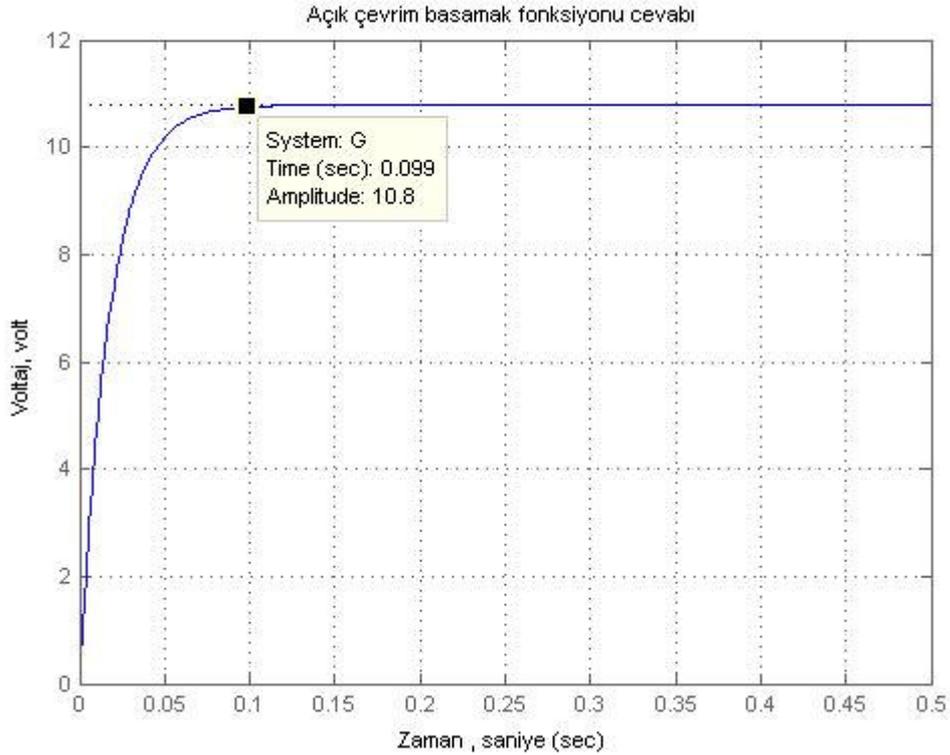
Matlab ortamında parametreler girilince elde edilen transfer fonksiyonu:

10.79

$$G(s) = \frac{10.79}{2.66e-006 s^2 + 0.0171 s + 1}$$

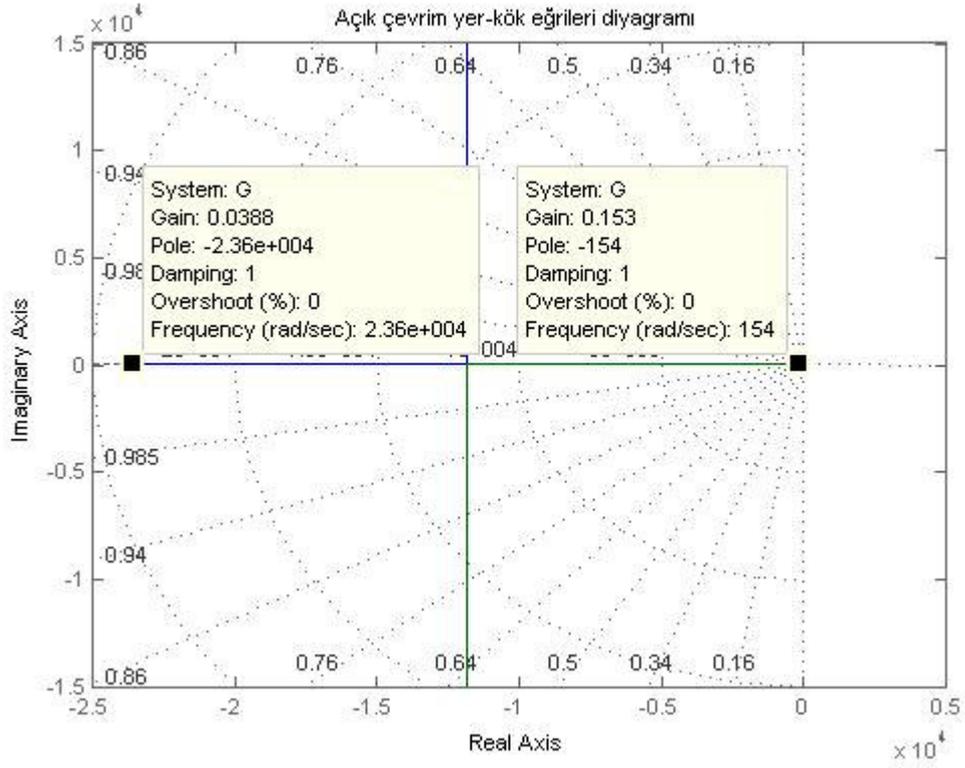
$$2.66e-006 s^2 + 0.0171 s + 1$$

Sonrasında matlab fonksiyonları kullanılarak motora ait açık çevrim ve kapalı çevrim motor özellikleri (basamak cevabı, bode ve nyquist diyagramları, kök-yer eğrileri ve ziegler-nichols) kullanılarak motor PID parametreleri elde edilmiştir. Bu motor transfer fonksiyonun belirtilen çıktıları aşağıdaki şekillerde gösterilmiştir. Şekil 4.1' de fırçasız doğru akım motoruna ait açık çevrim basamak cevabı sisteme ait açık çevrim fonksiyon kullanılarak çizdirilmiştir.

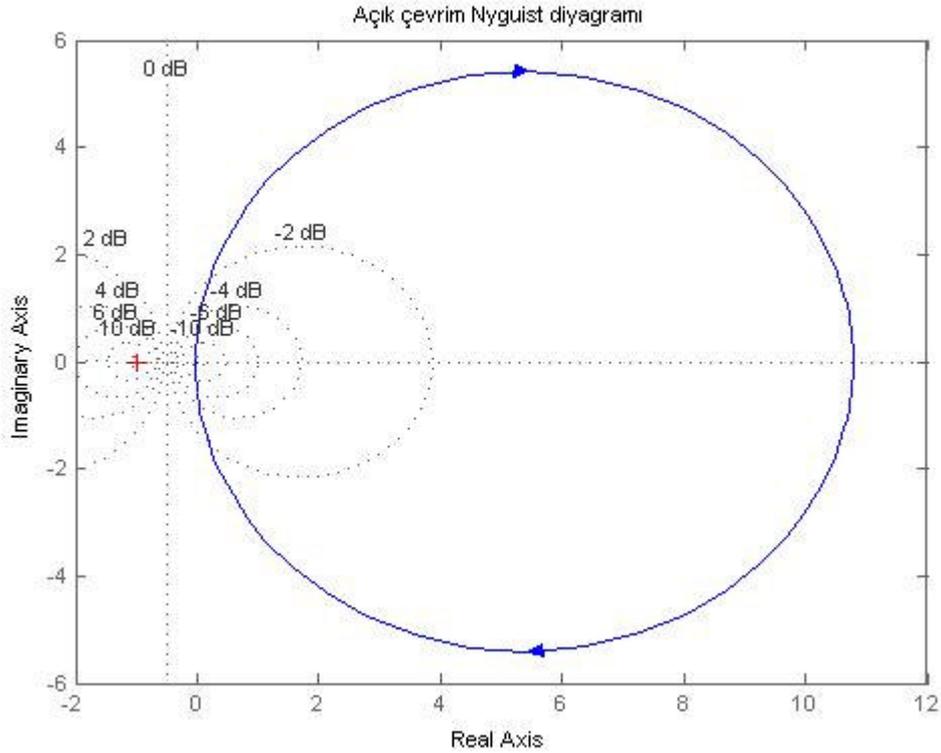


Şekil 4.1 Fırçasız doğru akım motoruna ait açık çevrim basamak cevabı

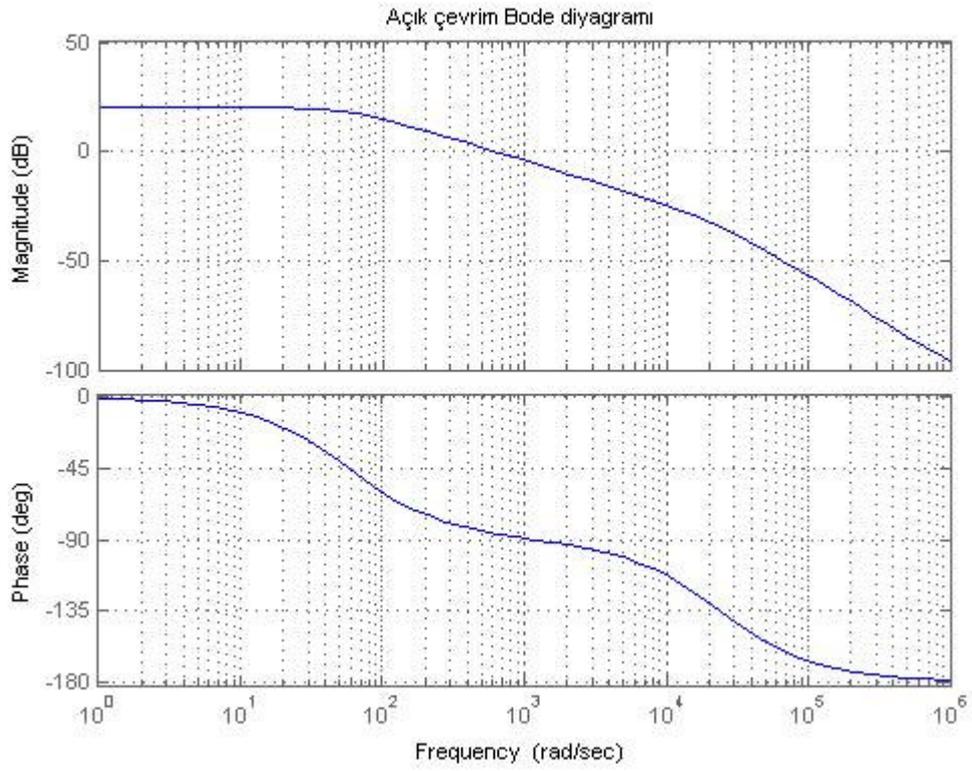
Şekil 4.2, Şekil 4.3, Şekil 4.4, Şekil 4.5 ve 4.6 da sistem hakkında daha fazla bilgi sahibi olabilmek için fırçasız doğru akım motoruna ait açık çevrim kök-yer eğrisi cevabı, ait açık çevrim Nyquist cevabı, açık çevrim Bode cevabı, kapalı çevrim kök-yer eğrisi, kapalı çevrim Nyquist cevabı yer almaktadır.



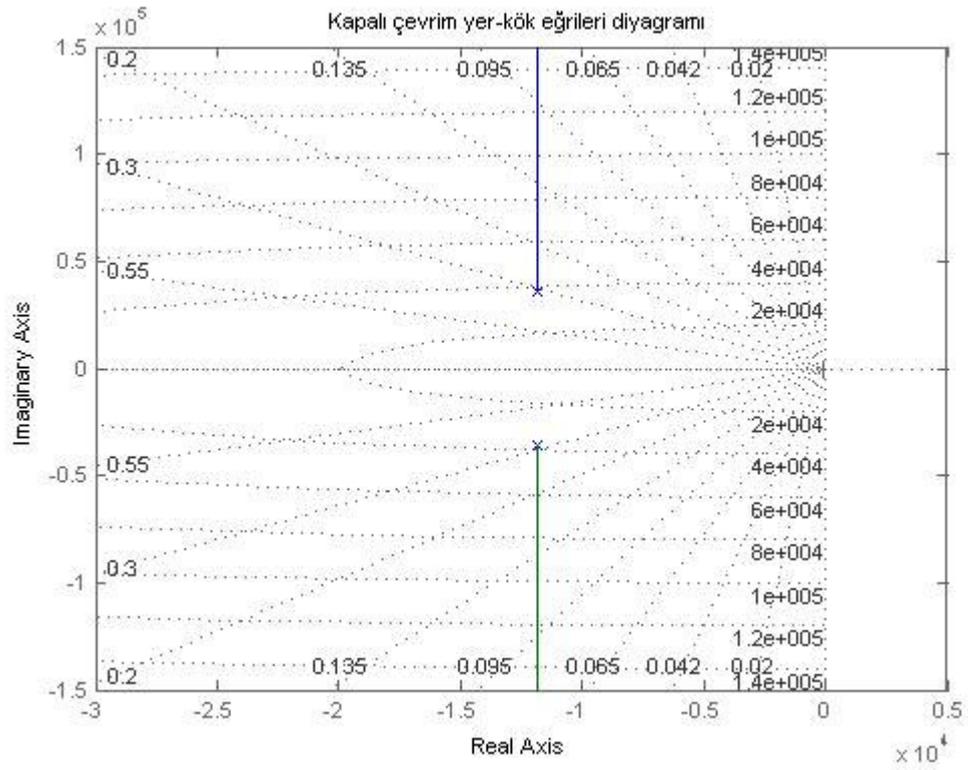
Şekil 4.2 Fırçasız doğru akım motoruna ait açık-çevrim kök-yer eğrisi cevabı



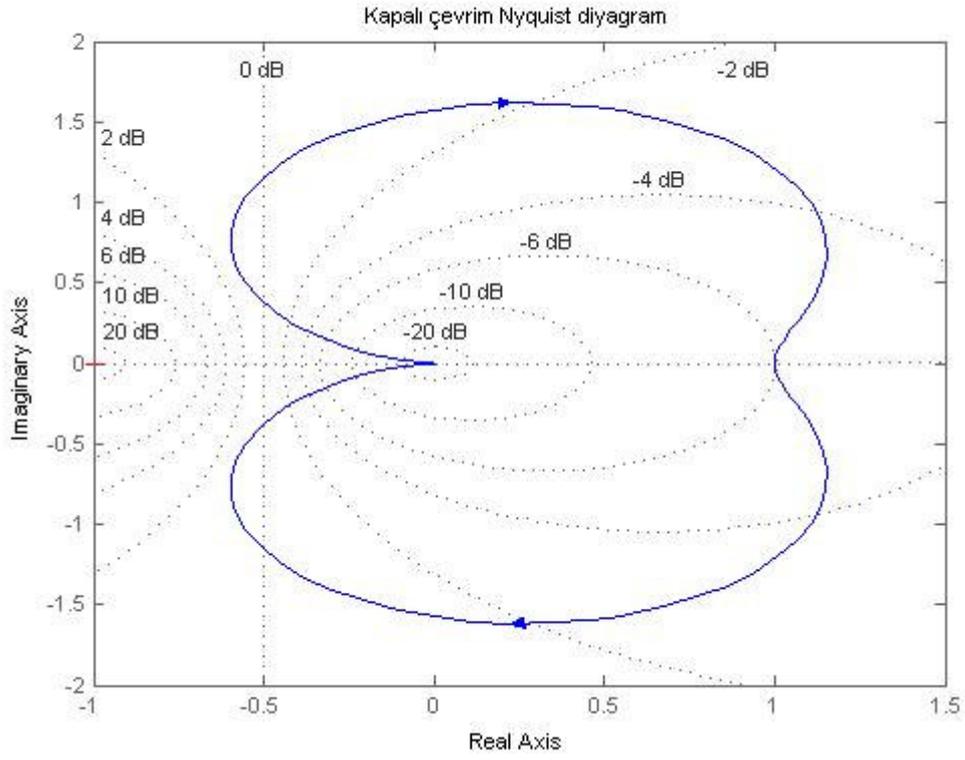
Şekil 4.3 Fırçasız doğru akım motoruna ait açık çevrim Nyquist cevabı



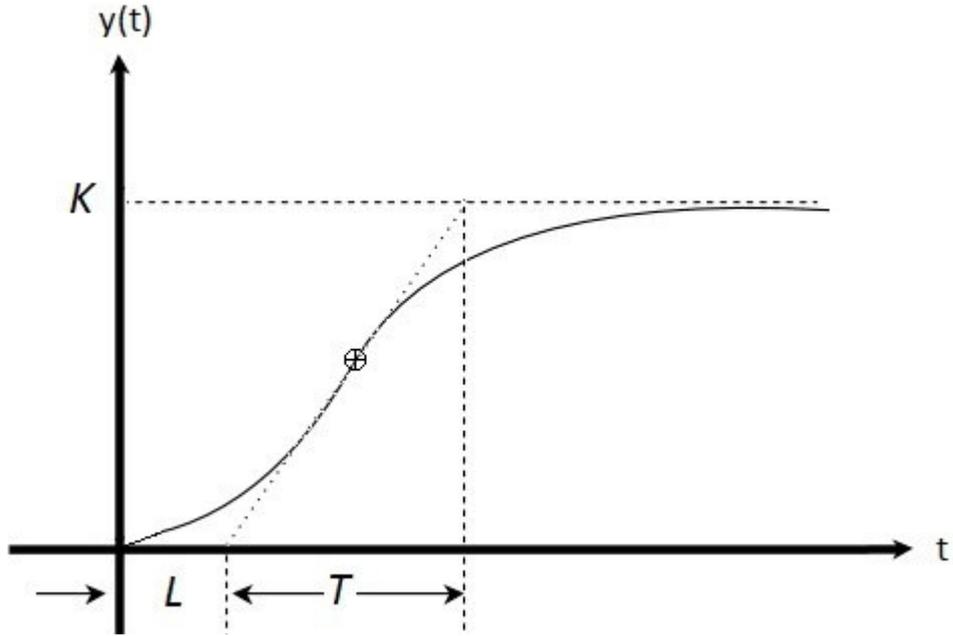
Şekil 4.4 Fırçasız doğru akım motoruna ait açık çevrim Bode cevabı



Şekil 4.5 Fırçasız doğru akım motoruna ait kapalı çevrim kök-yer eğrisi



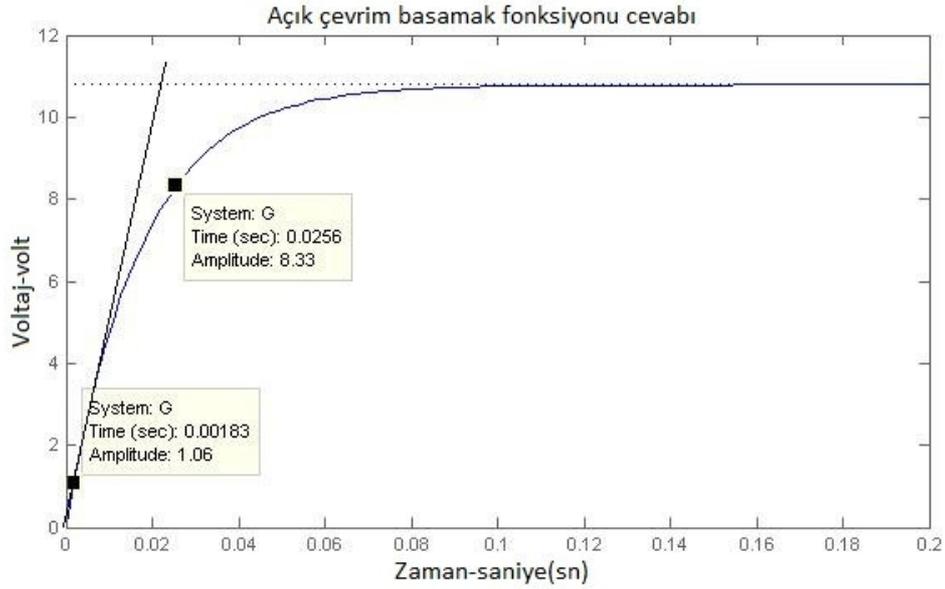
Şekil 4.6 Fırçasız doğru akım motoruna ait kapalı çevrim Nyquist cevabı



Şekil 4.7 PID parametrelerinin Ziegler Nichols ile belirlenmesi

Şekil 4.7' de PID parametreleri için gerekli olan zaman sürelerinin nasıl hesaplandığı gösterilmiştir. Sistemin basamak cevabı çizdirildikten sonra, PID katsayıları Ziegler Nichols kuralları gereği eşitlik 4.3' teki gibi belirlenir.

$$\begin{aligned}
 & K_P & T_I = \frac{K_P}{K_I} & T_D = \frac{K_D}{K_P} \\
 \text{PID} & K_P = 1,2 \times \frac{T}{L} & T_I = 2 \times L & T_D = 0,5 \times L
 \end{aligned} \tag{4.3}$$



Şekil 4.8 Motora ait PID parametrelerinin belirlenmesi

Şekil 4.8 den elde edilen bilgilerle motorlara ait parametreler $L=0,0025$ s ve $T=0,0261$ s olduğu gözlenir. Bu bilgiler ışığında PID katsayıları sırası ile 12,528 (K_P), 2505,6 (K_I) ve 0,01566 (K_D) bulunmuştur. Bu değerler daha sonra motor sürücüleri içerisinde kullanılmıştır. Hesaplamalar ise, aşağıdaki şekilde yapılmıştır.

$$T = 0,0261 \text{ s}$$

$$L = 0,0025 \text{ s}$$

$$T_I = 2 \times L \rightarrow T_I = 0,005 \text{ s} \quad \left| \quad T_D = 0,5 \times L \rightarrow T_D = 0,00125 \text{ s}$$

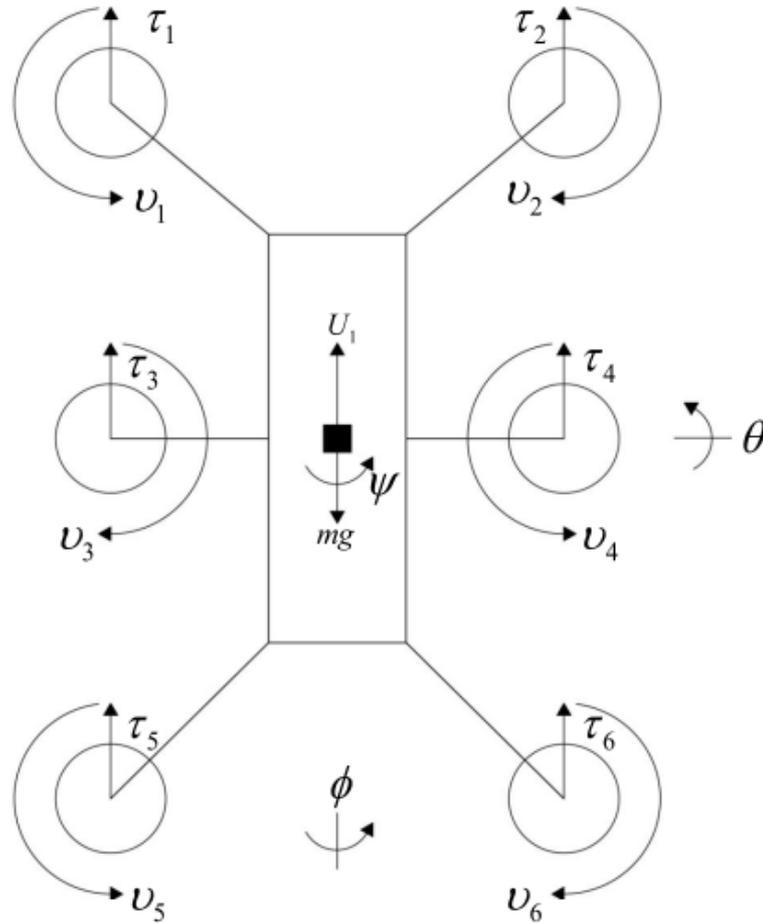
$$K_P = 1,2 \times \frac{T}{L} \rightarrow K_P = 1,2 \times \frac{0,0261}{0,0025} = 12,528$$

$$K_I = \frac{K_P}{T_I} \rightarrow K_I = \frac{12,528}{0,005} \rightarrow K_I = 2505,6 \text{ s}^{-1}$$

$$K_D = K_P \times T_D \rightarrow K_D = 12,528 \times 0,00125 \rightarrow K_D = 0,01566 \text{ s}$$

4.2 Basitleştirilmiş Sistem Dinamikleri

Altı rotorlu helikopterin Euler-Langrange Metoduna göre ayrıntılı olarak modelinin çıkarılması Ek B' de yer almaktadır. Bu bölümde basitleştirilmiş sistem dinamikleri kullanılarak hexacopter' in PID kontrolü açıklanmıştır. Basitleştirilmiş sistem dinamiklerinde çapraz sistem dinamiklerinin (kuplajları) olmadığı, itme ve çekme kuvvetlerinin sisteme etkisiz olduğu, yine ayrıntılı modellemede olduğu gibi sistemin homojen ve ağırlık merkezinin sistemin geometrik ortasında olduğu düşünülmüştür. Şekil 4.9' da, sisteme etkiyen her bir rotora ait τ itme gücünü (thrust), u eksenindeki döndürme momentini ve θ sistemin yunuslama (pitch) açısını, ϕ yuvarlanma (roll), ψ ise sapma -kafa açısı- (yaw) açılarını göstermektedir. Bu noktadan hareketle (4.4)' ten (4.8)' e kadar olan eşitliklerde sistemin nasıl modellendiği ve sistemin kontrol blok şemalarının nasıl oluşturulduğu gösterilmiştir [29].



Şekil 4.9 Altı rotorlu helikopterin gövde hareketleri ve uygulanan kuvvetler

$$\frac{d}{dt} \mathbf{x} = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \rightarrow \begin{array}{l} \text{Konum} \\ \text{Çizgisel hız} \\ \text{RPY Açılıarı} \\ \text{Açısal hız} \end{array} \quad (4.4)$$

Durum vektörü

Eşitlik (4.4)' de yer alan durum vektöründeki x, y, z yer eksenindeki konumu, $\dot{x}, \dot{y}, \dot{z}$ yer eksenindeki eksele çizgisel hızları, ϕ, θ, ψ gövde eksenindeki yuvarlanma, yunuslama, sapma açılarını, $\dot{\phi}, \dot{\theta}, \dot{\psi}$ gövde eksenindeki yuvarlanma, yunuslama, sapma açısal hızlarını göstermektedir.

$$\begin{aligned} u_1 &= [\tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5 + \tau_6] \quad \tau: \text{thrust} & u_1 &\rightarrow mg \\ u_2 &= l [\tau_5 - \tau_6] \quad l: \text{ağırlık merkezine olan uzaklık} & u_2 &\rightarrow \text{roll } (\phi) \\ u_3 &= l [\tau_5 - \tau_1] & u_3 &\rightarrow \text{pitch } (\theta) \\ u_4 &= [v_1 + v_4 + v_5 - (v_2 + v_3 + v_6)] \quad v: \text{moment} & u_4 &\rightarrow \text{yaw } (\psi) \end{aligned} \quad (4.5)$$

Eşitlik (4.5) yer alan kontrol işaretleri ise sırasıyla kalkış, yuvarlanma (roll), yunuslama (pitch) ve sapma (yaw) hareketlerini düzenleyen kontrol işaretleridir. Kalkış esnasında yuvarlanma, yunuslama ve sapma açı miktarının sıfır olduğu, sistemin dengede olduğu düşünülmüştür. Bu sebeple kalkış esnasında gövde açılarını düzenleyen kontrol işaretleride sıfırdır. u_1 vektörünün uygulandığı kalkış esnasında motorlar yarı devirli olarak yaklaşık 4200 devir/dakika da çalıştırılır. u_2, u_3, u_4 kontrol işaretlerinin uygulanabilmesi için sistemin havada yer eksenine göre bulunduğu noktadaki gövde açılarının sıfır noktasından uzaklaşması gerekir. Böyle bir durumda sistem gerekli kontrol vektörlerini hesaplayarak, ilgili itme (thrust) kuvvetlerini rotora iletir. Aracın tekrardan sıfır noktasına dönüşünü sağlar.

Eşitlik (4.6)' da ise sırası ile ilk kısımda gövde açısal konumları kontrol eden kontrol işaretleri, ikinci kısımda ise açısal hızlara ait değişkenler tanımlanmıştır. Geleneksel olarak yer koordinat sistemi Rxyz şeklinde tanımlanır. Sapma hareketini öne çıkarmak için Rxyz koordinatları yerine Rzxy koordinat sistemini kullanacak olursak eşitlik (4.6)' nin son kısmındaki dönel sistem karşımıza çıkar. Sistemde sapma miktarı sıfır ve yunuslama açısı çok çok küçük kabul edilir ise, sistemin yuvarlanma hareket denklemleri eşitlik (4.7)' nin birinci kısmındaki halini alır. Eşitlik (4.7)' nin ikinci kısmında tüm eksenlerdeki açı değerleri sıfır kabul edildiğinde sistem $g - u_1$ ivmesi ile kalkışını gerçekleştirir. Yapılan bu hareket tanımı sadece z eksenindeki kalkışa tekabül eder.

$$\mathbf{u} = [u_1, u_2, u_3, u_4]^T$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \frac{\sin\psi}{\cos\phi} & \frac{\cos\psi}{\cos\phi} & 0 \\ \sin\psi \tan\phi & \cos\psi \tan\phi & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$
(4.6)

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$\ddot{x} = -u_1 \cos\phi \sin\theta$$

$$\ddot{y} = u_1 \sin\phi$$

$$\ddot{z} = g - u_1 \cos\phi \cos\theta$$
(4.7)

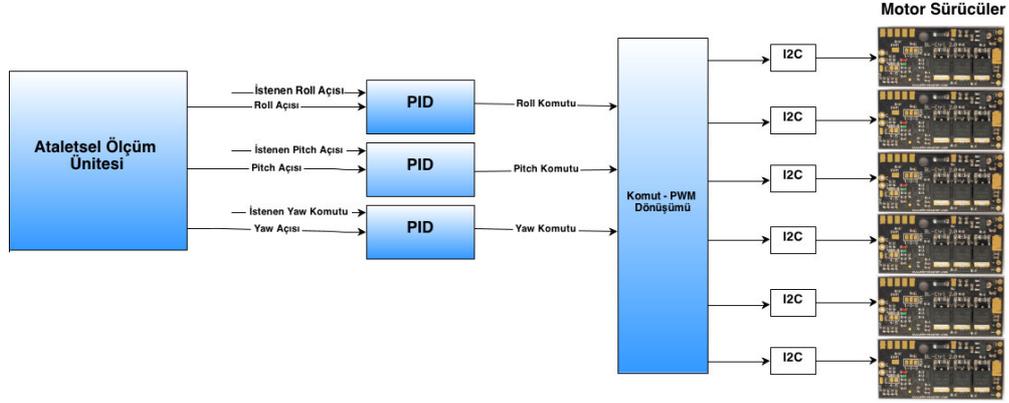
Son durumdaki sistem durum vektörü ise, kalkışta gövde açılarının sıfır olması durumunda eşitlik (4.8)'de de görüldüğü gibi $g - u_1$ ivmesi ile hareket eder.

$$\dot{\mathbf{x}} = \frac{d}{dt} [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \phi \ \theta \ \psi \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$$

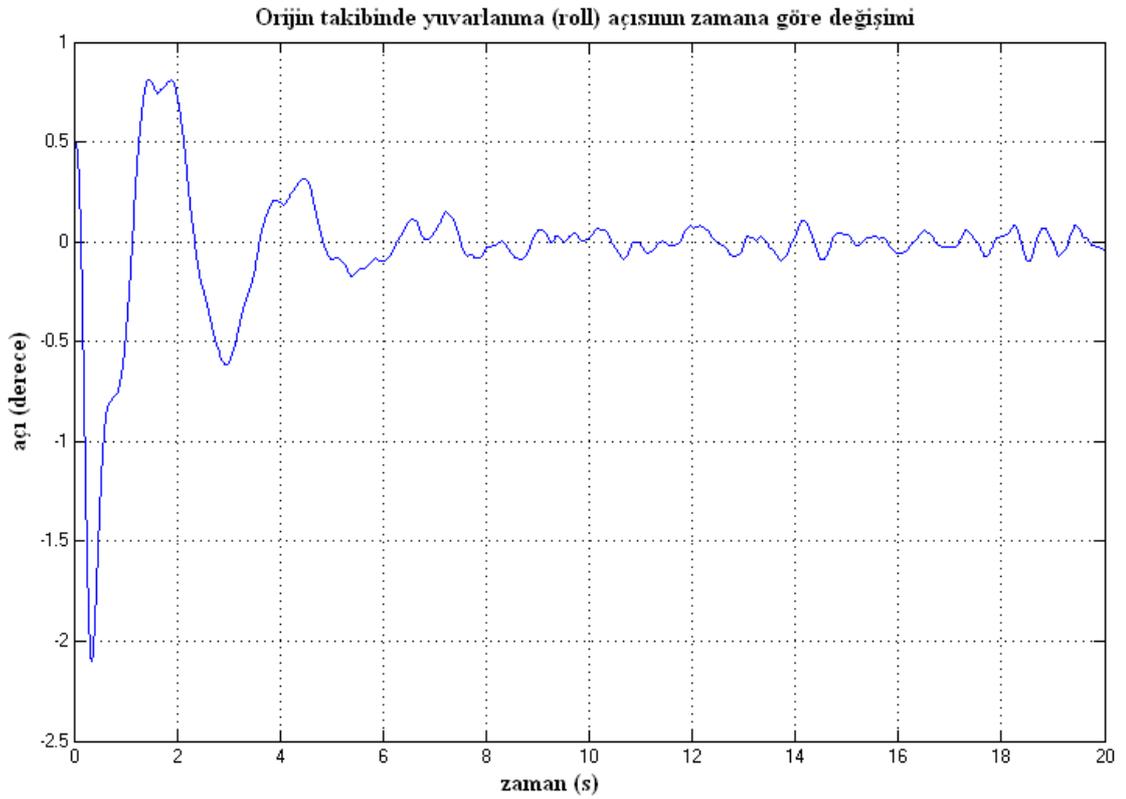
$$\ddot{\mathbf{x}} = [\dot{x} \ \dot{y} \ \dot{z} \ -u_1 \cos\phi \sin\theta \ u_1 \sin\phi \ g - u_1 \cos\phi \cos\theta \ \dot{\phi} \ \dot{\theta} \ \dot{\psi} \ u_2 \ u_3 \ u_4]^T$$
(4.8)

4.3 Hexacopter PID Çıktıları

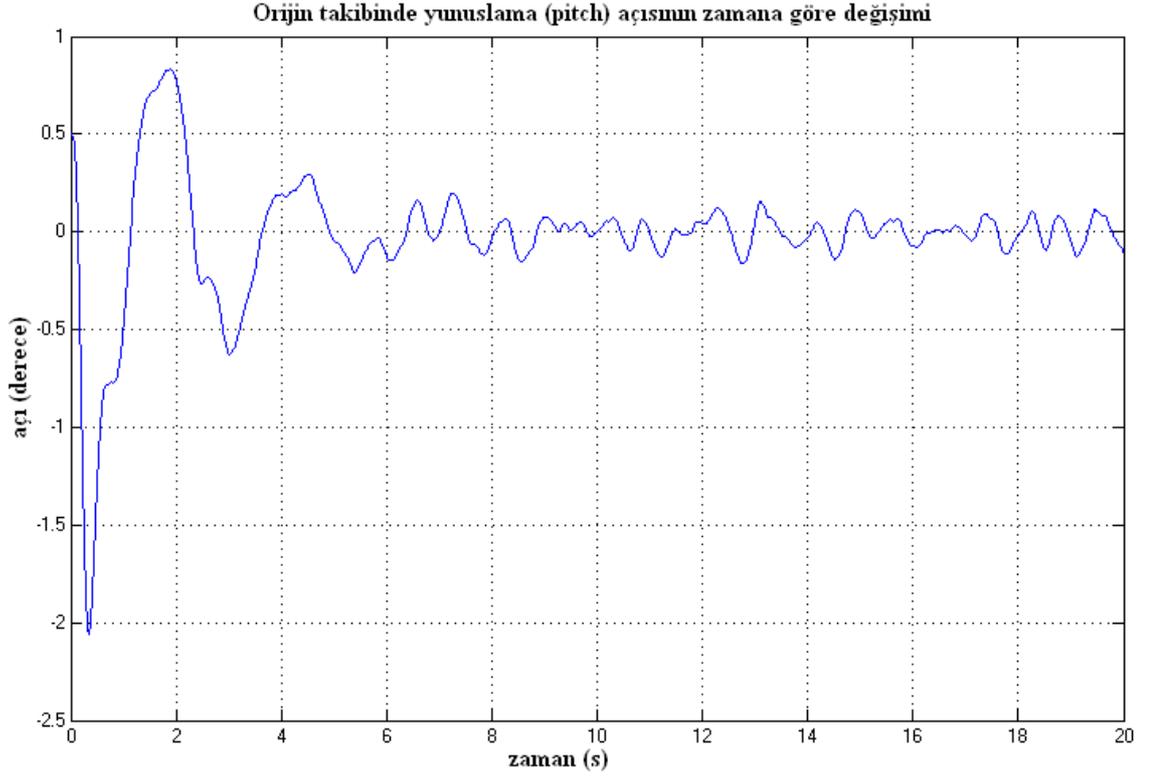
Hexacopter PID çevrimleri Şekil 4.10' da belirtildiği üzere tasarlanmıştır. Sistem MATLAB ortamında simüle edilmiş, gövde eksenleri açı değişimleri (yuvarlanma, yunuslama, sapma) Şekil 4.11, Şekil 4.12 ve Şekil 4.13' de görülebilir.



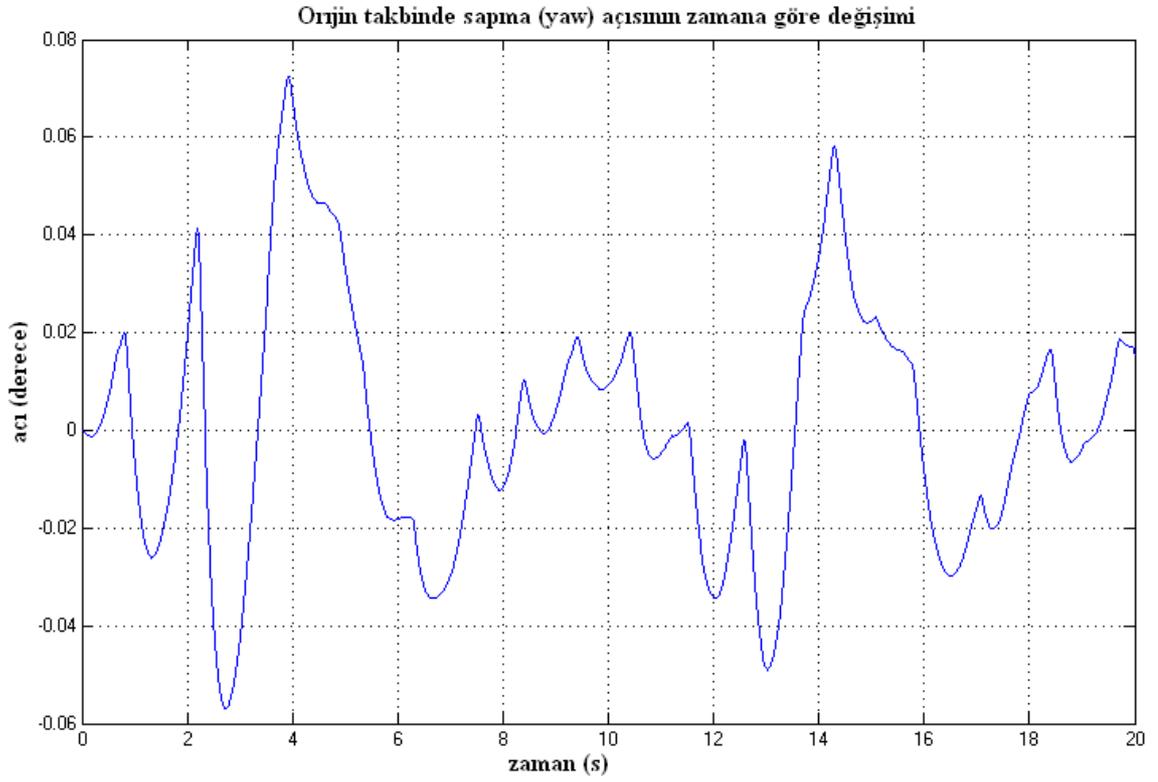
Şekil 4.10 Hexarator PID çevrimleri



Şekil 4.11 Orijin takibinde yuvarlanma (roll) açısının zamana göre değişimi



Şekil 4.12 Orijin takibinde yunuslama (pitch) açısının zamana göre değişimi



Şekil 4.13 Orijin takibinde sapma (yaw) açısının zamana göre değişimi

4.4 Model Öngörülü Kontrol

4.4.1 Çoklu Rotorlu Helikopterlerde Hareketin Modellenmesi

Ek-2 de de sistemin hareket denklemlerinin nasıl elde edildiği belirtilmiştir. Çoklu rotorlu helikoptere ait matematiksel model aşağıdaki gibidir.

Sisteme dengesizlik vektörleri de eklendiğinde sistemin son durumdaki modeli (4.9):

$$\begin{aligned}\ddot{\phi} &= \frac{J_r \dot{\theta} (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 + \Omega_5 - \Omega_6)}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\psi} \dot{\theta} + \frac{bl (\Omega_1^2 - \Omega_2^2)}{I_{xx}} \\ \ddot{\theta} &= \frac{J_r \dot{\phi} (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 + \Omega_5 - \Omega_6)}{I_{xx}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi} + \frac{bl (\Omega_1^2 - \Omega_2^2)}{I_{yy}} \\ \ddot{\psi} &= \frac{d (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2 + \Omega_5^2 - \Omega_6^2)}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\theta} \dot{\phi}\end{aligned}\quad (4.9)$$

4.4.2 Değişken Gecikme Zamanlı MPC ile Kontrol

Model Predictive Control (MPC) ile geleneksel kontrol yöntemleri arasındaki temel fark geleneksel kontrol yöntemleri önceden hesaplanmış kontrol kuralını, kontrol değerini kullanırken; MPC özellikle sınır şartları olan tipinde iteratif yöntemleri her bir örnekleme zamanında tahmin ufku sayısı kadar hesaplar. Bununla birlikte MPC' nin diğer yöntemlere karşı olan avantajı MPC' de sınır şartları doğrudan problemin yapısına eklenebilir; çok değişkenli ve/veya zamanla değişen kontrol düzenlerinin kontrolünde etkili olmasıdır [30], [32], [33], [35], [36], [39].

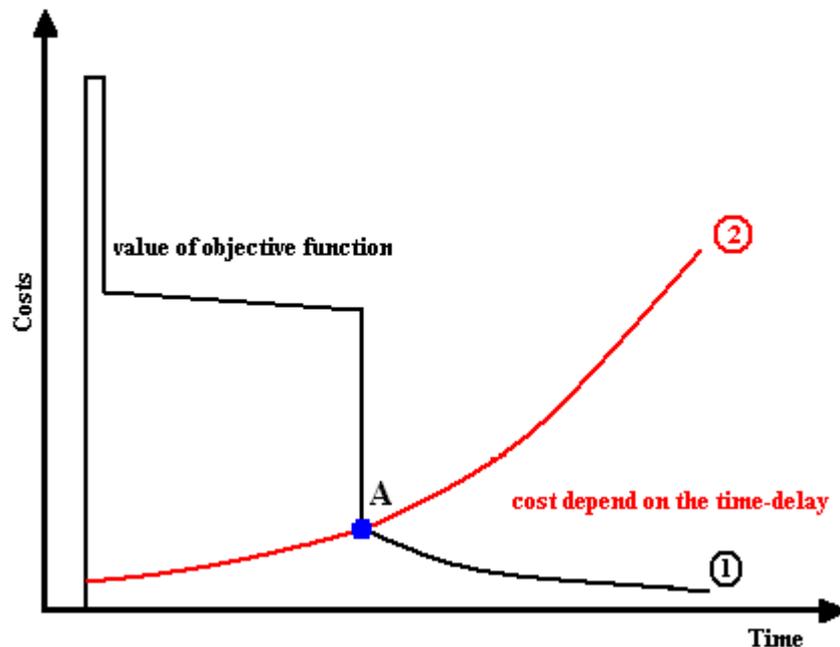
MPC' nin gerçekleşmesinde gerçek zamanlı kontrole her zaman ihtiyaç duyulur. Bugün MPC' nin gerçek zamanlı uygulamaları özellikle icra zamanı (execution time) problemleri açısından bir çok defa ele alınmıştır. Zamanla değişen icra zamanları, kontrol sistemlerine beraberinde kompanzasyonu kolay olmayan gecikmeleri getirirler. Özellikle, sınır şartlarının olduğu MPC modellerinde optimizasyon problemlerinin çözümü için gerekli iterasyon sayısı arttıkça gecikme süresi artar. Sistem örneklendiği noktadan uzaklaşır. Gecikme (latency), kontrol performansını iyi kompanze edilmediği sürece negatif etkiler. Bugüne kadar gerçek zamanlı uygulama açısından [30], [31], [34], [37] çalışmalarında incelenen bu konu, sınır şartları içeren MPC problemini

periyodik görevler halinde (periodic task) ele almıştır. Bu çalışmalarda işlenen ana fikir konveks optimizasyon probleminin çözümünde kullanılan iteratif programların erken sonuçlandırılmasının MPC' nin performans indeks' ini arttırabileceği söylenmiştir. Bununla birlikte iterative programların erken sonuçlandırılması (early termination) kararlılık şartlarını sağlarken, kontrol işaretlerinin tahmininde etkin sonuçlar ortaya koyabilir [31], [37].

4.4.3 Değişken Gecikme Zamanlı MPC:

Sınır şartları içeren bir MPC problemindeki temel sorun ne kadar süre optimizasyona devam edilmelidir. Bilindiği üzere optimizasyon sürecindeki her bir iterasyon için daha iyi bir optimum değer; daha iyi bir kontrol değeri elde edilir. Bununla birlikte, her bir iterasyonda Objective (hedef) Fonksiyonunun değeri azalırken, gecikmeye bağlı toplam maliyet (cost) fonksiyon değeri artar [31], [37].

Bugüne kadar bu alanda yapılan çalışma optimizasyonun, toplam maliyet fonksiyonunun artışa geçtiği noktada sonlandırılmasını ortaya koyar. Bu noktada, önerilen basit bir durdurma kriteridir. Bu kriter ile amaçlanan gecikmeye bağlı olan maliyetlerin belirli bir seviyede tutulmasıdır. Önerilen bu fikir ile ilgili bir örneklendirme Şekil 4. 14' te görülebilir.



Şekil 4.14 Önceki çalışmalarda önerilen durdurma kriteri [31]

Şekil 4.14' te sınır şartlı bir MPC' de kontrol sinyalini bulmada kullanılan optimizasyon süreci örneklendirilmiştir. Burada herbir iterasyon ile birlikte maliyet değerleri azalmaktadır (bir nolu eğri). İki nolu eğri ise kontrol sinyali elde edilinceye kadar geçen süreye bağlı gecikme maliyetlerinin artışı göstermektedir. Daha önceki yapılan çalışmada önerilen fikir; gecikmeye bağlı maliyet fonksiyonu artmaya başladığı anda optimizasyon algoritmasını optimum değerden önceki göreceli iyi bir noktada keserek kontrol sinyalini elde etmektir. Bir önceki çalışmada bu kesim tekniği ile ilgili ayrıntılı bir analiz yapılmamıştır. Çalışma sadece herhangi bir sistem için tecrübeye bağlı bir kesim tekniğini önerir. Bu durumda temel olarak iki soru akla gelir. İlk olarak izin verilen ve sonuçtaki gecikme zamanı optimizasyonu kesip, kontrol sinyalini sisteme uyguladığımız en iyi zaman mıdır? Yoksa bu gecikme zamanı sistem için geliştirilebilir mi?

4.4.4 Sınır Şartlı Bir MPC Problemi İçin Matematiksel Yöntemler

Önceki çalışmalar incelendiğinde olgunlaşmamış optimizasyon süreçlerinin active set yöntemleri ile ele alındığı görülür. Optimizasyon problemlerinin çözümünde active set yöntemleri kullanıldığında, optimum sonuç sınır şartlarının Objective Function' a eklenip çıkarılmasıyla bulunur. Bu yöntemde maliyet fonksiyonunun zamanla nasıl azaldığı tam olarak bilinmemektedir. Keza algoritma genel olarak optimum sonuca giden herhangi bir yolu kullandığından, zamanla değişimi çalışma sayısına ve sınır şartların maliyet fonksiyonuna eklenme sırasına göre değişir. İşte tüm bu faktörler active set yöntemi kullanıldığında optimum sonuca belirli bir zamanda ulaşmayı ve optimizasyon sürecinin ne zaman sonlandırılması gerektiğini zorlaştırır [31], [37].

MPC algoritmaları içerisinde yer alan optimizasyon süreçlerinde en çok karşılaşılan active set yöntemleri, simplex algoritması ve interior point yöntemleridir. Active set yöntemlerinin neden uygun olmadığı yukarıda açıklanmıştır [41], [43].

Optimizasyon sürecindeki başka bir yöntem ise simplex algoritmasıdır. Bu yöntemin zayıf bir worst case (en kötü durum) davranışı vardır. Bu algortmada çözüm için gerekli iterasyon sayısı bilinmeyenlerin sayısı arttıkça üstel olarak artar. Yüksek hesaplama gücü gerektirdiğinden ve Objective Function' ın iterasyon boyunca nasıl azaldığı net olmadığından, simplex algoritması MPC' deki optimizasyon probleminin çözümünde optimuma en yakın ve en az iterasyon gerektiren değeri belirlemede yardımcı olamaz.

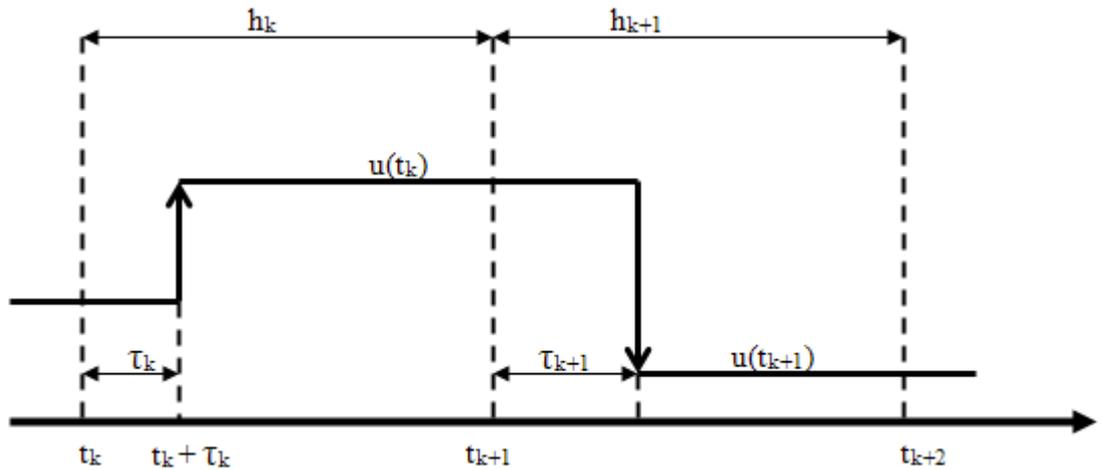
Bu çalışmada, interior point algoritmalarını kullanmanın başlıca üç temel nedeni vardır. İlk olarak algoritmanın karmaşıklığı her zaman için polinomsaldır. İkinci olarak, belirli bir iterasyonda optimum noktasından ne kadar uzak olduğumuzla ilgili bir tahmin yapmamıza imkan verir. Optimal değeri bulmada iterasyon sayısı bir üst değer ile sınırlandırılmıştır ve bilinen convex quadratic problemlerin çözümündeki en hızlı algoritmalarından bir tanesidir [42], [44].

Genellikle convex quadratic problemlerler interior point yöntemleri ile şu şekilde çözülür.

- Maximizasyon, minimizasyon problemlerin tanımlanması.
- Karush Kuhn Tucker şartları kullanılarak eşitsizlik sınır şartlarının hedef fonksiyonuna eklenmesi.
- Sistemin Newton adım yöntemi ile çözülmesi.

4.4.5 Gecikmeye Bağlı Suboptimal Çözümün Tek Bir MPC İçin Tanımı:

Tek bir MPC' deki iterasyona ve gecikmeye bağlı suboptimal çözümü tanımlamak için ilk önce gecikmeye sahip kontrol sinyali olan bir sistemi ele alalım. Burada kontrol sinyalindeki gecikme; kontrol sinyalini elde etmede kullanılan optimizasyon probleminin çözümü için gereken zamandır. Bu zaman sürekli zamandan ayırık zamana aktarılmalıdır. Şekil 4.15' te bu durum örneklendirilmiştir.



Şekil 4.15 Gecikme zamanına sahip bir kontrol sinyalinin modeli [38]

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t - \tau) \\
y &= Cx(t) \\
x(t_{k+1}) &= \phi(h_k)x(t_k) + \Gamma_1(h_k, \tau_k)u(t_{k-1}) + \Gamma_0(h_k, \tau_k)u(t_k) \\
y(t_k) &= Cx(t_k) \\
\begin{bmatrix} x(t_{k+1}) \\ u(t_k) \end{bmatrix} &= \underbrace{\begin{bmatrix} \phi(h_k) & \Gamma_1(h_k, \tau_k) \\ 0 & 0 \end{bmatrix}}_H \begin{bmatrix} x(t_k) \\ u(t_{k-1}) \end{bmatrix} + \underbrace{\begin{bmatrix} \Gamma_0(h_k, \tau_k) \\ I \end{bmatrix}}_J u(t_k)
\end{aligned} \tag{4.10}$$

(4.10) nolu formül grubu ile birlikte plant önce sürekli zaman eşitliği olarak ifade edilmiştir. Sürekli zaman durum uzay eşitliği ilk olarak ifade edilmiştir. Daha sonra bu gecikme ayırık zamana aktarılmıştır. Ayırıklaştırma işlemi [40] nolu referansda ayrıntılı olarak işlenmiştir. Son kısımda ise, durum uzay modelindeki sistem ayırık zamanda matris formunda gösterilmiştir. Aşağıda optimizasyon sürecinde gerekli olan maliyet fonksiyonu gösterilmiştir.

$$\begin{aligned}
J &= \int_{kh}^{(k+1)h} \begin{pmatrix} x(t) - x_{soll} \\ u(t - \tau) - u_{soll} \end{pmatrix}^T \begin{pmatrix} Q_C & N_C \\ N_C^T & R_C \end{pmatrix} \begin{pmatrix} x(t) - x_{soll} \\ u(t - \tau) - u_{soll} \end{pmatrix} \\
x_{soll}, u_{soll} &\rightarrow LQR \rightarrow 0 \quad N_C \rightarrow 0 \\
J &= \int_{kh}^{(k+1)h} \begin{pmatrix} x(t) \\ u(t - \tau) \end{pmatrix}^T \begin{pmatrix} Q_C & 0 \\ 0 & R_C \end{pmatrix} \begin{pmatrix} x(t) \\ u(t - \tau) \end{pmatrix} \quad J = \sum_{kh}^{(k+1)h} \begin{pmatrix} \underline{x}_k \\ \underline{u}_k \end{pmatrix} \begin{pmatrix} \underline{Q}_d & \underline{N}_d \\ \underline{N}_d^T & \underline{R}_d \end{pmatrix} \begin{pmatrix} \underline{x}_k \\ \underline{u}_k \end{pmatrix}
\end{aligned} \tag{4.11}$$

Burada ayırıklaştırma işlemini kolaylaştırmak için LQR tipi bir maliyet fonksiyonu kullanılmış ve N terimi sıfır olarak kabul edilmiştir. N terimi sıfır olarak kabul edilse bile ayırıklaştırma işleminin özelliğinden dolayı sıfırdan farklı bir N terimi sürecin sonunda oluşur. N teriminden kurtulmak için maliyet fonksiyonu geçiş ifadesinin yok edilmesi 'cross term elimination' işlemine tabi tutulur [45].

$$\begin{aligned}
\underline{M} &= \underline{N}_d \underline{R}_d^{-1} & \underline{\tilde{u}}_k &= \underline{u}_k + \underline{M}^T \underline{x}_k \\
\underline{\tilde{Q}} &= \underline{Q} - \underline{M} \underline{N}_d & \underline{\tilde{H}} &= \underline{H} - \underline{J} \underline{M}^T \text{ ve } \underline{\tilde{J}} = \underline{J} \underline{L}^{-1}
\end{aligned} \tag{4.12}$$

Ağırlık matrisi, Cholesky factorization işlemine tabi tutulur ve aşağıdaki şekilde bileşenlerine ayrılır. Sistemde ağırlık matrisinin seçilmesi (4.11), (4.13) nolu eşitlikte, Cholesky factorization işlemi sonucu yeni sistem matrisleri (4.13) nolu eşitlikte gösterilmiştir.

$$\begin{aligned}\tilde{\underline{Q}} &= \tilde{\underline{C}}' \tilde{\underline{C}} & \underline{R}_d &= \tilde{\underline{L}}' \tilde{\underline{L}} \\ \hat{\underline{u}}_k &= \tilde{\underline{L}}^{-1} \tilde{\underline{u}}_k & \tilde{\underline{x}}_{k+1} &= \tilde{\underline{H}} \tilde{\underline{x}}_k + \tilde{\underline{J}} \hat{\underline{u}}_k\end{aligned}\quad (4.13)$$

Tüm bu işlemlerden sonra MPC için gerekli olan maliyet fonksiyonu aşağıdaki şeklini alır(4.14).

$$J = \sum_{h=1}^{Np} \left(\tilde{\underline{C}} \tilde{\underline{x}}_k \right)^T \left(\tilde{\underline{C}} \tilde{\underline{x}}_k \right) + \hat{\underline{u}}_k^T \hat{\underline{u}}_k \quad (4.14)$$

Artık gecikme süresine sahip MPC regülatör problemi için dönüştürülmüş sistem eşitlikleri kullanılarak oluşturulabilir (4.15).

$$\begin{aligned}\forall k \geq Nu &\rightarrow \hat{\underline{u}}_k = \hat{\underline{u}}_{soll} = 0 \\ \tilde{\underline{X}}_{k+1} &= \tilde{\underline{H}} \tilde{\underline{X}}_k + \tilde{\underline{J}} \hat{\underline{u}}_k \\ \tilde{\underline{X}}_{k+2} &= \tilde{\underline{H}} \tilde{\underline{X}}_{k+1} + \tilde{\underline{J}} \hat{\underline{u}}_{k+1} = \tilde{\underline{H}}^2 \tilde{\underline{X}}_k + \tilde{\underline{H}} \tilde{\underline{J}} \hat{\underline{u}}_k + \tilde{\underline{J}} \hat{\underline{u}}_{k+1}\end{aligned}$$

$$\begin{aligned}\begin{bmatrix} \tilde{\underline{x}}_{k+1} \\ \vdots \\ \tilde{\underline{x}}_{k+Nu} \\ \tilde{\underline{x}}_{k+Nu+1} \\ \vdots \\ \tilde{\underline{x}}_{k+Np} \end{bmatrix} &= \begin{bmatrix} \tilde{\underline{H}} \\ \vdots \\ \tilde{\underline{H}} \\ \tilde{\underline{H}} \\ \vdots \\ \tilde{\underline{H}} \end{bmatrix} \underbrace{\tilde{\underline{x}}_k}_{\text{baş.değ.}} + \begin{bmatrix} \tilde{\underline{J}} \\ \vdots \\ \tilde{\underline{H}} & \tilde{\underline{J}} & \dots & \tilde{\underline{J}} \\ \tilde{\underline{H}} & \tilde{\underline{J}} & \dots & \tilde{\underline{J}} \\ \tilde{\underline{H}} & \tilde{\underline{J}} & \dots & \tilde{\underline{J}} \\ \vdots & \vdots & \vdots & \tilde{\underline{J}} \\ \tilde{\underline{H}} & \tilde{\underline{J}} & \dots & \tilde{\underline{J}} \end{bmatrix} \begin{bmatrix} \hat{\underline{u}}_k \\ \vdots \\ \hat{\underline{u}}_{k+Nu-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.15)\end{aligned}$$

$$\underline{\underline{x}}_p = \underline{\underline{A}}_p \underline{\underline{x}}_k + \underline{\underline{B}}_1 \hat{\underline{u}}_k^*$$

Bununla birlikte hedef fonksiyonu (Objective Function) MPC' nin içine yazılması ile birlikte aşağıdaki şeklini alır (4.16).

$$\begin{aligned}
J &= \underline{x}_p^T \tilde{Q} \underline{x}_p + \begin{pmatrix} \hat{u}_k^* \\ \underline{u}_k \end{pmatrix}^T \hat{u}_k^* \\
&= \begin{bmatrix} \tilde{C} \left(\underline{A}_p \underline{x}_k + \underline{B}_1 \hat{u}_k^* \right) \end{bmatrix}^T \begin{bmatrix} \tilde{C} \left(\underline{A}_p \underline{x}_k + \underline{B}_1 \hat{u}_k^* \right) \end{bmatrix} + \begin{pmatrix} \hat{u}_k^* \\ \underline{u}_k \end{pmatrix}^T \hat{u}_k^* \\
&= \begin{pmatrix} \hat{u}_k^* \end{pmatrix}^T \underbrace{\left[\underline{B}_1' \tilde{C}' \tilde{C} \underline{B}_1 + I \right]}_{\underline{T}} \begin{pmatrix} \hat{u}_k^* \end{pmatrix} + 2 \begin{pmatrix} \hat{u}_k^* \end{pmatrix}^T \underbrace{\left[\underline{B}_1' \tilde{C}' \left(\tilde{C} \underline{A}_p \underline{x}_k \right) \right]}_{\underline{c}} \\
&= \begin{pmatrix} \hat{u}_k^* \end{pmatrix}^T \underline{T} \begin{pmatrix} \hat{u}_k^* \end{pmatrix} + 2 \begin{pmatrix} \hat{u}_k^* \end{pmatrix}^T \underline{c}
\end{aligned} \tag{4.16}$$

4.4.6 Optimizasyon Problemi İçin Sınır Şartlarının Tanımlanması

Bilindiği üzere MPC sınır şartlarının olması durumunda quadratic programlama ile çözülür. Bu sebeple, bu adımda artık kontrol ufku 'control horizon' boyunca gerekli olan sınır şartları tanımlanmalı ve konveks quadratic optimizasyon problemi oluşturulmalıdır.

$$\begin{aligned}
\underline{u} &\leq \underline{u}_k \leq \bar{u} \\
\underline{u} &\leq \begin{bmatrix} \underline{u}_k \\ \vdots \\ \underline{u}_{k+h_c-1} \end{bmatrix} \leq \bar{u}
\end{aligned} \tag{4.17}$$

Maliyet fonksiyonuna uygulanan tüm işlemler cross term elimination, Cholesky factorization burada da uygulanır. Dönüştürülmüş sınır şartları eşitlikleri kullanılarak sınır şartları için gerekli olan MPC oluşturulur (4.18).

$$\begin{aligned}
\tilde{u}_k &= \underline{u}_k + \underline{M}^T \underline{x}_k & \underline{u}_k &= \tilde{u}_k - \underline{M}^T \underline{x}_k \\
\tilde{H} &= \underline{H} - \underline{J} \underline{M}^T & \underline{X}_{k+1} &= \tilde{H} \underline{x}_k + \underline{J} \tilde{u}_k
\end{aligned} \tag{4.18}$$

Sınır şartları üzerinde dönüşüm (cross term elimination ve Cholesky factorization) uygulamalarına devam edilir (4.19).

$$\begin{aligned}
 \underline{\underline{u}} &\leq \tilde{u}_k - \underline{M}^T \underline{X}_k \leq \bar{u} \\
 \underline{\underline{u}} + \underline{M}^T \underline{x}_k &\leq \tilde{u}_k \leq \bar{u} + \underline{M}^T \underline{x}_k \\
 \tilde{\underline{J}} &= \underline{J} \underline{L}^{-1} & \tilde{u}_k &= \frac{\hat{u}_k}{\tilde{L}} \\
 \tilde{\underline{H}} &= \underline{H} - \underline{J} \underline{M}^T & \underline{x}_{k+1} &= \tilde{\underline{H}} \underline{x}_k + \tilde{\underline{J}} \hat{u}_k \\
 \underline{\underline{u}} + \underline{M}^T \underline{X}_k &\leq \frac{\hat{u}_k}{\tilde{L}} \leq \bar{u} + \underline{M}^T \underline{X}_k \\
 \tilde{\underline{L}} (\underline{\underline{u}} + \underline{M}^T \underline{X}_k) &\leq \hat{u}_k \leq \tilde{\underline{L}} (\bar{u} + \underline{M}^T \underline{X}_k)
 \end{aligned} \tag{4.19}$$

Sınır şartları için de bir model öngörülü kontrol problemi oluşturulur (4.20).

$$\begin{aligned}
 \underline{x}_{k+1} &= \underline{H} \underline{x}_k + \underline{J} \hat{u}_k \\
 \underline{x}_{k+2} &= \underline{H} \underline{x}_{k+1} + \underline{J} \hat{u}_{k+1} = \underline{H}^2 \underline{x}_k + \underline{H} \underline{J} \hat{u}_k + \underline{J} \hat{u}_{k+1} \\
 &\vdots \\
 \underline{x}_{k+Nu-1} &= \dots
 \end{aligned}$$

$$\underbrace{\begin{bmatrix} \underline{x}_{k+1} \\ \vdots \\ \underline{x}_{k+Nu} \end{bmatrix}}_{\underline{x}_p} = \underbrace{\begin{bmatrix} \tilde{\underline{H}} \\ \vdots \\ \tilde{\underline{H}}^{Nu} \\ \underline{H} \end{bmatrix}}_{\underline{A}_p} \underbrace{\underline{x}_k}_{\text{baş.değ.}} + \underbrace{\begin{bmatrix} \tilde{\underline{J}} & & & \\ & \tilde{\underline{J}} & & \\ & & \tilde{\underline{J}} & \\ \underline{H} & \underline{J} & \dots & \underline{J} \end{bmatrix}}_{Nu(B1)} \underbrace{\begin{bmatrix} \hat{u}_k \\ \vdots \\ \hat{u}_{k+Nu-1} \end{bmatrix}}_{\left(\hat{\underline{u}}_k\right)^*} \tag{4.20}$$

$$\underline{x}_p = \underline{A}_{Nu} \underline{x}_k + \underline{B}_1 \hat{\underline{u}}_k$$

Sınır şartlarının son hali aşağıdaki gibidir (4.21).

$$\begin{aligned}
\tilde{L}(\underline{u} + \underline{M}^T \underline{X}_k) &\leq \hat{u}_k \leq \tilde{L}(\bar{u} + \underline{M}^T \underline{X}_k) \\
\tilde{L}\left(\underline{u} + \underline{M}^T \left(\underline{A}_{Nu} \underline{X}_k + \underline{B}_1 \hat{u}_k^*\right)\right) &\leq \left(\hat{u}_k\right)^* \leq \tilde{L}\left(\bar{u} + \underline{M}^T \left(\underline{A}_{Nu} \underline{X}_k + \underline{B}_1 \hat{u}_k^*\right)\right) \\
\frac{\tilde{L}\left(\underline{u} + \underline{M}^T \left(\underline{A}_{Nu} \underline{X}_k\right)\right)}{1 - \tilde{L}\underline{M}^T \underline{B}_1} &\leq \left(\hat{u}_k\right)^* \leq \frac{\tilde{L}\left(\bar{u} + \underline{M}^T \left(\underline{A}_{Nu} \underline{X}_k\right)\right)}{1 - \tilde{L}\underline{M}^T \underline{B}_1}
\end{aligned} \tag{4.21}$$

4.4.7 Problemin Path Following (Yol Takibi) Metodu ile Çözümü

MPC ve sınır şartları dönüştürülmüş sistem denklemleri ile oluşturulduğunda karşımıza convex quadratic problem çıkar. Bu tür problemler genellikle active set yöntemleri veya interior point yöntemleri kullanılarak çözülür. Buradaki çalışmada, gecikmeye bağlı suboptimal çözümü bulmada interior point yöntemi kullanılmıştır. Bu yöntemin avantajları:

- Arama yönü kriteri ile maliyet fonksiyonun her iterasyon ile birlikte azaldığını göstermek mümkünken simplex veya active set yöntemleri ile göstermek oldukça zordur. Çünkü; maliyet fonksiyonu belirli bir iterasyon uzunluğunda aynı kalabilir[41].
- Interior point yöntemleride iterasyon sayısı sınırlıdır. Bu bize herhangi bir iterasyon da optimum değere olan uzaklığı iterasyon sayısını bildirir [42].
- Bir iterasyon için gereken hesaplamalar sabit sayılı eşitliklerden oluştuğundan, iterasyon için gereken işlem zamanı mikroişlemcide sabittir.

Tüm bu avantajlar bize toplam maliyet fonksiyonunun zamanla değişen zaman gecikmesine bağlı olarak arttığı sırada hangi iterasyonda optimizasyonu bitirip suboptimal çözümü bulmamızı sağlar. Unutulmamalıdır ki, kontrol sinyalini hesaplamak için ne kadar fazla süre harcanırsa o oranla sistem durum değişkenlerinin ölçüldüğü ve örneklendiği noktadan uzaklaşacaktır.

Burada, bir yol takip yöntemi convex quadratic problemin çözümünde kullanılmıştır. Bugün akademik olarak kullanılan pek çok algoritma vardır. Bunlardan bazıları:

Newton-KKT [44], Yalmip (interior point çözücüsü) [47], QPC Quadratic Programming in C [48], Quadprog-middle scale algorithms [46] vb.

Bu çözücülerden QPC sınır şartlarını bu formda doğrudan çözebilirken, diğer çözücüler için sınır şartları $\underline{A}x \leq \underline{b}$ formatında veya $\underline{A}x \geq \underline{b}$ formatında girilmelidir. Bu şekli ile son sınır şartları ifadesi (4.22) ile belirtilmiştir.

$$\frac{\tilde{L}\left(\underline{u} + \underline{M}^T (\underline{A}_{Nu} \underline{x}_k)\right)}{1 - \tilde{L}\underline{M}^T \underline{B}_1} \leq \left(\hat{\underline{u}}_k\right)^* \leq \frac{\tilde{L}\left(\bar{\underline{u}} + \underline{M}^T (\underline{A}_{Nu} \underline{x}_k)\right)}{1 - \tilde{L}\underline{M}^T \underline{B}_1}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \left(\hat{\underline{u}}_k\right)^* \\ \left(\hat{\underline{u}}_k\right)^* \end{bmatrix} \geq \begin{bmatrix} \frac{\tilde{L}\left(\underline{u} + \underline{M}^T (\underline{A}_{Nu} \underline{x}_k)\right)}{1 - \tilde{L}\underline{M}^T \underline{B}_1} \\ \frac{\tilde{L}\left(\bar{\underline{u}} + \underline{M}^T (\underline{A}_{Nu} \underline{x}_k)\right)}{1 - \tilde{L}\underline{M}^T \underline{B}_1} \end{bmatrix} \quad (4.22)$$

Sınır şartları formülünde \underline{x}_k ile belirtilen değişkenlerin başlangıç değerleridir. Quadratik programlamadaki gösterimi kolaylaştırmak için, (4.22) ile ifade edilen sınır şartları ve maliyet fonksiyonu yerine (4.23)' teki eşiti quadratik programlamayı daha iyi açıklamak için kullanılmıştır[41]. Eşitlik (4.23) ve (4.40) arasında açıklanan algoritma [41] nolu referansta ayrıntılı bir şekilde açıklanmıştır.

$$\begin{aligned} & \frac{1}{2} x^T Qx + c^T x \\ \text{st. } & Ax \geq b \\ & x \geq 0 \end{aligned} \quad (4.23)$$

$$\begin{aligned} & \frac{1}{2} x^T Qx + c^T x \\ & Ax - w = b \\ & x, w \geq 0 \end{aligned} \quad (4.24)$$

Bir yol takibi yönteminde (path following method) yapılacak ilk iş eşitsizlik sınır şartlarını, eşitlik sınır şartlarına dönüştürmek ve daha sonra Langrange çarpanı kullanarak maliyet fonksiyonuna eklemektir. Daima pozitiflik şartı da maliyet fonksiyonuna eklendiğinde sistem önce (4,24) ve devamında (4.25)' teki halini alır.

$$\frac{1}{2}x^T Qx + c^T x - \mu \sum_j \log x_j - \mu \sum_j \log w_j \quad (4.25)$$

$$Ax - w = b$$

Sınır şartı (ları) Langrange çarpanları kullanılarak maliyet fonksiyonuna eklenir (4.26).

$$f(x, w, y) = \frac{1}{2}x^T Qx + c^T x - \mu \sum_j \log x_j - \mu \sum_j \log w_j - y^T (b - Ax + w) \quad (4.26)$$

Supreme noktaları için maliyet fonksiyonunun türevi alınıp gradient oluşturulduğunda (4.27);

$$\begin{aligned} c + Qx - \mu X^{-1}e - A^T y &= 0 \\ -\mu W^{-1}e + y &= 0 \\ b - Ax + w &= 0 \end{aligned} \quad (4.27)$$

Yukarıdaki sistem eşitliklerine $z = \mu X^{-1}e$ değişkeni tanıtilip, sistem eşitlikleri yeniden yazıldığında (4.28);

$$\begin{aligned} c + Qx - z - A^T y &= 0 \\ XZe &= \mu e \\ WZe &= \mu e \\ b - Ax + w &= 0 \end{aligned} \quad (4.28)$$

Bu eşitliklerde, X ve e matrislerinin kanonik formda olduğu unutulmamalıdır (4.29).

$$X = \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & x_n \end{bmatrix} \quad \text{ve} \quad e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (4.29)$$

Bu noktada, yukarıdaki sisteme 3 boyutlu uzayda adım yönü tanıtılır (4.30).

$$\begin{aligned}
c + Q(x + \Delta x) - (z + \Delta z) - A^T(y + \Delta y) &= 0 \\
(X + \Delta x)(Z + \Delta Z)e &= \mu e \\
(W + \Delta W)(Y + \Delta Y)e &= \mu e \\
b - A(x + \Delta x) + (w + \Delta w) &= 0
\end{aligned} \tag{4.30}$$

Yukarıdaki sistem eşitlikleri yeniden düzenlendiğinde (4.31);

$$\begin{aligned}
Q\Delta x + \Delta z + A^T \Delta y &= c + Qx - z - A^T y =: \sigma \\
Z\Delta x + X\Delta z &= \mu e - ZXe \\
W\Delta y + Y\Delta w &= \mu e - WYe \\
A\Delta x - \Delta w &= b - Ax + w =: \rho
\end{aligned} \tag{4.31}$$

Yukarıdaki sistem eşitlerinde $c + Qx - z - A^T y =: \sigma$ ve $b - Ax + w =: \rho$ olarak yazıldığında sistem aşağıdaki halini alır (4.32).

$$\begin{aligned}
A^T \Delta y + \Delta z - Qx &= \sigma \\
A\Delta x - \Delta w &= \rho \\
z\Delta x + x\Delta z &= \mu e - ZXe \\
W\Delta y - Y\Delta w &= \mu e - WYe
\end{aligned} \tag{4.32}$$

Burada sistem eşitliklerindeki Δz ve Δw değişkenleri Δx ve Δy cinsinden yazılır (4.33).

$$\begin{aligned}
\Delta z &= X^{-1}(\mu e - ZXe - Z\Delta x) \\
\Delta w &= Y^{-1}(\mu e - WYe - W\Delta y)
\end{aligned} \tag{4.33}$$

Bu noktada erişilen sistem eşitliği aşağıdaki gibidir (4.34).

$$\begin{aligned}
A^T \Delta y - (Z + Q)\Delta x &= \sigma - \mu X^{-1}e + z \\
A\Delta x + Y^{-1}W\Delta y &= \rho + \mu Y^{-1}e - w
\end{aligned} \tag{4.34}$$

Sistem eşitlikleri matris formunda yazıldığında (4.35) elde edilir.

$$\begin{bmatrix} -(X^{-1}Z+Q) & A^T \\ A & Y^{-1}W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c-A^T y-\mu X^{-1}e+Qx \\ b-Ax+\mu Y^{-1}e \end{bmatrix} \quad (4.35)$$

Artık yapılması gereken sadece bu denklem sisteminin çözümüdür. Bu, (4.36) ile de gösterilmiştir. Bu denklem sistemini çözerken karşımıza çıkan bir başka konu μ çarpanının seçilmesi ve adım uzunluğu θ ' nın belirlenmesidir.

$$\begin{aligned} \Delta x &= -(X^{-1}Z+Q)^{-1}(c-A^T y-\mu X^{-1}e+Qx-A^T \Delta y) \\ \Delta y &= Y^{-1}W(b-Ax+\mu Y^{-1}e-A\Delta x) \end{aligned} \quad (4.36)$$

Adım uzunluğu tek bir değişken için aşağıdaki gibi belirlenebilir (4.37).

$$\begin{aligned} x_j + \theta \Delta x_j &> 0 & j=1, 2, \dots, n \\ \frac{1}{\theta} &> -\frac{\Delta x_j}{x_j} & j=1, 2, \dots, n \end{aligned} \quad (4.37)$$

Bu seçim tüm değişkenlere uygulandığında (4.38) elde edilir.

$$\begin{aligned} \frac{1}{\theta} &> \max_{i,j} \left\{ -\frac{\Delta x_j}{x_j}, -\frac{\Delta w_i}{w_i}, -\frac{\Delta y_i}{y_i}, -\frac{\Delta z_j}{z_j} \right\} \\ \theta &= r \left(\max_{i,j} \left\{ -\frac{\Delta x_j}{x_j}, -\frac{\Delta w_i}{w_i}, -\frac{\Delta y_i}{y_i}, -\frac{\Delta z_j}{z_j} \right\} \right)^{-1} \end{aligned} \quad (4.38)$$

μ çarpanının seçimi: Eğer verilen nokta optimuma giden yolda yer alıyor ise μ , $x^T z$ veya $y^T w$ olarak seçilir. Eğer verilen nokta optimuma giden yolda yer almıyor ise μ , bu değerlerin ortalaması olarak alınır. Formül (4.39)' da δ sıfır ile bir arasındaki bir değeri ifade eden skalalama çarpanıdır ve çoğunlukla 1/10 olarak pratikte kullanılır.

$$\mu = \delta \frac{x^T z + y^T w}{n+m} \quad \left\{ \begin{array}{l} n: \#1. \text{ sınır şartına ait değişken vektörün uzunluğu} \\ m: \#2. \text{ sınır şartına ait değişken vektörün uzunluğu} \end{array} \right\} \quad (4.39)$$

Son olarak, MPC formülünün çözümü için gerekli olan bu algoritmaya bütün olarak yeniden bakıldığında (4.40);

Başlat $(x,y,z,t) > 0$

Optimal noktaya ulaşılan kadar aşağıdaki denklem sistemini çöz

$$\left\{ \begin{array}{l} \rho = b - Ax + w; \sigma = c - A^T y - z + Qx; \gamma = z^T x + y^T w; \mu = \delta \frac{\gamma}{n+m} \end{array} \right\}$$

$$\begin{bmatrix} -(X^{-1}Z + Q) & A^T \\ A & Y^{-1}W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c - A^T y - \mu X^{-1}e + Qx \\ b - Ax + \mu y^{-1}e \end{bmatrix}$$

$$\Delta z = X^{-1}(\mu e - XZe - Z\Delta x), \quad \Delta w = Y^{-1}(\mu e - YWe - W\Delta y) \quad (4.40)$$

$$\theta = r \left(\max_{i,j} \left\{ -\frac{\Delta x_j}{x_j}, -\frac{\Delta w_i}{w_i}, -\frac{\Delta y_i}{y_i}, -\frac{\Delta z_j}{z_j} \right\} \right)^{-1}$$

$$x + \theta \Delta x \rightarrow x, \quad y + \theta \Delta y \rightarrow y, \quad w + \theta \Delta w \rightarrow w, \quad z + \theta \Delta z \rightarrow z$$

}

4.4.8 Quadratic Programlama' nın Yakınsaması (Convergence)

Bir üst bölümde karesel programlama probleminin genel anlamda interior point yöntemi ile çözümü verilmiştir. Bu bölümde MPC için kontrol sinyalini bulmada kullanılan karesel programlama çözücüsünün yakınsaması ele alınacaktır. Bu çözücüde her bir iterasyonda maliyet fonksiyonun değeri iyileştirilmektedir. Ayrıca interior point yönteminde çözüm aranırken çözüm uzayında hep bir güvenilir yön araştırma kriteri kullanılır ki bu çözücülerin hepsinin yakınsama analizlerinde ele alınmıştır. Birlikte, her bir iterasyonda azalan maliyet fonksiyonu ve iterasyon sayısı sınır kriteri bize herhangi bir iterasyonda optimum noktadan ne kadar uzaklıkta olduğumuzu gösterir.

Bu çalışmada, [42] nolu referanstaki çözücü kullanılmıştır. Bu çözücünün düzenli yakınsama oranı aşağıdaki gibi (4.41)' de tanımlanır.

n : quadratic programlama da ki değişken sayısı, $\lambda > 0$ bariyer parametresi, $\alpha : \alpha < 1$,
 s^k : slackness vector, z^k : ortalama değer, $S^k : \text{diag}(s^k)$, $\Delta X : \text{diag}(\Delta x)$, $\mu = z^k$.

$$z^{k+1} \leq \left(1 - \frac{1}{4\sqrt{n}}\right) z^k$$

Lemma I. (4.41)

$$z^{k+1} \leq \left(1 - \frac{a}{\sqrt{n}} + \frac{a^2}{4n}\right) z^k$$

Proof:

$$1) \quad X^k \Delta s + \mu (X^k)^{-1} = \lambda e - X^k s^k$$

$$\begin{aligned} 2) \quad X^{k+1} s^{k+1} &= (X^k + \Delta X)(s^k + \Delta s) \\ &= X^k s^k + \Delta X s^k + X^k \Delta s + \Delta X \Delta s \\ &= X^k s^k + X^k \Delta s + \mu (X^k)^{-1} \Delta x + \Delta X s^k + \Delta X \Delta s \\ &= \lambda e + \Delta X ((s^k + \Delta s) - \mu (X^k)^{-1} e) \end{aligned} \quad (4.42)$$

(4.42)' deki bir ve iki nolu eşitlikten (4.43) nolu sonuca ulaşılır.

$$X^{k+1} s^{k+1} = \lambda e + \Delta X (X^k)^{-1} (\lambda e - \mu (X^k)^{-1} \Delta x - \mu e) \quad (4.43)$$

(4.43) nolu eşitlikte norm alındığında;

$$\begin{aligned} n z^{k+1} &= e^T X^{k+1} s^{k+1} \\ &= n \lambda + (\lambda - \mu) e^T (X^k)^{-1} \Delta x - \mu \left\| (X^k)^{-1} \right\|^2 \\ &= n \lambda - \frac{a z^k}{\sqrt{n}} e^T (X^k)^{-1} \Delta x - \mu \left\| (X^k)^{-1} \right\|^2 \\ &\leq n \lambda - \frac{a z^k}{\sqrt{n}} |e^T (X^k)^{-1} \Delta x| - \mu \left\| (X^k)^{-1} \right\|^2 \\ &\leq n \lambda - \frac{a z^k}{\sqrt{n}} \left\| e^T \right\| \left\| (X^k)^{-1} \Delta x \right\| - \mu \left\| (X^k)^{-1} \right\|^2 \\ &= n \lambda + a z^k \left\| (X^k)^{-1} \Delta x \right\| - \mu \left\| (X^k)^{-1} \right\|^2 \\ &= n \lambda + z^k (a \left\| (X^k)^{-1} \Delta x \right\| - \left\| (X^k)^{-1} \Delta x \right\|^2) \\ &= n \lambda + \frac{a^2 z^k}{4} \end{aligned} \quad (4.44)$$

(4.44)' de son eşitsizlik doğru olduğundan karesel terim maksimum değerine $\left\| (X^k)^{-1} \Delta x \right\| = \frac{\alpha}{2}$ de erişir. $\mu = z^k$ ve $\lambda = \left(1 - \frac{\alpha}{\sqrt{n}} \right) z^k$ olduğu görülür. Böylelikle (4.41)' in geçerliliği son adım olan (4.45)' te kanıtlanmıştır [42].

$$z^{k+1} \leq \left(1 - \frac{a}{\sqrt{n}} + \frac{a^2}{4n} \right) z^k \quad (4.45)$$

(4.41)' in geçerliliğini ispat etmekle amacımız maliyet fonksiyonunun her bir iterasyonda azaldığını göstermekti. Bununla birlikte, çözücü içindeki iterasyon sayısı [43]' te görülmektedir. Bu iterasyon sayısının yakınsaması kriteri uygun azalma çarpanı kullanılarak da farklı çözücülere uygulanabilir.

Lemma II. Bariyer değişkeni μ ve başlangıç değeri μ^0 ve $1-\theta$ faktörü ile güncelleniyorsa, $0 < \theta < 1$ olması durumunda $(1/\theta) \log(n\mu^0/\varepsilon)$ iterasyon sayısından sonra maliyet fonksiyonunun değeri $n\mu \leq \varepsilon$ olur [43].

Proof: Başlangıçtaki duality aralığı $n\mu^0$ ' dir. Her bir iterasyonda $1-\theta$ azaldığından k iterasyon sayısı sonrasında duality aralığı eğer $(1-\theta)^k n\mu^0 \leq \varepsilon$ küçük ise duality aralığı ε dan küçüktür [43].

Her iki tarafın logaritması alınır (4.46).

$$\begin{aligned} (1-\theta)^k n\mu &\leq \varepsilon \\ k \log(1-\theta) + \log(n\mu^0) &\leq \log \varepsilon \\ -\log(1-\theta) &\geq \theta \end{aligned} \quad (4.46)$$

Yukarıdaki ifadede son satır değerlendirildiğinde (4.47)

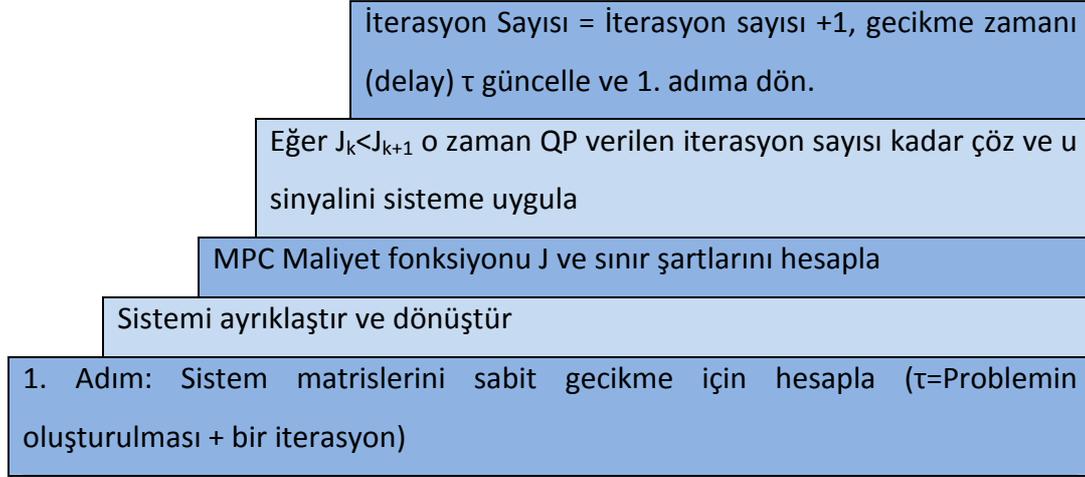
$$-\log(1-\theta) \geq \theta \quad (4.47)$$

Sadece

$$k\theta \geq \log(n\mu^0) - \log \varepsilon = \log \frac{n\mu^0}{\varepsilon} \quad (4.48)$$

olması durumunda (4.48) tamamen doğrudur. Böylelikle Lemma II açıklanmış olur.

4.4.9 Gecikme Zamanlı MPC' nin Bir Mikrodenetleyicide Uygulanması



Şekil 4.16 Önerilen Sistemin Mikrodenetleyiciye Uygulanması

1. Adımda, sistem matrislerini oluşturmak ve QP Problemini sabit iterasyon sayısı kadar çözmek için gerekli olan gecikme zamanı sabit zaman olarak gösterilmiştir. Daha sonra ayırklaştırma ve dönüştürme işlemleri sistem matrislerine uygulanır ve MPC problemi oluşturulur. Sonrasında hesaplanan maliyet fonksiyonunun bir önceki hesaplanandan daha yüksek olup olmadığına bakılır. Eğer toplam maliyet fonksiyonu hala azalıyor ise problem önceden belirlenen iterasyon sayısı kadar çözülür. Toplam maliyet fonksiyonunun artmaya başladığı noktada çözücü sonlandırılır ve son bulunan kontrol sinyali vektörünün ilk elemanı sisteme uygulanır.

Bu uygulama sistemde offline ve online olmak üzere iki farklı şekilde yapılabilir. Offline olarak uygulanmak istenirse verilen sisteme ait maliyet fonksiyonu eğrileri hesaplanır.

Sistem için optimum maliyet ve suboptimal iterasyon sayısı bulunur. Bu noktadan sonra maliyet fonksiyonunda optimum değer aramaya gerek yoktur. Çünkü; toplam maliyet değerleri artık kontrolcü için gerekli olan gecikme değerine bağlı olarak artmaya başlamıştır. Online olarak uygulamada ise her bir iterasyondan sonra karesel programlamadaki zamanla değişen zaman gecikmelerine bağlı olan maliyet fonksiyonu hesaplanarak bir önceki iterasyondaki bulunan maliyet fonksiyonu ile alt komutlar kullanılarak mikrodenetleyicide karşılaştırılır. Gecikmeye bağlı maliyetlerin artışı fark edildiğinde karesel çözücü kesintiye uğratarak bulunan kontrol sinyali sisteme uygulanır. Offline ve online uygulamadaki başlıca avantaj ve dezavantajlar offline

uygulama, online uygulamaya göre sistemi daha az kesintiye uğrattığından sistem daha hızlı cevap üretirken bozucu etkilere ve ani değişikliklere karşı olan toleransından kaybeder. Şekil 4.16' da önerilen sistem akışı resmedilmiştir.

4.4.10 İkinci Dereceden Bir Sistem İle Yapılan Örnek Çalışma

H_p : Tahmin ufku on iki örnekleme aralığı

H_u : Kontrol ufku dört örnekleme aralığı.

h : 20 ms örnekleme zamanı.

τ : %5-%50 örnekleme zamanı aralığında değişen zaman gecikmesi.

$x_1 - x_2$: Sistem durumları.

u : Kontrol sinyali.

$$-2 \leq u \leq 4$$

$I_c = [x_1; x_2; u]$: Başlangıç değerleri.

$$I_c = [3; 1; 0]$$

Plant:

$A_c = [-1 \ 0; 0 \ -2]$ sistem matrisi.

$B_c = [1; 1]$ sistem matrisi.

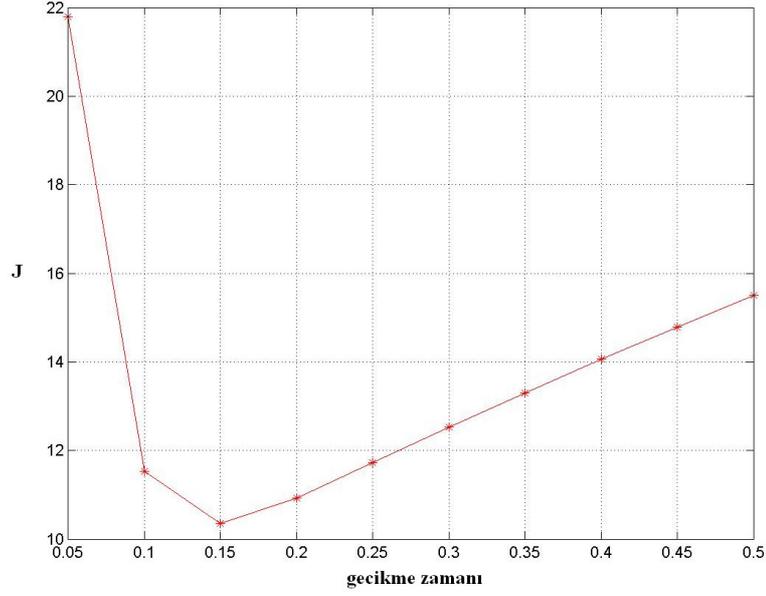
Maliyet fonksiyonu:

$Q_c = [1 \ 0; 0 \ 1]$: maliyet fonksiyonu ağırlık altmatrisi.

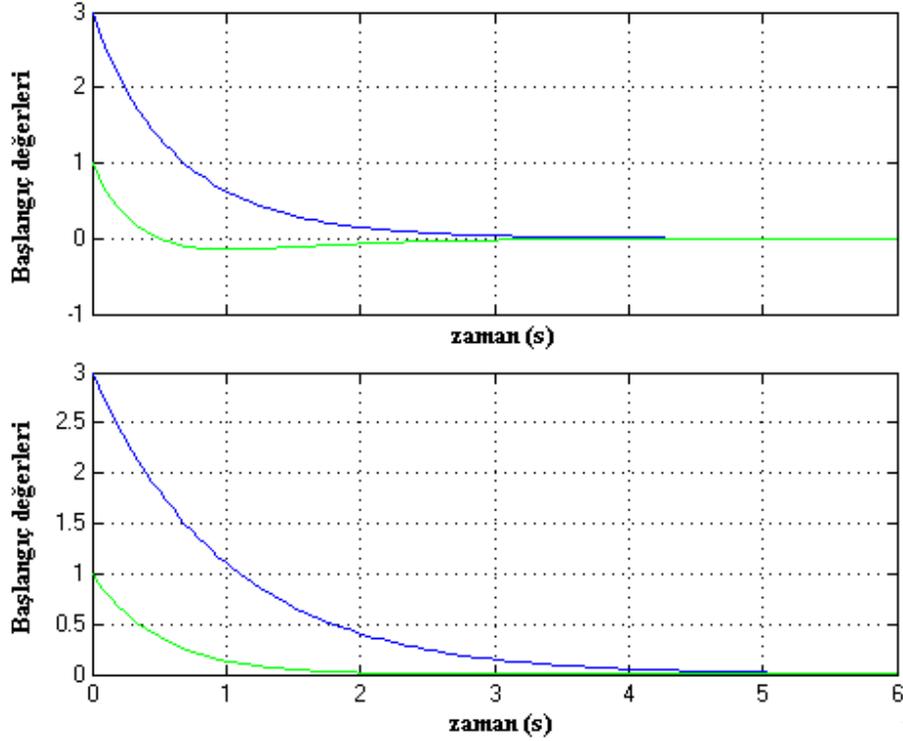
$N_c = [0; 0]$: maliyet fonksiyonu ağırlık altmatrisi.

$R_c = [1]$: maliyet fonksiyonu ağırlık altmatrisi.

$J_c = [Q \ N; N^T \ R]$: maliyet fonksiyonunun matris gösterimi.



Şekil 4.17 $J_D=f(x(\tau))$ Ayrık maliyet fonksiyonunun 'MPC' gecikmeye bağlı değişimi



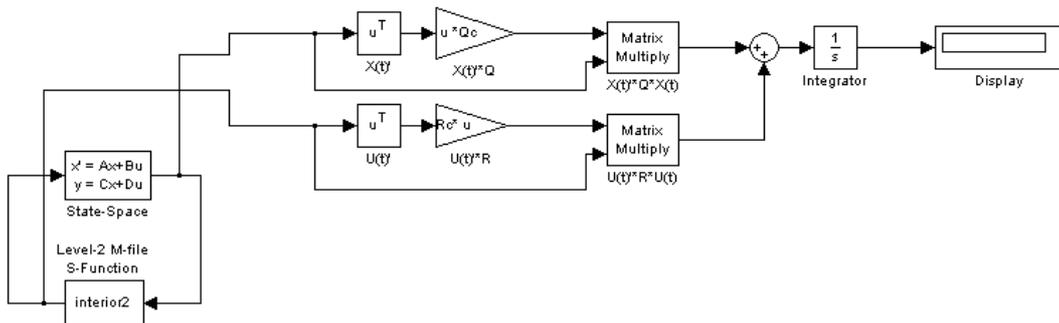
Şekil 4.18 Sistem durum çıkışlarının değişimi ($x_1; x_2$) (MPC-LQR)

Yukarıdaki şekilde üstteki şekil MPC' ye alttaki şekil ise klasik LQR' a aittir. Şekilde x eksenini zamanı belirtirken y eksenini durumların ($x_1; x_2$) değişimini göstermektedir. Şekilden de anlaşılacağı üzere sistem durumları önerilen MPC' de klasik LQR' a göre daha hızlı referans değerlerini takip etmektedir.

Kontrol sistemlerinde kontrol sinyali hesaplanıp sisteme uygulanıncaya kadar sistem bir noktadan farklı bir noktaya doğru hareket eder. Kontrol sinyalinin hesaplanmasındaki gecikmeler sistemi kontrol sinyali hesaplamalarını daha zorlu kılmaktadır. Burada bu

süreçteki amacımız sisteme ait kontrol sinyalini hesaplariken yaşanan gecikmeleri en aza indirmektir. Böylelikle sınır şartlarına sahip olan MPC' deki kontrol sinyalini hesaplamak için gerekli olan zamanla değişen zaman gecikmesi en aza indirilebilir.

Yukarıdaki sistemde ayrıklaştırılmış maliyet ve optimizasyon sürecindeki bir iterasyon %5 gecikme olarak kabul edilmiştir. Bu kabul optimizasyon çözücüsünün optimumu bulmak için 10 iterasyona ihtiyaç duymasından dolayı %50' ye kadar arttırılmıştır. Analizde görülmüştür ki ayrık maliyet fonksiyonu üçüncü iterasyondan itibaren zaman gecikmesine bağlı olarak artmaya başlamıştır. Bu sistem için optimizasyon çözücüsü üçüncü iterasyon sonunda sonlandırılmış ve bulunan kontrol sinyali sisteme uygulanmıştır. Ayrıklaştırılmış maliyet fonksiyonu değerleri Şekil 4.17 ve Çizelge 4.1' de görülebilir. Şekil 4.18' de sistem durumları klasik LQR ile zaman gecikmeli sınır şartlı MPC karşılaştırılmıştır. Orijin takibinde zaman gecikmeli sınır şartlı MPC, klasik LQR' a göre daha hızlıdır. Şekil 4.19' da maliyet fonksiyonu değerlerinin MATLAB ortamında hesaplanması görülebilir.



Şekil 4.19 MPC deki maliyet fonksiyonunun MATLAB ortamında hesaplanması

Çizelge 4.1 Zaman gecikmeli MPC' deki ayrık maliyet fonksiyonu değerlerinin iterasyon sayısına göre değişimi

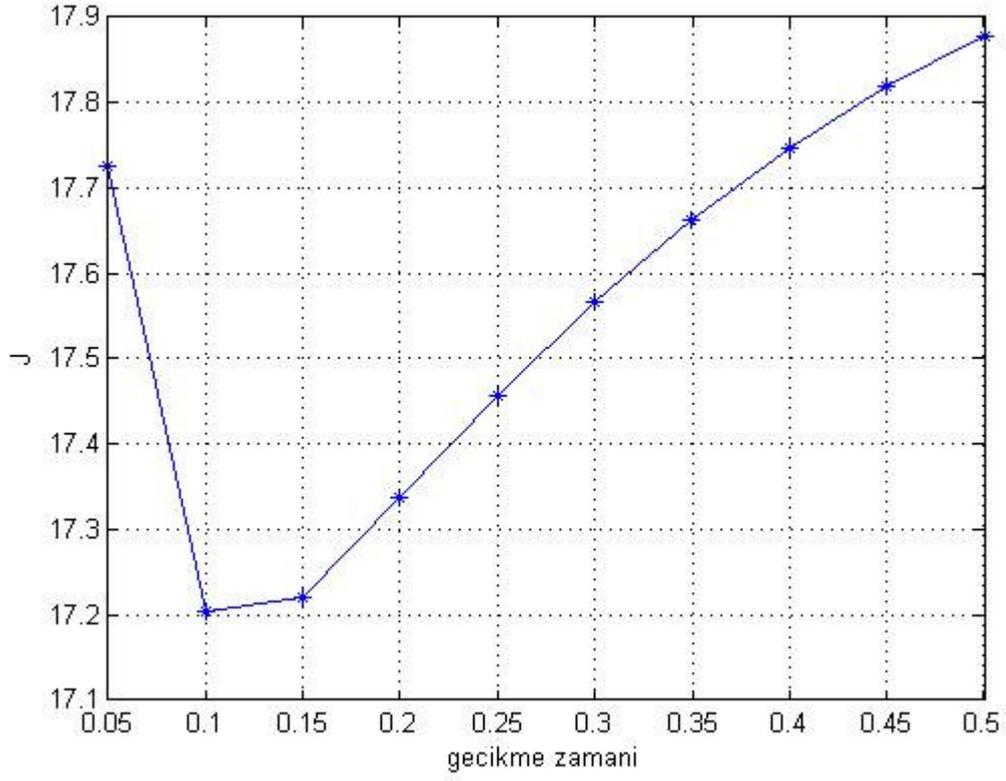
Iter. Sayısı	J_D MPC
1	21,9880
2	11,9092
3	10,5534
4	11,2672
5	11,9631
6	12,2703
7	13,3071
8	14,3183
9	15,1031
10	15,8042

Yukarıdaki şekilde hesaplanan maliyetlerin hesaplama yolu, tabloda ise hesaplanan maliyetlere ait değerler verilmiştir. Buna göre optimizasyon sürecinde maliyet fonksiyonu üçüncü iterasyonun sonucunda artmaya başladığından, optimizasyon çözücüsü bu noktada sonlandırılmıştır.

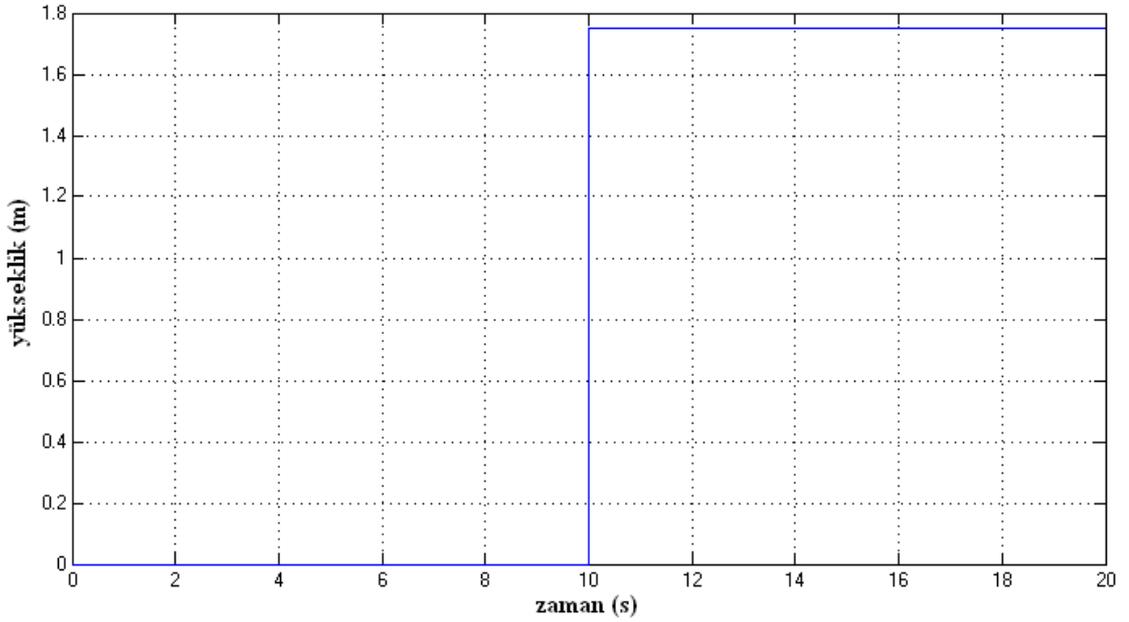
Bu çalışmada bir MPC probleminin çözümünde toplam maliyetin zaman gecikmesine bağlı olarak zamanla arttığı görülmüştür. Bu yüzden zaman gecikmeli MPC problemi için optimizasyonunu toplam maliyetin zaman gecikmesine bağlı olarak artmaya başladığında optimizasyon sürecinin sonlandırılması daha küçük değerlikli kontrol sinyallerinin bulunmasına imkan verebilir.

4.4.11 Önerilen Denetleyici Modelinin Hexacoptere Uygulanması

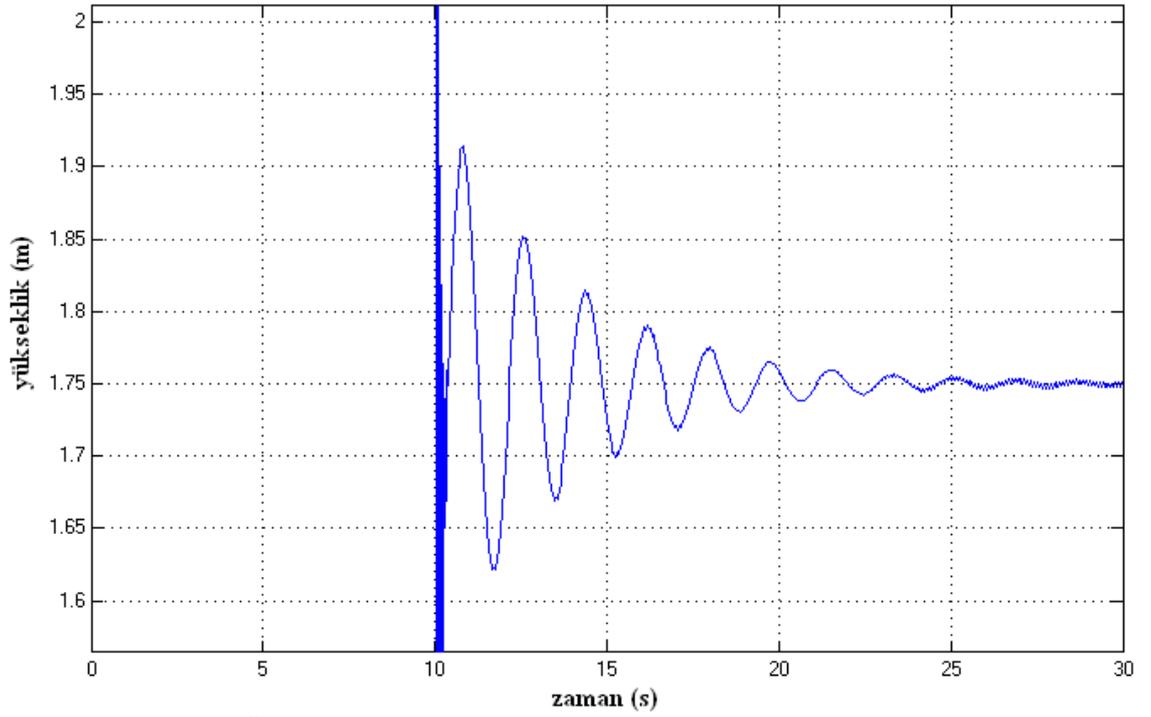
Önerilen yöntemin ve sistemin istenilen yükseklik referansı Şekil 4.21' de gösterilmiştir. Sistemde örnekleme zamanı 5 ms olarak alınmıştır. Sistemin zaman gecikmesi 2,5 ms olarak belirlenmiş, kontrol değerini bulmak için gerekli iterasyon sayısı ise 10' dur. Sistemde gecikmeye bağlı olarak ikinci iterasyondan sonra ön kesme tekniği uygulanarak kontrol sinyali elde edilmiş ve olgunlaşmamış hali ile sisteme uygulanmıştır. Şekil 4.20' de gecikme zamanının örnekleme zamanına oranı x eksenini ile gösterilmiş, y eksenini ile maliyet değerlerinin iterasyon sayısına bağlı olarak değişimi belirtilmiştir. Şekil 4.22' de sistem çıkışının ön kesmeli tekniğe bağlı olarak değişimi ifade edilmiştir. Buna göre simülasyon ortamında sistem 7 sn içerisinde istenilen yükseklik değeri olan 1, 75 m' ye oturmaktadır. Sistemdeki hata değerinin değişimi Şekil 4.23' de gösterilmiştir. MPC için H_p tahmin ve H_u kontrol ufukları 60 ve 20 olarak belirlenmiştir.



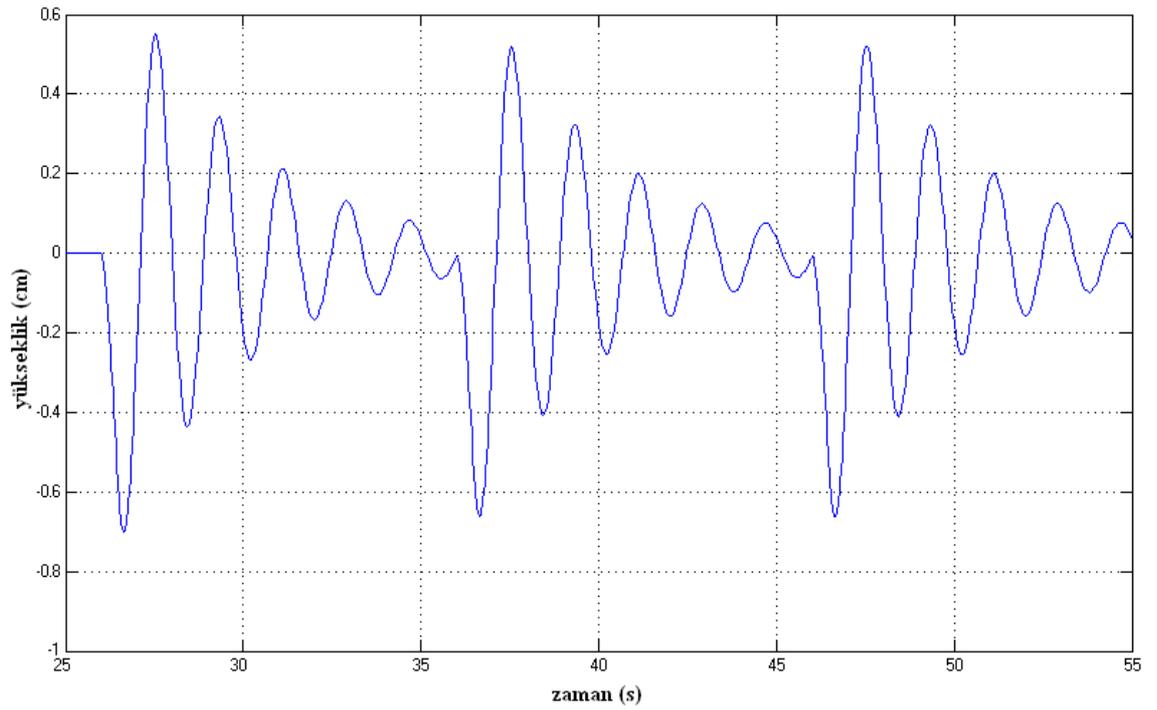
Şekil 4.20 Hexacopter sistemine uygulanan kontrol sinyal maliyet değerlerinin gecikme zamanına göre değişimi



Şekil 4.21 Sistem için belirlenen yükseklik değerinin zamana göre değişimi



Şekil 4.22 Ön kesmeli MPC ile sistemin istenilen yüksekliğe oturması



Şekil 4.23 Ön kesmeli MPC ile yükseklik değerinin sürekli haldeki değişimi

EŞ ZAMANLI KONUMLANDIRMA VE HARİTALAMA

5.1 Eş Zamanlı Konumlandırma ve Haritalama

Simultaneous Localization and Mapping (SLAM) yani eş zamanlı konumlandırma ve haritalama (EZKH) yönteminin temel amacı; otonom araçların, bilinmeyen bir konumda ve çevrede kendi konumu hesaplarırken, aynı anda bulunduğu konumun haritasını artımsal olarak çıkarmasıdır. İşte bu sorunun cevabı ile insansız araçlar tam bir işlevselliğe kavuşurlar. Ayrıca, konumlandırma ve haritalama problemleri insansız hava araçlarında otonom kontrolün temelini oluşturur.

Olasılıksal SLAM problemi ilk defa 1986 yılında IEEE Robotics and Automation konferansında ele alınmıştır. Bu yıllarda ilk defa SLAM probleminde kestirim yöntemleri denenmiştir. 1995 yılında ilk defa hareketli robotlarda kullanılmıştır. 1999' da International Symposium on Robotics' de SLAM probleminde Kalman Filtresi tabanlı çözümler ile yakınsama konusu ele alınmış, 2000' li yıllarda ise, konu hakkındaki çalışmalar ağırlıklı olarak veri birliği, veri karmaşıklığı ve algoritmaların uygulanmaları konusundaki zorluklar/fırsatlar konusunda yoğunlaşmıştır [50].

SLAM problemin ele alınmasındaki en önemli kırılma noktası konumlandırma ve haritalama problemlerinin ilk defa birlikte ele alınmasıyla olmuştur. Bu iki problem, ilk defa tek bir kestirim problemi olarak yer işaretleri (landmark) arasındaki korelasyon ile ifade edilmiştir. Korelasyon, genel olarak iki rastgele değişken arasındaki doğrusal ilişkinin yönü ve gücüdür [50].

SLAM problemi, insansız araçların, bir çevrenin haritasını çıkarması ve çıkardığı bu harita ile konumunu belirlemesidir. Bir SLAM algoritması aşağıdaki gibi ifade edilebilir [50]:

- X_k : Araçın konumunu ve yönelimi.
- u_k : k-1 zamanında uygulanan, k zamanına ve aracı X_k durumuna getirmeyi amaçlayan kontrol vektörü.
- m_i : i. yer işaretinin yerini ifade eden vektör
- z_{ik} : i yer işaretinden k zamanında aracın konumundan alınan gözlem
- Bununla birlikte $X_{0:k}$, $U_{0:k}$, m , $Z_{0:k}$ kümeleri tanımlanır.

SLAM problemi için gerekli birinci aşama olasılıksal dağılımdır. Olasılıksal dağılım bize yer işaretlerinin ve aracın durumunu bildirir. Genel olarak bu problemde bizden yinelemeli (recursive) yöntemine dayanan bir çözüm istenir. k-1 zamanında kestirim ile aracın kontrol, konum ve yer işaretleri gözlem çiftine dayanan dağılım tahmini yapılır. Bu dağılımın yapılabilmesi için bir geçiş ve gözleyici modeli; kontrol ve gözleyici girişlerinin etkilerinin tanımlanmasında gereklidir [50].

Gözleyici modeli Z_k ile araç konumunun ve yer işaretlerinin olduğu noktaları ifade eder.

$P(Z_k, X_k, m)$ araç ve haritanın tanımlı olması, gözlemlerin şartlı olarak bağımsız oldukları kabulü yapılır [50]. $P(X_k | X_{k-1}, u_k)$ hareket modeli ise araç için geçerli olan olasılık dağılımlarını durum geçişleri üzerine tanımlar. Bu tip algoritmalarda durum geçişlerinin Markov süreci olduğu kabul edilir ki X_k sadece X_{k-1} ' e bağlıdır ve u_k kontrolü haritadan ve gözlemlerden bağımsızdır. Bu durumda SLAM algoritması olasılıksal yapıda iki aşamalı yinelemeli bir yöntemle uygulanabilir [50].

1. Zaman güncellemesi:

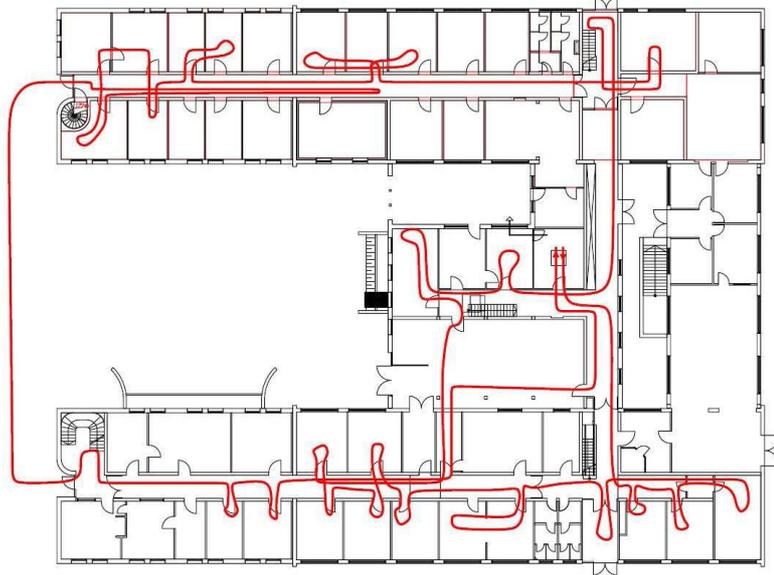
$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) = \int P(x_k | x_{k-1}, u_k) \times P(x_{k-1}, m | Z_{0:k}, U_{0:k-1}, x_0) dx_{k-1} \quad (5.1)$$

2. Ölçüm güncellemesi:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) = \left(\frac{P(z_k | x_k, m) P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k | Z_{0:k-1}, U_{0:k})} \right) \quad (5.2)$$

5.2 Başlıca Algoritma Yapıları

2D I-SLSJF (Iterated-Sparse local submap joining filter) yerel harita parçacıklarını birleştirerek en küçük kareler ve sparse filtres, yöntemlerine dayanan bir algoritmadır. Şekil 5.1' de algoritmaya ait örnek harita kestirimi görülmektedir [49], [51], [52].



Şekil 5.1 Iterated-Sparse local submap joining filter [51]

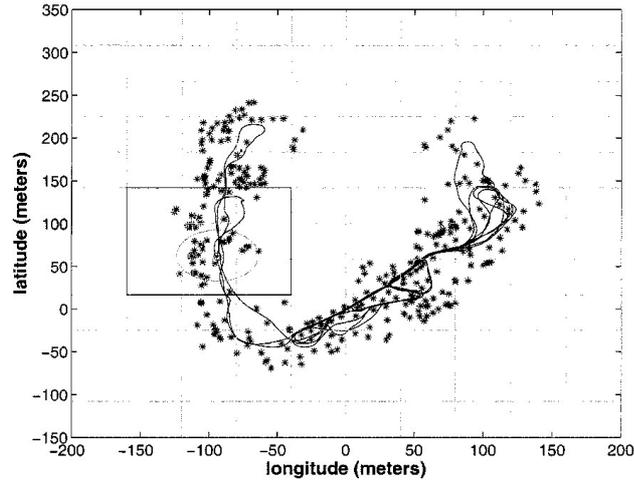
CAS Robot Navigation Toolbox: Genel kullanıcı lisanslı MATLAB ortamı için geliştirilmiş bir yazılımdır. Özellik çıkarımı, odimetri modeli ve veri ilişkisi stratejilerine imkan verir. Şekil 5.2' de algoritmaya ait örnek harita kestirimi görülmektedir [49], [53].



Şekil 5.2 Robot Navigation Toolbox ile elde edilen harita kestirimi [53]

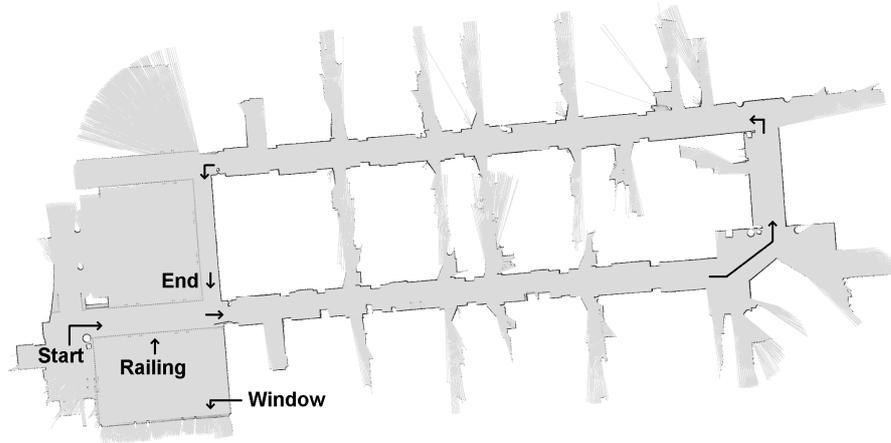
CEKF-SLAM (Compressed Extended Kalman Filter) Sıkıştırılmış Genelleştirilmiş Kalman Filtresi: Bu algoritma SLAM Kalman Filtresi uygulamalarındaki işlemel karmaşıklığı

azaltmak için geliştirilmiştir. Temeli SLAM algoritmalarında kullanılan durum vektörünü anlık olarak aktif yerel durum vektörü ve diğerleri olarak ikiye ayırmasıdır. Genelleştirilmiş Kalman Filtresi yardımı ile sadece aktif yerel durum vektörü güncellenir. Diğer durumlar için gerekli olan güncelleme bilgileri ise yardımcı katsayı matrislerine sıkıştırılır. Yerel durum değiştirildiğinde, bütün durumların güncellenmesi yine Genelleştirilmiş Kalman Filtresi'nde olduğu gibi icra edilir [49], [54]. Şekil 5.3' de algoritmaya ait örnek harita kestirimi görülmektedir.



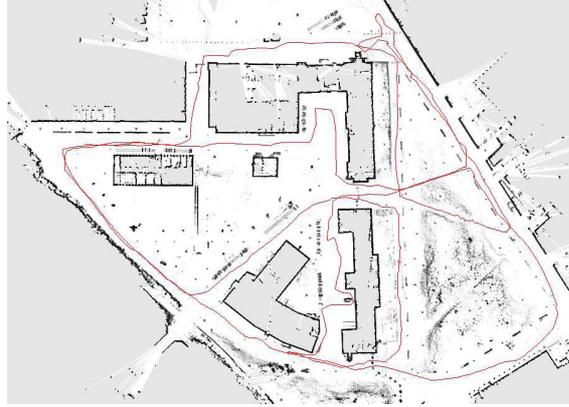
Şekil 5.3 CEKF-SLAM uygulaması ile harita kestirimi [54]

DP-SLAM (Distributed Particle): Bu algoritma tipinde konumlandırma ve harita çıkarma problemi yer işaretleri kullanılmadan çözülür. Bu durumda parçacık filtresinden elde edilen harita her bir örnekleme zamanında yeniden kopyalanır. Şekil 5.4' te algoritmaya ait örnek harita kestirimi görülmektedir [49], [55], [56].



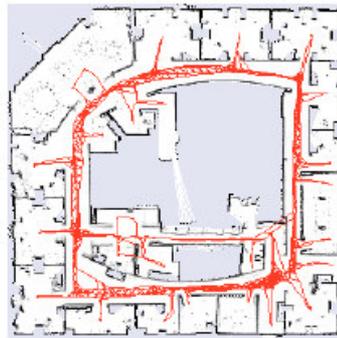
Şekil 5.4 Örnek bir DP-SLAM uygulaması [55]

Gmapping (Grid Mapping): Temeli Rao-Blackwellized parçacık filtresine dayanır. Bu yöntemde kendine ait olan haritayı taşıyan parçacık filtresinde kullanılan sayıyı azaltmak amaçlanır. Haritası çıkarılmak istenen bölge grid metodu ile küçük dört köşeli parçalara ayrılır. Sonrasında olasılık dağılım hesabında robotun hareketi ve son yapılan gözlemler kullanılır. Böylelikle robotun durumu hakkındaki belirsizlik azaltılırken, parçacık azaltılması ile oluşan problemler seçici yeniden örnekleme işlemleri kullanılarak ortadan kaldırılır. Şekil 5.5' te algoritmaya ait örnek harita kestirimi görülmektedir [49], [57], [58].

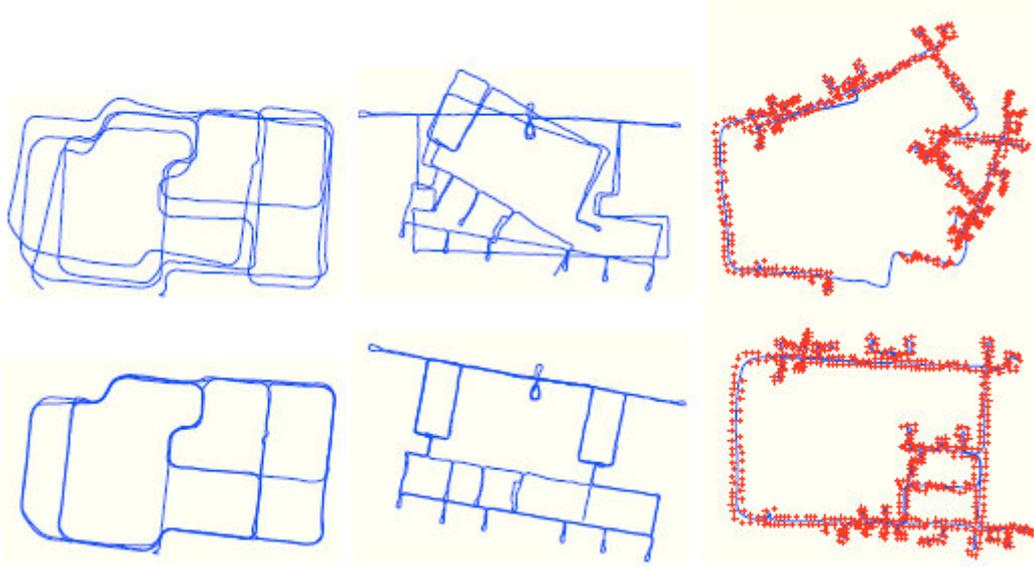


Şekil 5.5 Freiburg Üniversitesi kampüsündeki grid mapping uygulaması [57]

HOG-Man (Hierarchical Optimization for Pose Graphs on Manifolds): Manifoldlar üzerindeki Durum Grafları için Silsileli Eniyileme yönteminde robot sahada iken yöntem bütün alan yerine sadece kaba olarak alanın yapısını düzeltir. Bu şekilde robot sadece haritanın kısımlarını güncellerken, sadece parça haritaya ait topladığı bilgilerle ilgili sorguları gerçekleştirir. Yöntem hatanın azaltılması yaklaşımı ile robotun olduğu sahayı bir manifold olarak ele alır. Böylelikle harita çıkarımı sırasında durum uzay parametrelerinde oluşabilecek tekillik problemi ortadan kalkar. Şekil 5.6' da algoritmaya ait örnek harita kestirimi görülmektedir [49], [59].

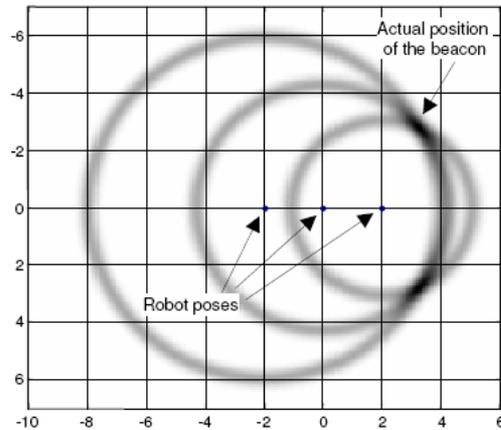


Şekil 5.6 HOG-Man yöntemiyle oluşturulmuş harita örnekleri [59]



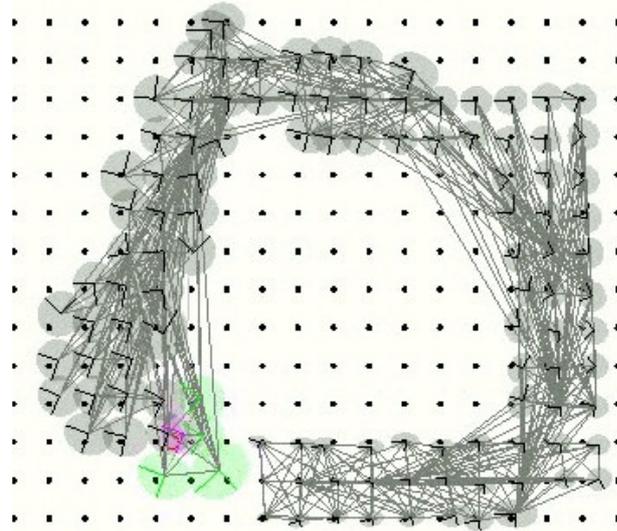
Şekil 5.8 Manifold Tool kit yöntemiyle oluşturulan haritalar [62]

RO-SLAM (Range-Only SLAM): Bu yöntemle ile SLAM problem sırasında oluşabilecek Bayes etkileşim problemine çözüm bulmak amaçlanmıştır. Robotun sahada kademeli hareketi sırasında önceki bilgilere sahip olmadan konumunun kestirimini hedefler. Aracın kademeli hareketi sırasındaki izleri tanımlamada ortamda sadece odometri ve mesafe sensöründen yararlanır. Veri ilişkisi problemi ile uğraşmaz. Öklidyen kordinatlarında her bir harita parçası için iz (parçacık) dağılımları arasındaki şartsal bağımsızlığı ortaya çıkararak Gauss toplamı elde eder. Böylelikle uygun bir olasılık gözlem modeli gecikmeye gerek olmadan online operasyon için elde edilir. Yöntemin devamında bu şekilde yapılan uygulamanın Monte-Carlo yaklaşımına göre daha iyi sonuç verdiği iddia edilmiştir [49], [63]. Şekil 5.9’ da algoritmaya ait örnek harita kestirimi görülmektedir.



Şekil 5.9 Range-only SLAM [63]

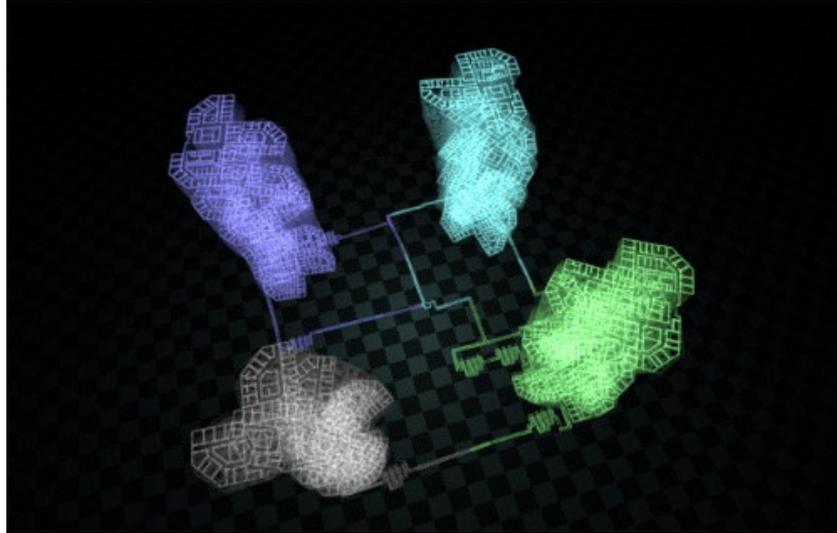
TJTF for SLAM (Thin Junction Tree Filters for SLAM): SLAM probleminin özüne yapılan vurguda bu algoritma öncelikle SLAM'ı hareketli robotlardaki temel bir problem olarak tanımlar. Bilindiği üzere robot sahada seyrüseferini gerçekleştirirken aynı zamanda etrafını saran çevrenin haritasını çıkarmalı ve bu haritada kendini konumlandırmalıdır. Bugüne kadar gerçekleştirilen en yaygın uygulama SLAM problemi için Kalman Filtresi yaklaşımıdır. Kalman Filtresi başlangıç için iyi bir yöntem olmakla birlikte algoritmik olarak SLAM problemini boyutlandırmada yeterince iyi olmadığı vurgulanmıştır. Özellikle Kalman Filtresi'nin olumsuz etkileri olan filtrenin zaman karmaşıklığı, tahmin edilen durumun boyutunun güncellenmesinin haritadaki yer işaretleri ile birlikte karesel artmasından bahsedilen bu algortmada yeni bir filtreleme tekniği olan ince birleşik ağaç tekniğinden (thin junction tree) bahsedilir. Bu filtreleme yönteminde aracın tahmin edilen durumunun izlenmesi hareketi ince bir birleşik ağaç şeklinde yapılır. SLAM problemine uygulandığında doğrusal uzay oluşturan ağaç filtreleri, aracın tahmin edilen durumunda doğrusal zaman filtreleme işlemi olarak yapar [49], [64], [65]. Şekil 5.10' da algortmaya ait örnek harita kestirimi görülmektedir.



Şekil 5.10 Thin Junction Tree Filters for SLAM yöntemi ile haritalama örneği [64]

Treemap: Ağaç Haritası algoritması SLAM problem için geliştirilmiş özellik çıkarımına dayanan bir algortmadır. Artımlı olasılıksal çıkarım algoritması olan Ağaç Haritası algoritması birden fazla düşük dereceli Gauss fonksiyonlarının çarpımından meydana gelen yüksek dereceli bir Gauss fonksiyonu olarak düşünülebilir. En büyük avantajlarından bir tanesinin farklı tip SLAM problemlerine uygulanabilir olmasıdır.

Ağaç Haritası algoritmasının en büyük katkısı oldukça etkin bir Gauss fonksiyonu çıkarımı yapabilmesidir. Harita üzerindeki iki boyutlu yaklaşık bir milyon özelliği saniyenin yarısında işleyebilmektedir. Kendi alanındaki benzer algoritmaları aynı işi yapan Ağaç Haritasının daha hızlı olduğu iddia edilir. Ağaç Haritası Algoritması iki katmandan oluşur. Arka planda koşan algoritma katmanında problem ait tüm zorluklar ele alınır. Daha öncede belirtildiği üzere birden fazla düşük dereceli Gauss fonksiyonlarının çarpımından oluşan yüksek mertebeli Gauss fonksiyonunda düşük mertebeli fonksiyon parçacıkları ölçümlere karşılık gelir. Algoritmadaki diğer katman işte bu ölçümleri düşük mertebeli Gauss fonksiyonlarına dönüştürür ve arka plana iletir ve uygulama ağırlıklı bir yaklaşım politası tanımlar. Ayrıca arka plan algoritması artımlı olarak ortalama sonuç Gauss fonksiyonunu lineerleştirilmiş düşük mertebeli Gausslar üzerinden hesaplar. Bu işlemlerin sonucunda elde edilen sonuç ağaç haritası sürücüsü tarafından harita kestirime dönüştürülür. Yayınlanan akademik çalışmalarda iki boyutlu özellik tabanlı SLAM, iki boyut - üç serbest dereceli durum ilişkisi tabanlı SLAM, odimetri olmadan üç boyutlu-altı serbest dereceli özellik tabanlı SLAM için geliştirilen uygulamalar ele alınmıştır [49], [66], [67]. Şekil 5.11' de algoritmaya ait örnek harita kestirimi görülmektedir.



Şekil 5.11 Ağaç haritası uygulaması [66]

5.3 Gmapping (Grid Mapping)

Grid Haritalamada kullanılan yöntemlerden biri olan Gmapping Rao Blackwellized Parçacık Filtresi tabanlı bir algoritmadır. Bu bölümdeki çalışma [57] nolu referansta bulunabilir. Rao-Blackwellized Parçacık Filtreleri ile ilgili bilgi [57], [58] kaynaklarından elde edilebilir. Gmapping algoritması SLAM problemi için haritayı gridlere ayırmakla birlikte, sadece robotun hareketi değil en son yapılan gözlemi de olasılıksal dağılım hesaplarına ilave eder. Dağılım fonksiyonun hesaplanmasında kullanılan parçacıkların sadece bir kısmı seçilerek hesaplamalar yapılır [57].

Gmapping algoritması başlıca iki ana kısımdan oluşur [57].

- Rao-Blackwellized Parçacık Filtreleri SLAM problemine uygulanırken birleşik sonraki durum (5.3),

$$P(x_{1:t}, m | z_{1:t}, u_{1:t-1}) \quad (5.3)$$

şeklinde hesaplanır. Burada:

$x_{1:t}$: robotun gezingesini,

m : haritayı,

$z_{1:t}$: gözlemleri,

$U_{1:t-1}$: odometri ölçümlerini ,

$p(x_{1:t}, m | z_{1:t}, U_{1:t-1})$ ifadesi ise gözlemlere ve odometri ölçümlerine bağlı olan şartlı olasılığı ifade eder. Bu durumdaki dağılım fonksiyonu (5.4):

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = P(m | x_{1:t}, z_{1:t}) \cdot P(x_{1:t} | z_{1:t}, u_{1:t-1}) \quad (5.4)$$

Bu formüldeki ilk kısım konuma ve gözlemlere bağlı olarak haritayı oluştururken, ikinci kısım ise gözlemlere ve odometri ölçümlerine bağlı olarak konumu hesaplar. Yine ilk kısım bilinen durumlarla haritalamayı ikinci kısım ise konum kestirimini ifade eder.

Bir önceki eşitlikteki çarpım ilk olarak robotun gezingesini hesaplar. Sonrasında bu gezingeye ait haritayı hesaplamayı mümkün kılar. Harita kestimi her zaman için robotun durum kestirimine bağlı olması, bu tip hesaplamayı etkin kılar. Bu yöntemle yapılan hesaplama literatürde sıklıkla Rao-Blackwellization olarak adlandırılır [57].

Eşitlik (5.4)' teki harita kestirimi $p(m|x_{1:t}, z_{1:t})$, bilinen durumlarla harita kestirimi tekniği (mapping with known poses) ile hesaplanır. Eşitliğin ikinci kısmındaki hesaplama için parçacık filtresi tekniği uygulanabilir. Buradaki her bir parçacık robotun potansiyel bir gezinesini gösterir. Yine herbir parçacık kullanılarak bireysel bir harita oluşturulur. Haritalar herbir parçacığa ait gezege ve gözlemlerden oluşturulur [57], [58].

Rao-Blackwellized (SIR) Parçacık Filtresi: Bilinen bir SIR parçacık filtresi algoritmasıdır. SIR; sampling (örnekleme), importance (önem), resampling (yeniden örnekleme) baş harflerinden oluşur. İlk defa dinamik Bayes Network problemlerini çözümlenmede istatistik de kullanılmıştır. Haritada bu algoritma ise artımsal olarak sensör gözlemlerini ve odometri ölçümlerini okumakta ve burdan hareketle robotun gezinesi hesaplanmaktadır [57], [59].

Algoritma yapısı başlıca dört adımda ifade edilebilir. Bunlar sırası ile [57]:

- Örnekleme: Kullanılacak yeni nesil parçacıklar $x_t^{(i)}$, bir önceki nesil yardımı ile hesaplanır $x_{t-1}^{(i)}$. Bu durum için öneri dağılım fonksiyonu π den yararlanır. En sıklıkla kullanılan dağılım fonksiyonu odometrik hareket modelidir.
- Ağırlıkların Öneminin İfadesi: Her bir parçacık için bir ağırlık önemi belirlenir. Önemlerin belirlenmesi sırasında önem örnekleme prensibine bağlı olarak bir ağırlık atanır (5.5).

$$w_t^{(i)} = \frac{p\left(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1}\right)}{\pi\left(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1}\right)} \quad (5.5)$$

- Yeniden örnekleme: Eldeki parçacıklar ağırlık değerleri ile orantılı olarak yeniden seçilirler. Bu adım parçacıkların sayısını sınırlandırmak ve etkinliklerini belirlemek açısından önemlidir. Sonuç itibari ile sadece sınırlı sayıda parçacık kullanılacaktır. Ayrıca yeniden örnekleme; hedef dağılımın, kestirilen dağılımdan farklı olduğu durumlarda parçacık filtresini kullanabilmeyi mümkün kılar.

- Harita kestirimi: Her bir parçacık için harita kestirimi, harita kestirim formülüne göre hesaplanır $p\left(m^{(i)} \mid x_{1:t}^{(i)}, z_{1:t}\right)$. Bu formüle göre harita kestirimi hesaplanırken

o örneğe (parçacığa) ait robot gezinteleri $x_{1:t}^{(i)}$ ve gözlemlerden $z_{1:t}$ yararlanır.

Bu algoritmanın uygulanması sırasında yeni bir gözlem yapıldığında gezintelerin ağırlıklarının yeniden gözden geçirilmesi gerekmektedir. Zamanla robotun aldığı yol, gezintenin uzunluğu artacağından, bu sürecin etkin bir şekilde uygulanması zorlaşır. Doucet ve Grisettinin çalışmalarında bu süreçte parçacıkların önem ağırlıklarının belirlenmesinde dağılım fonksiyonunun aşağıdaki özelliğe sahip olduğu düşünülür [57].

$$\pi\left(x_{1:t} \mid z_{1:t}, u_{1:t-1}\right) = \pi\left(x_t \mid x_{1:t-1}, z_{1:t}, u_{1:t-1}\right) \cdot \pi\left(x_{1:t-1} \mid z_{1:t-1}, u_{1:t-2}\right) \quad (5.6)$$

(5.5) ve (5.6) formülleri kullanılarak ağırlıklar; (5.7) ile hesaplanır [57].

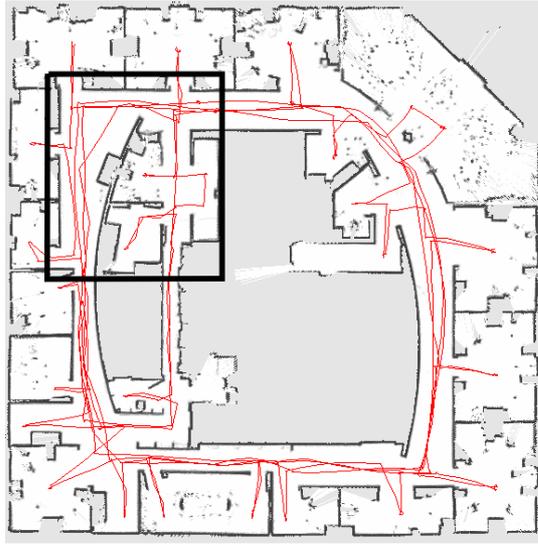
$$\begin{aligned} w_t^{(i)} &= \frac{p\left(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1}\right)}{\pi\left(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1}\right)} \\ &= \frac{\eta p\left(z_t \mid x_{1:t}^{(i)}, z_{1:t-1}\right) p\left(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1}\right) p\left(x_{1:t-1}^{(i)} \mid z_{1:t-1}, u_{t-2}\right)}{\pi\left(x_t^{(i)} \mid x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1}\right) \underbrace{\pi\left(x_{1:t-1}^{(i)} \mid z_{1:t-1}, u_{t-2}\right)}_{w_{t-1}^{(i)}}} \end{aligned} \quad (5.7)$$

$$\begin{aligned} w_t^{(i)} &= \frac{\eta p\left(z_t \mid x_{1:t}^{(i)}, z_{1:t-1}\right) p\left(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1}\right) p\left(x_{1:t-1}^{(i)} \mid z_{1:t-1}, u_{t-2}\right)}{\pi\left(x_t^{(i)} \mid x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1}\right) \underbrace{\pi\left(x_{1:t-1}^{(i)} \mid z_{1:t-1}, u_{t-2}\right)}_{w_{t-1}^{(i)}}} \end{aligned} \quad (5.8)$$

$$\propto \frac{p\left(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}\right) p\left(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1}\right)}{\pi\left(x_t \mid x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1}\right)}$$

Burada $\eta = 1 / p\left(z_t \mid z_{1:t-1}, u_{1:t-1}\right)$ tüm parçacıklar için eşit olan ve Bayes kuralından gelen normalizasyon çarpanıdır. Günümüzdeki pek çok parçacık filtresi uygulamasında (5.8)'deki tekrarlanan formül kullanılır [57].

Geliştirilmiş Öneri Dağılım Fonksiyonu: Parçacık filtreleri ve SLAM uygulamalarında kullanılan parçacıkların azaltılması riskini ortadan kaldırmak için hesaplamalarda daha etkin öneri yayılım fonksiyonları kullanılır. Gmapping tekniği geliştirilmiş öneri dağılım fonksiyonu ve adaptif yeniden örnekleme ile diğer tekniklerden SLAM uygulamalarında ayrılır [57]. Şekil 5.12 ve Şekil 5.13' de Intel araştırma laboratuvarında gerçekleştirilen Gmapping algoritması görülmektedir.



Şekil 5.12 Intel Research Lab içerisindeki Gmapping uygulaması [57]



Şekil 5.13 Intel Lab. Gmapping uygulamasında robot gezintileri [57]

5.4 Gmapping Geliştirilmiş Öneri Dağılım Fonksiyonu ve Hesaplanması

Öneri yayılım fonksiyonundan, algoritmanın tahmin altdöngüsü sırasında yeni nesil parçacıklar seçilmelidir. Öneri dağılım fonksiyonu, hedef dağılımına ne derece yakınsa filtre cevabıda o derece etkindir. Hedef dağılım fonksiyonundan doğrudan parçacıklar seçilebilseydi, parçacıkların ağırlıkları eşit olacak ve yeniden örnekleme ortadan kalkacaktı. Gmapping algoritmasında öneri dağılım fonksiyonu olarak odometrik hareket modeli kullanılmıştır. Odometrik hareket modeli bütün robot türlerinde kolay hesaplanabilir olduğundan SLAM algoritmalarında sıklıkla kullanılır. Bununla birlikte, parçacıkların önem ağırlıkları ise gözlem modeline göre hesaplanır. Bunun için eşitlik (5.8)' deki π ifadesi gözlem modeli ile değiştirilir (5.10) [57].

Öneri dağılım fonksiyonu, sensörlerin robot durum kestiriminden daha hassas olduğu durumlarda bize suboptimal bir çözüm sunar. Bu sorunun üstesinden gelmek için son sensör gözlemi z_t dikkate alınır. Sensör gözleminin öneri dağılım fonksiyonu ile bütünleştirilmesi ile gözlem benzerliklerinin anlamlı bölgelerinin örneklenmesi üzerine odaklanabilir. Doucet [57], [68] göre dağılım fonksiyonu (5.9);

$$p\left(x_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}\right) = \frac{p\left(z_t | m_{t-1}^{(i)}, x_t\right) p\left(x_t | x_{t-1}^{(i)}, u_{t-1}\right)}{p\left(z_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}\right)} \quad (5.9)$$

parçacık ağırlıklarının değişimine göre optimal öneri dağılım fonksiyonudur. Bu eşitlik kullanılarak ağırlıklar aşağıdaki eşitlikler ile yeniden hesaplanır [57].

$$\begin{aligned}
w_t^{(i)} &= w_{t-1}^{(i)} \frac{\eta p\left(z_t | m_{t-1}^{(i)}, x_t^{(i)}\right) p\left(x_t^{(i)} | x_{t-1}^{(i)}, u_{t-1}\right)}{p\left(x_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}\right)} \\
&\propto w_{t-1}^{(i)} \frac{p\left(z_t | m_{t-1}^{(i)}, x_t^{(i)}\right) p\left(x_t^{(i)} | x_{t-1}^{(i)}, u_{t-1}\right)}{p\left(z_t | m_{t-1}^{(i)}, x_t\right) p\left(x_t | x_{t-1}^{(i)}, u_{t-1}\right)} \\
&= w_{t-1}^{(i)} p\left(z_t | m_{t-1}^{(i)}, x_t^{(i)}, u_{t-1}\right) \\
&= w_{t-1}^{(i)} \int p\left(z_t | x'\right) p\left(x' | x_{t-1}^{(i)}, u_{t-1}\right)
\end{aligned} \tag{5.10}$$

Hareketli robotlar lazer gibi hassas sensörlerle modellendiğinde, lazerin hassaslığı gözlemlerde aşırı keskin benzerlik fonksiyon çıktılarına sebep vereceğinden geliştirilmiş öneri dağılım fonksiyonu kullanmak oldukça uygundur. Yer işaretleri bazlı SLAM probleminde geliştirilmiş öneri dağılım fonksiyonun hesaplanmasında [57], [70] Gauss yaklaşımı kullanan Rao-Blackwellized parçacık filtresi kullanılmıştır. Robotun durumu, her bir parçacık filtresi için Kalman Filtresi kullanan Gauss yaklaşımı ile kestirilmiştir. Buna rağmen bu çalışmada geliştirilmiş öneri dağılım fonksiyonun hesaplanmasında yer işaretlerinden farklı olarak yoğunlaştırılmış harita bölmeleri kullanılmıştır.

Yeni nesil örnekleri elde etmede veri üzerinde Gauss yaklaşımı kullanılmıştır. Yukarıda bahsedilen diğer yöntemden farklı olarak buradaki Gauss yaklaşımı hesaplanması sırasında gözlem benzerlik fonksiyonun anlamlı bölgesini tanımlamak için tara-eşle algoritması kullanılır. Tara-eşle algoritması sonucunda gözlem içindeki anlamlı bölgeler ortaya çıkartılarak hedef dağılım fonksiyonuna göre örneklenir ve geliştirilir. Sırası ile her bir K tane parçacık için Gauss parametreleri hesaplanır. Hesaplanan parametrelerin ortalama değerleri alınırken odometrik hareket modelinden faydalanılır. Gauss parametreleri sırası ile aşağıdaki formüller ile hesaplanır. İkinci formül (5.12) değişimi belirtirken ilk formül (5.11) ortalama ifade eder [57].

$$\mu_t^{(i)} = \frac{1}{\eta^{(i)}} \cdot \sum_{j=1}^K x_j \cdot p\left(z_t | m_{t-1}^{(i)}, x_j\right) \cdot p\left(x_j | x_{t-1}^{(i)}, u_{t-1}\right) \tag{5.11}$$

$$\Sigma_t^{(i)} = \frac{1}{\eta^{(i)}} \cdot \sum_{j=1}^K p\left(z_t | m_{t-1}^{(i)}, x_j\right) \cdot p\left(x_j | x_{t-1}^{(i)}, u_{t-1}\right) \cdot (x_j - \mu_t^{(i)})(x_j - \mu_t^{(i)})^T \quad (5.12)$$

Her bir parçacık için kullanılan normalizasyon faktörü (5.13) [57]:

$$\eta^{(i)} = \sum_{j=1}^K p\left(z_t | m_{t-1}^{(i)}, x_j\right) \cdot p\left(x_j | x_{t-1}^{(i)}, u_{t-1}\right) \quad (5.13)$$

Bu yolla optimum öneri dağılım fonksiyonu yeni nesil parçacıkları belirlemek için kullanılır. Öneri dağılım fonksiyonunu kullanarak ağırlıklar aşağıdaki şekilde (5.14) ile hesaplanır [57].

$$\begin{aligned} \omega^{(i)} &= \omega_{t-1}^{(i)} \cdot p\left(z_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}\right) \\ &= \omega_{t-1}^{(i)} \cdot \int p\left(z_t | m_{t-1}^{(i)}, x'\right) \cdot p\left(x' | x_{t-1}^{(i)}, u_{t-1}\right) dx \\ &\approx \omega_{t-1}^{(i)} \cdot \sum_{j=1}^K p\left(z_t | m_{t-1}^{(i)}, x_j\right) \cdot p\left(x_j | x_{t-1}^{(i)}, x_{t-1}\right) \\ &= \omega_{t-1}^{(i)} \cdot \eta^{(i)} \end{aligned} \quad (5.14)$$

Buradaki normalizasyon çarpanı yeni önerinin hesaplanmasında değinilen Gauss parametrelerinin hesaplanması sırasında kullanılan normalizasyon çarpanı ile aynıdır [57].

5.5 Yeniden Örnekleme

Parçacık filtrelerinin performansının belirlenmesinde ana değerlerden biriside yeniden örnekleme aşamasıdır. Yeniden örnekleme sırasında, daha düşük değerlikli önem ağırlıklarına sahip parçacıklar, daha yüksek değerlikli önem ağırlıklarına sahip parçacıklar ile değiştirilebilir. Bir taraftan hedef dağılımını kestirmede sınırlı sayıda parçacık kullanılacağından yeniden örnekleme filtre performansında hayati bir değere sahipken, diğer taraftan yeniden örnekleme iyi örnekleride filtreden çıkarabilir ki bunun sonucunda parçacık fakirleşmesi (güçsüzleşmesi) ortaya çıkar. Bu çalışma sırasında [57], [69] yer alan formülasyon (5.15) kullanılmıştır.

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\varpi^{(i)})^2} \quad (5.15)$$

Burada $\varpi^{(i)}$ parçacık i nin normalleştirilmiş değerini ifade eder [57].

Bu formülasyonun anlamı şu şekilde açıklanabilir: Eğer hedef dağılımından elde edilen örnekler kullanılsaydı bu örneklerin birbirine göre olan önem ağırlıkları örnekleme prensibine göre eşit olurdu. Hedef dağılımının daha kötü tahmini ise önem ağırlıklarının daha yüksek değişimi ile sonuçlanırdı. (5.15) nolu eşitlik önem ağırlıklarının dağıtılmasında bir ölçüt olarak kullanılır. Ayrıca parçacık kümelerinin hedefe ne kadar yaklaştığını belirlemede iyi bir ölçüttür. (5.15) nolu eşitlik kullanılarak burada yeniden örnekleme gerekliliği olup olmadığına karar verilir. Eğer Neff değeri atanan eşik değerinin yarısından daha küçük ise yeniden örnekleme süreci gerçekleştirilir. Pratikte kullanılan bu yöntemle yeniden örnekleme sayıları azaltılarak iyi parçacıkların ortadan kalkma ihtimali azaltılır. Bu sayede sadece gerçekten ihtiyaç duyulduğunda yeniden örnekleme süreci işleme sokulur [57].

5.6 Bütün Algoritmaya Yeniden Bir Bakış

Her bir zaman için yeni bir ölçüm çifti vardır (u_{t-1}, z_t) . Her bir parçacık için bireysel olarak öneri hesaplanır ve o parçacığı güncellemek için kullanılır. Bunun sonucunda aşağıdaki döngü oluşur [57]:

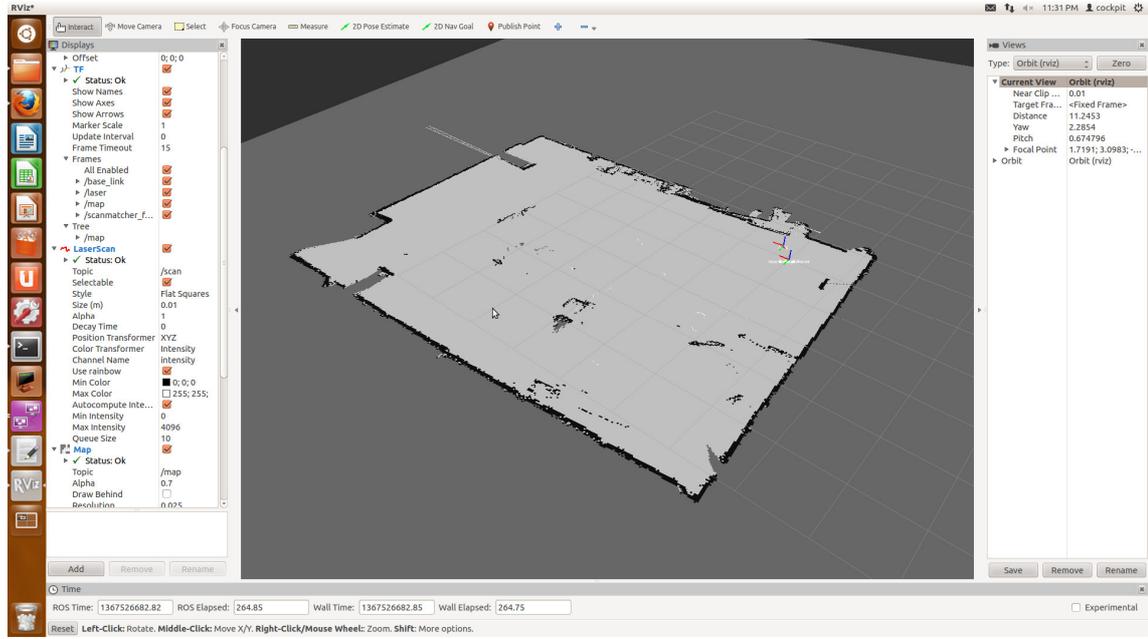
1. i parçacığına ait, robotun bir önceki konumu $x_{t-1}^{(i)}$ ve hareket modeli ölçümü u_{t-1} kullanılarak robotun durumu $x_t^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$. Burada \oplus operatörü konum birleştirme operatörüdür.
2. Başlangıç tahmini kullanılarak $x_t^{(i)}$, harita üzerindeki $m_{t-1}^{(i)}$ nokta temel alınarak bir tara-eşle algoritması çalıştırılır. Tara-eşle algoritması kullanılarak yapılan arama, $x_t^{(i)}$ çevresindeki belirli bir bölge ile sınırlıdır. Tara-eşle algoritmasının hata vermesi durumunda, robotun konumu ve ağırlıkları hareket modeline göre yeniden hesaplanır. Adım 3 ve 4 ihmal edilir.

3. Tara-eşle algoritması tarafından bildirilen robotun kestirilen konumu $\hat{x}_t^{(i)}$ etrafında bir aralık içerisinde belirli örnekleme noktaları seçilir. Bu noktalar temel alınarak, örneklenen konumlar için x_j hedef dağılımlarının $p(z_t | m_{t-1}^{(i)}, x_j) \cdot p(x_j | x_{t-1}^{(i)}, u_{t-1})$ geliştirilmesi ile öneri dağılımının ortalama ve kovaryans değerleri hesaplanır. Bu adım sırasında yine ağırlık çarpanı $\eta^{(i)}$ hesaplanır.
4. i parçacığına ait yeni konum $x_t^{(i)}$, geliştirilmiş öneri dağılımının Gauss yaklaşımı $(\mu_t^{(i)}, \Sigma_t^{(i)})$ ile çözümlenmesinden elde edilir.
5. Önem ağırlıkları güncellenir.
6. Yeni çizilen robot konumu $x_t^{(i)}$ ve gözlem z_t kullanılarak i parçacığına ait harita güncellenir.

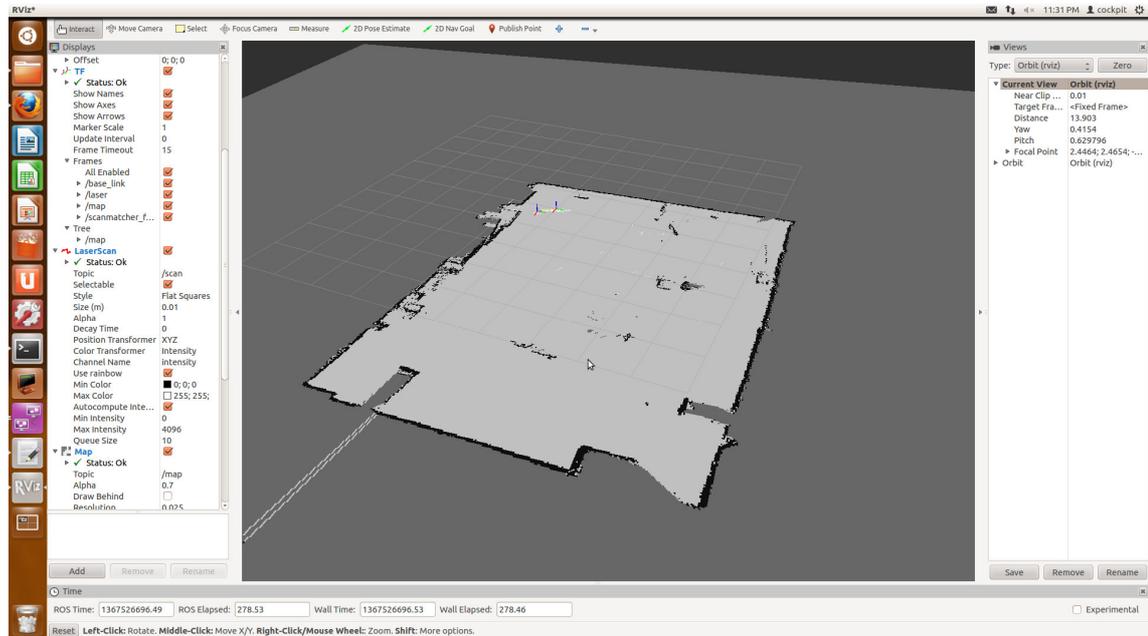
Yeni nesil örnekler hesaplandıktan sonra N_{eff} değeri baz alınarak yeniden örnekleme adımı icra edilir.

5.7 Algoritmanın Hexacoptere Uygulanması ve Saha Çıktıları

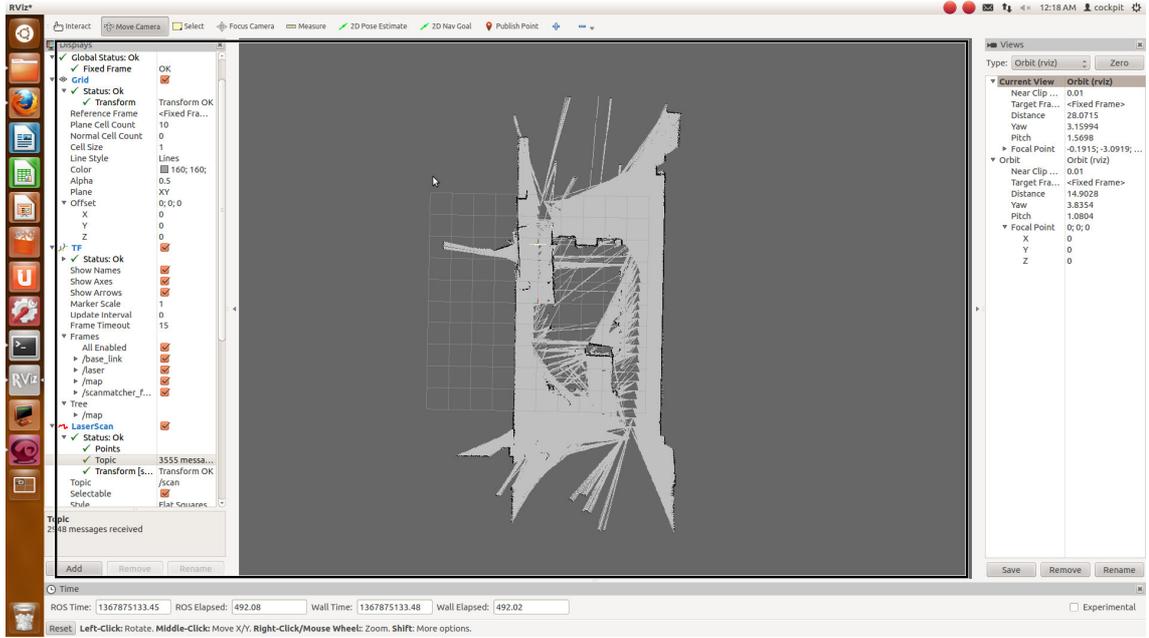
Algoritma C dilinde yazılarak mini PC üzerinde koşturulmuş, hexacopter gövdesi ile A009 araştırma laboratuvarı ve A blok zemin kat asansör çevresinin harita kestirimi yapılmıştır.



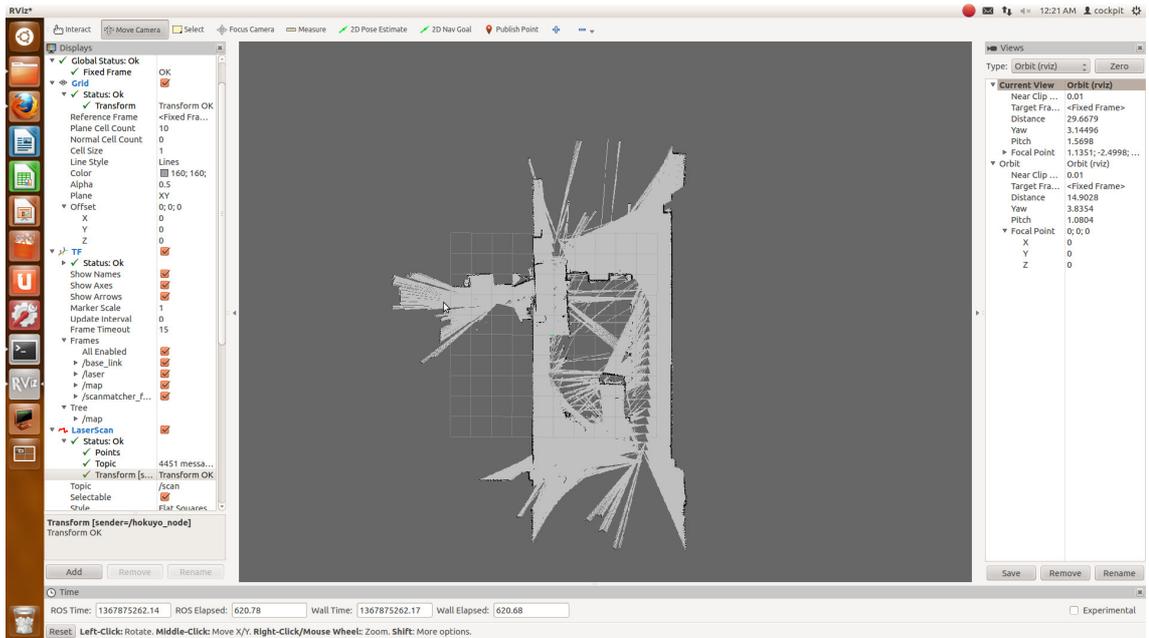
Şekil 5.14 Gmapping ile A009 araştırma laboratuvarının harita kestirimi



Şekil 5.15 Gmapping ile A009 araştırma laboratuvarında harita kestirimi-2



Şekil 5.16 Gmapping ile A Blok Zemin kat asansör çevresi harita kestirimi



Şekil 5.17 Gmapping ile A Blok Zemin kat asansör çevresi harita kestirimi-2

SONUÇ VE ÖNERİLER

Altı rotorlu bir helikopterde (hexacopter) denge ve yükseklik kontrolü için yeni bir kontrol algoritması isimli bu tezde ilk olarak hazır alınan gövde üzerinde yazılım ve kontrol arayüzü oluşturulmuştur. Daha sonra aracın denge PID, ve yükseklik problemi model öngörülü iki farklı kontrolcü tipi kullanılarak incelenmiştir. Uygulanan model öngörülü kontrolde klasik uygulamalardan farklı olarak bir kestirim tekniği önerilmiş, teorik analizi yapılmış ve uygulanmıştır. Aracın istenen irtifaya (0.2~6m) oturabilmesi sağlanmıştır. Aracın engelden kaçınması ve harita kestirimi için lazer sensörden elde edilen veriler kullanılmıştır. ROS arayüzünde geliştirilen uçuş platformu ile aracın ana merkezden istenilen koordinatlara gitmesi ve uzaktan kumanda ile kontrol edilebilmesi amaçlanmıştır. Uçuş platformu geliştirilmekte olup, araç seyrüsefer için deneme aşamasındadır.

Aracın uçuş platformu geliştirme arayüzü ROS' u ve İHA' larını tekrar hatırlayacak olursak, oto-kontrol mekanizmasına içeren tüm sistemler gerçek zamanlı sistem ve karar alma algoritmalarını işleyecek bir sistemden meydana gelir. Nasıl ki canlılarda ani hareketleri ve düzenli yapılan hareketleri gerçekleştiren omurilik ve ön beyin gibi sistemler varsa, aslında insansız hava araçlarında geliştirilen bu sistemlerin amacı doğaya olabildiğince yakınsayabilmektir. Bu modelleme için altı rotorlu model helikopter (hexacopter) üzerinde gerçek zamanlı bir mikrokontrolör kartı ve yüksek seviyeli algoritmaları işleyebilecek bir işletim sistemi (Ubuntu) bulunmaktadır. Mikrokontrolör ve işletim sistemini birbirinden ayıran ise; mikrokontrolörler, bir döngü çerçevesinde önceden belirlenen hayati fonksiyonları yerine getirirken (nefes alma,

kalp atışı...vb.); işletim sistemleri ise, isteğe bağlı başlangıcı ve bitişi her zaman net olmayan çevrimleri yerine getirir.

Helikopterde, kontrolü yapılan en önemli bölüm motorlardır. Yüksek seviyeli görevlerin durumu ne olursa olsun motor çalışması sistemi aksatmayacak ve havadaki dengeyi koruyacak şekilde devam etmelidir. Bunun için, bu sistem, atalet ölçüm ünitesinden gerçek zamanda sürekli olarak bilgi almaya devam etmektedir. Çoklu rotorlu helikopterde, ARM mimarisine sahip 32 bit STM32F4 mikrokontrolör uygulanmış ve gerçek zamanlı sistem bunun üzerine yapılandırılmıştır.

Yüksek seviyeli algoritmaları koşturmak için mini PC, 10X10 cm boyutlarını geçmeyen, 1.6 GHz' lik Atom işlemcisine sahip, linux tabanlı Ubuntu işletim sistemini çalıştırabilen bir bilgisayar kullanılmıştır. Çoklu rotorlu helikopterde koşturmak yüksek seviyeli algoritmalar; engel tanıma, harita kestirimi, lazer ve kameradan veri elde edilmesi işlemlerinde kullanılmıştır.

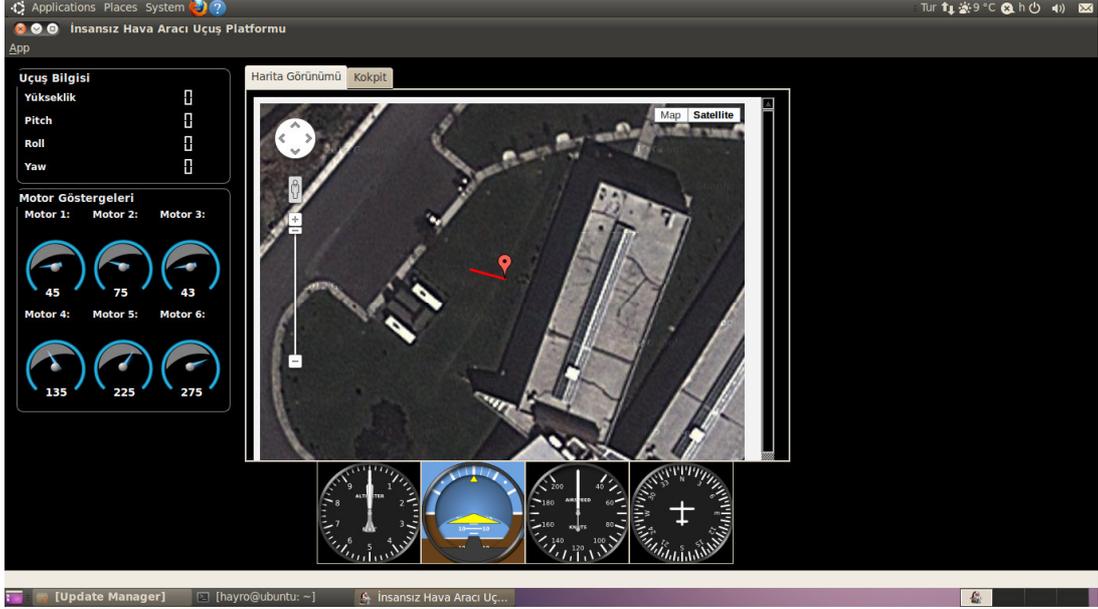
Gerçek zamanlı mikrokontrolör ve yüksek seviyeli çevrimleri çalıştıracak otonom araç modeli, farklı ortamlar için geliştirilmiş otonom araçlar ve robot tipleri için geliştirilmektedir. Bu geliştirilen yazılım mimarilerinin, etkin bir çalışma ortamı olan, özellikle akademik çalışmalar için, ROS yani Robot İşletim Sistemi tercih edilmiştir. ROS, bugün açık kaynak kodlu bir işletim sistemidir. Bununla birlikte, üzerinde çalışacağı gerçek bir işletim sistemi ortamına ihtiyacı vardır. Günümüzde, Ubuntu çoğunlukla, deneysel ortamda Mac ve Solaris işletim sistemlerinde çalışabilmekte ve bu sistemler üzerinde ROS uygulamaları geliştirilebilmektedir. Yine ROS üzerinde C++, Phyton, Java ile otonom araçlar için yazılım geliştirmek mümkündür. 2007'den günümüze geliştirilen ROS, insansız hava araçlarını oluşturacak tüm sensör tiplerine destek vermektedir.

ROS'un bir uygulama özelliği de, ağ temelli uygulamalara (robot ağları) olanak vermesidir. Uygun bir ağ yapısı kurularak, iki veya daha fazla robot senkron çalıştırılabilmektedir. Ağ üzerinden iletişim sayesinde bir robottan elde edilen veri, diğer robot için sanki kendi sensöründen gelen veriyi andırmaktadır. Hexacopter için geliştirilen arayüz ile hexacopter arasında haberleşme ve veri iletişimi de ROS'un bu özelliğinden faydalanılarak elde edilmiştir. Yapılandırılan bu ağ ile, uygulamada TCP/IP ve fiziksel iletişim katmanında, standart ISM bandı (Endüstriyel, Bilimsel ve Sağlık çalışmalarına yönelik frekans bandı) 2.4 GHz hızında kablosuz veri iletişimi sağlanmıştır.

Seyrüsefer (navigasyon) görevlendirmesi olarak tez çalışması içerisinde hedef noktasının koordinatları arayüzden otopilota gönderilmektedir. Yeni bir hedef noktası komutu arayüzden gönderildiğinde otopilot içerisinde yeni hedef noktası arayüzden son gönderilen hedef noktası olarak kabul edilir. Ardından hava aracı bu noktaya yönlendirilir.

Hedef noktası alındığında hava aracının o anda GPS üzerinden belirlenen koordinatı ile hedef noktası arasındaki açı hesaplanmaktadır. Bu açı (yaw), hava aracının baş yönünün döndürüleceği açıya eşittir. Böylelikle hava aracı yeni baş yönü komutuyla hedef noktasına ilerlemeye başlayacaktır. Otonom bir araç ile hedef takibi yapılabilmesi için öncelikle o aracın koordinatları bilinmeli, sonrasında da hedef noktanın koordinatları ile araç bu noktaya yönlendirilebilmelidir. Bu işlemi gerçekleştirebilmek amacıyla, GPS modülüyle alınan koordinatlar bu işlemde kullanılabilir. Otonom hava aracı hexacopter, 3 eksen, pozisyon (RPY) kontrolünü sağlayan otopilotuna gerekli açı komutlarını vererek sağ-sol, ileri-geri yönde hareket edebilmekte ve kendi eksenini etrafında dönebilmektedir. Hedef noktası takibinde ise aracın baş yönü kontrol edilerek o yöne doğru hareket etmesi sağlanmaktadır.

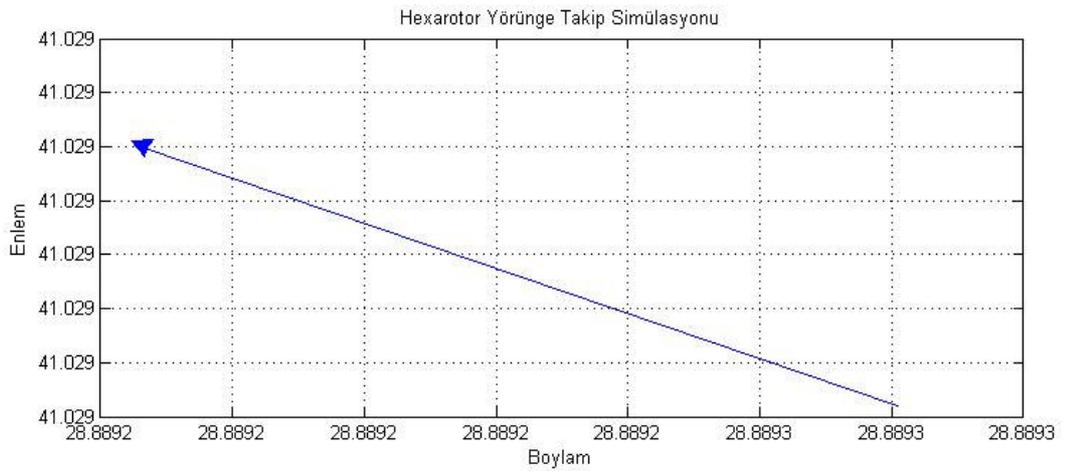
Belirlenen hedef noktası hava aracının hafızasına yüklendikten sonra iki nokta arasındaki açı hesaplanarak baş açısı komutu üretilmektedir. Kafa açısı ile hexacopter o anki baş yönü arasındaki fark, hedef noktası takibinde PID algoritmalarının hata girişini ifade etmektedir. PID algoritması ile bu hata oranı en düşük seviyede tutulmaya çalışılır. Böylelikle, hexacopter hedef noktasına yönlendirilmiş olur. Yapılan çalışmada tek bir hedef noktası takibi yapılmış olsa da otopilot sisteminin sahip olduğu koordinat belirleme ve hedefe yönelme özellikleri geniş bir potansiyel taşımaktadır. Koordinat bilgisinin elde edilmesiyle birlikte otonom hedef takibi, hava aracının dünya üzerindeki yerinin bulunabilmesi, birden fazla hava aracının koordineli bir şekilde uçuş gerçekleştirmesi ve filo kurulması gibi uygulamalara imkân sağlanabilmektedir. Çalışma bu açıdan ilerletilmeye de oldukça müsaittir. Şekil 6.1' de Hexacopter için geliştirilen deneme çalışması ve uçuş arayüzü, Şekil 6.2' de Hexacopter için uygulaması yapılan uzaktan kumanda, Şekil 6.3' te Hexacopter için simülasyon çalışması görülmektedir.



Şekil 6.1 Hexacopter için otonom uçuş arayüz platformu



Şekil 6.2 Hexacopter için kablosuz ağ üzerinden uzaktan kumanda platformu



Şekil 6.3 Hexacopter için deneme seyrüsefer (navigasyon) çalışması

KAYNAKLAR

- [1] Chen, H., Wang, X. M. ve Li, Y.,(2009). "A Survey of Autonomous Control for UAV", International Conference on Artificial Intelligence and Computational Intelligence, AICI '09, 7-8 Nov. 2009, 2:267-271.
- [2] Pachter, M. ve Chandler, P.R., (1998). "Challenges of autonomous control", IEEE Control Systems, 18(4):92-97, DOI: 10.1109/37.710883.
- [3] Ryan, A., Zennaro, M., Howell, A., Sengupta, R. ve Hedrick, J.K., (2004). "An overview of emerging results in cooperative UAV control", 43rd IEEE Conference on Decision and Control, 17 Dec. 2004, 1:602- 607, DOI: 10.1109/CDC.2004.1428700
- [4] Yan, J., Yan, L., Minai, A.A. ve Polycarpou, M.M., (2005). "Balancing search and target response in cooperative unmanned aerial vehicle (UAV) teams", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 36(3):571- 587, DOI: 10.1109/TSMCB.2005.861881
- [5] Sanca, A.S., Alsina, P.J. ve de Jesus F Cerqueira, J., (2010). "Dynamic Modeling with Nonlinear Inputs and Backstepping Control for a Hexarotor Micro-Aerial Vehicle", Robotics Symposium and Intelligent Robotic Meeting (LARS), 23-28 Oct. 2010 Latin American ,36-42, DOI: 10.1109/LARS.2010.14
- [6] Aviastar, A Shallow Saucer like Gondola, Lipowski Helikopteri, Breguet-Richet Helikopteri, De Bothezat Helikopteri,Ömichen Helikopteri, Brennan Helikopteri, Convertawings Model A Helikopteri, Yakovlev VVP-6 Helikopteri, Focke-Wulf Fw.61 Helikopteri, Curtis-Wright VZ-7, www.aviastar.org, 1 Ocak 2013. (Helikopter Tarihçesi)
- [7] Draganfly Innovations, Draganflyer, <http://www.draganfly.com/> 1 Şubat 2013. (İk Quadrotor tasarımlarından biri model helikopter üreticisi)
- [8] Pounds P. et al., (2006). "A practical quad-rotor robot", in Proc. Australasian Conference on Robotics and Automation (ACRA'06), Auckland, New Zealand.
- [9] Hoffmann, G., Rajnarayan, D.G., Waslander, S.L., Dostal, D., Jung S. J. ve Tomlin, C.J., (2004). "The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)", The 23rd Digital Avionics Systems Conference, DASC '04., 2:12:E.4, 121-10 , DOI: 10.1109/DASC.2004.1390847.
- [10] Samir Bouabdallah, (2007). Design and control of quadrotors with application to autonomous flying, These No 3727, EPFL.
- [11] Kemper, M., Control system for unmanned 4-rotor-helicopter, <http://www.google.com/patents/EP1901153A1?cl=en>, 1 Ocak 2013.
- [12] Tournier G. P.,(2006). Six Degree of Freedom Estimation Using Monocular Vision and Moir´e Patterns, Massachusetts Institute of Technology.

- [13] Clifton, C. A., (2005). Hybrid System Based Design for the Coordination and Control of Multiple Autonomous Vehicles, MSc. Thesis, Graduate School of Vanderbilt University.
- [14] ROS, ROS hakkında genel bilgi, www.ros.org, 01 Nisan 2013. (İşletim sistemi platformu web sitesi)
- [15] ST, stm32f4 resmi, www.st.com/stm32f4 ,01 Nisan 2013. (Alt seviye kontrol için MCU' nun üretici web sitesi)
- [16] Sparkfun, IMU 9 DoF Razor resmi, [https:// www. sparkfun.com/ products/ 10736](https://www.sparkfun.com/products/10736), 01 Nisan 2013. (IMU verilerini içeren üretici web sitesi)
- [17] Starlino, İvme ölçer çalışma prensibi resmi, [http://www. starlino. com/ imu_guide.html](http://www.starlino.com/imu_guide.html) 01 Nisan 2013. (IMU verilerini içeren üretici web sitesi)
- [18] Sparkfun, Jiroskop hakkında bilgi, [https://www. sparkfun.com/ pages/ accel_gyro_guide](https://www.sparkfun.com/pages/accel_gyro_guide) 01 Nisan 2013. (Jiroskop genel bilgi içeren üretici web sitesi)
- [19] Electronic Products, Manyetometre bozucu etkileri, [http://www. electronicproducts.com/SensorsandTransducers/Sensorsand Transducers/ agnetometer_basics_for_mobile_phone_applications.aspx](http://www.electronicproducts.com/SensorsandTransducers/SensorsandTransducers/agnetometer_basics_for_mobile_phone_applications.aspx), 01 Nisan 2013. (Manyetometre genel bilgi ve resim içeren üretici web sitesi)
- [20] Digi, Xbee bilgi, <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module>, 1 Nisan 2013. (Xbee hakkında bilgi ve verileri içeren üretici web sitesi)
- [21] Digi, Xbee resim, [http://www.pcbprotos.com/tr/21-kablosuz?Orderby =name &orderway=desc](http://www.pcbprotos.com/tr/21-kablosuz?Orderby=name&orderway=desc), 01 Nisan 2013. (Xbee resim için web sitesi)
- [22] Solarbotics, Ultrasonik sensör resmi, <https://solarbotics.com/product/40320/> 01 Nisan 2013. (Ultrasonik sensörüne ait resim için web sitesi)
- [23] Robotshop, USB çevirici modül resmi, <http://www.robotshop.com/sfe-ft232rl-usb-to-serial-5.html>, 01 Nisan 2013. (USB çevirici resim için web sitesi)
- [24] Rcgroups, kumanda alıcısı, <http://www.rcgroups.com/forums/showthread.php?t=1660632> 01 Nisan 2013. (Kumanda alıcısı resim için üretici web sitesi)
- [25] Futaba, Altı kanallı kumanda, [http://www.futaba-rc.com/ systems/ futk6900.html](http://www.futaba-rc.com/systems/futk6900.html), 01 Nisan 2013. (Kumanda hakkında bilgi ve resim içeren üretici web sitesi)
- [26] Mikrokofteri, Motor hız kontrolcüsü resmi, [http://www.mikrokofter.De /ucwiki/BL-Ctrl](http://www.mikrokofter.De/ucwiki/BL-Ctrl) 01 Nisan 2013. (ESC resim ve bilgi içeren üretici web sitesi)
- [27] RTOS, İşletim sistemi hakkında bilgi, <http://www.freertos.org/> 01 Nisan 2013. (İşletim sistemi hakkında bilgi içeren organizasyon sitesi)
- [28] Oludayo John Oguntoyinbo, O. J., (2009). Pid Control of Brushless DC Motor and Robot Trajectory Planning and Simulation with MATLAB SIMULINK Technology and Communication, Vaasan Ammattikorkeakoulu Vasa Yerkeshögskola, University of Applied Sciences.
- [29] Cowling, I. D., Yakimenko, O. A., Whidborne, J. F. ve Cooke, A. K., (2007). "A Prototype of an Autonomous Controller for a Quadrotor UAV," European Control Conference, 2-5 July 2007, Kos, Greece.
- [30] Morari, M. ve Lee, J. H., (1999). "Model predictive control: past, present and future", Computers and Chemical Engineering, 23:667–682.
- [31] Henriksson, D., Cervin, A., Åkesson, J. ve Årzén, K-E., (2002). "Feedback Scheduling of Model Predictive Controllers", 8th IEEE RealTime and

- Embedded Technology and Applications Symposium, San Jose, CA, September 2002.
- [32] Lee, S.M., Jeong, S.C., Ji, D.H. ve Won, S.C., (2011). "Robust model predictive control for LPV systems with delayed state using relaxation matrices," *American Control Conference (ACC)*, , June 29 -July 1 2011, 716-721.
- [33] Caruntu, C. F. ve Lazar, C., (2012). "Predictive compensation for network-induced time-varying delays," 16th International Conference on System Theory, Control and Computing (ICSTCC), 12-14 Oct. 2012, 1-6.
- [34] Valencia, F., Lopez, J.D., Marquez, A. ve Espinosa, J.J., (2011). "Moving horizon estimator for measurement delay compensation in model predictive control schemes," 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 12-15 Dec. 2011, 6678-6683, DOI: 10.1109/CDC.2011.6161346
- [35] Jianxiong, L., Yiming, F., Shengli, S., (2010). "Robust MPC algorithm for discrete-time systems with time-varying delay and nonlinear perturbations," 29th Chinese Control Conference (CCC), 29-31 July 2010, 3128-3133.
- [36] Pin, G., Filippo, M. ve Parisini, T., (2009). "Networked MPC for constrained linear systems: a recursive feasibility approach," Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009, 555-560, DOI: 10.1109/CDC.2009.5400031.
- [37] Henriksson, D., Anton Cervin, C., Åkesson, J. ve Årzén, K.-E., (2002). "On Dynamic Real-time Scheduling of Model Predictive Controllers". 41st IEEE Conference on Decision and Control, December 2002, Las Vegas, Nevada USA.
- [38] Martin Sanfridson, M., (2004). Quality of control and real-time scheduling allowing for time-variations in computer control systems, Doctoral thesis Department of Machine Design, Royal Institute of Technology Stockholm, Sweden. TRITA–MMK 2004:7 ISSN 1400-1179 ISRN/KTH/MMK--04/7—SE.
- [39] Maciejowski, J. M., (2002). Predictive control: with constraints, Pearson Education.
- [40] Izák, M., Görge, D. ve Liu, S., (2007). "On stability and Control of Systems with time varying sampling period and time delay", Proceedings of the 7th IFAC Sym-posium on Nonlinear Control Systems, 1056-1061.
- [41] Vanderbei, R. J., (2008). Linear Programming: Foundations and Extensions, International Series in Operations Research & Management Science, Springer.
- [42] Ye, Y., (1987). Further development of the interior algorithm for convex quadratic programming", Stanford University and Integrated Systems Inc., Stanford, CA.
- [43] Roos, C. et all, (2005). Interior Point Methods for Linear Programming, Second Edition, Springer.
- [44] Absil, P. A. ve Tits, A. L., (2006). "Newton-KKT interior-point methods for indefinite quadratic programming", Computational Optimization and Applications, Juli 2006, Netherlands, Springer.
- [45] Åström, K.-J. ve Wittenmark, B., (1990). Computer-Controlled Systems: Theory and Design. 2nd ed., Prentice-Hall. Englewood Cliffs, NJ.
- [46] MATLAB version 7.10.0. Natick, Massachusetts: The MathWorks Inc., 2010

- [47] Lofberg, J., (2004). "YALMIP : a toolbox for modeling and optimization in MATLAB," 2004 IEEE International Symposium on Computer Aided Control Systems Design, 4-4 Sept. 2004, 284-289, DOI: 10.1109/CACSD.2004.1393890.
- [48] Sigpromu, QPC hakkında bilgi, QPC <http://sigpromu.org/quadprog/index.html>, 1 Nisan 2009. (QPC quadratic programlama çözücüsüne ait web sitesi)
- [49] Openslam, SLAM hakkında bilgi, www.openslam.org, 01 Nisan 2013. (SLAM organizasyonu web sitesi)
- [50] Thrun, S., (2002). "Robotic mapping: A survey", in Exploring Artificial Intelligence in the New Millenium, Editors G. Lakemeyer and B. Nebel, Morgan Kaufmann, San Francisco, CA, USA.
- [51] Huang, S., Wang, Z., Dissanayake, G. ve Frese, U, (2008). "Iterated SLSJF: A sparse local submap joining algorithm with improved consistency", 2008 Australiasan Conference on Robotics and Automation, Canberra, December 2008.
- [52] Huang, S., Wang, Z. ve Dissanayake G., (2008). "Sparse local submap joining filter for building large-scale maps", IEEE Transactions on Robotics.
- [53] Arras K.O., "The CAS Robot Navigation Toolbox: Users Guide and Reference", Center for Autonomous Systems, KTH, September, 2004
- [54] Guivant, J. ve Nebot, E., (2001). "Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation", IEEE Transactions on Robotics and Automation.
- [55] Eliazar, A. ve Parr, R., (2003). "DP-SLAM: Fast, Robust Simultainous Localization and Mapping Without Predetermined Landmarks", IJCAI.
- [56] Eliazar, A. ve Parr, R., (2004). "DP-SLAM 2.0", IEEE 2004 International Conference on Robotics and Automation (ICRA 2004), 1314-1320.
- [57] Grisetti, G., Stachniss, C. ve Burgard, W., (2006). "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters", IEEE Transactions on Robotics.
- [58] Grisetti, G., Stachniss, C. ve Burgard, W., (2005). "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling", In Proc. of the IEEE International Conference on Robotics and Automation (ICRA).
- [59] Grisetti, G., Kuemmerle, R., Stachniss, C., Frese, U. ve Hertzberg, C., (2010). "Hierarchical Optimization on Manifolds for Online 2D and 3D Mapping", IEEE International Conference on Robotics and Automation (ICRA).
- [60] Olson, E. ve Agarwal, P., (2012). "Inference on networks of mixtures for robust robot mapping", Proceedings of Robotics: Science and Systems.
- [61] Wagner, R., Birbach, O. ve Frese, U., (2011). "Rapid Development of Manifold-Based Graph Optimization Systems for Multi-Sensor Calibration and SLAM", Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011),
- [62] Hertzberg, C., (2008). A Framework for Sparse, Non-Linear Least Squares Problems on Manifolds, Diploma Thesis, University of Bremen.
- [63] Blanco, J. L., Madrigal, J. A. F. ve Jimenez, J. G., (2008). "Efficient Probabilistic Range-Only SLAM", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008).

- [64] Paskin, M. A., (2003). "Thin Junction Tree Filters for Simultaneous Localization and Mapping", In the Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2003).
- [65] Thin Junction Tree Filters for Simultaneous Localization and Mapping, Technical Report, University of California, Berkeley, 2002
- [66] Frese, U. ve Schroeder, L., (2006). "Closing a Million-Landmarks Loop", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, 2006.
- [67] Frese, U., (2007). "Efficient 6-DOF SLAM with Treemap as a Generic Backend", Proceedings of the International Conference on Robotics and Automation, Rome, 2007.
- [68] Doucet., A., (1998). On sequential simulation-based methods for bayesian filtering, Technical report, Signal Processing Group, Dept. of Engineering, University of Cambridge.
- [69] Doucet, A., de Freitas, N. ve Gordan, N., (2001). Sequential Monte-Carlo Methods in Practice, Springer Verlag.
- [70] Montemerlo, M., Thrun, S., Koller, D. ve Wegbreit, B., (2003). "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges", In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), Acapulco, Mexico, 2003, 1151–1156.

PMBLDC Motor Parametreleri

Motor parametreleri ve deneyle elde edilen çıktıları aşağıdaki gibidir.

A-1 Motor Parametreleri

BRUSHLESS PERMANENT MAGNET DC MOTOR DESIGN

File: Setup1.res

GENERAL DATA

Rated Output Power (kW): 0.03
Rated Voltage (V): 8.4
Number of Poles: 16
Given Rated Speed (rpm): 10000
Frictional Loss (W): 0
Windage Loss (W): 0
Rotor Position: Outer
Type of Load: Constant Power
Type of Circuit: Y3
Lead Angle of Trigger in Elec. Degrees: 0
Trigger Pulse Width in Elec. Degrees: 120
One-Transistor Voltage Drop (V): 0
One-Diode Voltage Drop (V): 0
Operating Temperature (C): 75
Maximum Current for CCC (A): 3
Minimum Current for CCC (A): 0.1

STATOR DATA

Number of Stator Slots: 12

Outer Diameter of Stator (mm): 22.5

Inner Diameter of Stator (mm): 10

Type of Stator Slot: 3

Stator Slot

hs0 (mm): 0.3

hs1 (mm): 0.495974

hs2 (mm): 2.53603

bs0 (mm): 1

bs1 (mm): 2.7181

bs2 (mm): 1.35905

rs (mm): 0.3

Top Tooth Width (mm): 2.78016

Bottom Tooth Width (mm): 2.78016

Skew Width (Number of Slots) 0

Length of Stator Core (mm): 14

Stacking Factor of Stator Core: 0.95

Type of Steel: steel_1010

Slot Insulation Thickness (mm): 0

Layer Insulation Thickness (mm): 0

End Length Adjustment (mm): 0

Number of Parallel Branches: 1

Number of Conductors per Slot: 10 Type of Coils: 22

Average Coil Pitch: 1

Number of Wires per Conductor: 4

Wire Diameter (mm): 0.34

Wire Wrap Thickness (mm): 0

Slot Area (mm²): 6.76102

Net Slot Area (mm²): 6.34605

Limited Slot Fill Factor (%): 75

Stator Slot Fill Factor (%): 72.8643

Coil Half-Turn Length (mm): 18.0664

ROTOR DATA

Minimum Air Gap (mm): 0.25

Outer Diameter (mm): 27

Length of Rotor (mm): 14

Stacking Factor of Iron Core: 0.95

Type of Steel: steel_1010

Polar Arc Radius (mm): 11.5

Mechanical Pole Embrace: 0.7

Electrical Pole Embrace: 0.68755

Max. Thickness of Magnet (mm): 1

Width of Magnet (mm): 3.15128

Type of Magnet: NdFe35

Type of Rotor: 1

Magnetic Shaft: Yes

PERMANENT MAGNET DATA

Residual Flux Density (Tesla): 1.23

Coercive Force (kA/m): 890

Maximum Energy Density (kJ/m³): 273.675

Relative Recoil Permeability: 1.09981

Demagnetized Flux Density (Tesla): 0.932785

Recoil Residual Flux Density (Tesla): 1.23

Recoil Coercive Force (kA/m): 890

MATERIAL CONSUMPTION

Armature Copper Density (kg/m³): 8900

Permanent Magnet Density (kg/m³): 7400

Armature Core Steel Density (kg/m³): 7872

Rotor Core Steel Density (kg/m³): 7872

Armature Copper Weight (kg): 0.00700731

Permanent Magnet Weight (kg): 0.00546788

Armature Core Steel Weight (kg): 0.0249113

Rotor Core Steel Weight (kg): 0.00855185

Total Net Weight (kg): 0.0459383
Armature Core Steel Consumption (kg): 0.0513933
Rotor Core Steel Consumption (kg): 0.0428345 STEADY STATE PARAMETERS
Stator Winding Factor: 0.866025
D-Axis Reactive Inductance L_{ad} (H): 1.71633e-006
Q-Axis Reactive Inductance L_{aq} (H): 1.71633e-006
D-Axis Inductance L_1+L_{ad} (H): 1.2984e-005
Q-Axis Inductance L_1+L_{aq} (H): 1.2984e-005
Armature Leakage Inductance L_1 (H): 1.12677e-005
Zero-Sequence Inductance L_0 (H): 6.28484e-006
Armature Phase Resistance R_1 (ohm): 0.0431802
Armature Phase Resistance at 20C (ohm): 0.0355192
D-Axis Time Constant (s): 3.97482e-005
Q-Axis Time Constant (s): 3.97482e-005
Ideal Back-EMF Constant K_E (Vs/rad): 0.00927193
Start Torque Constant K_T (Nm/A): 0.00927186
Rated Torque Constant K_T (Nm/A): 0.0096509

NO-LOAD MAGNETIC DATA

Stator-Teeth Flux Density (Tesla): 1.02322
Stator-Yoke Flux Density (Tesla): 0.201323
Rotor-Yoke Flux Density (Tesla): 1.55381
Air-Gap Flux Density (Tesla): 0.938417
Magnet Flux Density (Tesla): 0.936839
Stator-Teeth By-Pass Factor: 0.000978612
Stator-Yoke By-Pass Factor: 7.86372e-005
Rotor-Yoke By-Pass Factor: 0.000168726
Stator-Teeth Ampere Turns (A.T): 3.0261
Stator-Yoke Ampere Turns (A.T): 0.13999
Rotor-Yoke Ampere Turns (A.T): 4.64661
Air-Gap Ampere Turns (A.T): 204.384
Magnet Ampere Turns (A.T): -212.125
Armature Reactive Ampere Turns at Start Operation (A.T): 3.93692

Leakage-Flux Factor: 1
Correction Factor for Magnetic
Circuit Length of Stator Yoke: 0.771837
Correction Factor for Magnetic
Circuit Length of Rotor Yoke: 0.499531
No-Load Speed (rpm): 8724.71
Cogging Torque (N.m): 0.0119773

FULL-LOAD DATA

The following results are of the maximum output.

Average Input Current (A): 1.53648
Root-Mean-Square Armature Current (A): 1.36662
Armature Thermal Load (A^2/mm^3): 8.73051
Specific Electric Loading (A/mm): 2.32005
Armature Current Density (A/mm^2): 3.76306
Frictional and Windage Loss (W): 0
Iron-Core Loss (W): 9.47086e-005
Armature Copper Loss (W): 0.241938
Transistor Loss (W): 0
Diode Loss (W): 0
Total Loss (W): 0.242033
Output Power (W): 12.6644
Input Power (W): 12.9064
Efficiency (%): 98.1247
Rated Speed (rpm): 8155.7
Rated Torque (N.m): 0.0148284
Locked-Rotor Torque (N.m): 0.0108123
Locked-Rotor Current (A): 0.010385

WINDING ARRANGEMENT

The 3-phase, 2-layer winding can be arranged in 3 slots as below: ACB
Angle per slot (elec. degrees): 240
Phase-A axis (elec. degrees): 120
First slot center (elec. degrees): 0

TRANSIENT FEA INPUT DATA

For Armature Winding:

Number of Turns: 20

Parallel Branches: 1

Terminal Resistance (ohm): 0.0431802

End Leakage Inductance (H): -2.90047e-008

2D Equivalent Value:

Equivalent Model Depth (mm): 14

Equivalent Stator Stacking Factor: 0.95

Equivalent Rotor Stacking Factor: 0.95

Equivalent Br (Tesla): 1.23

Equivalent Hc (kA/m): 890

Estimated Rotor Moment of Inertia (kg m²): 5.5671e-006

Hexacopter Hareket Denklemleri

B-1 Çoklu Rotorlu Helikopter Modelinin Çıkarılması

İzlenen çıkarım yöntemi sıralaması şu şekildedir [10]. Hareket matrisinin yazılması. Açısal hızların tanımlanması ve Euler-Lagrange Yöntemi ile Enerji denklemleri yardımı ile modelin çıkarımı. Buna göre model çıkarımı sırası ile:

Hareket Matrisi

$R(x, \phi)$, x-ekseni etrafında hareket

$R(y, \theta)$, y-ekseni etrafında hareket

$R(z, \psi)$, z-ekseni etrafında hareket olarak tanımlandığında her bir eksene ait hareket matrisleri (1), (2), (3);

$$R(x, \phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \quad (1)$$

$$R(y, \theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (2)$$

$$R(z, \psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Üç eksenin bir arada gösterildiği hareket matrisi aşağıdaki gibidir (4), (5).

$$R(\phi, \theta, \psi) = R(x, \phi)R(y, \theta)R(z, \psi) \quad (4)$$

$$R = \begin{pmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi \\ \sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \quad (5)$$

Açısal Hızlar

Tait-Bryan açılarının (ϕ, θ, ψ) zamanla değişimi süreksiz bir fonksiyondur. Bununla birlikte fiziksel olarak gyroscope larla ölçülen gövde açısal hızlardanda farklıdır. Bu durumda gövde açısal hızlarından, Tait-Bryan açısal hızlarına aşağıdaki dönüşüm matrisi ile geçilebilir (6).

$$R_r = \begin{pmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{pmatrix} \quad (6)$$

Matematiksel Modelleme

Çoklu rotorlu helikopter de Euler-Langrange yöntemi kullanılarak yapılan matematiksel modellemede aşağıdaki kabuller kullanılmıştır.

Yapı sabit ve simetriktir.

Gövde ağırlık merkezi ve orijini çakışıktır.

Pervaneler sabittir.

Langrange formülü ile $L = T - V$ (7);

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) \quad (7)$$

q_i genelleştirilmiş koordinatlar, Γ_i genelleştirilmiş kuvvetler, T kinetik enerji, V potansiyel enerji yi ifade eder.

Yine yer koordinatı ile $[\vec{X}, \vec{Y}, \vec{Z}]$, sistemin dönel koordinat eksenini $[\vec{x}, \vec{y}, \vec{z}]$ birbirine diktir. Bu durumda gövdenin herhangi bir noktası (8):

$$r_{x,y,z}(x, y, z) = R(\phi, \theta, \psi) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (8)$$

Eşdeğeri (9):

$$\begin{aligned} r_x(x, y, z) &= (\cos \psi \cos \theta)x + (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi)y + (\cos \psi \sin \theta \cos \phi + \sin \psi \cos \phi)z \\ r_y(x, y, z) &= (\sin \psi \cos \theta)x + (\sin \psi \sin \theta \sin \phi - \cos \psi \cos \phi)y + (\sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi)z \\ r_z(x, y, z) &= (-\sin \theta)x + (\cos \theta \sin \phi)y + (\cos \theta \cos \phi)z \end{aligned} \quad (9)$$

Zamana göre türevi ile ilgili çizgisel hızlar (10):

$$\begin{aligned} v_x(x, y, z) &= (-\sin \theta \cos \psi \dot{\theta} - \cos \theta \sin \psi \dot{\psi})x \\ &+ (-\cos \psi \cos \phi \dot{\psi} + \sin \psi \sin \phi \dot{\phi} - \sin \psi \sin \phi \sin \theta \dot{\psi} + \cos \psi \cos \phi \sin \theta \dot{\phi} + \cos \psi \sin \phi \cos \theta \dot{\theta})y \\ &+ (\cos \psi \sin \phi \dot{\psi} + \sin \psi \cos \phi \dot{\phi} - \sin \psi \cos \phi \sin \theta \dot{\psi} - \cos \psi \sin \phi \sin \theta \dot{\phi} + \cos \psi \cos \phi \cos \theta \dot{\theta})z \\ v_y(x, y, z) &= (-\sin \theta \sin \psi \dot{\theta} + \cos \theta \cos \psi \dot{\psi})x \\ &+ (-\sin \psi \cos \phi \dot{\psi} - \cos \psi \sin \phi \dot{\phi} + \cos \psi \sin \phi \sin \theta \dot{\psi} + \sin \psi \cos \phi \sin \theta \dot{\phi} + \sin \psi \sin \phi \cos \theta \dot{\theta})y \\ &+ (\sin \psi \sin \phi \dot{\psi} - \cos \psi \cos \phi \dot{\phi} + \cos \psi \cos \phi \sin \theta \dot{\psi} - \sin \psi \sin \phi \sin \theta \dot{\phi} + \sin \psi \cos \phi \cos \theta \dot{\theta})z \\ v_z(x, y, z) &= (-\cos \theta \dot{\theta})x \\ &+ (\cos \phi \cos \theta \dot{\phi} - \sin \phi \sin \theta \dot{\theta})y \\ &+ (-\sin \phi \cos \theta \dot{\phi} - \cos \phi \sin \theta \dot{\theta})z \end{aligned} \quad (10)$$

Eksenlere ait çizgisel hız ifadeleri yeniden yazıldığında (11), (12):

$$\begin{aligned}
 v_X(x, y, z) &= v_{Xx}x + v_{Xy}y + v_{Xz}z = (v_{Xx} \quad v_{Xy} \quad v_{Xz}) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \\
 v_Y(x, y, z) &= v_{Yx}x + v_{Yy}y + v_{Yz}z = (v_{Yx} \quad v_{Yy} \quad v_{Yz}) \begin{pmatrix} x \\ y \\ z \end{pmatrix} \\
 v_Z(x, y, z) &= v_{Zx}x + v_{Zy}y + v_{Zz}z = (v_{Zx} \quad v_{Zy} \quad v_{Zz}) \begin{pmatrix} x \\ y \\ z \end{pmatrix}
 \end{aligned} \tag{11}$$

Bu durumda gövde üzerinde herhangi bir noktadaki hızın karesi ifadesi:

$$\begin{aligned}
 v^2(x, y, z) &= v_X(x, y, z)^2 + v_Y(x, y, z)^2 + v_Z(x, y, z)^2 \\
 v^2(x, y, z) &= (v_{Xx} \quad v_{Xy} \quad v_{Xz}) \Lambda \begin{pmatrix} v_{Xx} \\ v_{Xy} \\ v_{Xz} \end{pmatrix} + (v_{Yx} \quad v_{Yy} \quad v_{Yz}) \Lambda \begin{pmatrix} v_{Yx} \\ v_{Yy} \\ v_{Yz} \end{pmatrix} \\
 &+ (v_{Zx} \quad v_{Zy} \quad v_{Zz}) \Lambda \begin{pmatrix} v_{Zx} \\ v_{Zy} \\ v_{Zz} \end{pmatrix}
 \end{aligned} \tag{12}$$

ile

$$\Lambda = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}$$

Bu ifade geliştirildiğinde (13), (14):

$$\begin{aligned}
v^2(x, y, z) &= x^2 \cdot (\cos^2 \theta \dot{\psi}^2 + \dot{\theta}^2) \\
&+ y^2 \cdot (\dot{\psi}^2 (\cos^2 \phi + \sin^2 \phi \sin^2 \theta) + \dot{\psi}(-2 \sin \phi \cos \phi \cos \theta \dot{\theta} - 2 \dot{\phi} \sin \theta) + \sin^2 \phi \dot{\theta}^2 + \dot{\phi}^2) \\
&+ z^2 \cdot (\dot{\psi}^2 (\sin^2 \phi + \cos^2 \phi \sin^2 \theta) + \dot{\psi}(2 \sin \phi \cos \phi \cos \theta \dot{\theta} - 2 \dot{\phi} \sin \theta) + \cos^2 \phi \dot{\theta}^2 + \dot{\phi}^2) \\
&+ 2xy \cdot (\dot{\psi}^2 \sin \phi \sin \theta \cos \theta + \dot{\psi}(\cos \phi \sin \theta \dot{\theta} - \sin \phi \cos \theta \dot{\phi}) - \cos \phi \dot{\phi} \dot{\theta}) \\
&+ 2xz \cdot (\dot{\psi}^2 \cos \phi \sin \theta \cos \theta + \dot{\psi}(-\cos \phi \cos \theta \dot{\phi} - \sin \phi \sin \theta \dot{\theta}) - \sin \phi \dot{\phi} \dot{\theta}) \\
&+ 2yz \cdot (-\dot{\psi}^2 \sin \phi \cos \phi \cos^2 \theta + \dot{\psi}(\sin^2 \phi \cos \theta \dot{\theta} - \cos^2 \phi \cos \theta \dot{\theta}) + \sin \phi \cos \phi \dot{\theta}^2)
\end{aligned} \tag{13}$$

$$\begin{aligned}
v^2(x, y, z) &= (y^2 + z^2)(\dot{\psi}^2 \sin^2 \theta - 2 \sin \theta \dot{\phi} \dot{\psi} + \dot{\phi}^2) \\
&+ (x^2 + z^2)(\dot{\psi}^2 \sin^2 \phi \cos^2 \theta + 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \cos^2 \phi \dot{\theta}^2) \\
&+ (x^2 + y^2)(\dot{\psi}^2 \cos^2 \phi \cos^2 \theta - 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \sin^2 \phi \dot{\theta}^2) \\
&+ 2xy \cdot (\dot{\psi}^2 \sin \phi \sin \theta \cos \theta + \dot{\psi}(\cos \phi \sin \theta \dot{\theta} - \sin \phi \cos \theta \dot{\phi}) - \cos \phi \dot{\phi} \dot{\theta}) \\
&+ 2xz \cdot (\dot{\psi}^2 \cos \phi \sin \theta \cos \theta + \dot{\psi}(-\cos \phi \cos \theta \dot{\phi} - \sin \phi \sin \theta \dot{\theta}) - \sin \phi \dot{\phi} \dot{\theta}) \\
&+ 2yz \cdot (-\dot{\psi}^2 \sin \phi \cos \phi \cos^2 \theta + \dot{\psi}(\sin^2 \phi \cos \theta \dot{\theta} - \cos^2 \phi \cos \theta \dot{\theta}) + \sin \phi \cos \phi \dot{\theta}^2)
\end{aligned} \tag{14}$$

Hız ifadesinin karesi kullanılarak kinetik enerji ifadesine geçilir.

Kinetik enerji ifadesi (15):

$$\begin{aligned}
T &= \frac{1}{2} \int y^2 + z^2 (R) dm(r) \cdot (\dot{\psi}^2 \sin^2 \theta - 2 \sin \theta \dot{\phi} \dot{\psi} + \dot{\phi}^2) \\
&+ \frac{1}{2} \int (x^2 + z^2) (R) dm(r) \cdot (\dot{\psi}^2 \sin^2 \phi \cos^2 \theta + 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \cos^2 \phi \dot{\theta}^2) \\
&+ \frac{1}{2} \int (x^2 + y^2) (R) dm(r) \cdot (\dot{\psi}^2 \cos^2 \phi \cos^2 \theta - 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \sin^2 \phi \dot{\theta}^2) \\
&+ \int xy (R) dm(r) \cdot (\dot{\psi}^2 \sin \phi \sin \theta \cos \theta + \dot{\psi}(\cos \phi \sin \theta \dot{\theta} - \sin \phi \cos \theta \dot{\phi}) - \cos \phi \dot{\phi} \dot{\theta}) \\
&+ \int xz (R) dm(r) \cdot (\dot{\psi}^2 \cos \phi \sin \theta \cos \theta + \dot{\psi}(-\cos \phi \cos \theta \dot{\phi} - \sin \phi \sin \theta \dot{\theta}) - \sin \phi \dot{\phi} \dot{\theta}) \\
&+ \int yz (R) dm(r) \cdot (-\dot{\psi}^2 \sin \phi \cos \phi \cos^2 \theta + \dot{\psi}(\sin^2 \phi \cos \theta \dot{\theta} - \cos^2 \phi \cos \theta \dot{\theta}) + \sin \phi \cos \phi \dot{\theta}^2)
\end{aligned} \tag{15}$$

ile gösterilir. Burada köşegen üzerinde yer almayan terimler diğerlerine göre bin mertebesinde daha küçük olduğundan ihmal edilmiştir. Bu durumda yeni ifade (16);

$$\begin{aligned}
 T &= \frac{1}{2} I_{xx} (\dot{\phi} - \dot{\psi} \sin \theta)^2 \\
 &+ \frac{1}{2} I_{yy} (\dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta)^2 \\
 &+ \frac{1}{2} I_{zz} (\dot{\theta} \sin \phi + \dot{\psi} \cos \phi \cos \theta)^2
 \end{aligned} \tag{16}$$

Yine aynı şekilde sistemin potansiyel enerjisi (17), (18):

$$V = g \int (-\sin \theta \cdot x + \sin \phi \cos \theta \cdot y + \cos \phi \cos \theta \cdot z) dm(r) \tag{17}$$

$$V = \int x dm(x) \cdot (-g \sin \theta) + \int y dm(y) \cdot (g \sin \phi \cos \theta) + \int z dm(z) \cdot (g \cos \phi \cos \theta) \tag{18}$$

Langrangian ifadesi her bir eksen için uygulandığında $L = T - V$ (19):

$$\begin{aligned}
 \tau_{\phi} &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \left(\frac{\partial L}{\partial \phi} \right) \\
 \tau_{\theta} &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \left(\frac{\partial L}{\partial \theta} \right) \\
 \tau_{\psi} &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \left(\frac{\partial L}{\partial \psi} \right)
 \end{aligned} \tag{19}$$

Geliştirilen ifadenin her bir harekete (roll-pitch-yaw) yorumlanması ile (20), (21):

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \left(\frac{\partial L}{\partial \phi} \right) &= \ddot{\phi} \cdot I_{xx} - \dot{\psi} \cdot \sin \theta \cdot I_{xx} - \dot{\psi} \dot{\theta} \cdot \cos \theta (I_{xx} + (I_{yy} - I_{zz})(2 \cos^2 \phi - 1)) \\ &+ \dot{\theta}^2 \cdot \frac{1}{2} \sin 2\phi (I_{yy} - I_{zz}) - \dot{\psi}^2 \cdot \frac{1}{2} \sin 2\phi \cos^2 \theta (I_{yy} - I_{zz}) \\ &+ \int y dm(y) \cdot (-g \cos \phi \cos \theta) + \int z dm(z) \cdot (g \sin \phi \cos \theta) \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \left(\frac{\partial L}{\partial \theta} \right) &= (\ddot{\theta} \cdot I_{yy} \cos^2 \phi - I_{zz} \sin^2 \phi) - (\dot{\psi} \cdot \frac{1}{2} \sin 2\phi \cos \theta \cdot (I_{yy} - I_{zz})) \\ &+ (\dot{\psi}^2 \cdot \frac{1}{2} \sin 2\phi (-I_{xx} + I_{yy} \sin^2 \phi + I_{zz} \cos^2 \phi)) \\ &+ (\dot{\theta} \dot{\phi} \cdot \sin 2\phi (I_{zz} - I_{yy})) + (\dot{\psi} \dot{\phi} \cdot \cos \phi (\cos 2\phi \cdot (I_{yy} - I_{zz}) + I_{xx})) \\ &+ \int x dm(x) \cdot (-g \cos \theta) - \int y dm(y) \cdot (-g \sin \phi \sin \theta) - \int z dm(z) \cdot (g \cos \phi \sin \theta) \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) &= (\dot{\psi} \cdot (\cos^2 \phi (I_{zz} \cos^2 \phi + I_{yy} \sin^2 \phi) + \sin^2 \theta I_{xx} - (\ddot{\phi} \cdot \sin \theta I_{xx})) \\ &+ (\ddot{\theta} \cdot \frac{1}{2} \sin 2\phi \cos \theta (I_{yy} - I_{zz})) + (\dot{\theta} \dot{\phi} \cdot \sin 2\phi (I_{xx} - I_{zz} \cos^2 \phi + I_{yy} \sin^2 \phi)) \\ &- (\dot{\psi} \dot{\phi} \cdot \sin 2\phi \cos^2 \phi \cdot (I_{yy} - I_{zz})) + (\dot{\theta} \dot{\phi} \cdot \cos \theta (I_{xx} + (2 \cos^2 \phi - 1)(I_{yy} - I_{zz}))) \\ &- \dot{\theta}^2 \cdot \frac{1}{2} \sin 2\phi \sin \theta (I_{yy} - I_{zz}) \end{aligned}$$

(20)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \left(\frac{\partial L}{\partial \phi} \right) = I_{xx} \dot{\omega}_x - (I_{yy} - I_{zz}) \omega_y \omega_z + \int y dm(y) \cdot (-g \cos \phi \cos \theta) \\ + \int z dm(z) \cdot (g \sin \phi \cos \theta)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \left(\frac{\partial L}{\partial \theta} \right) = -\sin \phi (\dot{\omega}_z I_{zz} - \omega_x \omega_y (I_{xx} - I_{yy})) \\ + \cos \phi (\dot{\omega}_y \cdot I_{yy} - \omega_x \omega_z (I_{zz} - I_{xx})) \\ + \int x dm(x) \cdot (-g \cos \theta) \\ - \int y dm(y) \cdot (-g \sin \phi \sin \theta) - \int z dm(z) \cdot (g \cos \phi \sin \theta)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) = -\sin \theta \cdot (\dot{\omega}_x I_{xx} - \omega_y \omega_z (I_{yy} - I_{zz})) \\ + \sin \phi \cos \theta \cdot (\dot{\omega}_y I_{yy} - \omega_x \omega_z (I_{zz} - I_{xx})) \\ + \cos \phi \cos \theta \cdot (\dot{\omega}_z I_{zz} - \omega_x \omega_y (I_{xx} - I_{yy})) \quad (21)$$

Euler açlarına ait bu ifadeler hız ve ivme bilgileri, gövdenin dönüşüm matrisi kullanılarak anlık hız ve ivme ile basitleştirilerek ifade edilebilir (23). Sırası ile sistemin dönüşüm matrisi (22):

$$R_r = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \quad (22)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} = I_{xx} \dot{\omega}_x - (I_{yy} - I_{zz}) \omega_y \omega_z$$

$$+ \int y dm(y).(-g \cos\phi \cos\theta)$$

$$+ \int z dm(z).(+g \sin\phi \cos\theta)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = -\sin\phi(\dot{\omega}_z I_{zz} - \omega_x \omega_y (I_{xx} - I_{yy}))$$

$$+ \cos\phi(\dot{\omega}_y I_{yy} - \omega_x \omega_z (I_{zz} - I_{xx}))$$

$$+ \int x dm(x).(-g \cos\theta)$$

$$- \int y dm(y).(-g \sin\phi \cos\theta) - \int z dm(z).(+g \cos\phi \sin\theta)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) = -\sin\theta(\dot{\omega}_x I_{xx} - \omega_y \omega_z (I_{yy} - I_{zz}))$$

$$+ \sin\phi \cos\theta(\dot{\omega}_y I_{yy} - \omega_x \omega_z (I_{zz} - I_{xx}))$$

$$+ \cos\phi \sin\theta(\dot{\omega}_z I_{zz} - \omega_x \omega_y (I_{xx} - I_{yy}))$$

(23)

Sisteme dengesizlik vektörleri de eklendiğinde sistemin son durumdaki modeli (24):

J_r	rotor ataleti
Ω	itici açısal hızı
I_{xx}, I_{yy}, I_{zz}	atalet momentleri
b	itme faktörü (thrust)
l	itici merkezinin kütle merkezine yatay uzaklığı
ϕ	yuvarlanma açısı
θ	yunuslama açısı
ψ	sapma açısı
d	çekiş faktörü (drag)

$$\begin{aligned}\ddot{\phi} &= \frac{J_r \dot{\theta} (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 + \Omega_5 - \Omega_6)}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\psi} \dot{\theta} + \frac{bl (\Omega_1^2 - \Omega_2^2)}{I_{xx}} \\ \ddot{\theta} &= \frac{J_r \dot{\phi} (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 + \Omega_5 - \Omega_6)}{I_{xx}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi} + \frac{bl (\Omega_1^2 - \Omega_5^2)}{I_{yy}} \\ \ddot{\psi} &= -\frac{d (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2 + \Omega_5^2 - \Omega_6^2)}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\theta} \dot{\phi}\end{aligned}$$

(24)

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Tarık Veli MUMCU
Doğum Tarihi ve Yeri : 09/04/1980 Altındağ
Yabancı Dili : İngilizce, Almanca, Fransızca
E-posta : tmumcu@yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Elektrik Mühendisliği	YTÜ	2005
Lisans	Elektrik Mühendisliği	YTÜ	2002
Lise	Elektronik Bölümü	Haydarpaşa A.T.L.	1998

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2002	YTÜ	Araştırma Görevlisi

YAYINLARI

Makale

1. Sub optimal solutions for time varying time delayed MPC controllers
Tarık Veli Mumcu, Kayhan Gulez ELEKTRONIKA IR
ELEKTROTECHNIKA, ISSN 1392-1215, VOL. XX, NO. X, 20XX
<http://dx.doi.org/10.5755/j01.eee>

Proje

1. Doç. Dr. Kayhan GÜLEZ, Arş. Gör. İbrahim ALIŞKAN, Doç. Dr. Ali Ahmed ADAM, Arş. Gör. Tarık Veli MUMCU 2010-04-02-KAP05 Model Bir Otonom Hava Aracı İçin (Helikopter) Yeni Kontrol Algoritmalarının Geliştirilmesi, Elektrik Mühendisliği 2010-05-15/2012-05-15