

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

HAVA ROBOTU İLE HEDEF KONUMLANDIRMA

Yüksek Lisans Tezi

SAMET ÖZABACI

EKİM 2019

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

HAVA ROBOTU İLE HEDEF KONUMLANDIRMA

Yüksek Lisans Tezi

SAMET ÖZABACI

DANIŞMAN
DR. ÖĞR. ÜYESİ HALUK BAYRAM

EKİM 2019

ONAY

İstanbul Medeniyet Üniversitesi, Fen Bilimleri Enstitüsü'nde Yüksek Lisans öğrencisi olan Samet Özabacı'nın hazırladığı ve jüri önünde savunduğu "Hava Robotu ile Hedef Konumlandırma" başlıklı tez başarılı kabul edilmiştir.

JÜRİ ÜYELERİ

Tez Danışmanı:

Dr. Öğr. Üyesi Haluk BAYRAM

Kurumu: İstanbul Medeniyet Üniversitesi

İMZA



Üyeler:

Dr. Öğr. Üyesi İbrahim Genç

Kurumu: İstanbul Medeniyet Üniversitesi



Prof. Dr. H. Işıl BOZMA

Kurumu: Boğaziçi Üniversitesi



Tez Savunma Tarihi: Ekim 2019

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim.

Samet ÖZABACI



Danışmanlığımı yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığımı taahhüt ederim.



Dr. Öğr. Üyesi Haluk BAYRAM

ÖNSÖZ/TEŞEKKÜR

Öncelikle lisansüstü eğitimim süresince ders ve tez çalışmalarında bilgi, deneyim ve görüşlerinden çokça istifade ettiğim, danışman hocam Sayın Dr. Öğr. Üyesi Haluk Bayram'a teşekkürlerimi sunarım.

Tez savunma jürisinde yer alan kıymetli hocalarım Dr. Öğr. Üyesi İbrahim Genç ve Prof. Dr. Işıl Bozma'ya katılımlarından dolayı teşekkür ederim.

Ayrıca çalışmamızda ihtiyaç duyduğumuz mekanik parçaların üretimde yardımlarını esirgemeyen YB Makine Teknik Müdürü sayın Haşmet Abacı'ya teşekkürlerimi sunarım.

Bu çalışma kısmî olarak TÜBİTAK 2232 programı 119C008 nolu proje kapsamında desteklenmiştir.

Son olarak lisansüstü eğitimim süresince onlara ayırmam gereken vakti eğitime ayırmamı hoş gören, maddi, manevi desteklerini her koşulda esirgemeyen sevgili aileme teşekkürlerimi bir borç bilirim.

İstanbul, Ekim 2019

Samet Özabacı

İÇİNDEKİLER

Sayfa

ONAY.....	iii
BİLDİRİM.....	iv
ÖNSÖZ/TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
KISALTMALAR.....	viii
TABLO LİSTESİ.....	x
ŞEKİL LİSTESİ.....	xi
ÖZET VE ANAHTAR KELİMELER.....	xiv
ABSTRACT AND KEYWORDS.....	15
1. GİRİŞ.....	16
1.1 Problem Tanımı.....	16
1.2 Literatür.....	17
1.3 Tezin Katkısı.....	18
1.4 Yaklaşım.....	19
1.5 Tezin Akışı.....	19
2. SİSTEM YAPISI.....	21
2.1. Hava Robotu Sistemi Bileşenlerinin Belirlenmesi.....	21
2.2. Hava Robotu Bileşenleri.....	28
2.2.1. Uçuş Kontrolcüsü.....	30
2.2.2. Robot İşlem Birimi.....	31
2.2.3. Küresel Konumlama Sistemi.....	31
2.2.4. Yazılım Tabanlı Radyo.....	32
2.2.5. Motor Hız Kontrolcüsü.....	33
2.2.6. Telemetry.....	34
2.2.7. Fırçasız Motor.....	35
2.2.8. Gövde.....	35
2.3. Hava Robotu Montajı.....	36
3. SENSÖR MODELLEME.....	40
3.1 Modelleme için Tasarlanan Sistem.....	40
3.1.1. Sinyal Alıcı Devresi.....	40
3.1.2. Radyo Verici Devresi.....	44
3.2 Moxon Anten.....	45
3.3 Sensör Verisi Toplama.....	48
4. YAKLAŞIM.....	52
4.1. Algılama Modeli ve Belirsizlik Ölçütü.....	52
4.2. En iyi Çevrimdışı Strateji.....	53
4.3. Çevrimiçi Stratejiler.....	55
4.3.1. Açığortay Üzerinde Ölçüm.....	56
4.3.2. Ölçüm Üçgeni'nin Bir Kenarı Üzerinde Ölçüm.....	59
4.3.3. Ölçüm Üçgeninin Kenarları Üzerinde Sıralı Ölçüm.....	61
5. SİMULASYON ve SITL TABANLI TESTLER.....	64
5.1. Simülasyon.....	64
5.2. SITL ile Tabanlı Testler.....	70

6. SONUÇ	75
KAYNAKÇA	76
EK 1. GQRX KURULUMU	79
EK 2. DRONEKIT ve DRONEKIT-SITL KURULUMU	80
EK 3. QGROUNDCONTROL KURULUMU.....	81
EK 4. MAVPROXY KURULUMU	82
EK 5. SITL KULLANILARAK SİMÜLASYON YAPILMASI.....	83
ÖZGEÇMİŞ.....	84



KISALTMALAR

ADC	: Analog Digital Converter
ASK	: Amplitude Shift Keying
BLE	: Bluetooth Low Energy
BNC	: Bayonet Neill–Concelman
CAN	: Controller Area Network
ESC	: Electronic Speed Control
FPU	: Floating Point Unit
GNSS	: Global Navigation Satellite System
GPS	: Global Positioning System
HDMI	: High Definition Multimedia Interface
I2C	: Inter Integrated Circuit
İHA	: İnsansız Hava Aracı
LiPo	: Lithium Polymer
MAh	: Milliamp Hour
MB	: Mega Byte
MHz	: Mega Hertz
OOK	: On – Off Keying
PCB	: Printed Circuit Board
PPM	: Pulse Position Modulation
PWM	: Pulse Width Modulation
RAM	: Random Access Memory
RF	: Radio Frequency
ROS	: Robot İşletim Sistemi (Robot Operating System)

SBC : Tek Karthı Bilgisayar (Single Board Computer)

SBUS : Serial BUS

SDR : Software Defined Radio

SDRAM : Synchronous Dynamic Random Access Memory

SITL : Software in the Loop

SMA : SubMiniature version A

SoC : System on Chip

SPI : Serial Peripheral Interface

UART : Universal Asynchronous Receiver/Transmitter

USB : Universal Serial BUS

TABLO LİSTESİ

Sayfa

Tablo 1. Sistem bileşenlerinin ağırlık listesi	22
Tablo 2. Sistem harici bileşenlerinin ağırlık listesi	23
Tablo 3. Hava robotunun bileşenleri ve görevleri	29
Tablo 4. Sensör modelleme alıcı devresi bileşenleri.....	40
Tablo 5. Açığortay Üzerinde Çalışan Algoritma.....	58
Tablo 6. Ölçüm üçgeninin bir kenarında çalışan algoritma	60
Tablo 7. Ölçüm üçgeninin kenarları üzerinde sıralı çalışan konumlandırma algoritması	62
Tablo 8. Açığortay algoritmasının robot üzerindeki gerçekleştirmesinin sözde-kodu.	71

ŞEKİL LİSTESİ

Sayfa

Şekil 1. Sistemin genel yapısı	21
Şekil 2. ZD550 4-rotorlu gövde	24
Şekil 3. Hava robotu kuvvet grafiği	25
Şekil 4. T-Motor 3515 veri tablosu (Machineto, 2019)	25
Şekil 5. Hava robotunun tüm ekipmanlar monte edilmiş hali	27
Şekil 6. Hava Robotu Donanım Diyagramı	28
Şekil 7. Pixhawk2 Cube uçuş kontrolcüsü	30
Şekil 8. Raspberry Pi 3B+	31
Şekil 9. Here2 GNSS	32
Şekil 10. RTL-SDR	32
Şekil 11. Gqrx programının ekran görüntüsü. RTL-SDR'dan alınan verilerin anlık gösterimi için kullanılmaktadır.	33
Şekil 12. Emax BL-Heli 80A motor sürücü	34
Şekil 13. 915 MHz telemetri	34
Şekil 14. T-Motor MT3515 650KV fırçasız motor	35
Şekil 15. ZD550 karbon fiber gövde	35
Şekil 16. (a) Hava robotunun önden görünüşü. (b) Hava robotunun üstten görünüşü	36
Şekil 17. İniş ayakları gövde montaj parçası	37
Şekil 18. Pervane adaptörü ve pulun montajlı hali	37
Şekil 19. Hava robotunun alt bölmesi	38
Şekil 20. Kıvılcım önleyici devre	38
Şekil 21. Pixhawk ve Raspberry Pi arasındaki bağlantı (Ardupilot, Ardupilot, 2019)	39
Şekil 22. Sensör modelleme alıcı devresi	41
Şekil 23. Sensör modelleme alıcı sistem yapısı	42
Şekil 24. Sensör modelleme yazılım ekran görüntüsü	42

Şekil 25. Sensör modelleme yazılım modülleri arasındaki ilişki	43
Şekil 26. 45 derece aralıkla alınan ölçüler. Hedef 0 dereceye gelecek şekilde alıcıdan 150 metre ileriye yerleştirilmiştir.....	43
Şekil 27. Radyo verici sistem yapısı.....	44
Şekil 28. Hedef olarak kullanılan radyo verici devresi.....	44
Şekil 29. Örnek Moxon ve Yagi anten karakteristik grafiği (Cebik, 2000).....	46
Şekil 30. 433 Mhz Moxon anten ölçüleri (Ac61a, 2019)	46
Şekil 31. (Sol Üst) BNC ve SMA konnektör bağlı Moxon anteni, (Sağ Üst) 3 boyutlu yazıcı ile üretilen motor-anten ara bağlantı parçası, (Sol Alt) motor üzerine bağlanmış ara bağlantı parçası, (Sağ Alt) çalışmakta olan sensör modelleme devresi.	47
Şekil 32. İstanbul Medeniyet Üniversitesi Kuzey Kampüsü ölçüm noktaları.....	48
Şekil 33. Mesafeye bağlı ölçüm grafikleri.....	49
Şekil 34. 150 metre mesafeden alınan ölçüm örneği.....	50
Şekil 35. (a) Algılama modelinin gösterimi. Hedef w konumunda ve ölçüm üçgeninin içinde bulunduğu bilinmektedir. Doğru doğrultu açısı, bilinmeyen ancak en fazla α kadar bozulmuştur. (b) İki doğrultu açısı ölçümü kesişiminin paralelkenar ile yaklaşılması. Şekillerde, hedefin konumu yıldız ile gösterilmiştir. (Bayram , Stefas, & Isler, 2018).....	52
Şekil 36. a) İkinci ölçümün alınabileceği bölge. b) Eniyi ölçüm yerleri (Bayram , Stefas, & Isler, 2018)	54
Şekil 37. Açığortay Üzerinden Çalışan Simülasyon Örneği	57
Şekil 38. Ölçüm üçgeninin sağ kenarı üzerinde çalışan simülasyon örneği.....	61
Şekil 39. Ölçüm üçgeninin kenarlarında sırayla ilerleyen simülasyon örneği	63
Şekil 40. $U^* = 314$ için rekabetçi oran – α grafiği	65
Şekil 41. $U^* = 314$ için lokalizasyon süresi – α grafiği.....	66
Şekil 42. $U^* = 314$ için ölçüm sayısı – α grafiği	67
Şekil 43. $U^* = 1256$ için rekabetçi oran – α grafiği	68
Şekil 44. $U^* = 1256$ için lokalizasyon süresi – α grafiği.....	69
Şekil 45. $U^* = 1256$ için ölçüm sayısı – α grafiği	69
Şekil 46. SITL ve Mavproxy terminal çıktısı	70

- Şekil 47.** SITL tabanlı test sonucu. Hava robotunun konumlandırma yaparken izlediği rota kırmızı çizgi ile gösterilmektedir.....72
- Şekil 48.** APM ile Hava Robotu Simülasyonu Program Çıktı Görüntüsü.....73



ÖZET VE ANAHTAR KELİMELER

HAVA ROBOTU İLE HEDEF KONUMLANDIRMA

Bu tezde, radyo sinyal kaynağının bir hava robotu ile konumlandırma problemi ele alınmıştır. Sabit irtifada hareket eden bir hava robotu, yerde bulunan bir radyo sinyal kaynağını, üzerindeki yönlü anten sayesinde konumlandırmaya çalışmaktadır. Burada konumlandırma işleminde, sadece hedefe olan doğrultu açısı kullanılmaktadır. Doğrultu ölçüm açısının, sınırlı gürültüye sahip olduğu varsayımı altında, üç farklı çevrimiçi konumlandırma stratejisi geliştirilmiştir. Geliştirilen bu stratejilerin performansları, en iyi çevrimdışı stratejisinin maliyet alt sınırı üzerinden rekabetçi oranları hesaplanarak analiz edilmiştir. Kapsamlı bilgisayar benzetimleri ile farklı parametre değerlerinin performans analizi yapılmıştır. Ayrıca konumlandırma işlemini yapabilecek, 4 rotorlu bir hava robotu üretilmiştir. Geliştirilen stratejilerden en iyi performansa sahip olan strateji, robot üzerinde gerçekleştirilerek, SITL kullanılarak test edilmiştir.

Anahtar Kelimeler: Hedef konumlandırma, Hava robotu, Çevrimiçi algoritmala

ABSTRACT AND KEYWORDS

TARGET LOCALIZATION WITH AN AERIAL ROBOT

This thesis addresses the problem of localizing a radio source using aerial robots. In this scenario, there is an aerial robot flying at a fixed altitude and a radio source locating in the ground. The goal of the robot is to localize the target using its directional antenna. In the localization process, the only input to the robot is noisy bearing measurements. It is assumed that bearing measurements are modelled using a bounded noise model. Under these assumptions, three online localization strategies are developed. The performance of these strategies are evaluated using competitive ratio, which uses the lower bound of the cost of the optimal offline strategy. The performance of the strategies is validated through a series of extensive computer simulations. Finally, the best of the developed strategies is implemented on the robot and tested using SITL.

Keywords: Target localization, Aerial robot, Online algorithms

1. GİRİŞ

Bu çalışma, radyo sinyal kaynağı lokalizasyonunun hava robotu ile konumlandırılmasını konu edinmektedir. Konumlandırma için yapılacak ölçümler uzaklık ve/veya doğrultu açısı tabanlı olabilir. Uzaklık ölçümleri, anten ile sinyal kaynağının bulunduğu ortamın farklılaşması nedeniyle, bozulmaya uğrayıp; elde edilen ölçümlerdeki uzaklık verisi de tutarlı olmayacaktır. Doğrultu açısı ölçümlerinde ise bu tür bir bozucu etki tüm açılara aynı derecede etkiyeceğinden, ölçüm performansını önemli derecede etkilemeyecektir. Bu nedenle, hedefe olan uzaklık verisi kullanılmayıp sadece doğrultu açısı kullanılarak hedef konumlandırma yaygın olarak kullanılmaktadır. Ancak burada önemli bir bilgi sağlayan uzaklık dikkate alınmadığından tasarlanacak stratejinin seçimi performansı büyük ölçüde etkileyecektir. Bu tür stratejiler ya çevrimiçi ya da çevrimdışı olarak çalışmaktadır. Çevrimiçi stratejiler, görev sırasında sonraki hareketi o ana kadar aldığı veriler üzerinden hesaplamaktadır. Bu da göreve başlamadan önce tüm hareketleri hesaplamak yerine görev sırasında hesaplama olanağı vermektedir. Çevrimdışı stratejilerde ise göreve başlamadan önce görev esnasında yapılacak tüm hareketler belirlenir. Çevrimdışı stratejiler, çevrimiçi stratejiler gibi görev sırasında alınan verileri kullanmadıklarından, görev başarımları çevrimiçi stratejilere göre farklılık gösterebilmektedir.

1.1 Problem Tanımı

Robot, konumlandırma yapacağı radyo sinyal kaynağından veri alabilecek uzaklıkta olsun. Robot, hedefe olan doğrultu açısını belirlemek için ölçümler alabilmektedir. Her bir doğrultu açısı ölçümü, hedefi içine olan ve tepe açısı 2α olan bir üçgen oluşturmaktadır. Konumlandırma, bu ölçüm üçgenlerinin kesişiminin alınması ile yapılmaktadır. Bu ölçümler sabit bir τ_0 süresi almaktadır. Ölçüm i 'nin alındığı yer s_i olarak tanımlanırsa, hedefi konumlandırma için alınan ölçüm yerleri kümesi $S = \{s_1, \dots, s_N\}$ olur. Problem, hedefin istenilen belirsizlik seviyesinde konumlandırılabilmesi için aşağıdaki maliyet fonksiyonunu minimum yapacak şekilde ölçüm yerlerinin belirlenmesidir:

$$\text{maliyet}(S) = \sum_{i=1}^{N-1} d(s_i, s_{i+1})/v + N\tau_{\delta}$$

Maliyet fonksiyonunun ilk kısmı, ölçüm noktalarına ulaşım için harcanan zamanı; ikinci kısmı ise ölçüm almak için harcanan toplam zamanı göstermektedir. $d(s_i, s_{i+1})$, s_i ve s_{i+1} ölçüm yerleri arasındaki uzaklığı; v , robotun hızını; τ_{δ} , her bir doğruyu açı ölçümü için harcanan zamanı; N ise toplam ölçüm sayısını ifade etmektedir.

1.2 Literatür

Hava robotları kullanarak radyo sinyali (RF) kaynağı konumlandırmanın yaban hayatı takibi (Bayram , Stefas, & Isler, 2018), WIFI router (Perkins, Dressel, Lo, & Enge, 2015), ZigBee sinyal kaynağı (Isaacs, Quitin, Carrillo, & Madhow, 2014), GPS Jammer sinyal kaynağı (Perkins, Dressel, Lo, & Enge, 2015) ve mobil telefon sinyal kaynağı (Dressel & Kochenderfer, 2018) tespit etmek gibi birçok uygulaması bulunmaktadır. Bu uygulama alanlarının hepsinde en kritik ve ortak nokta konumlandırma çalışmasının mümkün olan en kısa sürede gerçekleştirilmesidir. Örneğin arama kurtarma (Hoffmann, Waslander, & Tomlin, 2006) ve jammer konumlandırması gibi çalışmalarda konumlandırma süresi hayati önem taşımaktadır (Gibbons, 2013). Özellikle doğal afet anlarında yaralıların tespit edilmesi ve medikal müdahaleye olabildiğince hızlı ulaşmaları hayati önem taşımaktadır. Buna rağmen deprem, heyelan, çığ vb. afetler durumunda kazazedeleri tespit etmek, yardım ve tedavi götürmek oldukça zordur. Bu durumlarda arama kurtarma çalışmaları çok fazla zaman alır. Kimi zaman afet mahalli insanların kolaylıkla ulaşamayacağı yerlerde olabilir ve kazazedenin konumunun tespit edilmesi dahi, kazazede hayati tehlikeye girmeden önce mümkün olmamaktadır. Afet anlarının yanı sıra jammer konumlandırması da hayati önem taşımaktadır. Hava alanları, askeri alanlar ve gizlilik arz eden mekânlar çevresinde bulunan jammerlar çeşitli hava araçlarının, askeri araçların vb. haberleşmesini keseceğinden kazalara, hayati tehlikelere sebep olabilmektedir. Bundan dolayı olabildiğince hızlı ve otonom konumlandırma önem arz etmektedir. İnsan yaşamını riske atacak kadar ciddi sonuçlar doğurmasa da etkin ve hızlı konumlandırma yaban hayatı takibinde (Bayram , Stefas, & Isler, 2018) de büyük önem taşımaktadır. Takip edilen hayvanın davranışlarının izlenmesi ve koruma

faaliyetlerinin yüksek doğrulukla saptanabilmesi için lokalizasyon süresi kritik rol oynamaktadır. Bununla beraber hava robotlarının veya İHA'ların havada kalma sürelerinin kısıtlı olmasından dolayı konumlandırma süresi tüm çalışmalarda bir numaralı faktör haline gelmektedir.

Hava robotu ile RF sinyali tespit etmede genel olarak, hava robotu üzerine entegre mini bilgisayar ile birlikte çalışan RF sinyal alıcıya bağlı bir veya birden fazla anten vasıtasıyla alınan sinyalin doğrultu ve güç bilgisi hava robotunun anlık konumu birleştirilerek bir sonraki konum için çıkarımda bulunulur. Bu çıkarımlara sinyal kaynağına yeterince yaklaşıldığına dair sonuca varılana kadar devam edilir. Örneğin Bayram ve diğerleri (Bayram , Stefas, & Isler, 2018) DJI Matrice 100 hava robotu ile bir adet 3 element Yagi (Carr & Gippisley, Practical Antenna HandBook, 2012) anten ve RTL-SDR yazılım tabanlı radyo kullanarak her ölçüm konumunda açılı ölçümler olarak yaban hayatı takibi uygulamaları için konumlandırma yapmışlardır. Dressel ve Kochenderfer (Dressel & Kochenderfer, 2018) çalışmalarında Bayram ve diğerleri (Bayram , Stefas, & Isler, 2018) ile aynı hava robotu ve yazılım tabanlı radyo kullanarak, hava robotunun ön ve arkasına bağlı iki farklı Moxon antenini (Carr & Gippisley, Moxon Beam, 2012) ayrı sinyal alıcılara bağlayıp simültane ölçümler olarak sinyalin gelme yönü hakkında elde ettikleri çıkarımı, GPS ile elde ettikleri robotun mevcut konumu ile birleştirerek elde ettikleri veriler ile fırsatçı(greedy) arama tekniği kullanarak bir sonraki konum hakkında çıkarım yaparak sinyal kaynağına ulaşmaya çalışmışlardır. Yine Dressel ve Kochenderfer (Dressel & Kochenderfer, 2019) bir önceki çalışmalarında kullandıkları donanımı, yazılım tabanlı radyo bileşenini, daha güçlü olan HackRF ile değiştirerek bir diğer hava robotunu telemetri sinyali üzerinden takip etme çalışması yapmışlardır.

1.3 Tezin Katkısı

Tezde yapılan katkılar şu şekilde sıralanabilir:

- Hedef konumlandırma amaçlı uçan robotik sistemin mekanik ve elektronik olarak üretilmesi,
- Moxon anten kullanımı ile doğrultu açısı elde etme,

- Hedef konumlandırma için doğrultu açısı tabanlı çevrimiçi algoritmaların geliştirilmesi,
- Geliştirilen algoritmaların rekabetçi oran üzerinden performans analizinin yapılması,
- Geliştirilen algoritmaların uçan robotik sistem üzerinde gerçekleşip, SITL tabanlı deneylerinin yapılması.

1.4 Yaklaşım

Geliştirilen yaklaşımlar, doğrultu açı ölçümleri kullanarak konumlandırma yapmaktadır. Doğrultu açısında, sınırlı gürültü modeli olduğu kabul edilmiştir. Bu modelde, her bir ölçüm hedefin içinde bulunduğu bir üçgen vermektedir. Hedefin konumlandırılması alınan bu ölçüm üçgenlerinin kesişimlerinin alınması ile yapılmaktadır. Bu sınırlı gürültü modelini esas alarak, bir sonraki ölçüm yerini belirleyen çevrimiçi algoritmalar geliştirilmiştir:

- Ölçüm üçgeninin açıortayı üzerinde hareket tabanlı algoritma: Burada, robot aldığı doğrultu ölçümü üzerinde katlama-yarılama (doubling-halving) stratejisini kullanarak, bir sonraki ölçüm yerini belirlemektedir.
- Ölçüm üçgeninin sağ kenarı üzerinde hareket tabanlı algoritma: Burada, robot ölçüm üçgeninin sağ kenarı üzerinden, katlama-yarılama stratejisine göre bir sonraki ölçüm yerini belirlemektedir.
- Ölçüm üçgeninin sağ ve sol kenarları üzerinde hareket tabanlı algoritma: Robot, ölçüm üçgeninin bir sağ kenarı bir sol kenarında hareket edip, ölçüm yerini katlama-yarılama stratejisine göre belirler.

Burada hareket ederken kullanılan katlama-yarılama yaklaşımı sayesinde, hedef ne kadar uzakta olursa olsun, toplam ölçüm sayısı belli bir oranın içinde kalmaktadır. Aksi halde, sabit adım mesafesi kullanılmış olsaydı, uzaklığın artması ile sınırsız bir şekilde performans kötüleşecekti.

1.5 Tezin Akışı

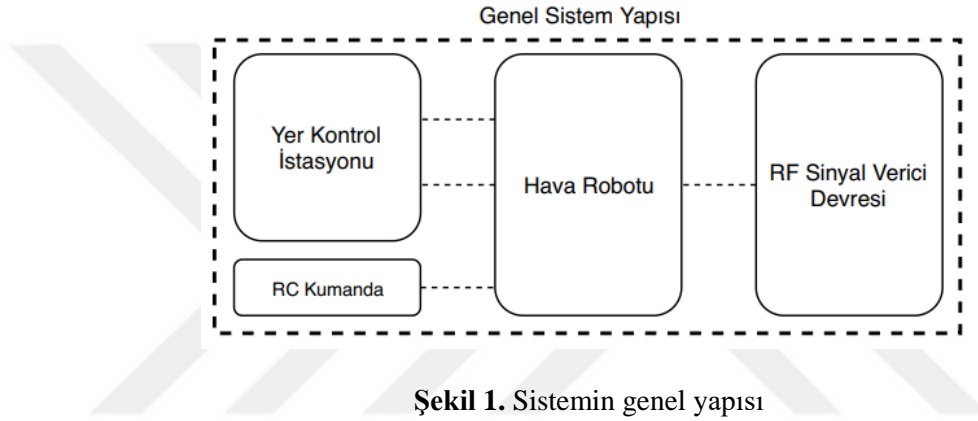
Tezin genel yapısı şöyledir: Bölüm 2’de hava robotunun tasarım aşamasında dikkat edilmesi gereken parametrelerden ve uygun bileşen seçiminden bahsedilmiş ve hava robotunu oluşturan bileşenler detaylı açıklanmıştır. Ayrıca tasarım ve montaj

aşamasında karşılaşılan ve çözülen problemlerden de bahsedilmiştir. Bölüm 3'te konumlandırma için kullanılan Moxon antenini modellemek, hava robotu olmadan da ölçüm yapabilmeyi sağlayan sensör modelinin yazılım ve donanım detayları verilmiştir. Yine 3. bölümde Moxon anten açıklanmış ve sensör model ile toplanan veriler değerlendirilmiştir. Bölüm 4'te konumlandırma için geliştirilen algoritmalar hakkında detaylar verilmiştir. Bölüm 5'te konumlandırma için geliştirilen algoritmalar, tasarlayıp üretilen hava robotu ve bu sistemlerin birlikte çalışmasının simülasyon ortamlarında gerçekleşmesi ve saha testlerinin detayları verilmiştir. Bölüm 6'da ise çalışmanın sonuçları tartışılmış ve olası gelecek çalışmalar hakkında bilgi verilmiştir. Ekler bölümünde ise tezi destekleyen ek materyallere yer verilmiştir.



2. SİSTEM YAPISI

Sistem, tüm bileşenleriyle hava robotu, yer kontrol istasyonu, RF sinyal verici devre ve RC kumandanadan oluşmaktadır. Sistemin genel yapısı Şekil 1’de görülmektedir. Lenovo Yoga 2 Pro dizüstü bilgisayar yer kontrol istasyonu olarak kullanılmıştır. Sistemde Radiolink AT9S SBUS destekli 10 RC kanal kumanda kullanılmıştır. Bir sonraki bölümde hava robotunu oluşturulan bileşenlerden, bileşenlerin seçimlerinden ve sistemin özelliklerinden detaylı olarak bahsedilecektir. RF sinyal verici devresinin detayları ise Bölüm 3.3.2’de verilecektir.



2.1. Hava Robotu Sistemi Bileşenlerinin Belirlenmesi

Hava robotu tasarlamaya başlamadan önce karar verilmesi gereken, birçok bağımlı parametre bulunmaktadır. Öncelikle parametreleri belirleyerek istenilen robotun niteliklerini belirledikten sonra, parametrelere en uygun bileşenler seçilerek robotun yapımına başlanmalıdır. Hava robotunun yapımına başlamadan önce belirlememiz gereken başlıklar:

- Hava robotunun amacının belirlenmesi
- Ne kadar sistem harici ağırlık taşıması gerektiği
- Ağırlığı kaldıracak, bileşenleri taşıyacak gövdenin seçimi
- Hava robotunu oluşturan tüm bileşenlerin toplam ağırlıklarının belirlenmesi
- Toplam ağırlığı kaldırıp manevra yapmaya olanak sağlayacak güçte motor ve pervane belirlenmesi
- Motorların çalışma aralığına uygun motor sürücülerin belirlenmesi

- Motorların çalışma gerilimine uygun, istenilen uçuş süresini sağlayabilecek pilin belirlenmesi

İlk maddeden başlayacak olursak hava robotumuzun amacı yazılım tabanlı radyo ve yönlü antenler kullanarak sinyal kaynağı konumlandırması yapmaktır. Bu amaç doğrultusunda kullanabileceğimiz iki farklı anten bulunmaktadır. Bunlar: Yagi ve Moxon anten. Bu çalışmada yapımı kolay, hafif, küçük ve düşük maliyetli olmasından dolayı Moxon anten kullanılmıştır. Ancak sistem, gelecek çalışmalarda farklı anten ve yükler ile kullanabilmek adına daha büyük ve ağır olan Yagi anten'e göre tasarlanmıştır

Hava robotunun kullanım amacı belirdikten sonra robotu çalıştıran sistem bileşenlerinin dışındaki parçalar ve bunların oluşturacağı yaklaşık ağırlık belirlenmelidir. Sinyal kaynağı konumlandırması yapılabilmesi için bir adet mini bilgisayara (Raspberry Pi 3B+), bir adet yazılım tabanlı radyoya, sinyalleri toplayacak antene ve gerekirse dönüştürücü ve kabloları ihtiyaç duyulmaktadır. Bu aşamada oluşturmak istediğimiz sistemin tahmini toplam ağırlığını belirlememiz gerekmektedir.

Tablo 1. Sistem bileşenlerinin ağırlık listesi

Bileşen	Ağırlık
Frame/Gövde	630 gr
İniş Ayakları	200 gr
4 x Motor	800 gr
4 x ESC	320 gr
4 x Pervane	60 gr
Lipo Pil	1300 gr
Pixhawk Cube	~100 gr
GNSS	49 gr
GPS Tutucu	~ 20 gr
RC Alıcı	~ 20 gr
Telemetry	~ 15 gr
Zigbee	~ 15 gr
Toplam	3630 gr

Tablo 1 ve Tablo 2’de sistem bileşenleri ve sistem harici bileşenlerin toplam ağırlığı 4755 gram olarak verilmiştir. Burada dikkat edilmelidir ki, bu liste ve ağırlık bilgileri sistemi toplamaya başlamadan önce oluşturulan tahmini verilerdir. İlerleyen bölümlerde sistem bileşenlerine kesin karar verdikten sonra bu listeler yeniden ve gerçek veriler ile oluşturulacaktır.

Tablo 2. Sistem harici bileşenlerinin ağırlık listesi

Bileşen	Ağırlık
Anten	~ 1000 gr
Ana Bilgisayar	~ 100 gr
RTL-SDR	~ 15 gr
Dönüştürücüler	~ 5 gr
Kablolar	~ 5 gr
Toplam	1125 gr

Tablo 1 ve Tablo 2’deki verilerin bir kısmı kullanmayı düşündüğümüz parçaların ağırlıkları kullanılarak, bazıları ise ortalama veriler kullanılarak oluşturulmuştur. Elde edilen ağırlık bilgisi maksimum değeri vermektedir. Bulunan değer, bu ağırlığa ulaşılacağı veya bu ağırlıkla uçulacağı anlamına gelmemektedir. Sistemi olası maksimum ağırlıklar göz önüne alınarak tasarlanmıştır ki hava robotu, olası tüm konfigürasyonlarda görevini zorlanmadan yerine getirebilsin.

Ağırlık bilgileri, ülkemiz içerisinde parçalara erişim, maliyet, fiyat ve performans değişkenleri değerlendirilerek oluşturulmuştur. Hava robotu 4 kanatlı (quad) olarak tasarlamaya karar verilip, Şekil 2’de görülen ZD550 kodlu gövdenin kullanılmasına karar verilmiştir.

Tahmini en yüksek kalkış ağırlığı ve hava robotunun sahip olacağı motor sayısını belirledikten sonra ihtiyaca uygun motor seçmelidir. Hava robotuna uygun motor seçmek için, motorun sahip olması gereken minimum itki gücü hesaplanmalıdır. En yüksek tahmini kalkış ağırlığı olan 4755 gr’ı 4’e (quad) bölersek, her bir kola bağlı

motorun uygulaması gereken itki miktarı $4755/4 = 1188$ gr olarak bulunur. Bulunan bu değer her bir motorun uygulaması gereken itkiyi belirtmektedir..



Şekil 2. ZD550 4-rotorlu gövde

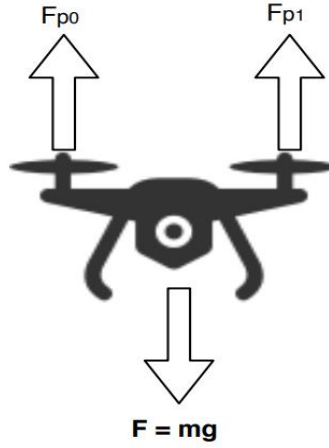
Şekil 5’de görülen hava robotunun toplam ağırlığı (F), her bir pervanenin uyguladığı toplam kuvvete $\sum_x F_{px}$ eşit olmalıdır. Ancak bu durum havada kalma işlemi için geçerlidir. Hava robotunun yükselebilmesi, manevra yapabilmesi ve özellikle ilk kalkışını yapabilmesi için motorların uyguladığı toplam itki $\sum_x F_{px}$ hava robotunun ağırlığından büyük olmalıdır ($\sum_x F_{px} > F_t$). Bu işlemi de bir katsayı ile ifade edecek olursak denklemimiz $F_t k = \sum_x F_{px}$ olmaktadır.

Hava robotunun zorlanmadan iniş kalkış ve manevra yapabilmesi için $k > 2$ olarak alınması önerilmektedir. (Alex, 2015) Ancak malzeme ağırlıklarının seçiminde tüm bileşenleri en üst sınırdan seçtiğimizden dolayı k değerini 1,5 almak yeterli olacaktır. k değerini 1.5 olarak tekrar hesapladığımızda her bir motorun sağlaması gereken toplam itki değeri 1780 gr olarak bulunur.

$$4755 * 1,5 = 7132 \text{ gr}, 7132 / 4 = 1780 \text{ gr}$$

Elde edilen itki değerini baz alarak ülkemizden kolayca satın alabileceğimiz, stok durumu çalışmalarımıza engel olmayacak, fiyat ve performans açısından da en uygun değeri verebilecek olan Tiger Motor MT3515 650KV fırçasız motoruna karar verilmiştir.

Motor seçiminde dikkat edilmesi gereken en önemli nokta Şekil 4’te de görüldüğü üzere üretici tarafından paylaşılan motorun kullanım kılavuzuna başvurmaktır. Seçilen motor ihtiyaç duyulan itkiyi sağlayıp sağlamadığına ve bu itkiyi sağlaması için gereken voltaj, akım ve pervane değerlerine bakılır.



Şekil 3. Hava robotu kuvvet grafiği

Şekil 4’te 3 rakamı ile gösterilen altı çizgili sütun, motorun 1 ile gösterilen sütundaki gerilim değeri ve 2 ile gösterilen sütundaki pervaneler ile tam güçte verebileceği itkiyi göstermektedir. Şekil 4’te de görüldüğü üzere motor, önerilen en düşük gerilim ve pervane ile dahi daha önce hesaplanan her bir motorun sağlaması gereken itki değeri olan 1780 gram değerine çok yakın performans sağlamaktadır.

Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (G)	RPM	Efficiency (G/W)	Operating temperature (°C)
MT3515 KV650	14.8 1	T-MOTOR 13*4.4CF 2	50%	4.3	64	650	4800	10.21	47
			65%	7.9	117	960	6000	8.21	
			75%	11.1	164	1280	6700	7.79	
			85%	14.8	219	1480	7500	6.76	
			100%	18.1	268	1690 3	8000	6.31	
		T-MOTOR 14*4.8CF 2	50%	5.6	83	840	4500	10.14	53
			65%	10.9	161	1300	5700	8.06	
			75%	15.1	223	1640	6300	7.34	
			85%	20.5	303	2000	6950	6.59	
			100%	24.8	367	2200 3	7300	5.99	
		T-MOTOR 15*5CF 2	50%	7.2	107	870	4300	8.16	60
			65%	13.8	204	1430	5300	7.00	
	75%		19	281	1800	5900	6.40		
	85%		25.4	376	2100	6500	5.59		
	100%		30.7	454	2400 3	6800	5.28		
	22.2 1	T-MOTOR 13*4.4CF 2	50%	8.7	193	1200	7000	6.21	61°C/150S
			65%	15.1	335	1830	8500	5.46	
			75%	20.8	462	2300	9500	4.98	
			85%	27.8	617	2750	10300	4.46	
			100%	33.2	737	3100 3	10800	4.21	

Notes: The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min.

Şekil 4. T-Motor 3515 veri tablosu (Machineto, 2019)

Şekil 4'teki değerlere istinaden, hava robotunda 22.2V gerilim sağlayabilen 6S 10.000 mAh 25C pil kullanılacaktır. Şekil 4'teki motor için 14" ve 15" karbon fiber pervaneler seçilmiştir. Yine motorun kullanım kılavuzuna bakacak olursak, 22.2V pil ile motorun verimi 14.8V seçeneklerinden daha yüksektir ve daha fazla itki sağlamaktadır. Bundan dolayı 22.2V pil tercih edilmiştir.

Seçilen pil ile yapabilecek maksimum uçuş süresini hesaplamanın birçok yöntemi bulunmaktadır. Bunun için ecalc (Ecalc, 2010) gibi birçok online hesaplama aracı da mevcuttur. Mevcut yöntemlerden bir tanesini açıklanarak uçuş süresi hesaplanacaktır. Öncelikle kullanılan pilin kapasitesi mili amper/saat'ten amper/dakika'ya çevrilir.

$$(10.000 \text{ mili amper/saat}) * 60 / 1000 = 600 \text{ amper/dakika}$$

Bu değer pilin tam kapasitesini göstermektedir. Lipo piller oldukça tehlikeli pillerdir. Kısa devre olması durumunda çok fazla akım vererek yanmaya başlarlar. Bundan dolayı kullanırken kullanım talimatlarına azami derecede uymak ve çok dikkatli olmak gerekmektedir. Lipo piller hücre başına maksimum dolum gerilimi olan 4.2V'un üzerinde şarj edilmemelidir ve hücre voltajları 3.3V'un altına indirilmemelidir. Aksi takdirde şişme yapabilir, pilin ömrü azalır ve hatta kullanılmaz hale gelebilir. Bu sebeplerden dolayı pili tam boşaltmadan kullanmamız gerekmektedir. Pilin kapasitesini %80 oranında kullanmak için mevcut 600 amper/dakika kapasitesininin 0.8 ile çarparak toplam kullanılabilir kapasiteyi hesaplıyoruz.

$$600 \text{ amper/dakika} * 0.8 = 480 \text{ amper/dakika}$$

Toplam uçuş süresini hesaplamak için toplam kalkış ağırlığının hesaplanması gerekmektedir. Şekil 5'te de görüldüğü üzere montajı tamamlanmış sistemin toplam ağırlığı 3900 gramdır. Her bir motora düşen ağırlık değerini bulmak için toplam ağırlığı hava robotunun kanat sayısına böldüğünde 975 gram değerini elde edildi. Şekil 4'te 22.2V seçeneğinin bulunduğu satırda *thrust* kolonuna bakacak olursak, motorumuz %50 güçte çalıştığında 1200 gram itki üreteceği görülmektedir. 22.2V çalışma gerilimi ve %50 çalışma aralığına ait satırda motorun 8.7A akım çektiği görülmektedir. Bu değer 1200 gram itki için olduğundan, sistemimizin ağırlığı için bu

değeri 8.4 olarak alıyoruz. Bu değer her bir motorun çekeceği akımı göstermektedir. Tüm motorların çekeceği toplam akım ise $(8.4*4) = 33.6$ amper'dir. Elde edilen değer toplam kapasiteye böldüğünde uçuş süresi elde edilmiş olur.

$$480 \text{ amper/dakika} / 33.6 \text{ amper} = 14.3 \text{ dakika}$$

Burada hesaplanan 14.3 dakika değeri daha önce de belirtilen hover, havada asılı kalma değeridir. Toplam uçuş süresi bu değer ile uyuşmayabilir. Uçuş süresi, uçuş alışkanlıklarına, yapılan manevralara, irtifaya, rüzgar, yükseklik gibi çevresel etmenlere de bağlı olarak değişiklik göstermektedir. Online araçlar kullanarak yapılan hesaplarda da benzer uçuş süreleri elde edilmiştir.



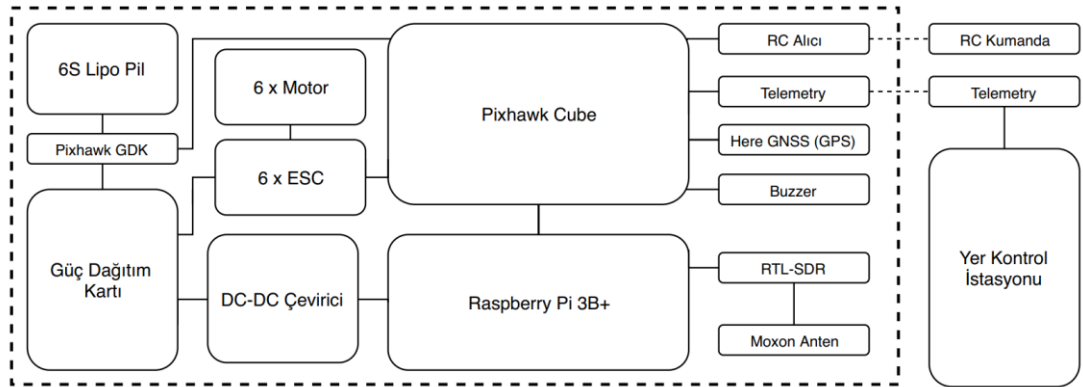
Şekil 5. Hava robotunun tüm ekipmanlar monte edilmiş hali

Dikkat edilmesi gereken bir diğer nokta ise ESC, motor sürücü seçimidir. ESC seçiminde de yine Şekil 4'teki motorun kullanım kılavuzundan faydalanıyoruz. Şekil 4'te *Amps* kolonundaki değerler, motorun o satırdaki gerilim ve pervane kombinasyonlarında çektiği akım değerini temsil etmektedir. Bu noktada motorların ihtiyaç duyacağı maksimum akım değerinden en az %25 yukarıda akım değerine sahip bir motor sürücü seçilmesi önerilir. Çünkü motorlar anlık olarak buradaki akım değerlerinin üstüne çıkabilir ve seçilen motor sürücü kayıp ve kalite kaynaklı tam olarak üzerinde yazan akım değerini sağlayamayabilir. Bunlardan dolayı her zaman ihtiyacımızın üzerinde değerlere sahip motor sürücü seçmek avantajımıza olacaktır. Çalışmada, Emax BLHeli 80A motor sürücü seçilmiştir. Motor sürücü seçiminde

dikkat edilmesi gereken bir diğer nokta ise; seçilen motor sürücünün uygulamaya uygun olmasıdır. Helikopter ve çoklu rotor hava araçları için özel tasarlanmış, çalışma frekansı minimum 400Hz ve üzerinde olan motor sürüler seçilmelidir. Sabit kanatlı uçaklar için tasarlanmış motor sürücülerin döner kanatlı hava araçlarında kullanılmaması gerekir. Aksi halde kırım yaşanabilir.

2.2. Hava Robotu Bileşenleri

Hava robotu, elektronik, elektro-mekanik ve mekanik olmak üzere üç alt başlıkta incelenebilecek birçok bileşenden meydana gelmektedir. Şekil 6’da hava robotunun tüm bileşenlerinin diyagramı verilmiştir. Tablo 3’te de hava robotunu oluşturan bileşenlerin ve görevlerinin listesi bulunmaktadır. Bir sonraki bölümde sırasıyla elektronik, elektro-mekanik ve mekanik bileşenler hakkında detaylı bilgi verilecektir.



Şekil 6. Hava Robotu Donanım Diyagramı

Tablo 3. Hava robotunun bileşenleri ve görevleri

Bileşen	Adet	Model	Görev
Hava Robotu			
Gövde	1	ZD550	Hava robotunun iskeletini oluşturmaktadır.
Otopilot	1	Pixhawk Cube	Uçuş kontrol bilgisayarı
GPS	1	Here GNSS	GPS uydularına bağlanarak otopilot için konum bilgisi üretmek
Ana bilgisayar	1	Raspberry Pi 3B+	Otonom uçuşlarda otopilot'u kontrol etmek, yazılım tabanlı radyo ile ölçümler almak
Motor	4	T-Motor MT3515 650KV Fırçasız Motor	Pervanelerin itki oluşturması için gerekli dairesel hareketi sağlamak.
Motor Sürücü	4	Emax BLHeli 80A	Motorları kontrol etmek
Pervane	4	14" Karbon Fiber	Motorların sağladığı dairesel kuvveti dikey itkiye çevirmek
Pil	1	6S 10000 mAh 25C Leopard- Power Lityum Polimer Pil	Hava robotuna ihtiyaç duyduğu enerjiyi sağlamak
Telemetri	1	915 Mhz Alıcı Modül	Yer istasyonuna hava robotunun teknik verilerini göndermek
RC Alıcı	1	Radiolink	RC kumandadan gelen sinyalleri otomatik pilot'a iletmek
Yazılım Tabanlı Radyo	1	RTL-SDR v3 R820T2 RTL2832U	Antenin aldığı sinyalleri I/Q data formatına çevirerek kaydetmek
Anten	1	El yapımı 433 Mhz Moxon Anten	Verici devreden sinyalleri almak

2.2.1. Uçuş Kontrolcüsü

Uçuş kontrolcüsü, üzerindeki uçuş kontrol yazılımı ve içerisinde bulunan dahili veya sonradan bağlanan harici sensörler yardımıyla kullanıcıdan gelen kontrol sinyalleri veya uçuş kontrolcüsünü komuta eden bilgisayardan gelen komutları işleyerek hava robotunun arzu edilen şekilde uçmasını sağlayan, hava robotunun en kritik bileşenidir. Cube uçuş kontrolcüsü (PX4, 2019), daha önceki ismiyle Pixhawk 2.1, Pixhawk projesinin (Pixhawk, Pixhawk, 2019) bir parçasıdır. Cube uçuş kontrolcüsü, NuttX RTOS (Gerçek zamanlı işletim sistemi) üzerinde PX4 (Pixhawk, PX4 AutoPilot, tarih yok) açık kaynaklı uçuş kontrol yazılımı çalışmaktadır. PX4 uçuş kontrol yazılımı, Pixhawk projesinin bir parçası olarak başlamış olsa da şuanda Dronecode projesinin altında geliştirilmeye devam edilmektedir. Dronecode, hava robotları için açık kaynaklı, güvenilir yazılımlar oluşturmak adına Linux Foundation tarafından başlatılan, kar amacı gütmeyen bir organizasyondur.



Şekil 7. Pixhawk2 Cube uçuş kontrolcüsü

Cube uçuş kontrolcüsünün teknik özelliklerine değinecek olursak; Cortex-M4F mimarisi üzerine kurulmuş FPU (Floating Point Unit) destekli, 32bit, 168MHz STM32F427 ana işlemciye sahiptir. Bunun yanı sıra 32 bit STM32F103 failsafe işlemcisi bulunmaktadır. 2 adet ana 1 adet de yedek olmak üzere toplam 3 adet ivmeölçer kontrolcü üzerinde bulunmaktadır. 256 KB RAM, 2MB Flash bellek, 14 PWM çıkışı, UART, I2C, SPI, CAN, PPM, SBUS ve 3.3V ADC gibi birçok bağlantı ve protokol desteği bulunmaktadır. Cube uçuş kontrolcüsü daha çok üreticiler ve ticari sistemler için tasarlandığından dolayı diğer Pixhawk modellerinden daha güçlü ve

stabildir. Hem donanım hem de yazılım olarak açık kaynak kodlu bir üründür. Dünya genelinde geniş bir kullanıcı kitlesine ve topluluk desteğine sahiptir.

2.2.2. Robot İşlem Birimi

Raspberry Pi, küçük boyutları uygun fiyatı ve en geniş topluluk desteği sunması ile dünya genelinde en çok kullanılan tek kart, düşük güçlü bilgisayardır. Raspberry Pi 3B+ (RaspberryPi.org, 2014), Cortex-A53 mimarisi üzerine kurulu 64bit, 4 çekirdek, 1.4GHz hızında çalışan Broadcom BCM2837B0 SoC'ye sahiptir. Bunun yanı sıra 1GB SDRAM, 2.4GHz ve 5GHz frekanslarında çalışan WiFi bağlantısı, Bluetooth BLE bağlantısı, Gigabit Ethernet bağlantısı, 4 adet USB portu, HDMI bağlantısı gibi birçok kabiliyetleri bulunmaktadır. Raspberry Pi, sistemimizde ana kontrol bilgisayarı olarak kullanılmaktadır.



Şekil 8. Raspberry Pi 3B+

2.2.3. Küresel Konumlama Sistemi

Küresel Konumlama Sistemi (GPS - Global Positioning System), ABD menşeli küresel konumlama sistemidir. GPS'in yanısıra Avrupa menşeli Galileo, Rusya menşeli GLONASS ve Çin menşeli BeiDou uydu tabanlı küresel konumlandırma sistemleri de mevcuttur. GNSS (Global Navigation Satellite System), küresel seyrüsefer uydu sistemi bu sistemlerin genel ismidir.

Hava robotunda, uçuş kontrolcüsünün üreticisi ProfiCNC'nin ürettiği, Pixhawk ile birlikte çalışan yüksek hassasiyetli Here2 GNSS kullanılmıştır. Here2, ARM Cortex-M4 tabanlı 32bit 72 MHz STM32F302 işlemciye sahiptir. İçerisinde ivmeölçer, jiroskop, pusula ve barometre sensörleri bulunmaktadır. UART, I2C ve CAN protokolleri ile çalışabilmektedir. Üzerinde bulunan buton ise güvenlik butonu olarak çalışarak hava robotu üzerinde ekstra bir buton konumlandırmaya gerek bırakmamaktadır. Ayrıca yukarıda saydığımız GNSS sistemlerinin hepsi ile çalışabilmektedir.



Şekil 9. Here2 GNSS

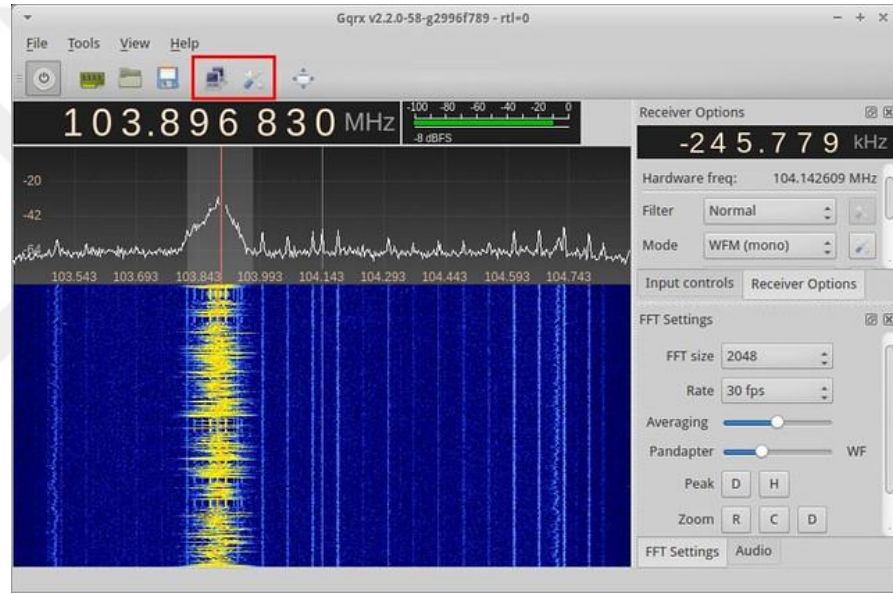
2.2.4. Yazılım Tabanlı Radyo

Yazılım tabanlı radyo (SDR - Software Defined Radio) dinlenen frekans aralığı ve frekans modülasyonu dinamik veya gömülü bir yazılım ile kontrol edilebilen küçük radyo alıcı donanımlardır.



Şekil 10. RTL-SDR

Yazılım tabanlı radyolar, klasik radyolar ile kıyaslanamayacak kadar geniş frekans aralığına sahiptir ve daha fazla modülasyon türünü desteklemektedir. Örneğin çalışmada kullanılacak yazılım tabanlı radyo (RTL-SDR) 500 KHz ile 1.7 Ghz arasında çalışmaktadır. 8bit ADC ve 2.4MHz (stabil) band genişliği sunmaktadır. RTL-SDR ile sinyalleri görüntülemek için Ubuntu işletim sistemi üzerinde *Gqrx* yazılımını kullandık. *Gqrx*'in kurulum bilgilerine EK 1'den ulaşabilirsiniz. Yazılım tabanlı radyolar yukarıda bahsettiğimiz frekans aralığı ile sınırlı değildir. Daha geniş frekans aralığına, daha yüksek ADC çözünürlük değerine sahip yazılım tabanlı radyolar da mevcuttur.



Şekil 11. Gqrx programının ekran görüntüsü. RTL-SDR'dan alınan verilerin anlık gösterimi için kullanılmaktadır.

Bunun yanı sıra yazılım tabanlı radyoların sadece alıcı değil aynı zamanda verici olarak kullanılabilen modelleri de mevcuttur. Bu modeller istenilen frekans aralığında istenilen içerikte yayın yapmasına, alınan sinyalleri kopyalayarak aynısını üretmesine de olarak vermektedir.

2.2.5. Motor Hız Kontrolcüsü

Hava robotunda kullanılan Emax BL-Heli ESC (Electronic Speed Controller) elektronik hız kontrolcüsü, bir diğer adıyla motor sürücüsü, özellikle döner kanatlı

platformlar için tasarlanmış. Motor sürücü, 80A sürekli akım geçişi 10sn için 100A akım geçişi değerleri ile 6S (25,2V)'a kadar çalışma gerilimi bulunmaktadır. Motor sürücüsü, RC kumanda veya servo test cihazları veya PWM sinyali üreten farklı cihazlar ile programlanabilmektedir. Program aracılığıyla motor çalışma yönü değişimi, düşük gerilim, sinyal kaybı, yüksek sıcaklık koruması gibi özellikleri programlanabilmektedir. Çalışma frekansı, frenleme ayarlaması ve çalışma eğrisi seçimi gibi gelişmiş özellikleri de bulunmaktadır (Emax, 2019).



Şekil 12. Emax BL-Heli 80A motor sürücü

2.2.6. Telemetri

Pixhawk'ın kendi telemetri sistemini ülkemizde temin etmek mümkün olmadığından, markette kolayca bulabileceğimiz 915MHz telemetri çifti kullanılmıştır. Sıklıkla kullanılan 433MHz telemetrisinin yerine 915MHz'de çalışan modelini seçmemizin sebebi, ölçüm yapan sistemimizin de 433MHz frekans bandında çalışmasından dolayı olası gürültülerin, örtüşmelerin önüne geçmektir.



Şekil 13. 915 MHz telemetri

2.2.7. Fırçasız Motor

Hava robotunda hobi ve havacılık sektöründe sıklıkla tercih edilen Tiger Motor MT3515 650KV kullanılmıştır. Motor yüksek verim ve düşük titreşim ile çalışarak üst düzey uçuş deneyimi sağlamaktadır. Ayrıca Şekil 4'te daha önce paylaşmış olduğumuz katalog değerlerine sahiptir.



Şekil 14. T-Motor MT3515 650KV fırçasız motor

2.2.8. Gövde

Hava robotunda, robotun iskeletini oluşturan gövde olarak Çin menşeli ZD550 karbon fiber gövde kullanılmıştır.



Şekil 15. ZD550 karbon fiber gövde

Şekil 15’de görüldüğü üzere katlanabilir kanatlı yapıya sahip bir gövdedir ancak katlama noktalarının altlarına motor sürücülerimizi konumlandırduğumuz için bu özelliği iptal edilmek durumunda kalmıştır. ZD550 kaliteli bir gövde olmamakla birlikte bir sonraki bölümde değineceğimiz bazı dezavantajları da bulunmaktadır. Bununla birlikte ülkemizde temin edilebilirlik, fiyat ve performans açısından bu çalışma için en uygun seçenek olarak karşımıza çıkmaktadır. ZD550, 630gram ağırlığa ve 550mm kol açıklığına sahiptir. En fazla 15inch pervane kullanılmasına olanak vermektedir. Demonte olarak temin edilip montajı yapılmaktadır.

2.3. Hava Robotu Montajı

Şekil 15’de görülen ZD550 gövdesi 2 adet 16 mm çapında iniş ayağına sahiptir. İniş ayakları ve gövde karbon fiber malzemeden oluşmaktadır ancak ayakları gövdeye bağlayan malzeme ince plastiktir.



Şekil 16. (a) Hava robotunun önden görünüşü. (b) Hava robotunun üstten görünüşü

Şekil 5’te görülen montajı tamamlanmış hava robotu 4kg’a yakın ağırlığa sahiptir. Bu ağırlık en hafif inişlerde dahi bileşim noktasındaki plastik malzemeyi çatlatmakta ve hasara sebep olmaktadır. Bundan dolayı ZD500’nin mevcut iniş ayakları yerine Şekil 16 (a)’da kanat altlarında görülen turuncu parçalar tasarlanarak üç boyutlu yazıcı aracılığıyla üretildi. İniş ayağı olarak da ekstra ağırlık taşıma problemi olmadığı için karbon fiber malzeme yerine kolay temin edilebilir, işlenmesi kolay, dayanıklı ve

karbon fibere oranla 1/20 oranda daha ucuz olan 20mm PVC boru iniş ayağı olarak tercih edildi



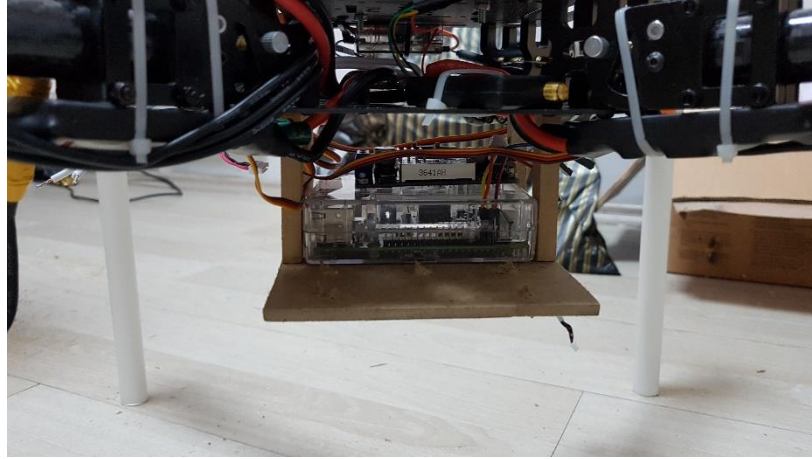
Şekil 17. İniş ayakları gövde montaj parçası

Şekil 5’de montajı tamamlanmış olarak görülen motorun pervane bağlantı aparatı 6mm mil çapına sahiptir. Pervane bağlantı mili boydan boya vida olarak işlenmediğinden, bu durum pervanelerin tam sıkılamamasına sebep olmaktadır. Bundan dolayı elimizdeki pervaneleri bağlamak için müdahalede bulunmamız gerekti.



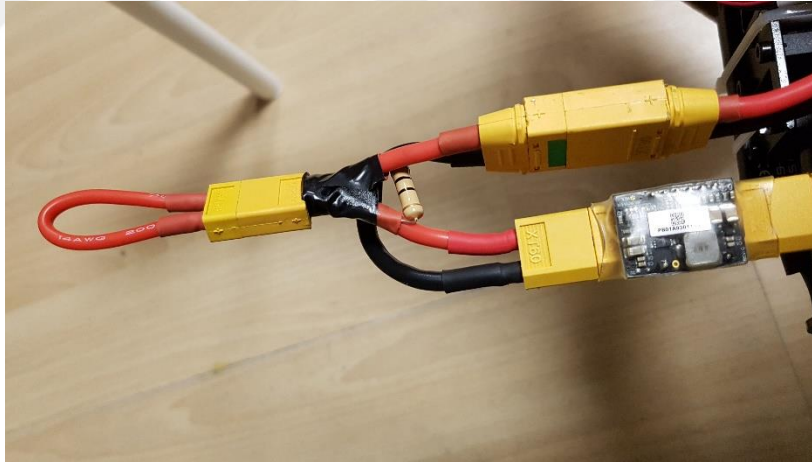
Şekil 18. Pervane adaptörü ve pulun montajlı hali

Pervaneleri motor bağlantı deliklerinin 4mm olan çaplarını motorun bağlantı aparatına uygun olarak masa matkabı ile 6mm’ye genişlettik. Sonrasında motor bağlantı aparatının milindeki boşluğu gidermek için iç çapı 6mm, uzunluğu 5mm pullar torna ile üretildi. Üretilen pullar boyandıktan sonra Şekil 18’deki gibi montajı tamamlandı.



Şekil 19. Hava robotunun alt bölümü

Mekanik problemler tamamlandıktan sonra hava aracının görevini yerine getirebilmesi için gereken Raspberry Pi, RTL-SDR ve Moxon Anten bileşenlerini taşıyacak bir bölme tasarlanıp ahşap malzemeler kullanılarak üretilerek, Şekil 19’da görülen hava robotunun alt tarafına montajı tamamlandı.



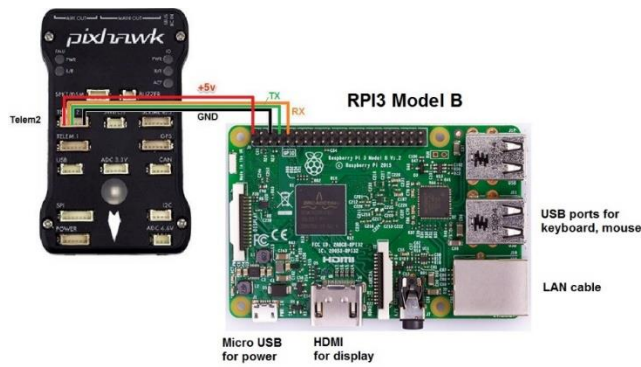
Şekil 20. Kıvılcım önleyici devre

Ohm kanununa göre, hava robotunda kullanılan 6S Lipo pilin gerilim değeri olan 25.2V, motor sürücülerdeki kondansatörlerin iç direnç değerine bölüldüğünde kondansatöre uygulanan anlık akım değerinin 500A’ın üzerinde olduğu görülmektedir. Bu da hava robotunda kullanılan 6S (25.2 V) pillerin sisteme her bağlandığında meydana gelen kuvvetli kıvılcımların sebebinin açıklanmaktadır. Bu kıvılcımlar enerji hattındaki konnektörlerin metal yapılarında hasara sebep olarak görevlerini olması gerektiği gibi yapamamasına ve hatta kırığa sebep olmaktadır. Bunun önüne geçmek

adına Şekil 20’de görülen, enerji hattına seri bağlı 10 Ohm 2W bir direnç ve bu dirence paralel kısa devre konnektörü kullanılarak kıvılcım önleyici devre oluşturulmuştur.

Şekil 20’de sol tarafta görülen XT60 kısa devre konnektörü sisteme enerji verileceği zaman takılı olmaması gerekmektedir. Hava robotuna pili bağladıktan sonra akım, direnç üzerinden geçerek motor sürücülerin üzerindeki kondansatörleri şarj etmeye başlayacaktır. Burada kullanılan direnç, devredeki toplam direnci mili Ohm seviyesinden, Ohm seviyesine yükselterek kondansatörlerin maruz kaldığı akımı 500A’den 2,5A’e düşürmekte ve kıvılcımları önlemektedir. Pili bağladıktan sonra kısa devre konnektörü de bağlanarak devre tamamlanır. Kısa devre konnektörünü bağladıktan sonra, akım her zaman direncin düşük olduğu yolu takip edeceği için kısa devre konnektörü üzerinden akmaya başlayacaktır.

Raspberry Pi ve Pixhawk arasında haberleşmeyi sağlamak adına Şekil 21’deki bağlantı oluşturulmuştur. Şekil 21’deki görsel Pixhawk 1’e ait olduğu için, hava robotunda kullanılan Cube uçuş kontrolcüsü pinlerinin bağlantı şeması sitesinden temin edilerek bağlantılar gerçekleştirilmiştir. Burada dikkat edilmesi gereken, Şekil 21’de gösterilen bağlantı Raspberry Pi’i, Pixhawk üzerinden enerjilendirme üzerine kurulu devreyi göstermektedir. Fakat Pixhawk’ın Raspberry Pi’e GPIO pinleri üzerinden sağlayacağı akım, Raspberry’i çalıştırmaya yetmemektedir. Bunun için Şekil 21’de kırmızı ile gösterilen VCC bağlantısını kullanmayarak Raspberry Pi harici olarak beslenmiştir.



Şekil 21. Pixhawk ve Raspberry Pi arasındaki bağlantı (Ardupilot, Ardupilot, 2019)

Son olarak Cube uçuş kartı ile bağlantı konnektörleri uyuşmayan telemetriyi uçuş kontrolcüsüne bağlamak için uygun konnektörler kullanılarak bağlantı kablosu yapılmıştır.

3. SENSÖR MODELLEME

Hedef konumlandırma görevinde, hedef üzerindeki radyo vericiden yayılan sinyallerin robot üzerindeki yönlü anten ve yazılım tabanlı radyo aracılığıyla okunarak alınan sinyal verisinin işlenmesi önemli bir adımdır. Bu işlem için öncelikle, alıcı olarak kullanılan SDR-anten sisteminin modellenmesinin yapılması gerekmektedir.

3.1 Modelleme için Tasarlanan Sistem

Sensör model sisteminin amacı; SDR-anten sistemini modellemek, antenin karakteristiğini, hassasiyetini belirlemek, doğrudan hava robotunu kullanmadan da ölçüm sistemi üzerinde çalışma yapabilmektir. Bunun için alıcı, verici ve kontrol devreleri tasarlanmıştır. Sensör modelinin oluşturulmasında kullanılan bu devreler bir sonraki bölümlerde detaylı olarak açıklanacaktır.

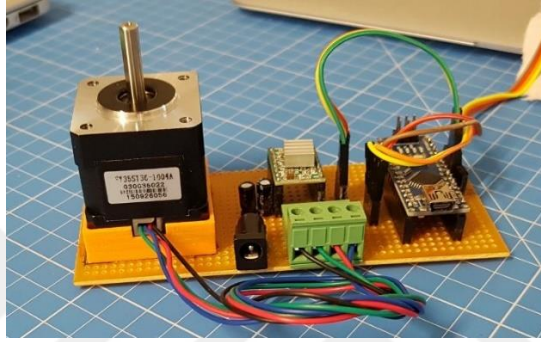
3.1.1. Sinyal Alıcı Devresi

Sinyal alıcı devresi, verici tarafından yayılan sinyali daha önce belirlenen mesafelerden farklı açılarda veya devamlı dönüş yöntemi ile kaydetmektedir. Bilgisayar üzerine kaydedilen bu sinyalin analizi ile antenin karakteristiği elde edilmektedir. Sistem, Tablo 4’de yer alan bileşenlerden oluşmaktadır.

Tablo 4. Sensör modelleme alıcı devresi bileşenleri

Bileşen	Model
Ana Bilgisayar	Lenovo Yoga2
Mikrodenetleyici	Arduino Nano
Step Motor Sürücü	A4988
Step Motor	Nema 14
Yazılım Tabanlı Radyo	RTL-SDR v3 R820T2 RTL2832U
Dönüştürücü	SMA-BNC
Konnektör	BNC
Anten	Moxon – 433MHz

Sensör modellemede kullanılan sinyal alıcı devresi Şekil 22’de gösterilmiştir. Devre, delikli plaket üzerine kurulmuştur. Mikrodenetleyici, motor sürücü ve step motordan meydana gelmektedir. Step motorun elektronik devreden ayrı olmaması, sistemin tek bir modül halinde çalışabilmesi için önem arz etmektedir. Step motorun delikli plaket üzerine sabitlenmesini sağlayacak ara eleman AutoDesk Fusion 360 yazılımı kullanılarak tasarlanmış ve üç boyutlu yazıcı kullanılarak üretilmiştir.

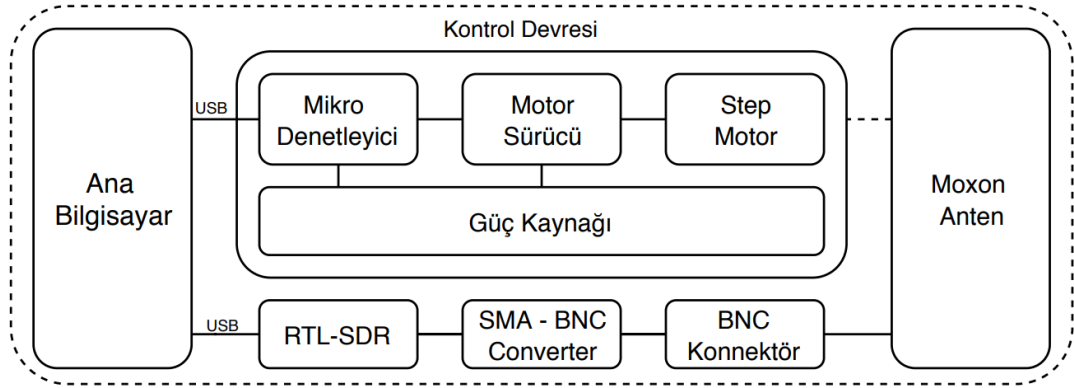


Şekil 22. Sensör modelleme alıcı devresi

Şekil 23’de gösterilen sistemin çalışmasından ve birleşenlerin görevlerinden bahsedecek olursak, ana bilgisayar, sensör modelleme yazılımını ve ona bağlı alt programları çalıştırmak, SDR ile alınan ölçümleri saklamak ve kontrolcü devresiyle haberleşmeyi sağlamakla görevlidir. Mikrodenetleyici, ana bilgisayar ile haberleşmeyi sağlayıp programdan gelecek veriler doğrultusunda motor sürücüyü ve dolayısıyla antenin dönüşünü sağlayan motoru kontrol etmekten sorumludur. SDR’in görevi ise ana programdan aldığı frekans, modülasyon, ölçüm süresi gibi parametreler doğrultusunda Moxon anten’den aldığı sinyalleri I/Q veri formatında USB üzerinden ana bilgisayara iletmektir. Analitik sinyal olarak da isimlendirilen I/Q data, kompleks değerli (imaginary) sinyalleri ifade etmek için kullanılan bir veri türüdür. (Kirkhorn, 1999) Şekil 31’de görülen dönüştürücü ve konnektör ise SDR ile Moxon anten arasında uygun bağlantıyı sağlamaktadır.

Ana bilgisayar üzerinde Python kullanılarak kodlanan sensör modelleme yazılımı çalıştırıldığında (sensorModel.py), kullanıcıyı Şekil 24’de verilen ekran

karşılmaktadır. Program, kullanıcıdan yapmak istediği ölçüm türüne ait parametreleri girmesini beklemektedir.



Şekil 23. Sensör modelleme alıcı sistem yapısı

Program, eğer belirli açılarda duraklayarak ölçüm yapmak isteniyorsa, yapılacak toplam ölçüm sayı bilgisinin, sürekli dönüş yöntemi ile ölçüm yapılacaksa da toplam ölçüm süresinin kullanıcı tarafından girilmesini beklemektedir. Programın gerek duyduğu parametreler girildikten sonra, Şekil 25’de yazılım modülleri arasında yapıya uygun olarak çalışmaya başlayacaktır.

```

sayment@sayment: ~/Desktop/Tez/tez_sensor_model
sayment@sayment:~/Desktop/Tez/tez_sensor_model$ python olcum_guncel.py

S-V-A-Y-M-E-N-T  M-O-D-E-L-L-E-M-E
Istanbul Medeniyet University Sensor Modeling Software

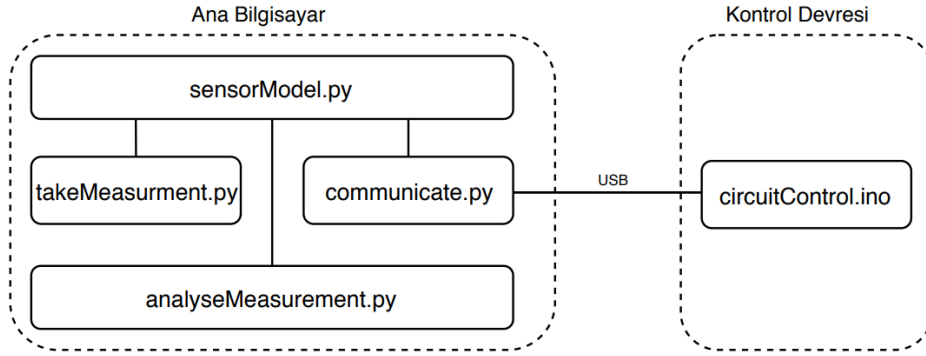
Yapmak istediğiniz testi seçin:
[+] 1- Adetli Ölçüm
[+] 2- Dairesel Ölçüm

```

Şekil 24. Sensör modelleme yazılım ekran görüntüsü

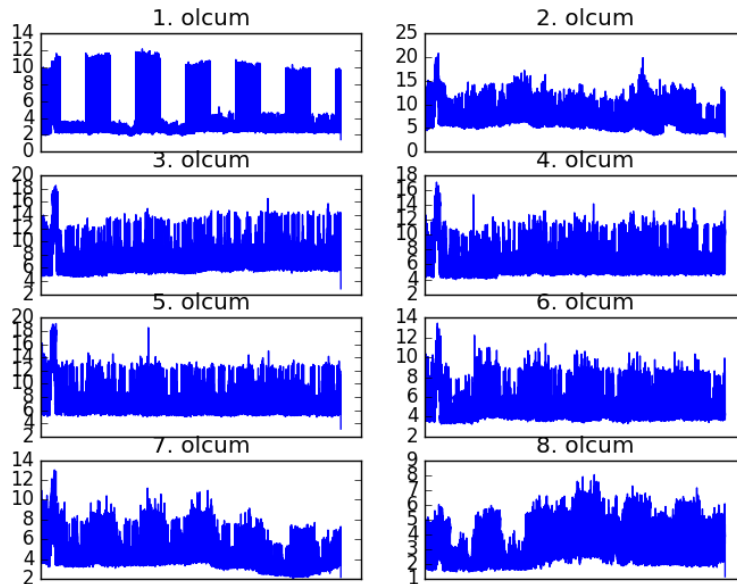
Çalışmaya başlayan program, kontrol devresi ile haberleşmeden sorumlu alt programı (communicate.py) kullanıcıdan aldığı dönüş sayısı, dönüş açısı, açısal hız gibi bilgiler ile çağırarak kontrol devresi üzerindeki mikrodenetleyiciyi sistemin çalışmaya

başladığına dair bilgilendirecektir. Kontrol devresinden onay sinyali aldıktan sonra sensör modelleme programı (sensorModel.py), ölçüm almak ile görevli programı (takeMeasurement.py) kullanıcıdan aldığı parametreler ile çağırarak SDR ile ölçümün alınmasını sağlayacaktır.



Şekil 25. Sensör modelleme yazılım modülleri arasındaki ilişki

İlk ölçüm tamamlandıktan sonra sensör modelleme programı kontrol devresini ilk ölçümün tamamlandığı, ikinci ölçüm için uygun pozisyona geçmesi için gereken verileri içeren paketi mikrodenetleyiciye gönderip, step motorun ölçüm için gereken ikinci konuma geçip onay kodunu içeren paketi geri göndermesi için bekleyecektir.

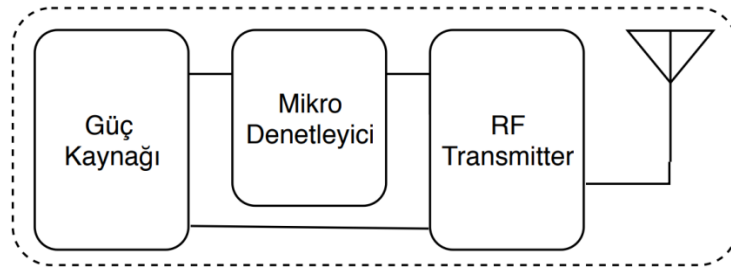


Şekil 26. 45 derece aralıkla alınan ölçüler. Hedef 0 dereceye gelecek şekilde alıcıdan 150 metre ileriye yerleştirilmiştir.

Aynı şekilde istenilen sayıdaki ölçümler tamamlandıktan sonra ana program ölçümleri analiz edecek programı (analyseMeasurement.py) çağırarak kaydedilen tüm ölçümleri grafiğe dökerek Şekil 26'dakine benzer çıktılar üretecektir.

3.1.2. Radyo Verici Devresi

Radyo verici devresi, mikrodenetleyici ve mikrodenetleyiciden gelen verileri kodlayıp radyo frekansına çeviren verici modülünden oluşmaktadır. Radyo verici sisteminin yapısı Şekil 27'de görülmektedir.



Şekil 27. Radyo verici sistem yapısı

Mikrodenetleyici olarak Arduino Nano, verici modülü olarak da üzerinde PT2262 encoder barındıran 433Mhz RF verici kullanılmıştır. Modül 18cm anten uzunluğuna sahiptir ve kullanım kılavuzuna göre uygun besleme ile 1 km'ye kadar kapsama alanı sağlamaktadır. Ancak saha deneylerinde 200 metreden sonra gönderilen sinyal bozulmaktadır. Ayrıca ASK ve OOK modülasyonlarını da desteklemektedir. Şekil 28'de görülen radyo verici devresi, breadboard üzerinde kullanılmıştır.



Şekil 28. Hedef olarak kullanılan radyo verici devresi

Verici devresinde mikrodenetleyicinin görevi sadece verici modüle periyodik sinyal göndermektir. Kullanılan modülasyon ve gönderilen verinin içeriği önemli değildir. Radyo verici sistemi, 4 Hz frekansında periyodik sinyal göndermektir.

3.2 Moxon Anten

Moxon anteni Les Moxon tarafından geliştirilen, temelde 2-elemanlı Yagi'ye benzeyen yönlü bir anten çeşididir. Moxon anten, elemanları birbirine kıvrılmış Yagi'ye benzetilse de, Yagi antene göre şu özelliklerinden (K8JHR, 2014) dolayı öne çıkmaktadır:

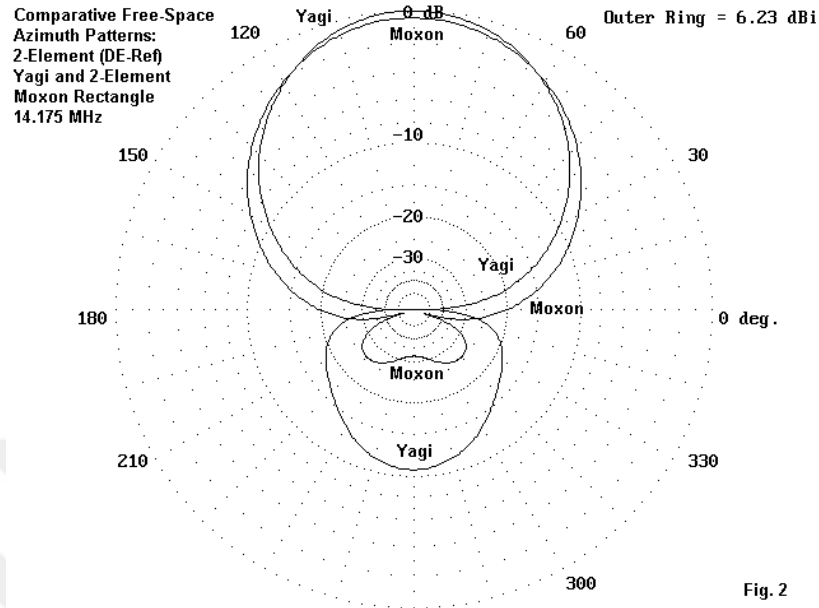
- 2-eleman Yagi'ye göre %33 daha küçüktür ve oldukça hafiftir.
- Temin etmesi kolay ve ucuz olan, bakır kablo gibi malzemeler kullanılarak dahi üretilebilir.
- Yatay veya dikey olarak kullanılabilir.
- Anteni ön ve arka tarafı arasında yönlülüğe ciddi katkıda bulunan, ortalama 25dB sinyal seviyesi farkı bulunmaktadır. Bu fark anten karakteristiğini de gösteren Şekil 29'da görülebilir
- Ekstra regülatöre gerek duymadan çalışabilen 50 ohm çıkışa sahiptir.
- Eğer yapım aşamasında antenin boyut bilgilerine uygun üretildiyse herhangi bir ayar veya konfigürasyon gerektirmemektedir.

Çalışmamızda Moxon anteni seçmemizdeki etkenler, kolaylıkla ve düşük maliyetle üretilebilmesi, küçük ve hafif olmasıdır.

Moxon anten yapımında en önemli nokta, anteni meydana getiren elemanların uzunluklarının ve bunlar arasındaki mesafelerin doğru şekilde ayarlanmasıdır. 433Mhz'de çalışacak antenin ölçülerinin belirlenmesinde 'Moxon Rectangle Generator' isimli programdan faydalanılmıştır. (Ac6la, 2019)

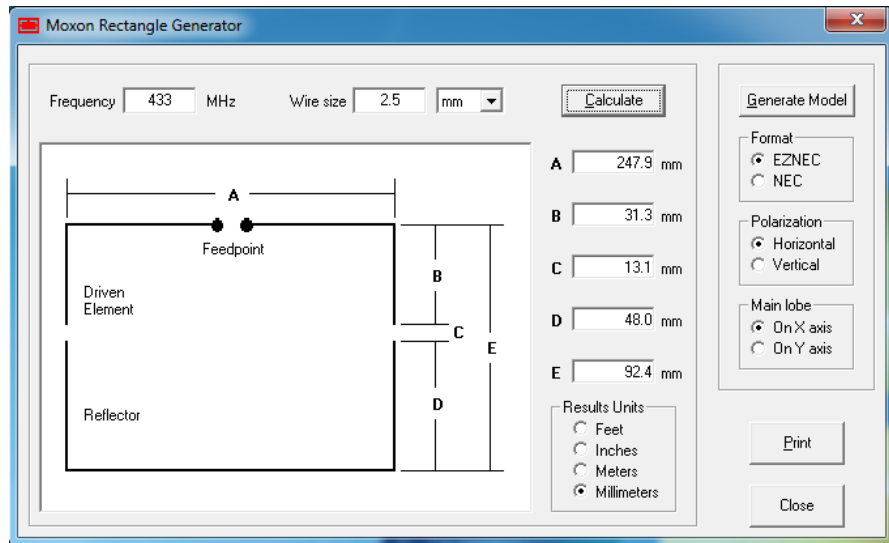
Program yardımıyla anten ölçüleri belirlendikten sonra 2.5mm bakır tel kullanılarak ölçülere uygun olarak üretilmiştir. Hafif olması için karton blok üzerine oturtulmuştur. Yapımı tamamlanan anten, Şekil 30'daki *Feedpoint* olarak gösterilen noktalardan BNC konnektöre bağlanmıştır. Bu noktalar antenin yönünü göstermektedir. Kullanılan

SDR'ın anten girişı SMA olduğundan dolayı BNC-SMA dönüştürücü kullanılarak antenin SDR'a bağlantısı tamamlanmıştır.



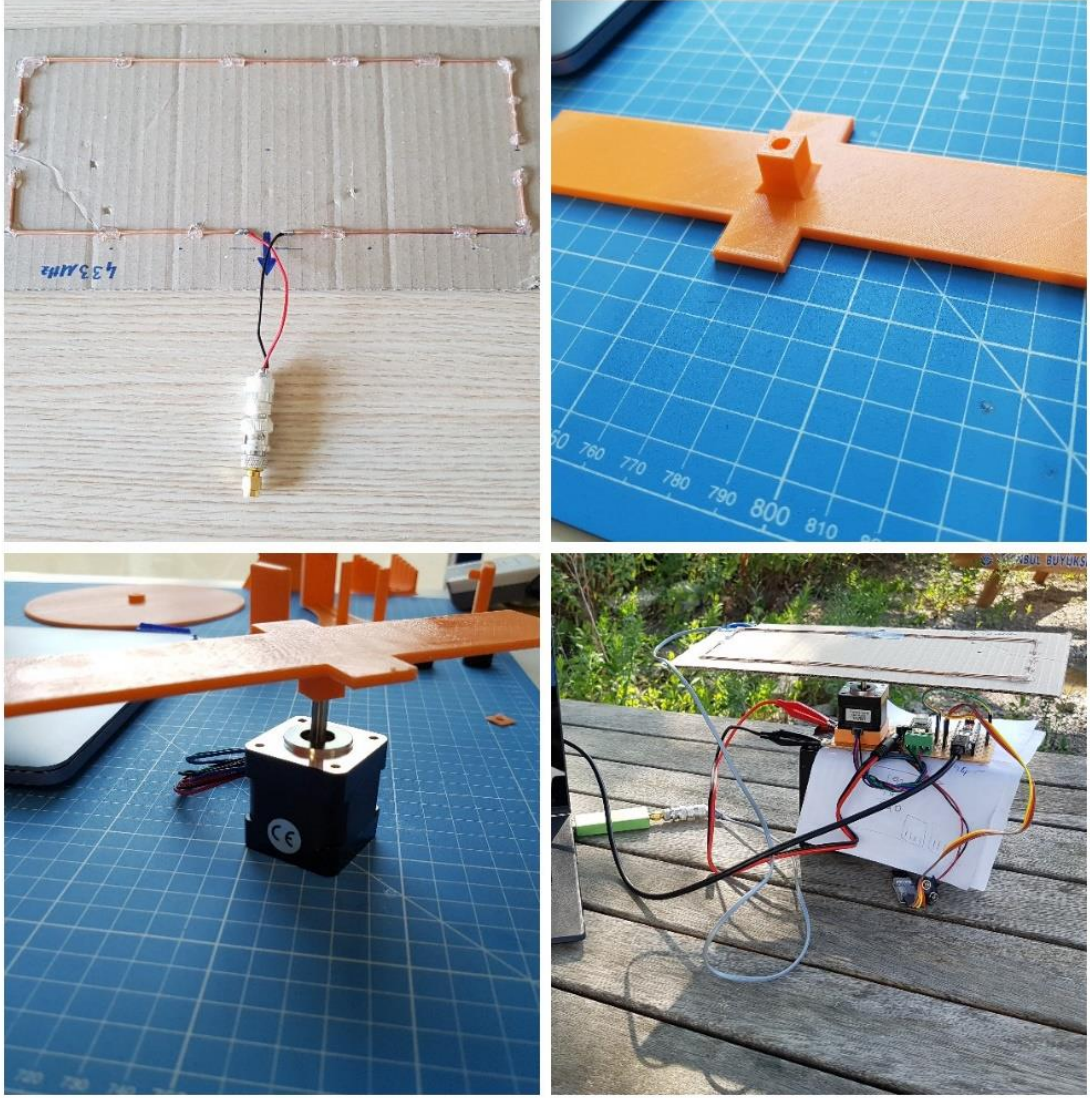
Şekil 29. Örnek Moxon ve Yagi anten karakteristik grafiđi (Cebik, 2000)

Yapımı tamamlanan antenin sensör model devremizin kontrol ettiđi step motora bağlantısını sağlamak için, step motorun mili ile anten arasında bağlantıyı sağlayacak bir ara modül, AutoDesk Fusion 360 yazılımı kullanılarak tasarlanmıştır. Tasarlanan parça üç boyutlu yazıcı ile üretilerek anten ile motor arasındaki bağlantı tamamlanmıştır.



Şekil 30. 433 Mhz Moxon anten ölçüleri (Acbla, 2019)

Yapımı tamamlanan antenin sensör modelleme devremizin kontrol ettiği step motora bağlantısını sağlamak için, step motorun mili ile anten arasında bağlantıyı sağlayacak bir ara modül, AutoDesk Fusion 360 yazılımı kullanılarak tasarlanmıştır. Tasarlanan parça üç boyutlu yazıcı ile üretilerek anten ile motor arasındaki bağlantı tamamlanmıştır.



Şekil 31. (Sol Üst) BNC ve SMA konnektör bağlı Moxon anteni, (Sağ Üst) 3 boyutlu yazıcı ile üretilen motor-anten ara bağlantı parçası, (Sol Alt) motor üzerine bağlanmış ara bağlantı parçası, (Sağ Alt) çalışmakta olan sensör modelleme devresi.

3.3 Sensör Verisi Toplama

Sensör modelleme sistemi kullanılarak veri toplama işlemi sonucu elde edilecek veriler ile anten karakteristiğinin elde edilmesi, toplanan verileri işleyerek doğrultu açısı elde eden programın geliştirilmesi ve bu işlemlerin hava robotu uçurmaya gerek kalmadan bağımsız olarak yapılabilmesi amaçlanmıştır.

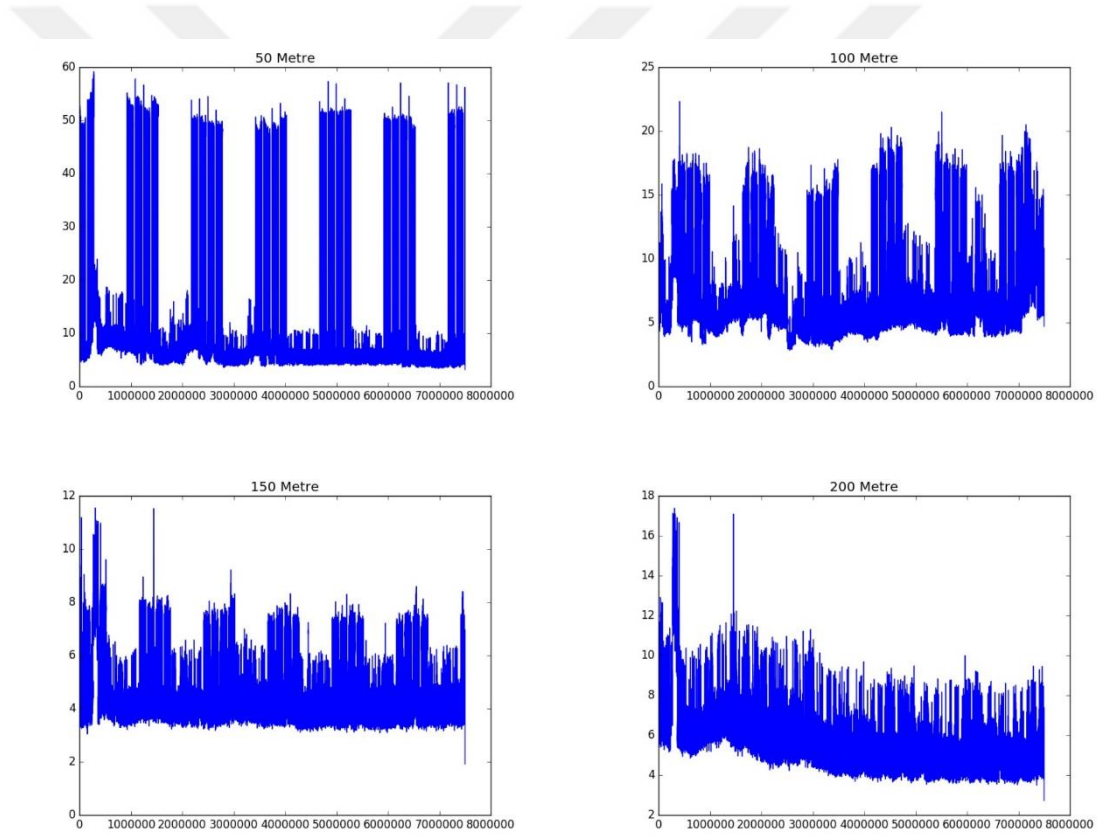


Şekil 32. İstanbul Medeniyet Üniversitesi Kuzey Kampüsü ölçüm noktaları.

Veri toplama işlemi için öncelikle Bölüm 3.1.2’de detayları verilen verici devresinin çalıştırılması gerekmektedir. Daha sonra Bölüm 3.1.1’de açıklanan alıcı devresi USB üzerinden Linux tabanlı bilgisayara bağlanmalı ve devreye üzerindeki güç bağlantı soketi kullanılarak enerji verilmelidir. Alıcı devresini kontrol eden yazılım, Linux üzerinde geliştirilip, işletim sistemi bağımlı komutlar çalıştırdığından devrenin bağlandığı bilgisayarın Linux tabanlı olması gerekmektedir. Alıcı devresini kontrol eden sensör model yazılımı (sensorModel.py) terminal üzerinden çağırıldığında Şekil 24’de görülen arayüze sahip program çalışmaktadır. Program iki çeşit ölçüm yapmaya izin vermektedir. Birincisi, hava robotunda da kullanılan ölçüm yöntemini modelleyen adetli ölçüm modeli, diğeri ise dairesel ölçüm modelidir. Adetli ölçüm modelinde sistem, ilk doğrultuda ölçüm aldıktan sonra, kullanıcıdan alınan ölçüm adedine bağlı olarak hesaplanan açı kadar dönüş yaparak bir sonraki doğrultuya geçmekte ve tekrar ölçüm almaktadır. Sistem bu davranışını bir tam tur tamamlanana kadar devam etmektedir. Dairesel ölçüm modelinde ise sistem, kullanıcı tarafından verilen ölçüm süresi ile motorun dakikadaki tur sayısını (rpm) ve yazılım tabanlı radyonun sinyal kaydı yapacağı süreyi belirler. Antenin hareketi, ölçüm ile aynı andan başlayarak istenilen süre boyunca bir tam tur devam ederek ölçüm tamamlanır.

Veri toplama işlemi Medeniyet Üniversitesi Kuzey Kampüsünde gerçekleştirilmiştir. Şekil 32’de belirtilen konumda verici devresi çalıştırıldıktan sonra uzaklık ile belirtilen konumlarda alıcı devresi ve kontrol yazılımı kullanarak adetli ve dairesel ölçümler alınmıştır. Adetli ölçümlerde α ; $\pi/3$, $\pi/4$, $\pi/6$ alınarak veri toplanmıştır. Adetli ölçümde her bir doğrultudaki ölçüm süresi 3 saniyedir. Dairesel ölçümde ise 5 ve 10 saniye periyotlu ölçümler alınmıştır. Ölçümler, anten doğrultusu vericiye dik konumda başlamaktadır.

Şekil 33’de gösterilen konumlarda sırasıyla vericiden 50, 100, 150 ve 200m uzaklıklarda alınmış ölçüm sonuçlarını göstermektedir. Ölçümlerde α , $\pi/3$ alınmıştır ve her bir ölçüm 3 saniye sürmüştür.

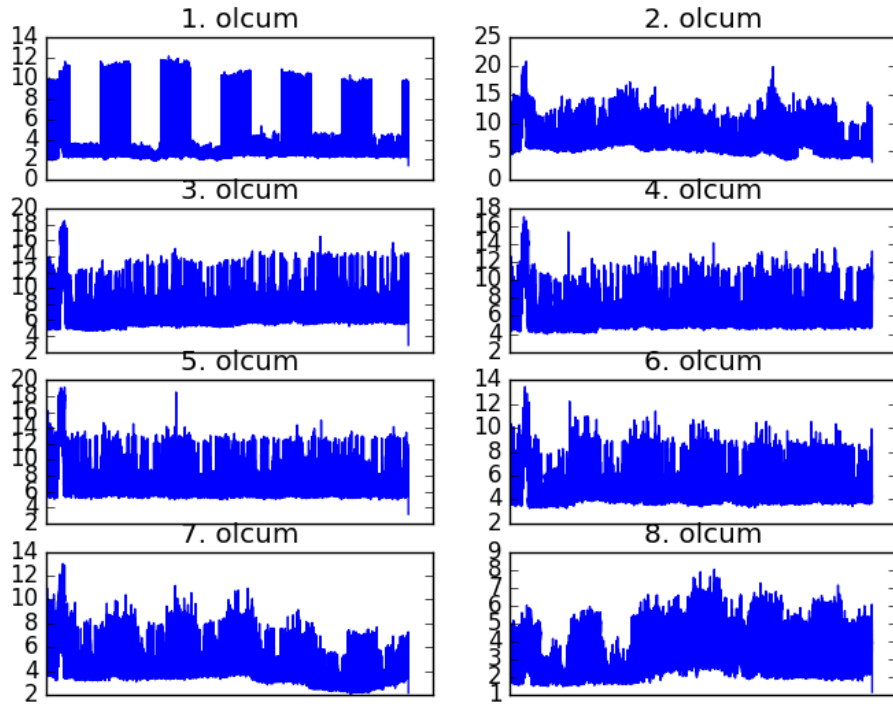


Şekil 33. Mesafeye bağlı ölçüm grafikleri

Şekil 33’e göre vericiden uzaklaştıkça sinyallerdeki bozulmaların arttığı ve alınan sinyalin zayıfladığı gözlemlenmektedir. 150 metreye kadar olan uzaklıklarda alınan ölçümlerde sinyaller açıkça gözükmekte; 200 metrede sinyal kaybolmaya, okunurluğu

düşmeye başlamaktadır. 200m’de ise okunan sinyalden anlamlı veri çıkarmak oldukça zorlaşmıştır.

Bu durum büyük oranda kullandığımız vericinin yeterli gücü sağlayabilecek kalitede olmamasından kaynaklanmaktadır. Daha kaliteli verici kullanılarak mesafe artırılabilir. Bunun yanı sıra kullandığımız yazılım tabanlı radyo sınırlı kabiliyetlere sahip, düşük kaliteli bir üründür. Daha kaliteli ve giriş gücü yüksek yazılım tabanlı radyoların kullanılması da alınan sinyalin kalitesini etkileyecek ve dolayısıyla sistemin performansını artıracaktır. Son olarak sitemde kullandığımız anten el yapımı olduğundan kaynaklı hassas üretilememiştir. Bu da kalite ve verimi düşürmektedir.



Şekil 34. 150 metre mesafeden alınan ölçüm örneği

Anteni, daha yüksek hassasiyet sağlayan üretim teknikleri kullanılarak üretmek hassasiyetini artıracak, daha kalın kablo veya iletkenler kullanmak da antenin gücünü artırarak alınan sinyalin kalitesi ve sistemin verimini artırabilir. Toplanan veriler göstermektedir ki, mevcut verici, yazılım tabanlı radyo ve anten sistemi kullanılarak

200m yarıçapında bir alanda çalışma yapılabilir. Daha uzak mesafelerden alınan sinyaller çıkarım yapılmasını zorlaştıracak ve hata oranını artıracaktır.

Şekil 34’de verici devresine 150 metre mesafeden α , $\pi/4$ değeri için alınan örnek ölçüm sonucu görülmektedir. Anlamlı veri çıkarabildiğimiz en uzun mesafe olan 150 metre mesafede yaptığımız ölçümler sonucunda anten için kullanılacak en sağlıklı α değeri, $\pi/4$ ’dür.

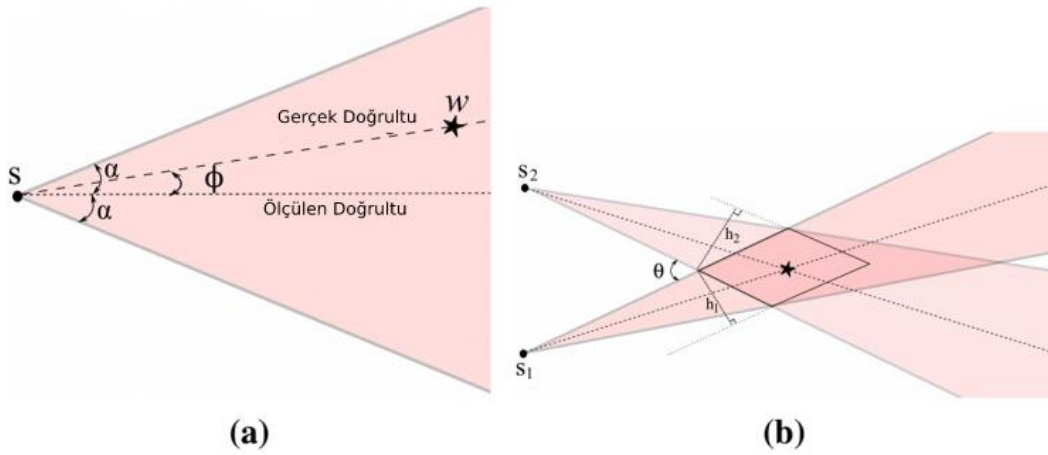


4. YAKLAŞIM

Konumlandırma işlemi için üç farklı algoritma geliştirilmiştir. İlk olarak algoritmalarda kullanılan algılama modeli ve belirsizlik ölçütü formülasyonu verilecektir. Sonrasında, geliştirilen algoritmaların performans analizi için en iyi çevrimdışı stratejinin maliyeti için bir alt sınır ortaya koyulacaktır. Sonraki bölümde geliştirilen algoritmaların detayları verilecektir.

4.1. Algılama Modeli ve Belirsizlik Ölçütü

Doğrultu açısı ölçümü üzerinde gürültü olmaması halinde, sadece iki ölçüm ile hedefin konumu belirlenebilir. Ancak gerçek hayatta tüm ölçümler üzerinde bir belirsizlik ve gürültü bulunmaktadır. Bu nedenle, hedef konumlandırma için geliştirilen yaklaşıma geçmeden önce doğrultu açı ölçümündeki belirsizlik ölçütü ve algılama modeli tanımlanacaktır.



Şekil 35. (a) Algılama modelinin gösterimi. Hedef w konumunda ve ölçüm üçgeninin içinde bulunduğu bilinmektedir. Doğru doğrultu açısı, bilinmeyen ancak en fazla α kadar bozulmuştur. **(b)** İki doğrultu açısı ölçümü kesişiminin paralelkenar ile yaklaşılması. Şekillerde, hedefin konumu yıldız ile gösterilmiştir. (Bayram , Stefas, & Isler, 2018)

Konumlandırma yaklaşımımız doğrultu açısı ölçümleri üzerindeki gürültünün, sınırlı olduğunu varsaymaktadır. Sınırlı gürültü (bounded noise) modeli, sensör modeli tam/doğru olmadığı zaman, rassal modellere göre avantaj sağlar (Isler & Bajcsy, 2006) (Tokekar & Isler, 2013). Bu modelde, sensör, α sınırlı gürültüsü ile bozulmuş doğrultu

açısı sağlar Şekil 35. Gerçek doğrultu ile ölçülen doğrultu arasındaki açı farkı φ , en fazla α kadar olabilir. Hedefin, ölçümün oluşturduğu 2B üçgenin içerisinde olduğu garanti edilmektedir.

Konumlandırma işlemi, ölçümler alındıkça bu ölçümlerin kesişimi alınarak yapılır. Sonuçta oluşan kesişim, konumlandırma belirsizliğini (localization uncertainty) ifade eder. İki doğrultu ölçümünün kesişimi Şekil 35b’de görülmektedir. Şekilde, koyu renkli alan, hedefin içinde bulunduğu alanı göstermektedir.

Ölçüm sayısı arttıkça kesişimin matematiksel gösterimi karmaşıklaşmaktadır. Karmaşıklığı azaltmak için, kesişimi bir paralelkenar (parallelogram) ile yaklaşıklayarak sadeleştirme yapabiliriz. Böylece w konumunda bulunan bir hedef için s_1 ve s_2 yerlerinden alınan iki ölçümün konumlandırma belirsizliği U , paralelkenarın alanı olarak verilebilir. (Bayram , Stefas, & Isler, 2018)

$$U(s_1, s_2, w) = \frac{h_1 h_2}{|\sin\theta|} \approx \frac{(2d(s_1, w)\tan\alpha)(2d(s_2, w)\tan\alpha)}{|\sin\theta|}$$

$$= (2\tan\alpha)^2 \frac{d(s_1, w)d(s_2, w)}{|\sin\theta|}$$

Denklem 1

Burada $d(a, b)$ ile a ve b noktaları arasındaki Öklid uzaklığı ifade edilmektedir. $(2\tan\alpha)^2$ terimi hedef-sensör geometrisine bağlı değildir. Denklemdeki geri kalanı, bir hedefin konumunun iki doğrultu ölçümü ile kestiriminde, belirsizlik ölçmek için kullanılan GDOP (geometric dilution of precision) (Kelly, 2003) formülüne benzemektedir.

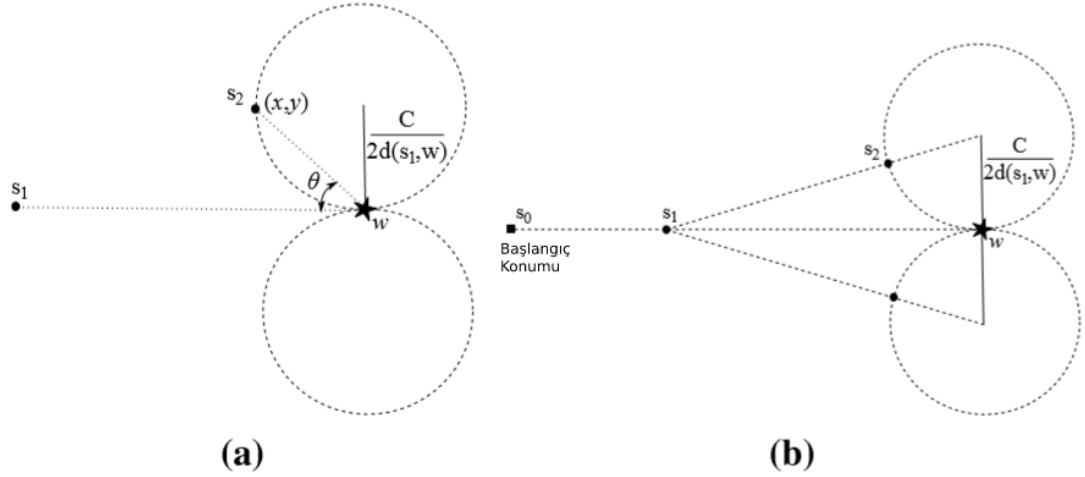
4.2. En iyi Çevrimdışı Strateji

En iyi çevrimdışı (optimal offline) strateji, hedefin konumunun bilindiği durumda hangi konumlardan ölçüm alınarak hedefin istenilen belirsizlik seviyesinde konumlandırılacağını belirleyen stratejidir. Burada bu stratejiyi tasarlamak yerine, bu stratejinin konumlandırma maliyeti için bir alt sınır elde edilecektir. Bu alt sınır,

geliştirilen algoritmaların rekabetçi oranının (competitive ratio) bulunarak performansının analiz edilmesinde kullanılacaktır.

Konumlandırma, alınan ölçümlerle elde edilen ölçüm üçgenlerinin kesişimi ile yapılmaktadır. Sonlu bir kesişim alanı elde etmek için, farklı iki yerden ve hedefe göre doğrudan (colinear) olmayan en az iki ölçümün alınması gerekmektedir. Dolayısıyla, en iyi çevrimdışı strateji de en az iki ölçüm alınmalıdır.

Önerme 1 (Bayram , Stefas, & Isler, 2018) :İlk ölçümün s_1 'den alındığını ve hedefin konumlandırmasındaki belirsizliğin en fazla U^* olduğunu kabul edelim. İkinci ölçüm s_2 'nin de (x, y) koordinatında alındığını ve hedefin $(0,0)$ koordinatına yerleştirildiğini kabul edelim. Denklem 1'deki belirsizlik ölçütü kullanılarak, tüm (x, y) noktaları şu eşitsizlik ile tanımlanır: $x^2 + \left(y \pm \frac{C}{2d(s_1, w)}\right)^2 \leq \left(\frac{C}{2d(s_1, w)}\right)^2$. Burada $C = \frac{U^*}{(2 \tan \alpha)^2}$.



Şekil 36. a) İkinci ölçümün alınabileceği bölge. b) Yeni ölçüm yerleri (Bayram , Stefas, & Isler, 2018)

Önerme 1'deki eşitsizlik ikinci ölçümün alınabileceği ölçüm bölgesini göstermektedir. Bu ölçüm bölgesi, Şekil 36(a)'da görüldüğü üzere yarıçapı $\frac{C}{2d(s_1, w)}$ olan iki disk ile ifade edilebilir.

Önerme 1’de ilk ölçüm yeri ile robotun başlangıç konumunun aynı olduğu varsayılmıştır. Ancak böyle olmaması halinde robotun ilk ölçüm yerine doğru ilerlemesi ve oradan ilk ölçümü alması gerekmektedir. Bu durum Şekil 36(b)’de gösterilmiştir. Robot ilk ölçüm yerinden ölçümü aldıktan sonra, ölçüm bölgesinde kendisine en yakın yere giderek ikinci ölçümü alır.

Konumlandırma maliyeti, hem hareket zamanını hem de ölçüm zamanını içermektedir. En iyi maliyet OPT için, OPT_U hareket ile harcanan zamanı gösterirken, $OPT_ö$ ölçüm almak için harcanan zamanı göstermektedir.

Teorem 1 (Bayram , Stefas, & Isler, 2018): Robot başlangıçta s_0 , hedef ise w konumunda bulunmaktadır. En iyi çevrimdışı çözümün maliyeti $OPT = OPT_U + OPT_ö \geq d(s_0, w) - (\sqrt{2} - 1)\sqrt{C} + 2\tau_ö$ değerinden büyüktür. Burada $\tau_ö$ bir ölçüm için harcanan zamanı göstermektedir.

4.3. Çevrimiçi Stratejiler

Çevrimiçi (online) algoritmalar, probleme dair veri setinin tamamından başlangıçta haberdar olmadan çalışan algoritmalarlardır. Çevrimiçi algoritma, veri geldikçe bir sonraki adımda ne yapacağına karar verir. Buna karşın çevrimdışı algoritmalar ise başlangıçta probleme dair tüm verilere sahip olarak çalışan algoritmalarlardır.

Çevrimiçi algoritmaların performans analizinde, aynı probleme daha önceden bilinen değerler verilerek en iyi çevrimdışı çözüme dair maliyet için bir alt sınır elde edilir. Bu alt sınır, en iyi çözümün ulaşabileceği en düşük maliyeti göstermektedir. Çevrimdışı bir algoritma için alt sınır olan bir değer, aynı problemi çözmeye çalışan çevrimiçi bir algoritma için referans alınabilir. Çevrimiçi algoritmanın performansı, bu algoritma ile elde edilen sonuçların maliyetinin, çevrimdışı algoritmanın alt sınırına oranlanmasıyla tespit edilebilmektedir. Bu değer literatürde *rekabetçi oran* (competitive ratio) olarak ifade edilir.

Çalışmada hedefin konumunun belirlenmesi için doğrultu açısı ölçümlerine ihtiyaç vardır. Bu doğrultu açısı ölçümleri, hedefi içine alan üçgenler vermektedir. Bu üçgenlerin kesişiminden de hedefin konumlandırılması yapılmaktadır. Dolayısıyla

doğrultu açılı ölçümlerinin yapılacağı yerlerin seçimi, konumlandırma performansını etkilemektedir. Ölçüm yerlerinin belirlenmesinde üç çevrimiçi strateji geliştirilmiştir.

4.3.1. Açılırtay Üzerinde Ölçüm

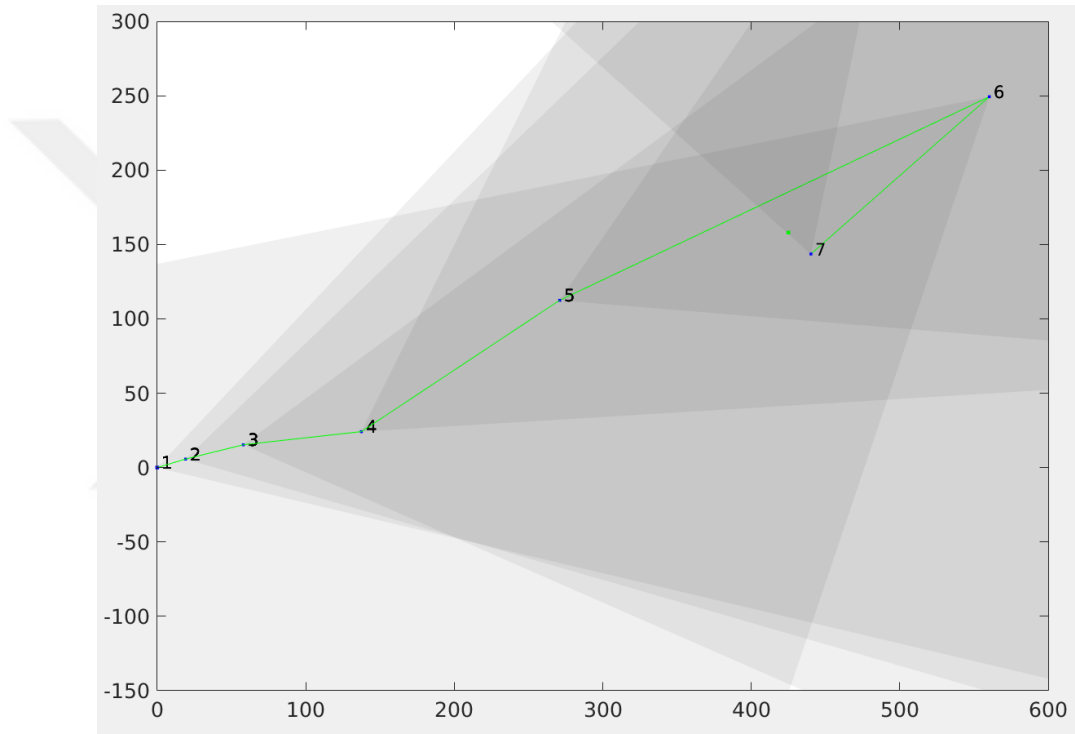
Bu stratejide, doğrultu açılı ölçüm yerlerinin belirlenmesinde, alınan son ölçüm üçgeninin açılırtayı kullanılmaktadır. Şöyle ki; hedefe yaklaşılan kadar açılırtay üzerinde bir önceki adımın iki katı olacak şekilde hareket edilir ve bu noktadan yeni bir ölçüm alınır. Hedef konumlandırılmayıp, ıskalanırsa artık katlama yerine adım uzaklığında yarılama yapılır. Yani artık açılırtay üzerinde önceki adımın yarısı kadar hareket edilir.

Konumlandırma algoritmamızı çalıştırmak için bazı değişkenleri tanımlayıp hedefimize uygun değerler ile programı başlatmamız gerekmektedir. Algoritmada kullanılan bu değişkenleri açıklamak gerekirse; α , ölçüm yapılacak açılı aralığı ile ölçüm üçgeninin tepe açısının belirlenmesinde kullanılmaktadır. Ölçüm üçgeninin tepe açısı 2α değerindedir. α değeri sonuca doğrudan etki etmektedir. α 'nın değerini büyütmek toplam ölçüm sayısını dolayısı ile ölçüme harcanan süreyi düşürecektir. Ancak α 'nın büyümesi ölçüm üçgeni alanının büyümesine sebep olarak ölçümün doğruluğunu azaltacak, ölçüm sayısını artıracaktır. Bu sebeplerden dolayı kullanılan antenin hassasiyeti ve hava robotunun maksimum havada kalış süresi göz önünde bulundurularak sisteme uygun α değeri seçilmelidir.

Tablo 5'de değinilmeyen maksimum mesafe değişkeni ise maksimum ölçüm mesafesini belirlemektedir. Yine bu değişken ve α yardımıyla ölçüm üçgeninin yan kenarları tespit edilmektedir. Her ölçüm noktasında oluşturulan ölçüm üçgeni ile, bir önceki nokta eğer ilk ölçüm noktası ise, ilk noktada oluşturulan ölçüm üçgeninin kesişim noktaları alınarak oluşan kesişim noktaları arasında kalan alan hesaplanır. Bu alan U değişkeni ile ifade edilir. Eğer bir önceki konum ilk ölçüm konumu değilse daha önceki konumlardan elde edilen paralelogram ile mevcut konumda elde edilen ölçüm üçgeninin kesişimleri kullanılarak kesişim alanı hesaplanır. Bu kesişim alanı hedefin içerisinde olması muhtemel alanı temsil etmektedir.

Bir diğerk değışken ise konumlandırma işleminin gerçekleşip gerçekleşmediğinin belirlenmesi için kullanılan, istenilen belirsizlik alanını ifade eden U^* değışkenidir. U değeri U^* değerdinden küçük olduğunda konumlandırma işleminin gerçekleştiğı varsayılarak program sonlandırılır.

Geliştirilen çevrimiçi algoritmaların tümünde katlama-yarılama (doubling-halving) metodu kullanılmıştır. Bu metod hedefe olan doğrultu açısını baz alarak bir sonraki konum ile mevcut konum arasındaki mesafeyi belirlemede kullanılmaktadır.



Şekil 37. Açıortay Üzerinden Çalışan Simülasyon Örneğı

Açıortay üzerinde ilerleyerek çalışan metotta, elde edilen sinyaller işlenerek belirlenen *doğrultu* olduğu gibi kullanılmakta, hava robotu Şekil 37'de de yeşil çizgiler ile ifade edilen ölçüm üçgeninin açıortay'ını *doğrultu* alıp ilerleyerek konumlandırma işlemi yapmaktadır. Şekil 37'yi açıklamak gerekirse; bu deneyde U^* 10 m yarıçapındaki dairenin alanını temsil eden 314 m^2 , minimum ilerleme miktarı 20 metre atanarak katlama stratejisi ile teste başlanmıştır. Şekilde rakamlar ile ifade edilen her bir nokta ölçüm noktalarını temsil etmektedir. Her ölçüm noktası ise ölçüm üçgeninin tepe noktasını teşkil eder. Şekil 37'de görüldüğü gibi her bir noktada oluşturulan ölçüm

üçgenleri farklı yoğunluk seviyeleri ile işaretlenmiş ve kesişimleri ölçüm üçgenlerinden daha koyu olarak ifade edilmiştir. 1. ölçüm noktasından itibaren her bir ölçümde katlama stratejisi uyarınca ilerleme miktarı bir önceki adımın iki katı olarak alınmıştır. 6. adımda ise hedef nokta geçildiği için strateji yarılama olarak değiştirilmiş ve devam eden adımlar bir öncekilerin yarısı olacak şekilde ayarlanmıştır. Anlaşılabilirliği artırmak adına görselde belirtilmemekle birlikte bu işlem, her adımda ölçülen belirsizlik alanının U istenen belirsizlik alanından U^* daha küçük oluncaya kadar devam etmiştir. Şekil 37'deki üçgenler görüldüğü üzere, alınan ölçümler ile oluşturulan ölçüm üçgenlerinin açıortayı doğrultu kabul edilerek, o doğrultu üzerinden stratejinin belirlediği kadar ilerleyerek bir sonraki ölçüm noktası tespit edilmiştir.

Tablo 5. Açıortay Üzerinde Çalışan Algoritma

Algoritma	Açıortay Üzerinde Ölçüm
1:	$metod \leftarrow katlama$
2:	$mesafe \leftarrow MinimumMesafe$
3:	Ölçüm al
4:	Ölçümleri analiz et, doğrultu açısını bul
5:	Ölçüm üçgenini hesapla
6:	Doğrultu açısı üzerinden bir sonraki ölçüm noktasına git
7:	Adım 3-5 tekrarla
8:	Bir önceki ölçüm noktasında elde edilen ölçüm üçgeni ile bu noktadaki ölçüm üçgeninin kesişimlerini al (U)
9:	while $U > U^*$ do
10:	$adım \leftarrow adım + 1$
11:	if $strateji == yarılama$ and $mesafe < MinimumMesafe$ then
12:	$strateji \leftarrow katlama$
	end if
13:	if $doğrultu > \pi/2$ and $strateji == katlama$ then
14:	$strateji \leftarrow yarılama$
	end
15:	if $strateji == katlama$ then
16:	$mesafe \leftarrow mesafe * 2$
17:	else
18:	$mesafe \leftarrow mesafe/2$
	end if
19:	$doğrultu$ üzerinden $mesafe$ kadar ilerleyerek bir sonraki noktaya git
20:	Ölçüm al ve $doğrultu$ bul
21:	$doğrultu$ ve α 'yı kullanarak ölçüm üçgenini hesapla
22:	Bir önceki kesişim paralelogramı ile bu $adım$ 'da elde edilen ölçüm üçgeninin kesişimini bul (U)
	end while

Tablo 5’te 6. ve 24. satırlarda belirtilen *ölçüm al* ifadesi, hava robotunun bulunduğu konumda α kadar dönerek, her dönüşünde RTL-SDR vasıtasıyla verici frekansında sinyal okuması yapıp okuduğu sinyalleri, okuma doğrultusunu göz önünde bulundurarak kaydetmesi işlemidir. 8. ve 24. satırlarda *geçen analiz et, doğrultu bul* ifadeleri ise kaydedilen her bir sinyalin işlenerek sinyallerden vericinin olası yönüne dair bilgi çıkarılması, olası doğrultu açısının hesaplanması işlemidir. 12. ve 25. satırlarda geçen *ölçüm üçgeni* ifadesi ise Şekil 37’de her biri farklı renk yoğunluğu ile gösterilen, tepe noktaları ölçüm noktaları olan üçgenleri temsil etmektedir. Bu üçgenlerin tepe açısı 2α değerindedir ve yan kenarları da daha önceden kodda tanımlanmış maksimumUzaklık değeri ve α kullanılarak hesaplanır.

Program *katlama* stratejisi ile başlayıp *doğrultu* açısı $\pi/2$ değerini geçinceye kadar aynı strateji ile devam eder. *Doğrultu*’nun $\pi/2$ değerini geçmesi, hava aracının hedefi geçtiğini ifade etmektedir ve strateji, hedefe daha hassas yaklaşmak ve konumlama yapabilmek adına *yarılama* olarak değiştirilir.

17. satırda ifade edilen kontrol mekanizması bu durumu ifade etmektedir. Satır 15’deki kontrol mekanizması ise, *strateji* yarılama döndükten sonra hava robotunun ilerleme değerini ifade eden *mesafe* değişkeni programda önceden tanımlanan, satır 5’te de belirtilen MinimumMesafe değişkeninden küçük olması durumunda *strateji*’nin tekrar *katlama* olarak değiştirilmesini ifade etmektedir. Çoğu zaman bu aşamaya gelmeden konumlandırma işlemi gerçekleşmektedir.

4.3.2. Ölçüm Üçgeni’nin Bir Kenarı Üzerinde Ölçüm

Bir önceki bölümde açıklanan açıortay üzerinde ilerleyen yöntemden farklı olarak, bu yöntemde ölçüm üçgeninin açıortayı değil, üçgenin bir kenarı *doğrultu* kabul edilerek o *doğrultu* üzerinde ilerlenerek konumlandırma işlemi yapılmıştır. Bu yöntemde her bir ölçüm konumunda ölçüm üçgenlerinin aynı kenarının seçilmesi önemlidir. Şekil 38’de görülen yeşil çizgiler, ölçüm üçgenlerinin sağ kenarları üzerindedir ve seçilen kenar doğrultu kabul edilerek bir sonraki noktaya ilerlenir. Bu yöntemde ölçüm üçgeninin sağ kenarı seçilmiştir. Şekil 38’de de hava robotu, bir önceki algoritmada olduğu gibi ilk ölçüm noktasından itibaren katlama stratejisi ile başlamaktadır. Her

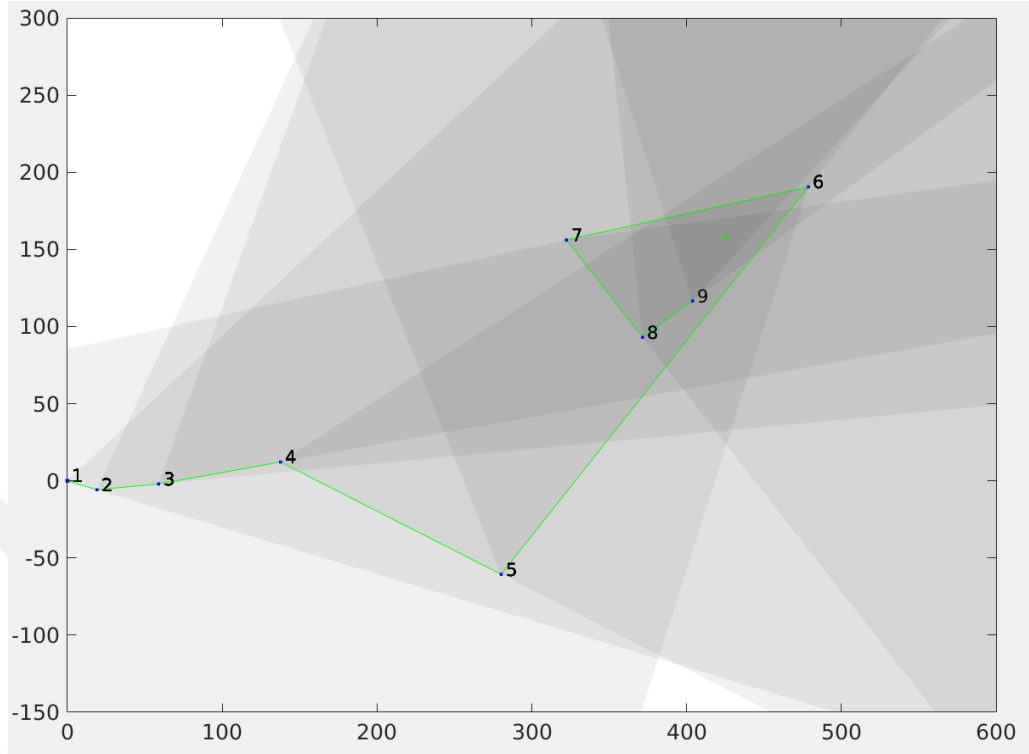
noktada, bir öncekinin iki katı ilerleme mesafesi ile hareket ederek bir önceki algoritmada olduğu gibi 6. ölçüm noktasında hedef geçilmiştir.

Tablo 6. Ölçüm üçgeninin bir kenarında çalışan algoritma

Algoritma	Ölçüm Üçgeninin Bir Kenarı Üzerinde Ölçüm
1:	$metod \leftarrow katlama$
2:	$mesafe \leftarrow MinimumMesafe$
3:	Ölçüm al
4:	Ölçümleri analiz et, doğrultu açısını bul
5:	Ölçüm üçgenini hesapla
6:	Doğrultu açısı üzerinden bir sonraki ölçüm noktasına git
7:	Adım 6-9 tekrarla
8:	Bir önceki ölçüm noktasında elde edilen ölçüm üçgeni ile bu noktadaki ölçüm üçgeninin kesişimlerini al (U)
9:	while $U > U^*$ do
10:	$adım \leftarrow adım + 1$
11:	if $strateji == yarılama$ and $mesafe < MinimumMesafe$ then
12:	$strateji \leftarrow katlama$
	end if
13:	if $doğrultu > \pi/2$ and $strateji == katlama$ then
14:	$strateji \leftarrow yarılama$
	end
15:	if $strateji == katlama$ then
16:	$mesafe \leftarrow mesafe * 2$
17:	else
18:	$mesafe \leftarrow mesafe/2$
	end if
19:	$doğrultu \leftarrow doğrultu + \alpha$
20:	$doğrultu$ üzerinden $mesafe$ kadar ilerleyerek bir sonraki noktaya git
21:	Ölçüm al ve $doğrultu$ bul
22:	$doğrultu$ ve α 'yı kullanarak ölçüm üçgenini hesapla
23:	Bir önceki kesişim paralelogramı ile bu $adım$ 'da elde edilen ölçüm üçgeninin kesişimini bul (U)
	end while

Hedef geçildiğinden dolayı stratejisini yarılama olarak değiştirerek, sonraki noktalar için hareket mesafesini yarıya indirerek ilerlemiştir. Bu yöntem, ölçüm üçgeninin bir kenarını takip ettiği için hedefin etrafını dolanarak konumlandırma eğilimi göstermektedir. Bir önceki bölümde kullanılan açortay metodundan farklı olarak bu yöntemde, Tablo 6'da da 23. satırda ifade edilen, hava robotunun ölçüm üçgenin kenarlarından bir tanesini $doğrultu$ açısı olarak kabul etmesi ve o doğrultu üzerinde

ilerleyerek konumlandırma yapmaktadır. Bunun dışında kalan yerler her iki metotta da aynıdır.



Şekil 38. Ölçüm üçgeninin sağ kenarı üzerinde çalışan simülasyon örneği

4.3.3. Ölçüm Üçgeninin Kenarları Üzerinde Sıralı Ölçüm

Bu algorithmada bir önceki yöntemde olduğu gibi ölçüm üçgeninin sürekli bir kenarı üzerinden hareket etmek yerine, her iki kenarı da sırayla kullanılarak konumlandırma yapılmıştır. Şekil 39’da görüldüğü üzere, algoritma ilk konumdan itibaren her konumda oluşturduğu ölçüm üçgenlerinin önce sol sonra sağ kenarını doğrultu kabul ederek konumlandırma işlemi bitene kadar sırayla her iki kenarı da kullanarak ilerlemiştir. Şekil 39’u açıklamak gerekirse; bu algoritma da bir öncekilerdeki ile aynı parametreleri kullanarak çalışmaktadır. İlk noktadan itibaren oluşturulan ölçüm üçgenlerinin kenarları sırasıyla doğrultu kabul ederek katlama stratejisi uyarınca ilerlemiş, 7. noktada, diğer algoritmalarda olduğu gibi hedef geçildiği için stratejisini yarılama olarak değiştirerek devam etmiştir. Stratejisi değişmesine karşın metot sabit kalmış ve ölçüm üçgenlerinin kenarlarından sırasıyla ilerlemeye devam etmiştir.

Bir önceki algoritmaya ek olarak bu metotta, Tablo 7’de 23. satırda ifade edilen kontrol mekanizması eklenmiştir. Ölçüm alınan adım sayısının 2’ye göre modu alınarak, doğrultu seçilmiştir.

Tablo 7. Ölçüm üçgeninin kenarları üzerinde sıralı çalışan konumlandırma algoritması

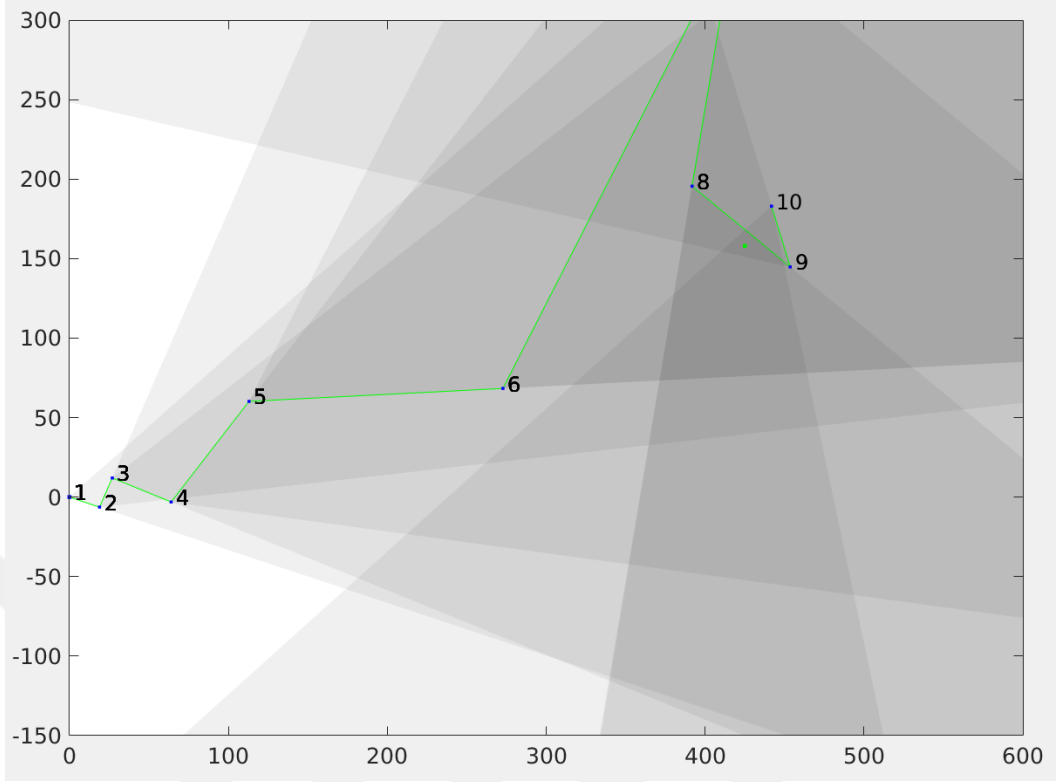
Algoritma Ölçüm Üçgeninin Kenarları Üzerinde Sıralı Ölçüm

```

1: metod ← katlama
2: mesafe ← MinimumMesafe
3: Ölçüm al
4: Ölçümleri analiz et, doğrultu açısını bul
5: Ölçüm üçgenini hesapla
6: Doğrultu açısı üzerinden bir sonraki ölçüm noktasına git
7: Adım 3-5 tekrarla
8: Bir önceki ölçüm noktasında elde edilen ölçüm üçgeni ile bu noktadaki ölçüm
   üçgeninin kesişimlerini al (U)
9: while  $U > U^*$  do
10:   adım ← adım + 1
11:   if strateji == yarılama and mesafe < MinimumMesafe then
12:     strateji ← katlama
13:   end if
14:   if doğrultu >  $\pi/2$  and strateji == katlama then
15:     strateji ← yarılama
16:   end if
17:   if strateji == katlama then
18:     mesafe ← mesafe * 2
19:   else
20:     mesafe ← mesafe / 2
21:   end if
22:   if adım % 2 == 0 then
23:     doğrultu ← doğrultu +  $\alpha$ 
24:   else
25:     doğrultu ← doğrultu -  $\alpha$ 
26:   end if
27:   doğrultu üzerinden mesafe kadar ilerleyerek bir sonraki noktaya git
28:   Ölçüm al ve doğrultu bul
29:   doğrultu ve  $\alpha$ 'yı kullanarak ölçüm üçgenini hesapla
30:   Bir önceki kesişim paralelogramı ile bu adım'da elde edilen ölçüm
   üçgeninin kesişimini bul (U)
31: end while

```

İkinin katı olan konumlarda hava robotunun ölçüm üçgeninin sol kenarını, ikinin katı olmayan konumlarda ise ölçüm üçgeninin sağ kenarını doğrultu açısı kabul ederek konumlandırma işlemi gerçekleştirilmiştir.



Şekil 39. Ölçüm üçgeninin kenarlarında sırayla ilerleyen simülasyon örneği

5. SİMULASYON VE SITL TABANLI TESTLER

Bu bölümde, geliştirilen algoritmaların başarımlarını ve hava robotunun otonom çalışma performansını test edebilmek adına yapılan Matlab simülasyonları ve SITL tabanlı testler ve sonuçları hakkında detaylı bilgi paylaşılacaktır.

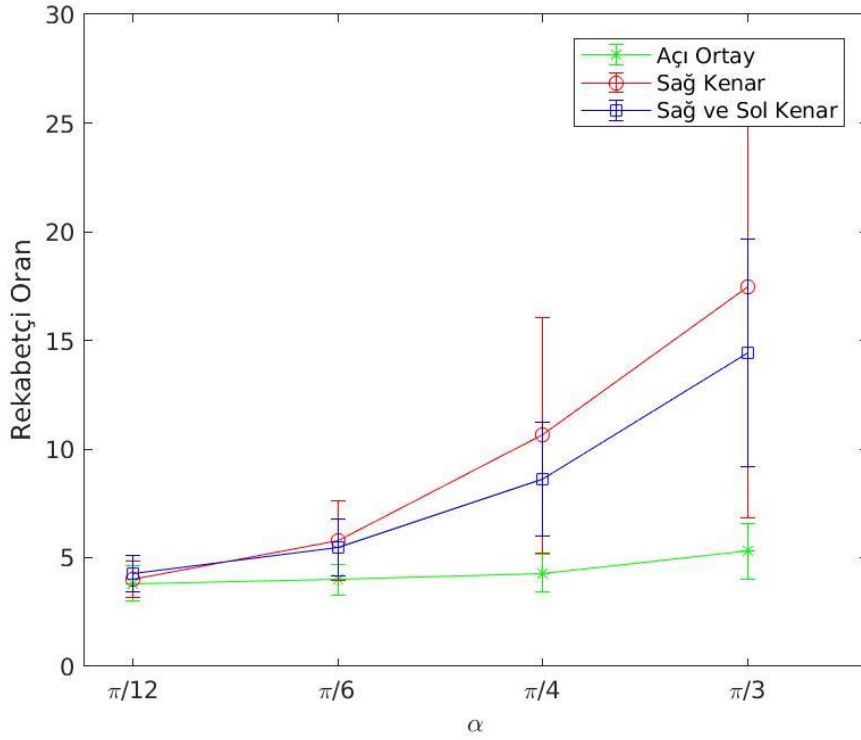
5.1. Simülasyon

Sistem, hava robotu kontrolünden sorumlu yazılım, ölçümleri alıp işleyen ve alınan sinyallere göre sinyal kaynağının doğrultusunu belirleyen yazılım ve sinyal işleyen yazılımdan gelen veriler ile kullanılan algoritma doğrultusunda ölçüm noktalarını belirleyip konumlandırma işlemine karar veren yazılım olmak üzere üç ana bölümden oluşmaktadır. Her bölüm öncelikle simülasyon ortamlarında, daha sonra saha testlerinde ayrı ayrı test edilip, doğruluklarına karar verildikten sonra bir araya getirilip yine simülasyon ortamında doğrulandıktan sonra sahada test edilmiştir.

Bölüm 4.3 altında açıklanan algoritmalar, Matlab kullanılarak bilgisayar benzetimleri yapılmıştır. Simülasyonlarda sinyal vericinin konumu ön tanımlı olarak programa verilmektedir. Hedef ile hava robotunun arasındaki doğrultu (mutlak doğrultu/true bearing) hesaplandıktan sonra Bölüm 4.1’de açıklanan ölçüm üçgeni modelini elde edebilmek adına mutlak doğrultuya rastgele atanan gürültü eklenerek ölçülen doğrultu gerçeğe en yakın şekilde modellenmeye çalışılmıştır. Doğrultusya eklenen gürültü en fazla $\pm\alpha$ kadardır.

Bölüm 4.1’de detaylarının açıklandığı algoritmalar, 10 m/s ilerleme hızı ve her bir ölçüm açısında 6sn harcandığı varsayılarak simülasyonlar gerçekleştirilmiştir. 6sn, hava robotunun ölçüm alıp analiz etmesi ve bir sonraki doğrultu değerine ulaşması için geçen toplam süreyi ifade etmektedir. Algoritmalar, rastgele belirlenmiş 100 adet hedef nokta için, her bir nokta 100 kez tekrarlanacak şekilde $\pi/3$, $\pi/4$, $\pi/6$, $\pi/12$ olmak üzere 4 farklı α değeri için test edilmiştir. Ayrıca 2 farklı U^* ölçütüyle testler yapıp sonuçlar rekabetçi oran, ölçüm zamanı, ölçüm süresi değişkenleri baz alınarak üretilmiştir. Test sonuçlarını paylaşacak olursak, U^* (istenen belirsizlik) değeri yarı çapı 10m olan dairenin alanını temsil eden 314 alındığında aşağıdaki sonuçlar üretilmiştir. Şekil 40’da, Bölüm 4.2’de açıklanan rekabetçi oranın α ’ya göre değişim

grafığı verilmiştir. Grafikte, geliştirilen üç algoritmanın da sonuçları dört farklı α değeri için karşılaştırmalı olarak üretilmiştir. Grafikte de görüldüğü üzere algoritmalar arasında en iyi performansı açıortay üzerinden ilerleyerek konumlandırma yapan yöntem vermektedir. Ölçüm üçgeninin her iki kenarını kullanan algoritma ise tek bir kenarını kullanan algoritmaya göre daha iyi sonuç vermiştir.

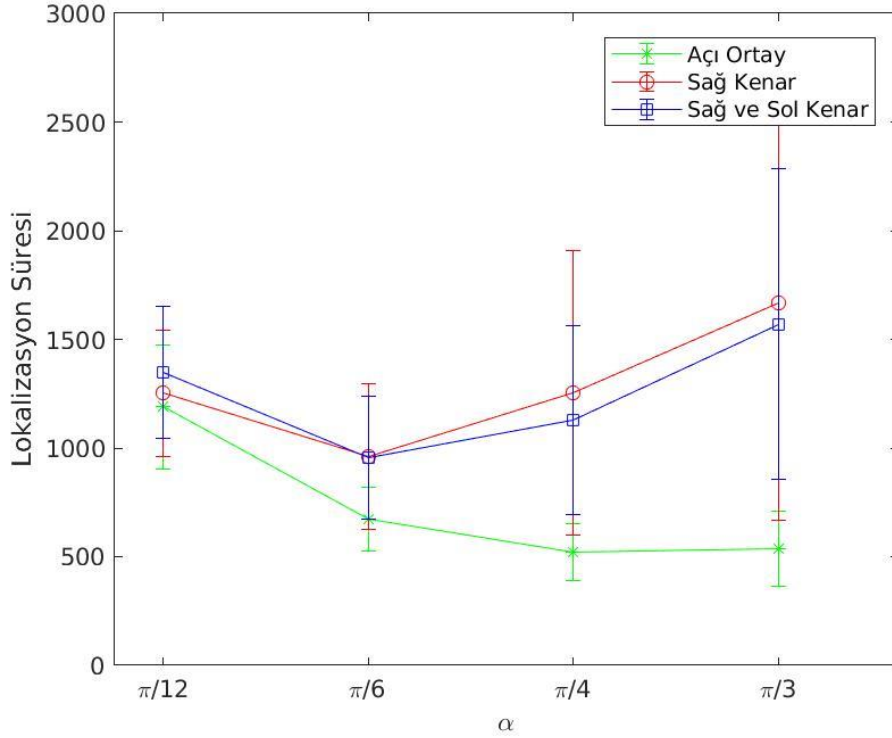


Şekil 40. $U^* = 314$ için rekabetçi oran – α grafığı

Şekil 40 için α değerindeki değişime göre grafığı yorumlayacak olursak; α , $\pi/12$ gibi küçük bir değer alındığında üç algoritma da birbirine çok yakın ve başarı oranı yüksek sonuçlar üreterek konumlandırma için kullanılan algoritma farkını ortadan kaldırmaktadır. α 'yı arttırdığımızda algoritmalar arasındaki başarı oranı farkı artmaktadır. Sonuç olarak α 'yı ne kadar düşük alırsak başarı oranımız o kadar artacak ve algoritmalar arasındaki fark azalacaktır.

Fakat $\pi/12$ gibi küçük değerler saha deneyleri için gerçekçi olmayacaktır. Kullanılacak antenler bu derece hassas ölçüm yapmaya izin verecek seviyede değildir. Ve her bir noktada ölçüm için harcanan süre diğer α değerlerine göre çok fazla olacağından, hava robotunun havada kalma süresi içerisinde konumlandırmanın gerçekleştirilememesine

sebeplendirilebilir. Ancak α 'yı düşürmek diğer parametreleri de etkileyecektir. Bu etkiler ileriki grafiklerde değineceğiz.

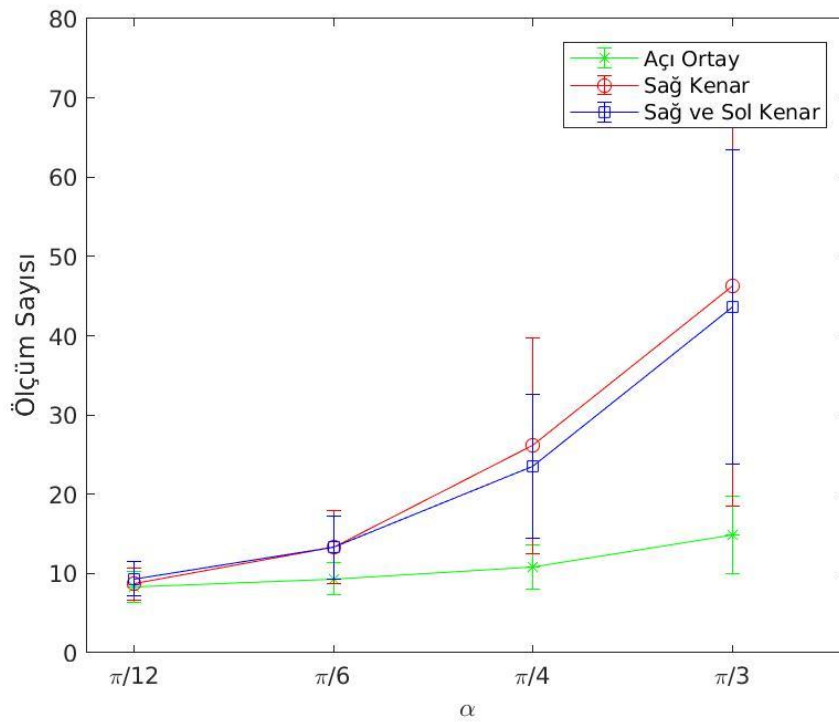


Şekil 41. $U^* = 314$ için lokalizasyon süresi – α grafiği

Şekil 41’de üç algoritmanın da lokalizasyon sürelerinin α ’ya göre grafiği verilmiştir. Grafikte görüldüğü üzere en iyi performansı yine açıortay metodu vermiştir. Ölçüm üçgeninin her iki kenarını kullanan algoritma ile tek bir kenar üzerinde ilerleyen algoritma çok yakın sonuçlar vermiş; hatta $\pi/6$ değerinde aynı sonuçları üretmişlerdir. Bir önceki paragrafta değinildiği üzere, α ’yı arttırmak algoritmanın başarımını arttırsa da bunun yan etkileri de olmaktadır. Şekil 41’de görüldüğü üzere, açıortay metodu için α küçüldükçe lokalizasyon süresi artmaktadır. Her iki kenarı kullanan ve tek bir kenar üzerinden ilerleyen algoritmalar için ise α değerini düşürmek, $\pi/6$ ’ya kadar konumlandırma süresini düşürse de, $\pi/6$ ’dan daha küçük değerler vermek konumlandırma süresini artırmaktadır. Her iki kenarı ve tek bir kenarı kullanan algoritmalar için en iyi lokalizasyon süresi sağlayan α değeri $\pi/6$ ’dır. Buna karşın tüm α değerlerinde açıortay metodu daha iyi konumlandırma süresi sağlamaktadır. En iyi

konumlandırma süresi sunan açıortay metodu ise en düşük hassasiyet sağlayan $\pi/3$ değerinde en düşük konumlandırma süresine ulaşmıştır.

Bunun temel sebebi; α 'yı düşürdüğümüzde her bir noktada alınan ölçüm sayısının ve buna bağlı olarak ölçüm ve analiz sürelerinin azalmasından kaynaklanmaktadır. Sonuç olarak α 'yı artırmak, algoritmanın başarımını artırsa da hava robotunun performansını artırır diyemeyiz. Hava robotları için en kritik parametrelerden birisi uçuş süresi olduğundan dolayı en iyi α değerini seçmek gerekmektedir.

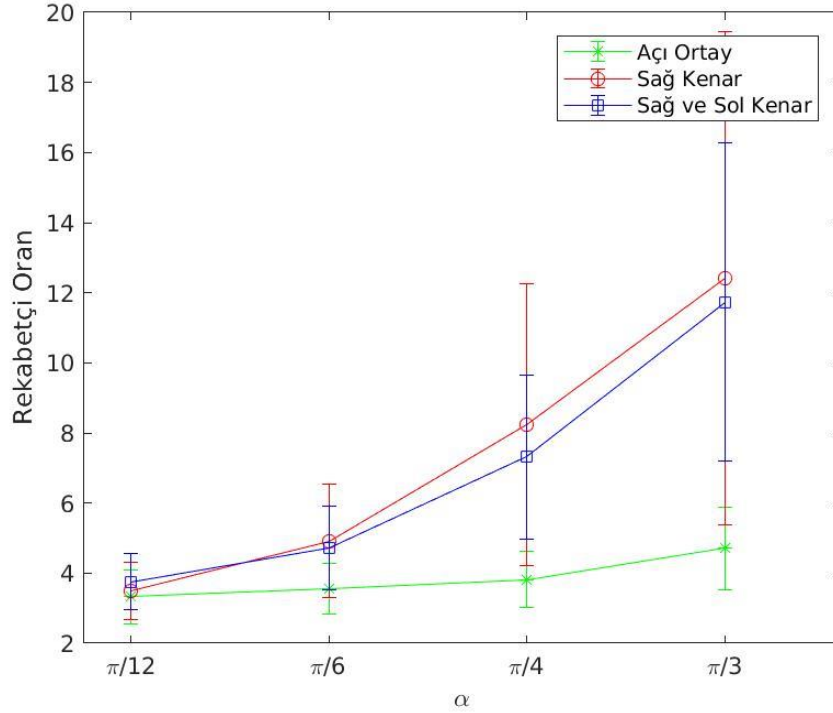


Şekil 42. $U^* = 314$ için ölçüm sayısı – α grafiği

Şekil 42’de görüldüğü gibi ölçüm sayısı- α grafiğinde de en iyi performansı açıortay metodu vermiştir. Bir önceki grafiklerde olduğu gibi α 'yı $\pi/12$ gibi küçük değerlere indirmek algoritmaların performansını birbirine yakınsatarak metod farkını ortadan kaldırmaktadır. Lokalizasyon süresi grafiğinde olduğu gibi; her iki kenarı kullanan algoritma ile tek bir kenarı kullanan algoritma birbirine çok yakın sonuçlar üretmiş ve $\pi/6$ değerinde aynı sonuçları vermiştir. Açıortay metodunu kendi içerisinde α değişkeni üzerinden inceleyecek olursak; α 'nın artması ölçüm sayısı üzerinde çok

büyük farklara sebep olmamıştır. Diğer metotlarda ise α arttıkça bir önceki değere göre ölçüm sayısında belirgin artışlar söz konusudur.

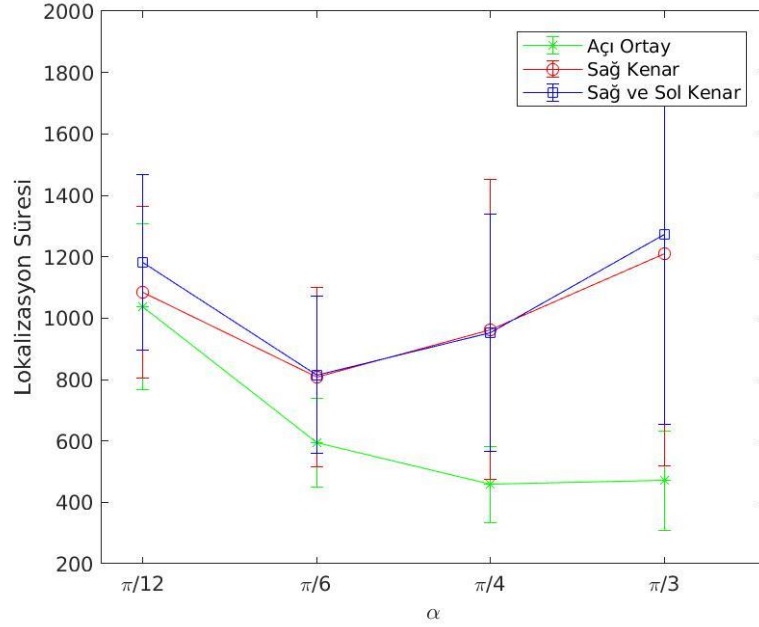
Sonuç olarak, U^* 'ın 314 alındığı simülasyon sonuçlarına göre, açıortay üzerinden ilerleyen algoritma en iyi sonuçları üretmektedir.



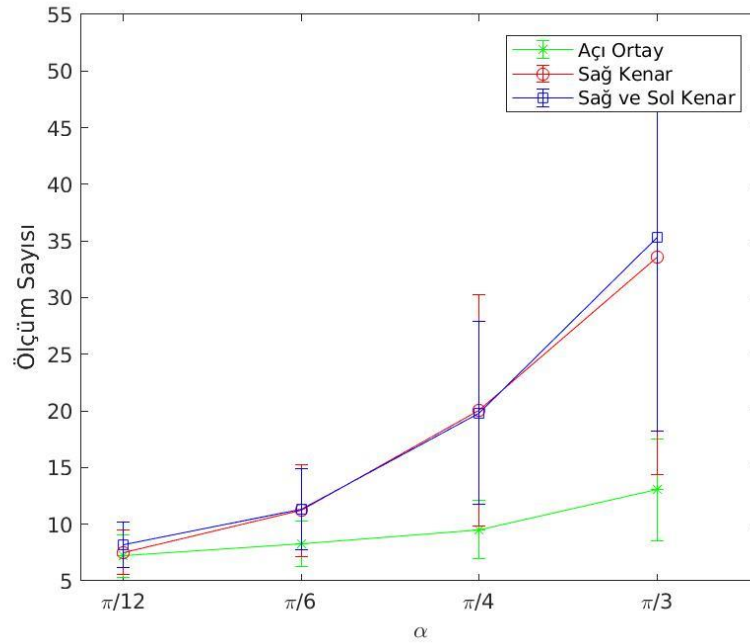
Şekil 43. $U^* = 1256$ için rekabetçi oran – α grafiği

Yukarıda değerlendirilen tüm simülasyon sonuçları, U^* (istenilen belirsizlik) değeri 1256 alınarak tekrar edilip U^* 'ın sonuçlara etkisi gözlemlenmiştir. Şekil 43'ü incelediğimizde grafiğin eğilimine bakacak olursak; Şekil 40'daki ile benzer eğilim göstermektedir. Ancak y eksenindeki rekabetçi oran değerlerini her iki grafikte de incelediğimizde U^* 'ı 1256 aldığımızda algoritmaların başarımının arttığı gözlemlenmiştir. Ancak algoritmanın başarımının artması sistemin başarısının arttığı anlamına gelmemektedir. U^* 'ı 1256 almamız bir önceki değere göre 4 katı büyük bir alan için lokalizasyon yaptığımızı göstermektedir ve bu da hata oranının 4 kat daha fazla olabileceğini göstermektedir. Bu da her iki grafik incelendiğin Şekil 43'de her bir nokta için varyansın Şekil 40'a kıyasla artışından anlaşılabilir. Şekil 44'e baktığımızda da Şekil 41 ile benzer eğilim ile karşılaşmaktayız. Bu grafikte de bir önceki rekabetçi oran grafiğinde olduğu gibi performans beklendiği üzere arttığı

görülmektedir. Yine bu grafikte de bir öncekinde olduğu gibi U^* 'ın artışı kaynaklı varyansdaki artış gözlemlenmiştir. Şekil 45 de önceki grafiklerdekine benzer çıktılar üretmiştir. Burada U^* 'ı büyütmek, özellikle her iki kenarı kullanan ve bir kenarı kullanan algoritmalar arasındaki farkı azaltarak algoritmaların çok yakın, hatta benzer sonuçlar vermelerine sebep olmuştur.



Şekil 44. $U^* = 1256$ için lokalizasyon süresi – α grafiği

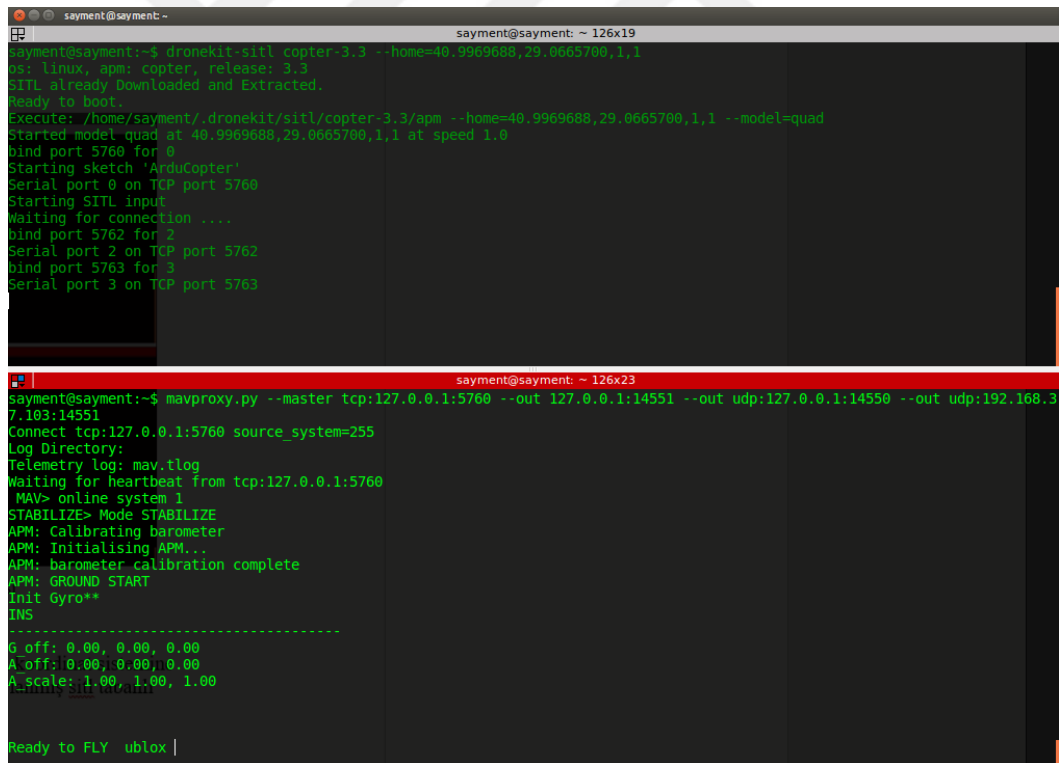


Şekil 45. $U^* = 1256$ için ölçüm sayısı – α grafiği

U^* 'ı artırmak ölçüm süresini ve ölçüm sayısını düşürmüş olarak gözükse de, bu sonuçlar istenilen alanın çok büyük bir değer olmasından dolayı algoritmanın istenilen duruma daha hızlı ulaşmasından kaynaklanmaktadır. Ancak dikkat edilmelidir ki bu U^* değeri için hata oranı sahada kullanamayacağımız kadar yüksektir.

5.2 SITL ile Tabanlı Testler

SITL (Software in the Loop), ArduPilot tabanlı uçuş kontrolcüsünün simülasyonunun yapılmasına olanak sağlayan bir yazılımdır. Bu simülasyon yazılımı sayesinde hiçbir donanıma gerek kalmadan hava robotu üzerindeki uçuş kontrol bilgisayarı simülasyon ortamında gerçekleştirilerek hava robotunun tüm özelliklerine sahada çalışıyormuş gibi erişerek çalışma olanağı sunmaktadır.



```
sayment@sayment:~$ dronekit sitl copter-3.3 --home=40.9969688,29.0665700,1,1
OS: linux, apm: copter, release: 3.3
SITL already Downloaded and Extracted.
Ready to boot.
execute: /home/sayment/.dronekit/sitl/copter-3.3/apm --home=40.9969688,29.0665700,1,1 --model=quad
started model quad at 40.9969688,29.0665700,1,1 at speed 1.0
bind port 5760 for 0
Starting sketch 'ArduCopter'
Serial port 0 on TCP port 5760
Starting SITL input
Waiting for connection ....
bind port 5762 for 2
Serial port 2 on TCP port 5762
bind port 5763 for 3
Serial port 3 on TCP port 5763

sayment@sayment:~$ mavproxy.py --master tcp:127.0.0.1:5760 --out 127.0.0.1:14551 --out udp:127.0.0.1:14550 --out udp:192.168.3
7.103:14551
Connect tcp:127.0.0.1:5760 source_system=255
Log Directory:
Telemetry log: mav.tlog
Waiting for heartbeat from tcp:127.0.0.1:5760
MAV> online system 1
STABILIZE> Mode STABILIZE
APM: Calibrating barometer
APM: Initialising APM...
APM: barometer calibration complete
APM: GROUND START
Init Gyro**
INS
-----
G_off: 0.00, 0.00, 0.00
A_off: 0.00, 0.00, 0.00
A_scale: 1.00, 1.00, 1.00

Ready to FLY ublox |
```

Şekil 46. SITL ve Mavproxy terminal çıktısı

Bu çalışmada Dronekit-SITL (Dronekit, Dronekit-SITL, 2014) kullanarak hava robotunun simülasyonu gerçekleştirilmiştir. Simülasyon terminal üzerinden başlatılır

network aracılığı ile diğer programlar ile haberleşmektedir. Şekil 46'daki üst terminalde görüldüğü üzere Dronekit-SITL 5760. port üzerinden yayın yapmaya başlamıştır. SITL'in 5760. port üzerinden yaptığı bu yayın, Mavproxy (Ardupilot, MAVProxy, 2019) yardımıyla alınarak istenilen IP ve portlara yönlendirilmesi sağlanmıştır. Şekil 46'daki alt terminalde görülen Mavproxy komutu lokal'deki 5760. porta bağlanarak o port üzerindeki yayını yine lokal'de 14550 ve 14551. portlar ile 192.168.37.103 IP'sinin 14551. portuna iletmektedir. Bu işlem farklı yazılımlar ile simülasyona bağlanıp veri okumamıza ve simülasyona veri göndererek kontrol etmemizi sağlamaktadır.

Tablo 8. Açırtay algoritmasının robot üzerindeki gerçekleştirilmesinin sözde-kodu

Algorithm Açırtay Algoritması ile Çalışan Hava Robotu Algoritması

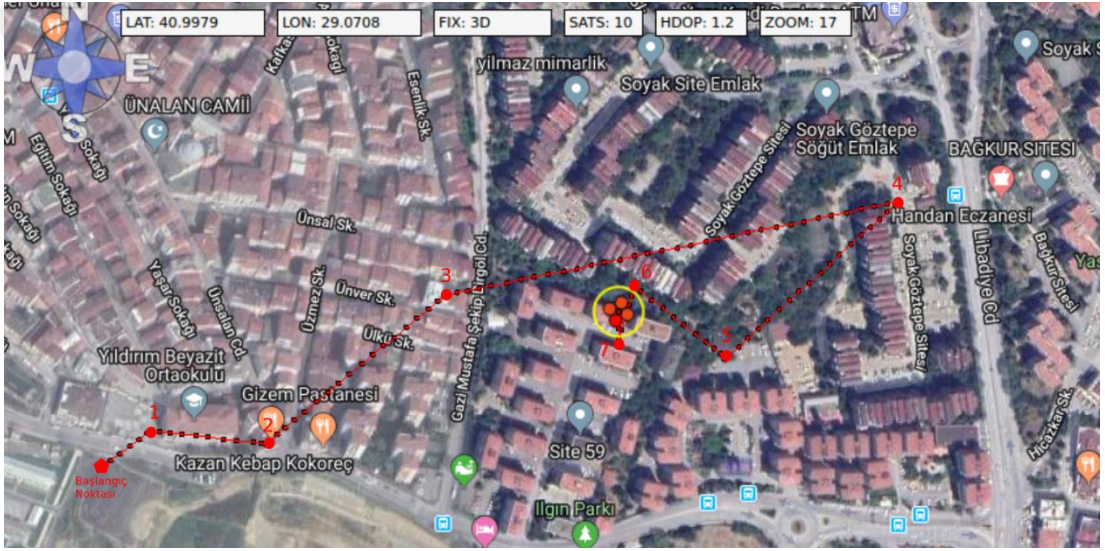
```

1: metod ← katlama
2: mesafe ← MinimumMesafe
3: Hava Robotuna Bağlan
4: Motorları Çalıştır (Arm)
5: Hava Robotunu İstenilen İrtifaya Çıkar (Takeoff)
6: Hava Robotu Başlangıç Doğrultusuna Çevir (Yaw)
7: Ölçüm alma fonksiyonunu çağır
8: Sinyal analiz fonksiyonunu çağır, (doğrultu) hesapla
9: Açırtay üzerinde ölçüm yeri belirle
10: Bir sonraki ölçüm noktasını koordinatlarını hesapla
11: Hava robotunu sonraki konuma götüren fonksiyonu çağır (waypoint nav)
12: Adım 7-10 Tekrarla
13: while  $U > U^*$  do
14:   adım ← adım + 1
15:   if strateji == yarılama and mesafe < MinimumMesafe then
16:     strateji ← katlama
17:   end if
18:   if doğrultu >  $\pi/2$  and strateji == katlama then
19:     strateji ← yarılama
20:   end if
21:   if strateji == katlama then
22:     mesafe ← mesafe * 2
23:   else
24:     mesafe ← mesafe / 2
25:   end if
26:   Hava robotunu sonraki konuma götüren fonksiyonu çağır
27:   Ölçüm alma fonksiyonunu çağır
28:   Sinyal analiz fonksiyonunu çağır, (doğrultu) hesapla
29:   Açırtay üzerinde ölçüm yeri belirle
30: end while

```

Lokal'deki 14550 portu, yer kontrol istasyonu yazılımı olan Apm'i simülasyona bağlamak için, yine lokal'deki 14551 portu ise hava robotunu kontrol eden yazılımın simülasyona bağlanması içindir. Diğer IP ise kodun ve simülasyonun çalıştığı bilgisayardan bağımsız olarak çalışan bir tableten farklı bir yer kontrol istasyonu ile simülasyonu izlemek için yazılmıştır.

Hava robotu üzerinde çalışan algoritma, Bölüm 4'te detayları verilen yaklaşımlara ek olarak, hava robotu üzerinde çalışan algoritma; hava robotu kontrolü ve sinyal işleme alt bölümlerini de barındırmaktadır. Hava robotu üzerinde çalışan algoritma Tablo 8'de verilmiştir.

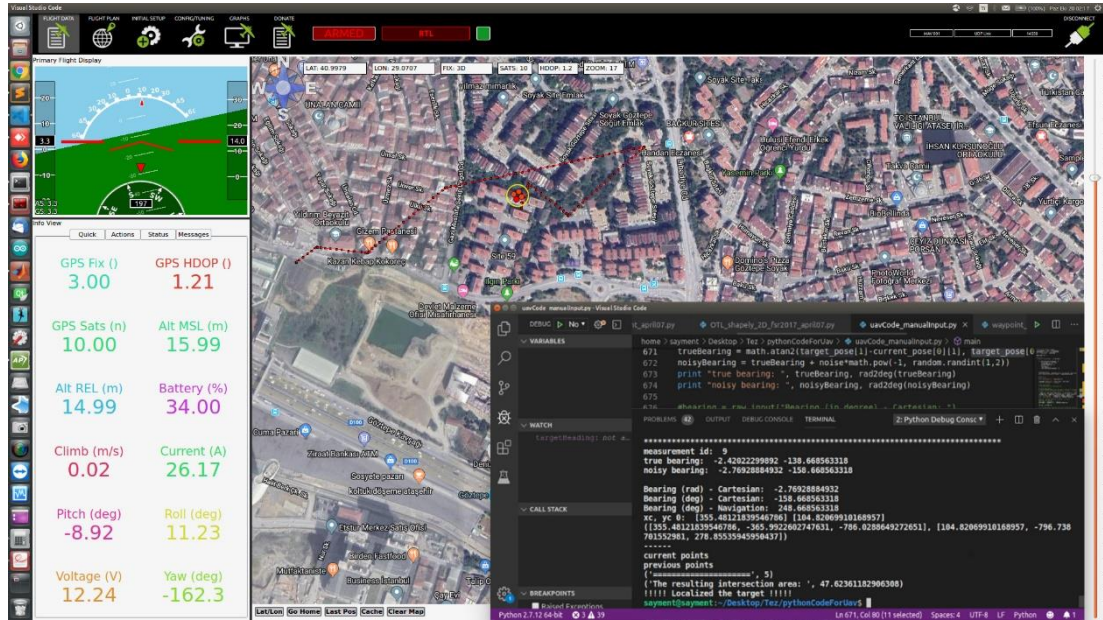


Şekil 47. SITL tabanlı test sonucu. Hava robotunun konumlandırma yaparken izlediği rota kırmızı çizgi ile gösterilmektedir.

Tablo 8'de 3. satırda geçen *hava robotuna bağlan* ifadesi, hava robotu üzerindeki kontrol biriminin (RaspberryPi) seri port üzerinden otomatik pilot ile bağlantı kurması işlemidir. Bağlantı kurulduktan sonra otomatik pilot kontrolleri yapıp, uçuş modunu ayarlayarak motorları çalıştırır ve hava robotunu istenilen irtifaya yükseltir. Hava robotu tüm işlemlerini sabit irtifada yapmakta, çalışma esnasında irtifa değiştirilmemektedir. 6. satırda belirtilen *başlangıç doğrultusu* kuzeye göre $\pi/2$ 'lik doğrultuyu ifade etmektedir. Bu doğrultu kartezyen koordinat sisteminde 0° 'ye karşılık gelmektedir. Bu işlemin amacı kartezyen koordinat sistemi ile küresel konumlama sistemi arasında geçiş sağlamak ve programın daha sade ve verimli çalışmasıdır. 7.

satırda belirtilen sinyal alma fonksiyonu, Bölüm 3.1.1’de belirtilen adetli ölçüm alma yöntemi ile çalışmaktadır. Başlangıç doğrultusunda ölçüm alınıp α kadar hava robotunun doğrultusu değiştirilir ve tekrar ölçüm alınarak bu işlem bir tur tamamlanıncaya kadar devam ettirilir. Alınan ölçümler işlenip doğrultu belirlendikten sonra Bölüm 4.3.1’de detayları verilen yöntem ile kartezyen koordinat sisteminde bir sonraki ölçüm noktası belirlenir. Kartezyen koordinat sisteminde belirlenen nokta baz alınarak hava robotu için küresel konumlandırma sistemine göre bir sonraki ölçüm noktasının koordinatları belirlenir. Tekrar ölçüm alıp Bölüm 4.3.1’de açıklanan yöntem kullanılarak, kartezyen koordinat sisteminde oluşturulan ölçüm üçgenlerinin alanı (U), U^* ’dan (istenen belirsizlik seviyesi) büyük oluncaya kadar işlemlere devam edilir.

Önceki bölümlerde detayları verilen U^* 314, $\alpha \pi/6$, minimum uzaklık 20m ve hedef nokta kartezyen koordinat sisteminde göre x ekseninde 350, y ekseninde 100 konumunda olacak şekilde ayarlanmıştır. SITL tabanlı hava robotu simülasyonu sonucu Şekil 47’de görülmektedir. Bu simülasyonda Bölüm 4’te analizleri yapılmış olan geliştirdiğimiz algoritmalar içerisinde en iyi performansı veren açıortay üzerinde ilerleyen algoritma test edilmiştir.



Şekil 48. APM ile Hava Robotu Simülasyonu Program Çıktı Görüntüsü

Şekil 47'deki grafiği detaylı inceleyecek olursak; Bölüm 3.1'de detayları açıklanan katlama-yarılama metodu açıkça görülmektedir. Hava aracı görselin sol altındaki konumdan hareketine başlayarak *katlama* stratejisine uyarınca her adımda bir önceki adımın iki katı kadar ilerlemiş, hava aracı hedefi geçtikten sonra ise stratejisini *yarılama* olarak değiştirerek ilerleme miktarını bundan sonraki her adımda bir öncekine göre yarıya düşürerek konumlandırma işlemini başarıyla gerçekleştirmiştir.

Yer kontrol istasyonu yazılımının ve çalışan programın çıktıları Şekil 48'de detaylı olarak görülmektedir. Hava aracı, bir önceki bölümde değinilen, Matlab simülasyon sonuçları ile de örtüşerek 9 konumda hedefi lokalize etmiştir. Bu sonuç Şekil 47'deki program çıktısında da gözükmemektedir.



6. SONUÇ

Bu tezde, hava robotu kullanarak bir radyo sinyal kaynağının konumlandırılması problemi ele alınmıştır. Robot, üzerindeki Moxon yönlü anten sayesinde hedefe olan doğrultu açısını elde edebilmektedir. Uzaklık bilgisi güvenilir olmadığından sadece doğrultu açısı ölçümleri kullanılmıştır. Buradaki doğrultu açı ölçümlerinde, sınırlı gürültü modeli esas alınmıştır. Bu modelde, hedefin ölçüm üçgeninin içerisinde olduğu garanti edilmektedir. Hedefin konumlandırılması, alınan ölçümlerden elde edilen bu ölçüm üçgenlerinin kesiştirilmesiyle yapılmaktadır. Dolayısıyla ölçüm yerlerinin seçimi konumlandırma için önemli olmaktadır. Üç farklı çevrimiçi konumlandırma stratejisi geliştirilmiştir: i) Açığortay üzerinde hareket eden strateji, ii) Ölçüm üçgeninin bir kenarı üzerinde hareket eden strateji, iii) Ölçüm üçgeninin sağ ve sol kenarları üzerinde hareket eden strateji.

Geliştirilen bu stratejilerin performansları, en iyi çevrimdışı stratejisinin maliyetinin alt sınırı üzerinden rekabetçi oranları hesaplanarak analiz edilmiştir. Kapsamlı bilgisayar benzetimleri ile farklı parametre değerleri kullanılarak performans analizi yapılmıştır. Bu sonuçlara göre açığortay üzerinde hareket eden strateji en iyi performansa sahiptir.

Ayrıca konumlandırma işlemini yapabilecek, 4 rotorlu bir hava robotu üretilmiştir. Bunun otonom olarak hareket edebilmesi için gerekli yazılım altyapısı Python dili ile DroneKit kütüphanesi kullanılarak geliştirilmiştir. Geliştirilen stratejilerden en iyi performansa sahip olan strateji, DroneKit SITL kullanılarak test edilmiştir.

Gelecek çalışma olarak, sistemin üretimi yapılan robot üzerinde testleri yapılacaktır. Geliştirilen sistemin Orman Bakanlığı tarafından yapılan yaban hayatı takip işlemlerine entegre edilmesi planlanmaktadır. Ayrıca, hedefin hareket etmediği varsayımı ortadan kaldırılıp, hareketli hedef halinde çevrimiçi algoritma geliştirilecektir.

KAYNAKÇA

- Ac6la. (2019). *Moxon Rectangle Generator*. <https://ac6la.com/moxgen1.html> adresinden alındı
- Alex, S. (2015). *How to choose the right motor for your multicopter drone*. Drone Test: <https://www.dronetrest.com/t/how-to-choose-the-right-motor-for-your-multicopter-drone/568> adresinden alındı
- Ardupilot. (2019). Ardupilot: <http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html> adresinden alındı
- Ardupilot. (2019). *MAVProxy*. Ardupilot: <http://ardupilot.github.io/MAVProxy/html/index.html> adresinden alındı
- Bayram , H., Stefan, N., & Isler, V. (2018). Aerial radio-based telemetry for tracking wildlife. *International Conference on Intelligent Robots and Systems (IROS)* (s. 4723-4728). IEEE.
- Carr, J., & Gippisley, G. (2012). Moxon Beam. *Practical Antenna HandBook* (s. 306-307). içinde McGrawHill.
- Carr, J., & Gippisley, G. (2012). *Practical Antenna HandBook*. *Three-Element Yagi Antennas* (s. 282-285). içinde McGrawHill.
- Cebik, L. B. (2000). *Notes on the Moxon Rectangle Pattern*. <http://www.antentop.org/w4rn1.001/moxpat.html> adresinden alındı
- Dressel, L., & Kochenderfer, M. (2018). Efficient and Low-cost Localization of Radio Signals with a Multicopter UAV. *AIAA Guidance, Navigation, and Control Conference*, (s. 1845).
- Dressel, L., & Kochenderfer, M. (2019). Hunting Drones with Other Drones: Tracking a Moving Radio Target. *International Conference on Robotics and Automation (ICRA)*, (s. 1905-1912). Montreal.
- Dronekit. (2014). *Dronekit-SITL*. <https://github.com/dronekit/dronekit-sitl> adresinden alındı
- Dronekit. (2014). *Qgroundcontrol*. https://docs.qgroundcontrol.com/en/getting_started/download_and_install.html adresinden alındı

- Ecalc. (2010). *the most reliable RC Calculator on the Web*. Ecalc:
<https://www.ecalc.ch/> adresinden alındı
- Emax. (2019). *EMAX User Instruction for BLHeli Series ESC*. Tme:
<https://www.tme.eu/Document/959503479e9fa7b0e61049c62875b2e2/BLHeli%20Instruction.pdf> adresinden alındı
- Gibbons, G. (2013). *FCC Fines Operator of GPS Jammer That Affected Newark Airport GBAS*. Inside GNSS: <https://insidegnss.com/fcc-fines-operator-of-gps-jammer-that-affected-newark-airport-gbas/> adresinden alındı
- Hoffmann, G., Waslander, S., & Tomlin, C. (2006). Distributed Cooperative Search Using Information - Theoretic Costs for Particle Filters, with Quadrotor Applications. *AIAA guidance navigation and control conference*, (s. 6576). Colorado.
- Isaacs, J., Quitin, F., Carrillo, L., & Madhow, U. (2014). Quadrotor Control for RF Source Localization and Tracking. *International Conference on Unmanned Aircraft Systems (ICUAS)*, (s. 244-252). Orlando.
- Isler, V., & Bajcsy, R. (2006). The Sensor Selection Problem for Bounded Uncertainty Sensing Models. *Transactions on Automation Science and Engineering* , 372-381.
- K8JHR. (2014). *6 Meter Band MOXON-STYLE*. K8JHR.COM:
https://k8jhr.com/files/6_m_moxon_build_report_1b____.pdf adresinden alındı
- Kelly, A. (2003). Precision dilution in triangulation based mobile robot position estimation. *Intelligent Autonomous Systems*, (s. 1046-1053).
- Kirkhorn, J. (1999). *Introduction to IQ-demodulation of RF-data*.
- Machineto. (2019). *T-motor high quality brushless motor MT3515-- KV400/650KV*. machineto.com: <http://www.machineto.com/t-motor-high-quality-brushless-motor-mt3515-kv400-650kv-10237085> adresinden alındı
- Perkins, A., Dressel, L., Lo, S., & Enge, P. (2015). Antenna Characterization for UAV Based GPS Jammer Localization. *28th International Technical Meeting of the Satellite Division of The Institute of Navigation*, (s. 1684-1695).
- Pixhawk. (2019). *Pixhawk*. Pixhawk: <https://pixhawk.org/> adresinden alındı
- Pixhawk. (tarih yok). *PX4 AutoPilot*. <https://px4.io/> adresinden alındı

PX4. (2019). *Cube Flight Controller*.

https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk-2.html adresinden alindi

RaspberryPi.org. (2014). *Raspberry Pi 3 Model B+*. RaspberryPi.org:

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> adresinden alindi

Tokekar, P., & Isler, V. (2013). Sensor placement and selection for bearing sensors with bounded uncertainty. *International Conference on Robotics and Automation (ICRA)* (s. 2515-2520). IEEE.



EK 1. GQRX KURULUMU

Sistemde Gqrx'e dair daha önceden kalan kurulum, paket, kütüphane varsa silinir:

- `sudo apt-get purge --auto-remove gqrx`
- `sudo apt-get purge --auto-remove gqrx-sdr`
- `sudo apt-get purge --auto-remove libgnuradio*`

Gqrx, sdr ve gnuradio ile alakalı kaynaklar (repository) yüklenip sistem güncellenir:

- `sudo add-apt-repository -y ppa:bladerf/bladerf`
- `sudo add-apt-repository -y ppa:myriadrfd/drivers`
- `sudo add-apt-repository -y ppa:myriadrfd/gnuradio`
- `sudo add-apt-repository -y ppa:gqrx/gqrx-sdr`
- `sudo apt-get update`

Son olarak Gqrx'in yüklemesi yapıp terminal veya arayüz aracılığı ile çalıştırılır.

- `sudo apt-get install gqrx-sdr`

EK 2. DRONEKIT VE DRONEKIT-SITL KURULUMU

Dronekit-SITL'i python-pip yükleme aracı ile kurulmaktadır. Öncelikle python-pip kurulur:

- `sudo apt-get install python-pip`

Sonrasında python-pip kullanılarak Dronekit ve Dronekit-SITL kurulur:

- `pip install dronekit`
- `pip install dronekit-sitl`



EK 3. QGROUNDCONTROL KURULUMU

Ubuntu işletim sisteminde bulunan modem manager uygulaması, seri port kullanan uygulamaları kesintiye uğratabilmektedir. Bundan dolayı öncelikle modem manager uygulamasını sildikten sonra, seri port kullanabilmek için mevcut kullanıcıya kullanım hakkı verilmesi gerekmektedir. QGroundControl, gstreamer uygulamasına ihtiyaç duyduğu için bu programın kurulması gerekmektedir:

- `sudo usermod -a -G dialout $USER`
- `sudo apt-get remove modemmanager -y`
- `sudo apt install gstreamer1.0-plugins-bad gstreamer1.0-libav -y`

Sonrasında QGroundControl'un sayfasından (Dronekit, Qgroundcontrol, 2014) programın imajı indirilerek çalıştırılabilir hakkı verilir. Sonra terminal veya grafik arayüz aracılığı ile çalıştırılır:

- `chmod +x ./QGroundControl.AppImage`
- `./QGroundControl.AppImage`

EK 4. MAVPROXY KURULUMU

DroneKit-SITL'in dięer programlar ile baęlantısını saęlayan Mavproxy, biręok pakete ihtiyaę duymaktadır. Bu paketler, ařaęıdaki komutlar ile kurulur:

- `sudo apt-get install python3-dev`
- `sudo apt-get install python3-opencv`
- `sudo apt-get install python3-wxgtk3.0`
- `sudo apt-get install python3-pip`
- `sudo apt-get install python3-matplotlib`
- `sudo apt-get install python3-pygame`
- `sudo apt-get install python3-xml`
- `sudo apt-get install python3-yaml`

Paketler yklendikten sonra `python-pip` kullanarak `mavproxy`'nin yklemesi tamamlanır:

- `pip install MAVProxy`

EK 5. SITL KULLANILARAK SİMÜLASYON YAPILMASI

Yeni bir terminal sayfası açılarak Dronekit-SITL çalıştırılır. Dronekit-SITL komutundan sonra SITL'in kullanacağı hava aracının türü belirtilir. Döner kanatlı 4 rotorlu araç kullanacağımız için copter-3.3 seçeneği ile çalıştırılır. Gerektiğinde Şekil 46'daki gibi simülasyonun başlayacağı koordinatlar parametre olarak da verilebilir.

- dronekit-sitl copter-3.3

Şekil 46'da üst taraftaki terminalde görüldüğü gibi Dronekit-SITL 5760 nolu portta çalışmaktadır. 5760 nolu portta çalışan bu uygulama ile haberleşebilmek ve veri akışını sağlamak için Mavproxy kullanmamız gerekiyor.

- mavproxy.py --master tcp:127.0.0.1:5760 --out 127.0.0.1:14551 --out 127.0.0.1:14550

Yukarıdaki kod parçası ile Mavproxy lokaldeki 5760 nolu porta (Dronekit-SITL) bağlanıp, burada çalışan uygulamayı yine lokaldeki 14550 ve 14551 nolu portlara yönlendirmekte, böylece haberleşmelerini sağlamaktadır. 14550 nolu port, QGroundControl'ün Dronekit-SITL'e bağlanıp hava robotu ile bağlantı kurup veri alış verişi sağlaması için kullanılmaktadır. 14551 nolu port ise hava robotunu kontrol eden programın Dronekit-SITL'e bağlanıp, hava robotunu kontrol ettiği, veri aldığı porttur.

Bu aşamadan sonra Python kullanarak Dronekit ile geliştirilen programda Dronekit'in connect metodu kullanılarak program içerisinde hava robotu simülasyonuna bağlanılabilir.

- connect("udp:127.0.0.1:14551", wait_ready=True)

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Samet Özabacı

Uyruğu: T.C.

Doğum Tarihi ve Yeri: 26 Eylül 1992, Kastamonu

Elektronik Posta: sametozabaci@gmail.com

EĞİTİM

Derece	Kurum	Mezuniyet Yılı
Lisans	Sakarya Üniversitesi, Teknoloji Fakültesi Mekatronik Mühendisliği Bölümü	2016
Yüksek Lisans	İMU, Lisansüstü Eğitim Enstitüsü, Elektrik Elektronik Mühendisliği ABD	...

İŞ TECRÜBESİ

Tarih	Kurum	Görev
2016-2019	Türk Telekom	Test Mühendisi

YABANCI DİLLER

İleri düzeyde İngilizce, başlangıç düzeyinde Almanca