

T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**İNSANSIZ SU ÜSTÜ ARAÇLARININ YOL TAKİBİ İÇİN
KULLANILAN GÜDÜM YASALARININ OPTİMİZASYONU VE
YENİ BİR ALGORİTMANIN GELİŞTİRİLMESİ**

DOKTORA TEZİ

Osman ÜNAL

Enstitü Anabilim Dalı : MAKİNE MÜHENDİSLİĞİ

Tez Danışmanı : Doç. Dr. Nuri AKKAŞ

Ortak Danışman : Dr. Öğr. Üyesi Gökhan ATALI

Mayıs 2024

T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

İNSANSIZ SU ÜSTÜ ARAÇLARININ YOL TAKİBİ İÇİN
KULLANILAN GÜDÜM YASALARININ OPTİMİZASYONU VE
YENİ BİR ALGORİTMANIN GELİŞTİRİLMESİ

DOKTORA TEZİ

Osman ÜNAL

Enstitü Anabilim Dalı : MAKİNE MÜHENDİSLİĞİ

Bu tez 10/05/2024 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği ile kabul edilmiştir.

JÜRİ	BAŞARI DURUMU
Jüri Başkanı: Prof. Dr. Sinan Serdar ÖZKAN	BAŞARILI
Üye: Doç. Dr. Nuri AKKAŞ	BAŞARILI
Üye: Doç. Dr. Barış BORU	BAŞARILI
Üye: Doç. Dr. Mehmet ÖNAL	BAŞARILI
Üye: Doç. Dr. Hüseyin ELÇİÇEK	BAŞARILI

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Osman ÜNAL
10/05/2024

TEŞEKKÜR

Doktora eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren danışman hocalarım Doç. Dr. Nuri AKKAŞ ve Dr. Öğr. Üyesi Gökhan ATALI' yla birlikte, fikirleriyle çalışmamı geliştiren ve destek olan çok değerli hocam Prof. Dr. Sinan Serdar ÖZKAN'a ayrı ayrı teşekkürlerimi sunarım.

Ayrıca bu çalışmanın maddi açıdan desteklenmesine olanak sağlayan Sakarya Uygulamalı Bilimler Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığına (Proje No: 132-2023) teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
KISALTMALAR	v
SİMGELER	vi
TABLolar LİSTESİ.....	vii
ŞEKİLLER LİSTESİ.....	viii
ÖZET.....	x
ABSTRACT	xi
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
LİTERATÜR TARAMASI	4
2.1. Literatürde Yer Alan GÜDÜM Yasaları	5
2.2. Literatürde Yer Alan Optimizasyon Yöntemleri.....	11
BÖLÜM 3.	
İNSANSIZ SU ÜSTÜ ARACININ OPTİMİZASYON YÖNTEMLERİ KULLANARAK GÜDÜMÜ	16
3.1. Desen Arama Optimizasyon Yönteminin Havuç Kovalama GÜDÜM Yasasına Uyarlanması	16
3.2. Benzetimli Tavlama Optimizasyon Yönteminin Doğrusal Olmayan GÜDÜM Yasasına Uyarlanması	26
3.3. Vekil Model Optimizasyon Yönteminin Vektör Alanı GÜDÜM Yasasına Uyarlanması	39
3.4. Muhakeme Esaslı GÜDÜM Yasası.....	47

BÖLÜM 4.

ARDUİNO TABANLI BİR İNSANSIZ SU ÜSTÜ ARACININ TASARIMI..... 51

4.1. İSÜA' nın Dış Tasarımı	51
4.1.1. Fırçasız motorlar	51
4.1.2. Fırçasız motor sürücüleri	53
4.1.3. Tekne gövdesi	55
4.2. İSÜA' nın İç Tasarımı	58
4.2.1. Arduino nano.....	60
4.2.3. NRF24L01 2.4 GHz kablosuz haberleşme modülü	62
4.2.3. GY-NEO6MV2 GPS modülü	65
4.2.4. QMC5883 GY-271 3 eksen dijital pusula sensörü	66
4.3. Kablosuz Haberleşme Sistemi.....	67

BÖLÜM 5.

İNSANSIZ SU ÜSTÜ ARACININ DENETİMİ..... 70

5.1. Klasik PID ile İSÜA' nın Denetimi	74
5.2. Yüksek Mertebeden Sayısal İntegral ve Türev Alma Yöntemleri Kullanarak İSÜA' nın Denetimi	78

BÖLÜM 6.

BULGULAR VE TARTIŞMA..... 82

6.1. Sayısal Benzetimden Elde Edilen Bulgular	82
6.2. Deneysel Çalışmadan Elde Edilen Bulgular	87

BÖLÜM 7.

SONUÇ VE ÖNERİLER..... 93

KAYNAKLAR 96

EKLER 102

EK A. İSÜA' nın Kablosuz Haberleşme Sistemi için Gerekli Açık Kaynak Kodlar.....	102
---	-----

EK B. GY-NEO6MV2 GPS Modülünün Açık Kaynak Kodları.....	106
EK C. QMC5883 GY-271 3 Eksen Dijital Pusulanın Açık Kaynak Kodları	109
EK D. NRF24L01 2.4 Ghz Kablosuz Haberleşme Modülünün Alıcısı için Kullanılan Açık Kaynak Kodu	110
EK E. NRF24L01 2.4 Ghz Kablosuz Haberleşme Modülünün Vericisi için Kullanılan Açık Kaynak Kodu	113
EK F. PID Kontrolcü için Kullanılan Açık Kaynak Kod.....	116
EK G. Tasarlanan İSÜA' nın Test Sürüşü Verileri	117



KISALTMALAR

BTOY	: Benzetimli Tavlama Optimizasyon Yöntemi
DAOY	: Desen Arama Optimizasyon Yöntemi
DOGY	: Doğrusal Olmayan GÜdüm Yasası
GAOY	: Genetik Algoritma Optimizasyon Yöntemi
HKGY	: Havuç Kovalama GÜdüm Yasası
İSÜA	: İnsansız Su Üstü Araçları
MEGY	: Muhakeme Esaslı GÜdüm Yasası
VAGY	: Vektör Alanı GÜdüm Yasası
VMOY	: Vekil Model Optimizasyon Yöntemi

SİMGELER

d	: Yol İzleme Hatası
dt	: Sayısal Benzetimde Kullanılan Zaman Adımı Büyüklüğü
p	: İSÜA' nın Konumu
v	: İSÜA' nın Hızı
W_i	: İSÜA' nın Temas Noktaları
δ	: Havuç Kovalama Optimizasyon Yönteminin Kritik Parametresi
φ	: Referans Yolun Eğiminin Açı Değeri
Ψ	: İSÜA' nın Rota Açısı
$\Psi_{\max}$	: İSÜA' nın Maksimum Dönüş Hızı

TABLolar LİSTESİ

Tablo 3.1 : Akıllı vektör tabanlı güdüm yasası kullanılarak yol izleme hatalarının düşürülmesi..	46
Tablo 3.2 : Uyarlamalı zaman adımlarını kullanan MEGY' nın literatürdeki diğer güdüm yasaları ile karşılaştırılması.....	50
Tablo 6.1 : Uyarlamalı zaman adımlarını kullanan MEGY' nın literatürdeki diğer güdüm yasaları ve optimizasyon yöntemleri ile karşılaştırılması.	83
Tablo 6.2 : Tasarlanan İSÜA' nın test sürüşü verilerinden bir kısmı	88
Tablo G.1 : Tasarlanan İSÜA' nın test sürüşünden elde edilen veriler	117

ŞEKİLLER LİSTESİ

Şekil 2.1 : Havuç kovalama algoritmasının iki boyutlu geometrik yapısı	6
Şekil 2.2 : Doğrusal olmayan güdüm yasasının iki boyutlu geometrik yapısı.....	7
Şekil 2.3 : Doğrusal olmayan güdüm yasasının türettiği küçük yarıçaplı (r) daireler nedeniyle saldırgan bir navigasyon.....	8
Şekil 2.4 : Doğrusal olmayan güdüm yasasının türettiği büyük yarıçaplı (r) daireler nedeniyle miskin bir navigasyon.....	9
Şekil 2.5 : Desen arama optimizasyon yöntemi	12
Şekil 2.6 : Vekil model optimizasyon yönteminin akış şeması	14
Şekil 3.1 : $\delta=4$ metre olması durumunda tutarsız bir navigasyon.....	19
Şekil 3.2 : $\delta=11$ metre olması durumunda tutarlı ve kararlı bir navigasyon.....	19
Şekil 3.3 : Küçük δ değeri nedeniyle agresif navigasyon.	22
Şekil 3.4 : Büyük δ değeri nedeniyle miskin navigasyon	22
Şekil 3.5 : İlk görev için optimal δ değerinin aranması	25
Şekil 3.6 : İkinci görev için optimal δ değerinin aranması.	26
Şekil 3.7 : ADTDOGY algoritmasının ürettiği optimum dairelerin netice verdiği kararlı bir navigasyon.....	38
Şekil 3.8 : Klasik vektör alanı güdüm yasası.	39
Şekil 3.9 : Modifiye edilmiş vektör alanı güdüm yasası.....	40
Şekil 3.10 : İlk görev için vektör alanı güdüm yasasında yer alan optimum ϕ_e , d_B , k ve α değerlerinin aranması.	45
Şekil 3.11 : İkinci görev için vektör alanı güdüm yasasında yer alan optimum ϕ_e , d_B , k ve α değerlerinin aranması.	46
Şekil 3.12 : Uyarlamalı zaman adımı boyutları üzerine temellendirilmiş Muhakeme Esaslı Güdüm Yasası (MEGY)	48
Şekil 3.13 : Sabit zaman adımları kullanan havuç kovalama güdüm yasası.....	49
Şekil 4.1 : Yüksek devirli fırçasız motor ve pervanesi	52
Şekil 4.2 : Fırçasız motor sürücüsünün iç yapısı	54
Şekil 4.3 : Fırçasız motor sürücüsünün mikroişlemci ve fırçasız motorlar ile bağlantısının devre şeması	55
Şekil 4.4 : Kablosuz haberleşme için kullanılan harici antenler	56

Şekil 4.5 : Tasarlanan İSÜA' nın üst güvertesi.....	57
Şekil 4.6 : Tasarlanan İSÜA' nın alt gövdesi.....	58
Şekil 4.7 : Tasarlanan İSÜA' nın iç yapısının görünümü	59
Şekil 4.8 : İç tasarımda yer alan elektronik kartların devre şeması	60
Şekil 4.9 : Arduino nano kartının ön yüzü	61
Şekil 4.10 : Arduino nano kartının arka yüzü	62
Şekil 4.11 : NRF24L01 2.4 GHz kablosuz haberleşme modülü	63
Şekil 4.12 : NRF24L01 2.4 GHz kablosuz haberleşme modülü için adaptör	64
Şekil 4.13 : GY-NEO6MV2 GPS modülü	65
Şekil 4.14 : QMC5883 GY-271 3 eksen dijital pusula sensörü	67
Şekil 4.15 : Tasarlanan İSÜA' nın kumanda ve harici bilgisayar ile haberleşme sistemi.....	68
Şekil 5.1 : Arduino nano ve harici sistemler arasındaki veri akışı.....	71
Şekil 5.2 : Arduino nano mikroişlemcinin iç yapısında gerçekleşen veri akışı	72
Şekil 5.3 : PID denetleyicinin iç yapısı.....	74
Şekil 5.4 : Birinci mertebeden sayısal integral alma tekniği.....	76
Şekil 5.5 : Birinci mertebeden sayısal türev alma tekniği.....	77
Şekil 5.6 : Yüksek mertebeden sayısal integral alma tekniği	79
Şekil 5.7 : Yüksek mertebeden sayısal türev alma yönteminde kullanılan değerler..	80
Şekil 6.1 : Vekil model optimizasyon yönteminin uyarlamalı zaman adımlarını kullanan muhakeme esaslı güdüm yasasına uygulanması.....	84
Şekil 6.2 : Genetik algoritmanın havuç kovalama güdüm yasasına uyarlanması neticesinde optimizasyon yönteminin yerel minimum noktaya takılması.....	85
Şekil 6.3 : Tasarlanan İSÜA' nın test sürüşünün yapıldığı havuz	87
Şekil 6.4 : Tasarlanan İSÜA' nın test sürüşü	92

İNSANSIZ SU ÜSTÜ ARAÇLARININ YOL TAKİBİ İÇİN KULLANILAN GÜDÜM YASALARININ OPTİMİZASYONU VE YENİ BİR ALGORİTMANIN GELİŞTİRİLMESİ

ÖZET

Bu çalışmada, Arduino tabanlı bir İnsansız Su Üstü Aracı (İSÜA) tasarlanmıştır. İSÜA'nın tasarımında, uydudan konum verisi almak için GY-NEO6MV2 GPS modülü, anlık olarak rota açısını belirlemek için QMC5883 GY-271 3 eksen pusula sensörü ve çift yönlü veri aktarımı yapabilmek için NRF24L01 2.4 GHz kablosuz haberleşme modülü kullanılmıştır. İSÜA'nın hareketini sağlayan iki adet fırçasız motor, ısınmalarını engellemek amacıyla teknenin su ile doğrudan temas eden dış kısmına entegre edilmiştir. İSÜA'nın optimal güdümü ve denetimi için literatürde yer alan yol takip algoritmalarının sayısal benzetimi yapılmıştır. Havuç kovalama güdüm yasasına (carrot chasing guidance law), doğrusal olmayan güdüm yasasına (nonlinear guidance law) ve vektör alanı güdüm yasasına (vector field guidance law) ait kritik parametreleri optimize edebilmek için genetik algoritma (genetic algorithm), desen arama (pattern search), benzetimli tavlama (simulated annealing) ve vekil model (surrogate) optimizasyon teknikleri ayrı ayrı kullanılmıştır. Böylece yol takip hataları minimuma indirilerek farklı optimizasyon yöntemlerinin mevcut güdüm yasaları üzerindeki performansı incelenmiştir. Ayrıca Arduino tabanlı kartlara kolayca gömülebilen, etkili ve kararlı yeni bir güdüm yasası tasarlanmıştır. Muhakeme esaslı güdüm yasası (MEGY) olarak isimlendirilen bu yeni yol takip algoritması, İSÜA'nın belirlenen rotadan uzaklığını ve rotadan sapma açısını göz önünde bulundurarak yol takip hatalarını diğer yöntemlere göre daha çok azaltmaktadır. Küresel optimizasyon yöntemleri, geliştirilen MEGY'ne entegre edilerek hangi optimizasyon yönteminin en az yol izleme hatasını verdiği belirlenmiştir. Ayrıca MEGY, kullandığı uyarlamalı zaman adımları sayesinde optimizasyonun ve sayısal benzetimin tamamlanması için geçen süreyi klasik yöntemlere göre ciddi oranda düşürmektedir. Bunun yanı sıra, yüksek mertebeden sayısal türev alma ve yüksek mertebeden sayısal integral alma teknikleri, literatürde yaygın olarak kullanılan PID denetleyicisine uyarlanarak sayısal hatalar azaltılmıştır. Bu çalışmada geliştirilen güdüm yasası ve denetleyici, Arduino tabanlı İSÜA'ya gömülmüştür. Dijital pusula sensöründen gelen anlık rota açısı ile GPS modülünden gelen konum bilgisi giriş verileri ve fırçasız motorların denetiminin sağlandığı PWM 5 ve 6 pinleri çıkış verileri olacak şekilde kapalı bir döngü oluşturulmuştur. Son olarak tasarımı, güdümü ve denetimi tamamlanan İSÜA'nın test sürüşleri yapılarak, gerçek hayatta uygulanabilirliği kanıtlanmıştır.

Anahtar Kelimeler: İnsansız Su Üstü Araçları, Güdüm Yasaları, Küresel Optimizasyon Yöntemleri, Yüksek Mertebeden Türev Alma Teknikleri

OPTIMIZATION OF GUIDANCE LAWS USED FOR PATH FOLLOWING OF UNMANNED SURFACE VEHICLES AND DEVELOPMENT OF A NOVEL ALGORITHM

ABSTRACT

In this study, an Arduino-based Unmanned Surface Vehicle (USV) was designed. GY-NEO6MV2 GPS module to receive location data from the satellite, QMC5883 GY-271 3-axis compass sensor to instantly determine the route angle, and NRF24L01 2.4 GHz wireless communication module to provide bi-directional data transfer were used in the designed USV. The two brushless dc motors that move the USV are integrated to the outer part of the boat that comes into direct contact with the sea water in order to prevent them from overheating. Numerical simulations of path following algorithms in the literature have been developed for optimal guidance and control of the USV. Genetic algorithm, pattern search, simulated annealing and surrogate optimization techniques were separately used to optimize the critical parameters of the carrot chasing guidance law, nonlinear guidance law and vector field guidance law. Thus, cross tracking errors were minimized and the performance of different optimization methods on traditional guidance laws was examined. Additionally, a novel effective and stable guidance law that can be easily embedded into Arduino-based boards has been designed. This novel path following algorithm, called the discernment-based guidance law (DBGL), reduces cross tracking errors more than other methods by taking into account the distance of the USV from the predefined path and the angle of deviation from the route. Global optimization methods were integrated into the developed DBGL and it was determined which optimization method gave the least cross tracking error. In addition, DBGL significantly reduces the time taken to complete the optimization and numerical simulation compared to classical methods, owing to the adaptive time steps it uses. In addition, numerical errors have been reduced by adapting higher-order numerical derivative and higher-order numerical integration techniques to the PID controller, which is widely used in the literature. The guidance law and controller developed in this study have been embedded in the Arduino-based USV. A closed loop has been completed with the input data (instantaneous course angle coming from the digital compass sensor and the location information coming from the GPS module) and the output data (PWM pins 5 and 6 which control the brushless motors). Finally, the USV, whose design, guidance and control was completed, has been tested and its applicability in real life has been proven.

Keywords: Unmanned Surface Vehicles, Guidance Laws, Global Optimization Methods, Higher Order Derivative Techniques

BÖLÜM 1. GİRİŞ

Günümüz teknolojisinin her geçen gün daha da ilerlemesiyle birlikte, insansız araçların sivil ve askeri sahalarda kullanımı giderek artmaktadır. Son zamanlarda, insansız hava araçları, özellikle askeri alanda çok çeşitli kritik görevleri yerine getirerek, insansız araçların yaşamımızdaki önemini ortaya koymuştur. Diğer yandan insansız deniz araçları da, yüksek kapasiteli yük taşıyabilme ve çok az enerji tüketerek uzun süre görevde kalma gibi önemli avantajlara sahip olduklarından, hayatımızda en az insansız hava araçları kadar ehemmiyetlidirler.

İnsansız deniz araçları, su altı araçları ve su üstü araçları olmak üzere temelde iki kısma ayrılırlar. İnsansız su altı araçları, hassas sensörleri ile deniz altında üç boyutlu olarak hareket edebilen deniz araçlarıdır. İnsansız Su Üstü Araçları (İSÜA) ise, su yüzeyinde ilgili kritik operasyonları otonom olarak gerçekleştirebilen deniz araçlarıdır. Günümüzde, İSÜA hassas otonom yetenekleri, yüksek manevra kabiliyetleri, küçük boyutları, düşük maliyetleri, gizli operasyonları güvenle yerine getirebilme gibi özellikleri sebebiyle birçok alanda önemli görevler üstlenmektedir. Deniz ve okyanuslarda bulunan keşfedilmemiş kaynakların incelenmesinde, liman ve sahillerin güvenlik açısından denetlenmesinde, deniz kirliliğinin araştırılması ve kontrol edilmesinde, deniz kazaları olması durumunda yük ve insanların kurtarılmasında, savaş durumunda düşmana karşı deniz yoluyla caydırıcı ve etkin bir taaruzda İSÜA tercih edilmektedir [1-7]. İSÜA' nın yukarıda baheseldiği gibi birçok alanda kritik öneme sahip olması, bu konunun tercih edilme motivasyonu olmuştur.

İSÜA' nın otonom hareketi için kullanılan kontrol stratejileri, deniz araçlarının yerine getirdikleri göreve göre farklılık göstermektedir. Zaman kısıtlaması olmadığından dolayı, İSÜA' nın otonom hareketinde en yaygın olarak kullanılan kontrol stratejilerinden biri yol takip algoritmalarıdır. Yol takip algoritmaları navigasyon, güdüm ve denetim olmak üzere üç alt sistemden oluşmaktadır. İSÜA, bu üç sistemin birbiriyle sürekli veri-alış verişi yaparak uyum içinde çalışmasıyla otonom hareketini

gerçekleştirir. Navigasyon, uydudan ve çevreden konum bilgisi, rota açısı gibi verilerin alınmasını sağlar. GÜDÜM, navigasyondan aldığı verileri kullanarak İSÜA' nın referans yola kilitlenmesi için denetim sistemine komutlar verir. Denetim ise güdüm yasası ve deniz aracının dümen veya sevk sistemi ile köprü kurarak, İSÜA' nın istenilen manevraları yapmasını sağlar. Yol takip algoritmasına ait bu üç alt sistem incelendiğinde güdüm yasası deniz aracında, beyin görevini üstlenmektedir.

İSÜA' nın güdüm yasaları ile ilgili mevcut çalışmalar incelendiğinde, havuç kovalama güdüm yasası (carrot chasing), doğrusal olmayan güdüm yasası (nonlinear), mutlak hedef ve görüş hattı güdüm yasası (pure pursuit and line-of-sight, plos) ve vektör alanı güdüm yasası (vector field) literatürde en yaygın olarak kullanılan yöntemler olduğu görülmüştür. İncelenen mevcut güdüm yasalarının hepsi, kullanıcı tarafından bir veya birden daha fazla deterministik olmayan bazı parametrelerin tanımlanmasını gerektirmektedir. GÜDÜM algoritmaları için kritik öneme sahip bu deterministik olmayan parametreler için sabit sayı yada deneme yanılma yöntemi ile belirlenen değerler kullanıldığı görülmüştür [8-13]. Kararlı ve minimum hata ile yol takibinin sağlanması için bu kritik değer, İSÜA' nın fiziksel özelliğine ve takip edilmesi istenilen referans yolu oluşturan hedef noktalarının sayısına aynı zamanda konumuna göre farklılık göstermektedir. Ayrıca her bir güdüm yasası için de bu kritik parametreler farklılık göstermektedir. Farklı manevra kabiliyetine ve farklı hızlara sahip tüm gemi tiplerinin farklı referans yollarında ideal ve kararlı yol takibini sağlamaları için deneme yanılma yöntemi ile bu kritik parametrelerinin bulunması çok zordur.

Bu çalışmanın temel amaçlarından birincisi, literatürde yer alan güdüm yasalarına farklı optimizasyon yöntemleri entegre ederek deterministik olmayan bu kritik parametreleri optimize etmek ve yol izleme hatalarını en aza indirmektir. Herbir güdüm yasasının sahip olduğu deterministik olmayan parametrelerin en uygun ve ideal değerlerini bulmak için literatürde en sık kullanılan genetik algoritma (genetic algorithm), desen arama (pattern search), benzetimli tavlama (simulated annealing) ve vekil model (surrogate) isimli optimizasyon yöntemleri kullanılmıştır. Optimizasyon yöntemleri ile birleştirilmiş bu yenilikçi yol takip algoritmaları, farklı gemi tiplerinin değişik yol haritaları için güdüm yasaları içerisinde yer alan kritik parametreleri otomatik olarak optimize edebilmektedir.

Bu tezin ikinci amacı, Arduino gibi programlama kartlarına bile kolaylıkla gömülebilen ve düşük hesaplama maliyetini netice veren özgün bir güdüm yasası geliştirmektir.

Geliştirilen bu özgün güdüm yasasına Muhakeme Esaslı Güdüm Yasası (MEGY) ismi verilmiştir. Geliştirilen bu güdüm yasasına da optimizasyon teknikleri uygulanarak kendisine özgü olan kritik parametreler optimize edilmiştir.

Bu çalışmanın üçüncü amacı, sayısal benzetimlerde sıklıkla karşılaşılan kararsız ve osilasyonlu navigasyona sebep olan problemleri ortadan kaldıracak bir stabilite şartının geliştirilmesidir. Geliştirilen bu stabilite şartı sayesinde, İSÜA' nın hızına ve sayısal benzetimde kullanılan zaman adımına göre güdüm algoritmalarında kullanılan kritik parametrenin alt limitleri tesbit edilebilmektedir. Böylece kararlı bir seyir elde edilmekle birlikte osilasyonun sebep olabileceği çapraz izleme hatalarının önüne de geçilebilmektedir.

Bu çalışmanın dördüncü amacı, arduino tabanlı, uydudan aldığı konum verilerini kullanabilen, gerektiği zaman merkezi kontrol birimi ile veri alış-verişinde bulunabilen çift fırçasız motor ile tahrikli bir İSÜA' nın prototip modelini üretmektir. Tasarlanan bu İSÜA, önceden kullanıcı tarafından belirlenen su üstündeki hedef noktalarını dışarıdan müdahale olmadan otonom olarak yakalayabilmektedir.

Bu çalışmanın beşinci ve son amacı ise, İSÜA' rının denetimini sağlayan klasik kontrolcülere yüksek mertebeden türev alma yöntemlerini [14-17] uygulayarak hesaplamarda meydana gelen sayısal hataları azaltmaktır.

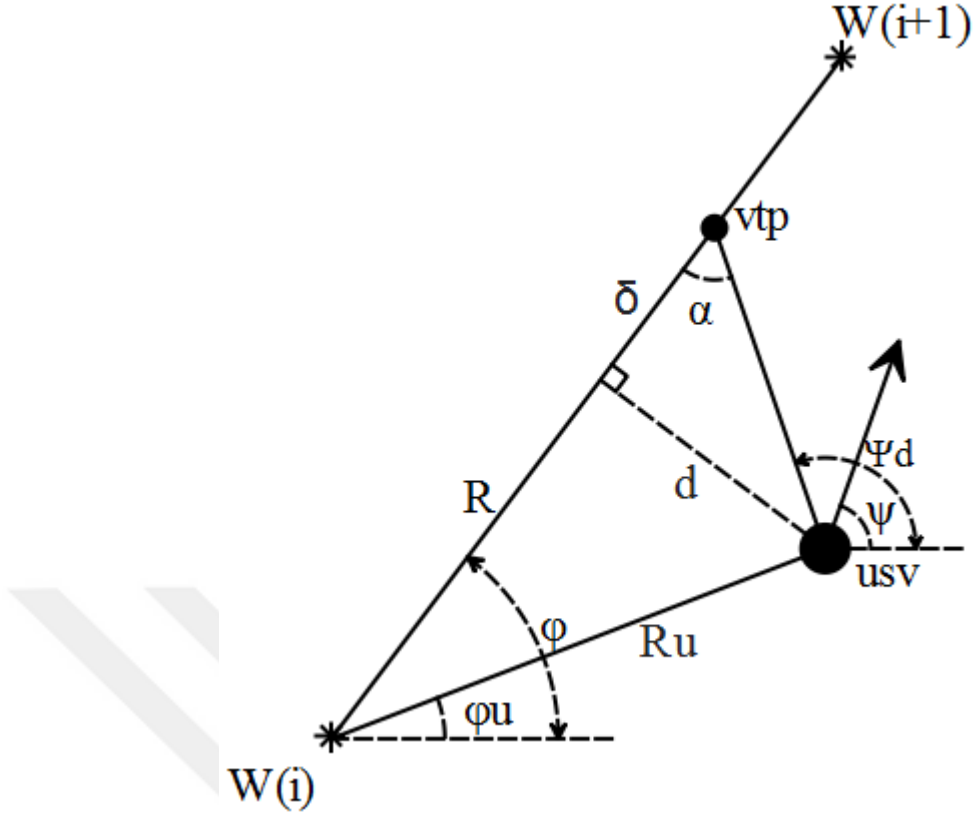
BÖLÜM 2. LİTERATÜR TARAMASI

Literatürde İSÜA' nın hareket kontrolü için yaygın olarak kullanılan üç farklı strateji bulunmaktadır. Bunlar yol izleme, yörünge izleme ve nokta stabilizasyonu yöntemleridir [18]. Yol izleme algoritması, çapraz izleme hatalarını ve rota açısı sapmalarını en aza indirmek için deniz aracını referans yolu takip etmeye zorlar. Yol izleme algoritmasında zaman dikkate alınmaz [19,20]. Ancak yörünge takip algoritmasında zamansal doğruluk önemli bir rol oynar ve bu algoritmanın temel amacı deniz aracının önceden belirlenen referans yol üzerinde bulunan her noktayı istenilen zamanlarda yakalamasıdır [21,22]. Nokta stabilizasyon algoritması ise deniz aracının konumunu ve rota açısını, zaman sınırlaması olmaksızın önceden tanımlanmış referans konum ve açısına ulaşmasını sağlar. Sondaj gemilerinin ve otonom kenetlenmeye ihtiyaç duyan deniz araçlarının sabit hedef görevleri için bu algoritma kullanılmaktadır [23,24]. Son zamanlarda yol izleme ile ilgili çalışmalar, zaman kısıtlaması olmaması nedeniyle birçok araştırmacının ilgisini çekerek çalışma konusu olarak tercih edilme sebebi olmuştur [25-31]. Yol izleme algoritmaları, istenilen referans yolu doğru bir şekilde takip edebilmek için navigasyon, güdüm ve denetleyici olmak üzere üç alt sistemden oluşur. Navigasyon, İSÜA' nın konumunu uydudan aldığı sinyallerle belirler ve İSÜA' nın çevresi ile ilgili verileri toplar [32]. Güdüm algoritmaları, referans yol üzerindeki optimum hedef noktalarını belirler ve navigasyon sistemi tarafından elde edilen verileri kullanarak önceden tanımlanmış referans yolu yakalamak için İSÜA' nın en iyi rota açısını ayarlamaya çalışır [33]. Denetleyici ise, güdüm sisteminin gönderdiği komutları göz önünde bulundurarak İSÜA' nı iki boyutta en az hata ile hareket ettirebilme rolünü üstlenir. Denetleyici, navigasyondan aldığı verileri girdi verileri olarak kullanarak, geminin motorlarına yada dümene çıktı komutlarını gönderir ve deniz aracının otonom kontrolünü sağlar [34]. Yol izleme algoritmasını oluşturan bu üç alt sistem arasında, önceden tanımlanmış referans yolu mümkün olduğu kadar küçük çapraz izleme hatasıyla yakalayabilmek için güdüm sistemi kritik bir öneme sahiptir.

2.1. Literatürde Yer Alan Güdüm Yasaları

Temelde, güdüm yasaları geometrik tabanlı yaklaşımlar [35,36] ve kontrol tabanlı yaklaşımlar [37] olmak üzere ikiye ayrılmaktadır. Geometrik tabanlı algoritmaların programlama kartları ile İSÜA' na uygulanması, kontrol tabanlı yaklaşımlara göre daha kolaydır. Havuç kovalama güdüm yasası (carrot chasing), doğrusal olmayan güdüm yasası (nonlinear), mutlak hedef ve görüş hattı güdüm yasası (pure pursuit and line-of-sight, plos) ve vektör alanı güdüm yasası (vector field) literatürde en sık kullanılan geometrik tabanlı yol izleme algoritmalarıdır [38-42]. Bu yöntemlerin hepsi sanal hedef noktaları belirlerler. İSÜA, önceden tanımlanmış referans yolu yakalamak ve nihai hedef noktaya ulaşmak için bu sanal hedef noktalarını takip eder.

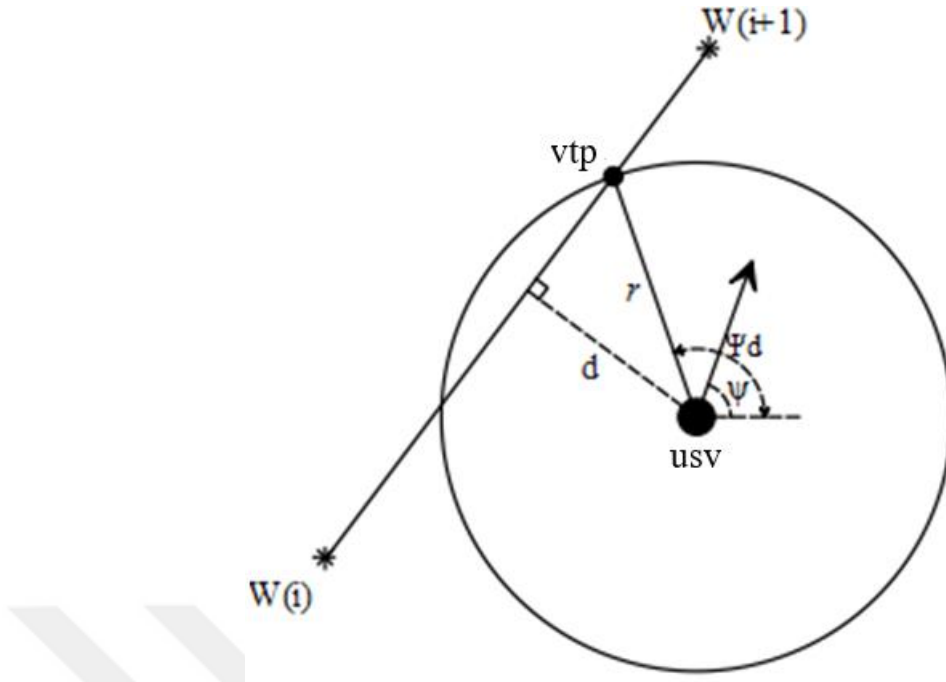
Havuç kovalama güdüm yasasında, tıpkı bir tavşanın bir havucu kovalaması gibi İSÜA da havuç kovalama algoritması tarafından önceden tanımlanmış yol üzerinde oluşturulan sanal hedef noktalarını sürekli olarak takip eder. Havuç kovalama güdüm yasası, İSÜA' nın önceden tanımlanmış referans yola en kısa mesafeyi oluşturan doğru ile referans yolunu oluşturan doğrunun kesişmesi ile oluşan noktadan hedef noktaya doğru kullanıcı tarafından belirlenen sabit bir mesafe eklenmesi ile belirlenen noktayı sanal hedef nokta varsayarak deniz aracının otonom seyrini sağlar [43]. İSÜA' nın rota açısı bu sanal hedef noktalarına göre ayarlanır. Bu nedenle, doğru ve kararlı bir navigasyonun sağlanabilmesi için sanal hedef noktalarının isabetli bir şekilde belirlenmesi gerekmektedir. Şekil 2.1' de havuç kovalama algoritmasının iki boyutlu geometrik yapısı gösterilmektedir. Şekil 2.1' de $W(i)$ önceki veya başlangıç hedef noktasını, $W(i + 1)$ ulaşmak istenilen hedef noktasını, $W(i)$ ile $W(i + 1)$ arasındaki mesafe önceden tanımlanmış yolu, noktasal usv sembolü insansız su üstü aracını temsil eder. vtp , havuç kovalama algoritması tarafından oluşturulan sanal hedef noktası anlamına gelir. Yol izleme hatası d harfi ile gösterilmiştir ve önceden tanımlanmış yol ile İSÜA arasındaki minimum mesafe olarak tanımlanır. Ψ , İSÜA' nın anlık rota açısını temsil eder. Ψ_d , sanal hedef noktası hedef alınarak yakalanılması istenilen rota açısıdır.



Şekil 2.1: Havuç kovalama algoritmasının iki boyutlu geometrik yapısı.

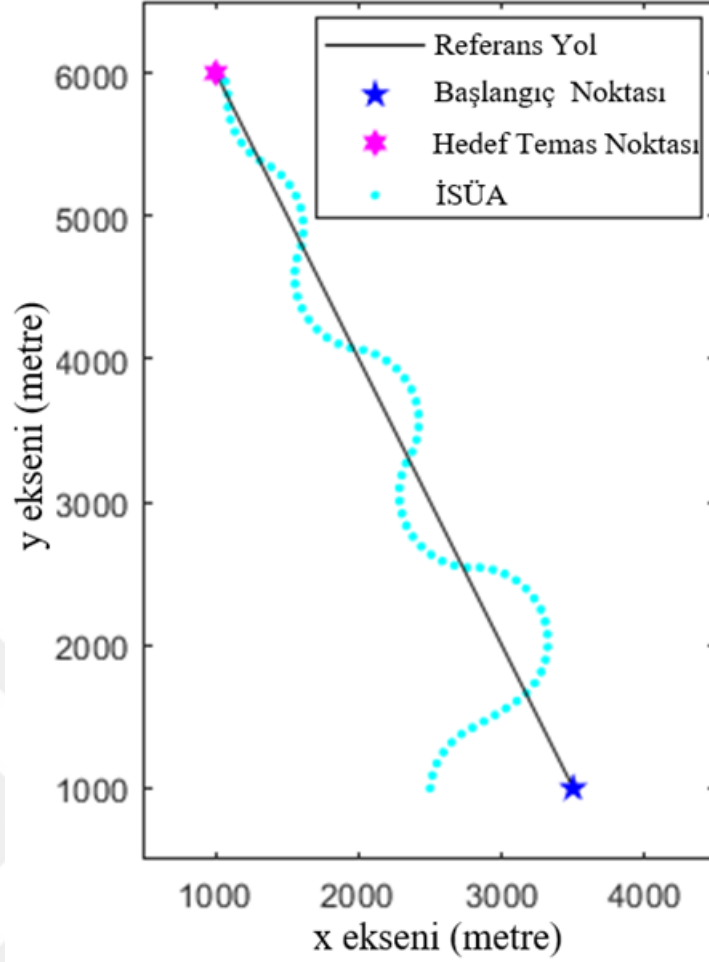
Havuç kovalama güdüm yasasının en temel amacı, yakalanılması istenen rota açısı ile İSÜA' nın anlık rota açısı arasındaki farkı azaltmak ve yol izleme hatalarını mümkün olan en kısa sürede minimuma indirmektir.

Doğrusal olmayan güdüm yasası, İSÜA' nın bulunduğu konumu merkez kabul eden çember oluşturur. Çember ile önceden tanımlanmış referans yolu oluşturan doğrunun kesişmesinden oluşan iki noktadan nihai varış noktasına yakın olan noktayı sanal hedef nokta olarak seçerek deniz aracının güdümünü tayin eder. Çember doğruyu teğet keserse tek kesişim sanal hedef noktası olur. Çemberin yarıçapının ne kadar olacağı kullanıcı tarafından belirlenir. Seçilen yarıçap, İSÜA' nın önceden tanımlanmış yola olan en kısa mesafeden daha kısa ise çember ile referans yol kesişmez ve sanal hedef noktaları oluşturulamaz. Bu durumda yol izleme algoritması çalışmaz ve deniz aracının otonom seyri sağlanamaz [44-47]. İSÜA' nın konumu her zaman adımında değiştiği için aynı yarıçapa sahip daire tekrar tekrar oluşturulur ve bu dairenin önceden tanımlanan referans yol ile kesişmesiyle oluşan yeni sanal hedef noktalar belirlenir. İSÜA bu sanal hedef noktaları takip ederek hedef temas noktaya ulaşmaya çalışır. Şekil 2.2 doğrusal olmayan güdüm yasasının geometrik şemasını göstermektedir.



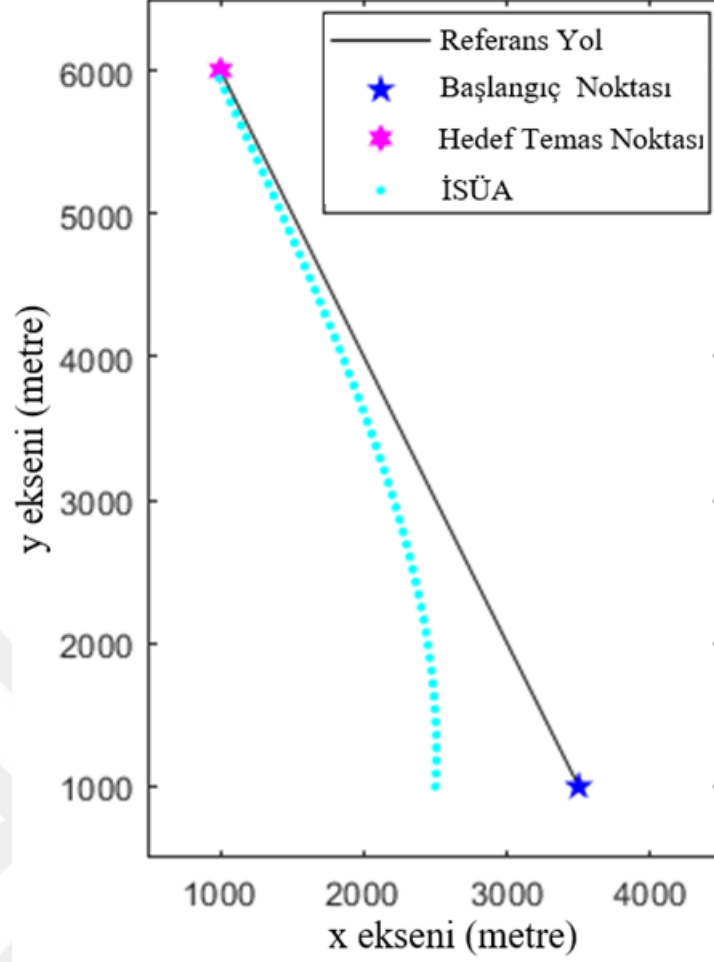
Şekil 2.2: Doğrusal olmayan güdüm yasasının iki boyutlu geometrik yapısı.

Şekil 2.2' de $W(i)$ ve $W(i+1)$ sırasıyla başlangıç noktasını ve hedef temas noktasını temsil etmektedir. Başlangıç ve hedef temas noktaları arasındaki çizgi önceden tanımlanmış referans yoldur. usv ve vtp, insansız su üstü aracı ve sanal hedef noktası anlamına gelir. d , r , Ψ ve Ψ_d sırasıyla yol izleme hatasını, dairenin yarıçapını (deterministik olmayan parametre), İSÜA' nın başlangıç rota açısını ve İSÜA' nın sanal hedef noktasına ulaşmak için istenen rota açısını sembolize eder. Doğrusal olmayan güdüm yasasının amacı, İSÜA' nın önceden tanımlanmış referans yolu hassas bir şekilde takip etmesi ve en az yol izleme hatasıyla en kısa sürede hedef temas noktaya ulaşmasıdır. İSÜA' nın rota açısı sanal hedef noktalarına göre değiştirilir. Bu sebeple, hassas ve güvenilir navigasyonu temin etmek için sanal hedef noktaları dikkatlice belirlenmelidir. Sanal hedef noktalarının konumu doğrudan güdüm yasasının oluşturduğu dairelerin yarıçapına bağlıdır. Bu nedenle, dairelerin yarıçapı (r), doğrusal olmayan güdüm yasasında anahtar parametredir. İSÜA' nın otonom seyrinin doğruluk hassasiyeti öncelikle buna bağlıdır.



Şekil 2.3: Doğrusal olmayan güdüm yasasının türettiği küçük yarıçaplı (r) daireler nedeniyle saldırgan bir navigasyon.

Eğer dairenin yarıçapı yol izleme hatasına eşitse veya dairenin yarıçapı çok küçükse (yol izleme hatasına yakınsa), Şekil 2.3' te olduğu gibi saldırgan bir navigasyon gözlemlenebilir. Bu durumlarda İSÜA önceden tanımlanmış referans yola hızlı bir şekilde ulaşır; ancak önceden tanımlanmış yolu kararlı bir şekilde takip edemez. İlk temastan hemen sonra önceden tanımlanmış yoldan ayrılır. Salınımlı hareket gözlemlenir. Salınımlı navigasyon ise toplamda büyük yol izleme hatalarına neden olur. Dairenin yarıçapı, yol izleme hatasıyla karşılaştırıldığında çok büyükse Şekil 2.4' te olduğu gibi miskin bir navigasyon elde edilir. İSÜA önceden tanımlanmış referans yolu hızlı bir şekilde yakalayamaz. Önceden tanımlanmış yolla ilk temas çok uzun zaman alır. Bu durum da yine toplamda büyük yol izleme hatasına yol açar. Bu sebeple, doğrusal olmayan güdüm yasasına girilen dairenin yarıçapının büyüklüğü, istikrarlı ve kararlı bir seyir için kritik öneme sahiptir.



Şekil 2.4: Doğrusal olmayan güdüm yasasının türettiği büyük yarıçaplı (r) daireler nedeniyle miskin bir navigasyon.

Mutlak hedef ve görüş hattı güdüm yasası, kullanıcı tarafından belirlenen iki farklı katsayıyı İSÜA' nın takip etmesi istenilen referans yola olan en kısa mesafesi ve İSÜA' nın hedef noktadan sapma açısı ile ayrı ayrı çarparak İSÜA' nın referans yolu takip edebilmesi için gerekli olan rota açısının belirlenmesini sağlar [48,49]. İSÜA' nın referans yola çok uzak olması durumunda kullanıcı tarafından belirlenen ilk katsayı ve referans yol ile İSÜA' nın arasındaki mesafenin çarpımı etkin rol oynayacağından, mutlak hedef ve görüş hattı güdüm yasası İSÜA' nı referans yolu bir an önce yakalamasına zorlayacaktır. İSÜA' nın nihai hedef noktasından sapma açısı çok fazla olması durumunda ise mutlak hedef ve görüş hattı güdüm yasası İSÜA' nın rota açısını referans yolun eğim açısına yaklaştırmaya çalışacaktır. Bu iki parametrenin (İSÜA' nın takip etmesi istenilen referans yola olan en kısa mesafesi ve İSÜA' nın hedef noktadan sapma açısı) ortak katkıları ve kullanıcı tarafından girilen iki katsayı göz önünde bulundurularak, İSÜA' nın referans yolu en az hata ile takip ederek istenilen hedef

noktasını yakalaması mutlak hedef ve görüş hattı güdüm yasası ile sağlanmaktadır. Rotadan sapma açısının ve aradaki mesafenin mutlak değerleri birbirinden uzak değerler olduğundan kullanıcı tarafından belirlenen bu iki katsayının doğru ve tutarlı seçilmesi, mutlak hedef ve görüş hattı güdüm yasasının düzgün çalışması için kritik bir öneme sahiptir. Aksi durumda, kararsız ve yanlış bir seyrin gözlemlenmesi kaçınılmazdır.

Geometrik yol izleme algoritmalarından biri olan vektör alanı güdüm yasası, etkin performansı ve programlama kartlarına uygulama kolaylığı nedeniyle birçok araştırmacı tarafından büyük bir ilgi görmüştür [50]. Vektör alanı güdüm yasası, kullanıcı tarafından tanımlanan dört adet parametreye ihtiyaç duyar. İlk parametre geçiş bölgesi sınırının mesafesidir. İSÜA' nın önceden tanımlanmış referans yoldan ne kadar uzak olduğunu belirlemek için kullanılır. İSÜA' nın önceden tanımlanmış referans yola dik mesafesi, sınır mesafeden büyük olduğunda (İSÜA önceden tanımlanmış referans yoldan çok uzaktadır), İSÜA önceden tanımlanmış referans yolu bir an önce yakalayabilmesi için rotasını mümkün olduğunca referans yola dik olacak şekilde ayarlamaya çalışır. Bu sayede yol izleme hataları en aza indirilir ve İSÜA' nın referans yolu hızlı bir şekilde yakalaması sağlanır. Sınır mesafesi olması gerekenden daha büyük seçilirse İSÜA, referans yolu yakalamak için olması gerekenden daha erken dönmeye başlayacaktır. Bu durum, yol izleme hatalarını önemli ölçüde artırmaktadır. İSÜA' nın önceden tanımlanmış referans yola olan dikey mesafesi sınır mesafesinden küçük veya ona eşit olduğunda (İSÜA önceden tanımlanmış yola yakınsa), İSÜA' nın rotası, yolu istikrarlı bir şekilde takip edecek şekilde kademeli olarak döndürülür. Sınır mesafesi olması gerekenden daha küçük seçilirse İSÜA kararlı bir şekilde dönemez ve önceden tanımlanmış yolun karşı cephesine geçer ve salınımlı bir navigasyon gözlemlenir. Bu nedenle sınır mesafesi, İSÜA' nın minimum yol izleme hatasıyla önceden tanımlanmış yolu tutarlı bir şekilde yakalayarak hedefe ulaşmasını sağlayan kritik bir parametredir.

Vektör alanı güdüm yasasının kullanıcı tarafından tanımlanan diğer üç parametresi de İSÜA' nın navigasyonunu etkiler. İkinci parametre, geçiş bölgesine giriş açısını temsil eder. Bu parametrenin değer aralığı 0 derece ile 90 derece arasındadır. Giriş açısı 90 derece seçilirse yol izleme hataları minimuma düşürülmüş olur ancak İSÜA geçiş bölgesine keskin bir şekilde girer ve referans yolu yakalaması zorlaşır. Giriş açısı küçüldükçe İSÜA geçiş bölgesine daha yumuşak manevra yaparak girer. Ancak bu durum büyük yol izleme hatalarına neden olabilir. Üçüncü ve dördüncü parametreler

İSÜA' nın geçiş bölgesinden referans yola doğru yaptığı dönüşün rota açısını kontrol eder.

İSÜA' nın rota açısı, geleneksel vektör alanı rehberlik yasasında yer alan Denklem 2.1 ve Denklem 2.2 kullanılarak belirlenmektedir. Denklem 2.1, İSÜA geçiş bölgesinin dışındayken ve Denklem 2.2 ise İSÜA geçiş bölgesinin içindeyken rota açısını hesaplamak için kullanılır [9,50].

$$\Psi_d = \varphi - \varphi_e \rho \quad (2.1)$$

$$\Psi_d = \varphi - \varphi_e \left(\frac{\rho d}{d_B} \right)^k - \left(\frac{k v \varphi_e}{\alpha d_B^k} \right) (\rho d)^{k-1} \sin(\Psi_p) \quad (2.2)$$

Denklem 2.1 ve 2.2' de yer alan φ önceden tanımlanmış yolun açısını temsil eder (başlangıç noktası ile hedef temas noktasını birleştiren doğrunun açısı). d_B , geçiş bölgesinin sınır mesafesidir (kullanıcı tarafından tanımlanması gereken ilk parametre). φ_e geçiş bölgesine giriş açısıdır (kullanıcı tarafından tanımlanması gereken ikinci parametre). α ve k , geçiş bölgesi içindeyken İSÜA' nın dönüşünü kontrol eden kullanıcı tarafından tanımlanan üçüncü ve dördüncü parametrelerdir. d yol izleme hatasıdır. v , İSÜA' nın hızıdır. Ψ_p önceki zaman adımıdaki İSÜA' nın rota açısını temsil eder. ρ , İSÜA' nın önceden tanımlanmış referans yolun hangi tarafında olduğunun bir göstergesidir. ρ -1, 0 veya 1 olabilir. Eğer İSÜA tam olarak referans yolun üzerinde ise bu değer 0 olacaktır. ρ aşağıdaki Denklem 2.3' e göre hesaplanır.

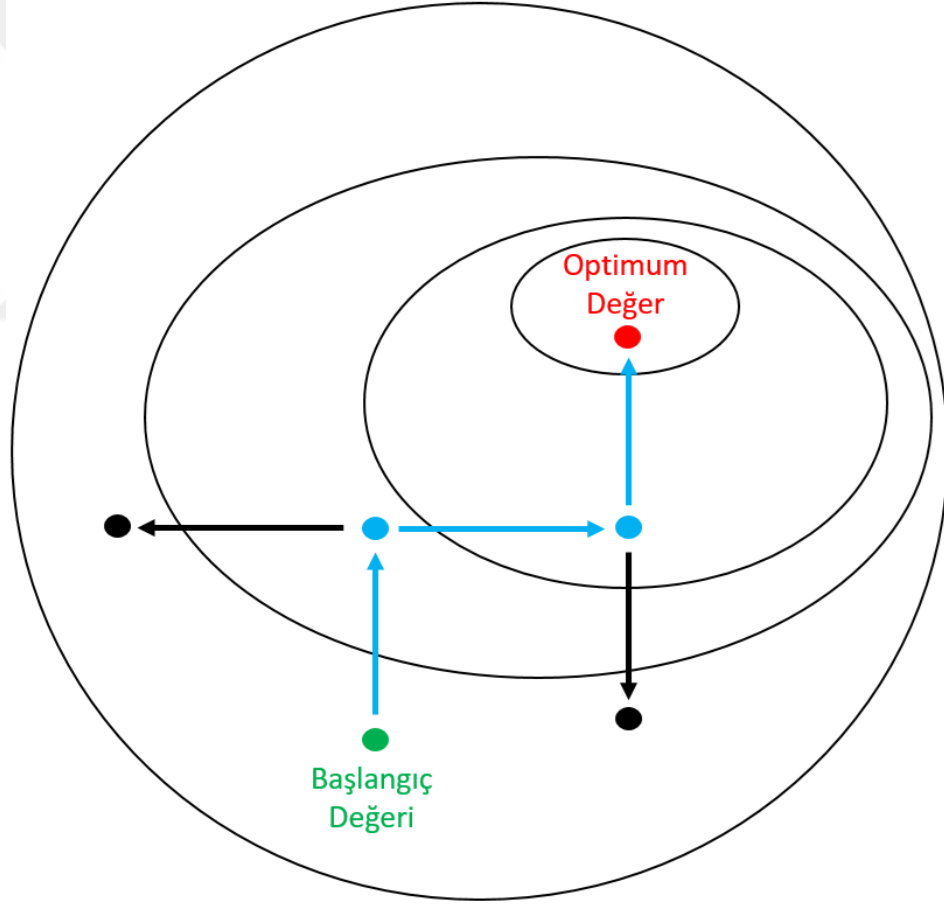
$$\rho = \text{sign}[(W_2 - W_1) \times (p - W_1)] \quad (2.3)$$

Denklem 2.3' te bulunan p , W_1 ve W_2 sırasıyla İSÜA' nın konumunu, ilk veya bir önceki temas noktasını ve hedef temas noktasını temsil eder.

2.2. Literatürde Yer Alan Optimizasyon Yöntemleri

Yukarıda bahsedilen literatürdeki güdüm yasaları kullanıcı tarafından belirlenen parametrelere ihtiyaç duyarlar. Havuç kovalama güdüm yasasında bir tane, doğrusal olmayan güdüm yasasında bir tane, mutlak hedef ve görüş hattı güdüm yasasında iki tane ve vektör alanı güdüm yasasında dört tane kritik parametrenin girilmesi gerekmektedir. Bu parametreler; deniz aracının hızı, maksimum manevra kabiliyeti, deniz aracının referans yola olan uzaklığı, hedef noktaların birbirine göre konumu ve

hedef noktaların sayısı gibi birçok durum göz önünde bulundurularak her bir güdüm yasasına özgü olarak seçilmelidir. Deneme yanılma yöntemi ile bu kritik parametreleri belirlemek zor olmakla birlikte ideal kararlı rotayı bu şekilde tutturmak neredeyse imkânsızdır. Bu durum ciddi çapraz izleme hatalarına sebep olmaktadır. Bu sebeple her bir güdüm yasası, parametrelerinin optimize edilmesine ihtiyaç duyar. Tek parametrelili havuç kovalama ve doğrusal olmayan güdüm yasası gibi algoritmaların parametre optimizasyonu kısa bir sürede yapılabilirken, vektör alanı güdüm yasası gibi çok parametrelili algoritmaların optimizasyonu uzun sürmekte ve hesaplama maliyetlerini önemli miktarda arttırabilmektedir. Literatürde desen arama (pattern search) [51], benzetimli tavlama (simulated annealing) [52], vekil model (surrogate) [53] ve genetik algoritma (genetic algorithm) [54] gibi çok sayıda optimizasyon teknikleri vardır.



Şekil 2.5: Desen arama optimizasyon yöntemi.

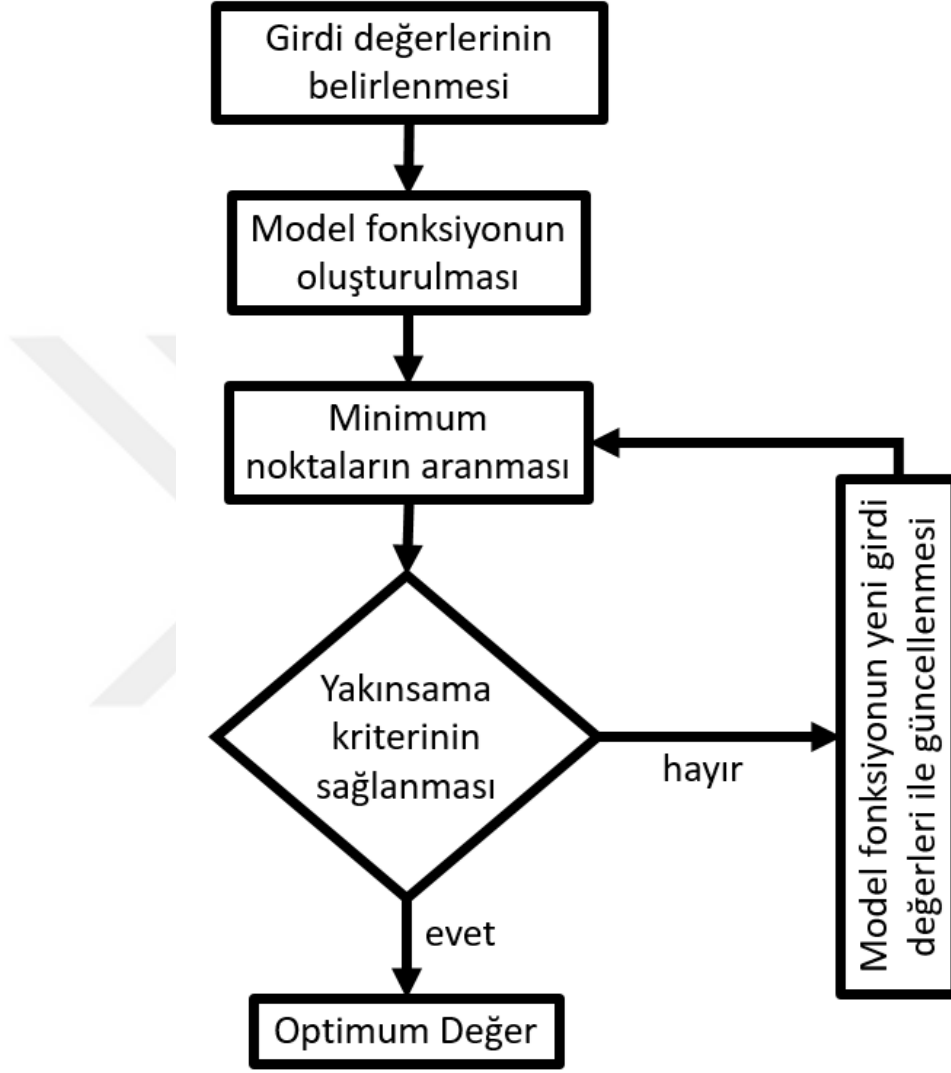
Desen arama optimizasyon yöntemi Hooke ve Jeeves tarafından ilk kez 1961 yılında literatüre girmiştir [55]. Desen arama, türevsiz bir optimizasyon yöntemidir ve hedef fonksiyonun türevlenebilirliğinin veya sürekliliğinin olmadığı durumlarda da

kullanılabilir [56]. Desen arama algoritması bir yakınsama teorisidir ve olasılıkların iki yada daha çok boyutlu bir analiz uzayında en iyi eşleşmeyi sağlayan yani en düşük hata değerine sahip çözümü bulmaya çalışır. Şekil 2.5, desen arama optimizasyon yöntemini göstermektedir. Şekil 2.5’ te yer alan yuvarlak kontur çizgileri, dışarıdan içeriye doğru değerlerin azaldığını ifade eder. En içeride bulunan kırmızı değer, minimum değerdir. Desen arama algoritması bu değere ulaşmaya çalışır. Algoritma öncelikle, Şekil 2.5’ te yeşil renk ile gösterilen bir başlangıç değeri oluşturur. Başlangıç noktasından, farklı noktalara dallanarak minimum değere yakanmaya çalışır. Algoritma, deneme noktalarından sayısal değeri küçük olana doğru hareket eder ve bu hareket Şekil 2.5’ te mavi renkle gösterilen başarılı adımdır. Büyük olan değerler, başarısız hamlelerdir ve bunlar Şekil 2.5’ te siyah renk ile gösterilmiştir.

Benzetimli tavlama yada benzetilmiş tavlama algoritması, metallerin tavlama işleminden esinlenerek [57] kullanıcı tarafından seçilen hedef fonksiyonunun küresel minimum noktasına yaklaşmak için tasarlanmış sezgisel bir optimizasyon tekniğidir [58-59]. Benzetimli tavlama, metallerin ısıtılması ve ardından yavaşça soğutulması temeline dayanır [60]. Isıtılan metallerin sıcaklığı azaldığında, metalin iç parçacıkları her sıcaklıkta bir denge durumuna gelir[61]. Benzetimli tavlama, ısı yükseldikçe, en küçük yerel minimum noktayı bulmak için komşu bölgelere yönelecektir. Yavaş yavaş sıcaklık azalmaya başladığında ise en düşük yerel minimum noktasında durmaya çalışacaktır [62]. Bu yaklaşım, hedef fonksiyonun küresel minimum noktasını isabetli bir şekilde bulunmasını sağlamaktadır [63].

Vekil model optimizasyon yönteminde, kullanıcı tarafından minimum noktası araştırılan hedef fonksiyonundan alınan veriler ile güncellenen temsili bir fonksiyon türetilir [64]. Mühendislik uygulamalarında kullanılan hedef fonksiyonları, fiziksel sistemlerin ölçümlerinden veya bilgisayar simülasyonlarından gelebilir. Çoğu durumda, bu tür hedef fonksiyonların doğrudan optimizasyonu, yani optimizasyon rutinlerinin doğrudan bu işlemlere uygulanması pratik değildir. Bunun sebeplerinden biri, simülasyona dayalı amaç fonksiyonlarının genellikle süreksiz ve türevlenemeyen fonksiyonlar olmasıdır. Diğer bir sebep ise, ölçüm veya simülasyonların yüksek hesaplama maliyetleridir. Mevcut teknoloji ile bilgisayarların bilgi işlem gücünün artmasına rağmen, mühendislikte kullanılan hedef fonksiyonlarının değerlendirilmesi ve işlemleri birkaç saat, gün ve hatta haftalar süren simülasyon sürelerini netice vermektedir. Bu süreksiz, türevlenemeyen ve yüksek hesaplama maliyeti gerektiren hedef fonksiyonların uygun

ve etkili bir şekilde minimum noktalarının bulunması, temsili bir fonksiyon türetilerek gerçekleştirilebilir. Vekil model optimizasyon algoritmasında, orijinal hedef fonksiyonunun optimizasyonunun yerini, hesaplama maliyeti açısından ucuz, sürekli ve türevi alınabilir temsili fonksiyonun optimizasyonu ve güncellenmesi alır [65].



Şekil 2.6: Vekil model optimizasyon yönteminin akış şeması.

Şekil 2.6, vekil model optimizasyon yönteminin akış şemasını göstermektedir. Algoritma öncelikle, kullanıcının belirlediği sınırlar içerisinde rastgele örnek girdi parametreleri oluşturur. Girdi parametreleri, hedef fonksiyona gönderilerek çıktı parametreleri hesaplanır. Elde edilen girdi ve çıktı değerleri kullanılarak interpolasyon ile veya başka yöntemler ile model fonksiyon oluşturulur. En düşük çıktı değerini netice veren girdi parametresine yakın yeni değerler türetilir. Bu değerler de hedef fonksiyonuna gönderilir ve yeni değerler ile model fonksiyon güncellenir. Güncellenen

model fonksiyonu minimum yapan deęerler araştırılır. Durdurma kriteri saęlanıncaya kadar anlatılan bu döngüler devam eder. Sonuçlar durdurma kriterini saęladığında, algoritma sona erer ve optimum deęer elde edilmiř olur.

Genetik algoritma bir biyolog ve mühendis olan Holland tarafından ilk kez 1975 yılında kullanılmıřtır [66]. Canlılardaki genetik kod prensibini kullanarak karmařık problemleri etkili bir řekilde çözümlmeyi amaçlamaktadır. Genetik algoritmalar, bir çözüml uzayındaki her noktayı, kromozom olarak adlandırılan ikili bit dizisi biçiminde kodlamaktadır. Bu noktaların her birine ait bir uygunluk deęeri vardır. Dolayısıyla, tek bir nokta yerine noktalar kümesini muhafaza etmektedir. Genetik algoritmalar, uygunlukların hesaplanması, çözümlerin kodlanması, seçim, mutasyon ve çaprazlama operatörlerinin uygulanması adımlarını içermektedir. Bu operatörlerden faydalanarak yeni bir popölasyon oluřturmaktadır [67]. Oluřan popölasyon ile çeřitlilik saęlanmış olup bu çeřitlilik arasından küresel minimum noktalar bulunmaya çalışılmaktadır.

Bu optimizasyon yöntemlerinin her biri yukarıda bahsedilen güdüm yasaları ile entegre edilerek parametre optimizasyonu yapılabilir. Vektör alanı gibi çoklu parametrelili güdüm yasalarına, yüksek hesaplama maliyetine sahip olan hedef fonksiyonlar için uyumlu olan vekil model optimizasyon yöntemini uygulamak daha isabetli olacaktır.

BÖLÜM 3. İNSANSIZ SU ÜSTÜ ARACININ OPTİMİZASYON YÖNTEMLERİ KULLANARAK GÜDÜMÜ

Bu bölümde İSÜA' nın otonom navigasyonu; havuç kovalama güdüm yasası (carrot chasing), doğrusal olmayan güdüm yasası (nonlinear) ve vektör alanı güdüm yasası (vector field) isimli algoritmalara optimizasyon yöntemleri uygulanarak geliştirilmiştir. Mevcut güdüm yasalarından esinlenilerek muhakeme esaslı güdüm yasası (MEGY) isimli özgün bir algoritma türetilmiştir.

3.1. Desen Arama Optimizasyon Yönteminin Havuç Kovalama Güdüm Yasasına Uyarlanması

Deneyisel çalışmalar maliyetli ve zaman aldığından, bu çalışmada incelenen güdüm yasaları sayısal benzinler ile sınanacaktır. İSÜA' nın sayısal benzetimlerinin yapılabilmesi bazı temel matematiksel denklemlerin türetilmesi gerekmektedir. İSÜA' nın dinamik modelinin ve kinematiğinin denklemleri türetilirken deniz aracının sabit bir hıza sahip olduğu ve dalga-rüzgar gibi dış kuvvetlerin olmadığı varsayılmaktadır. Ayrıca İSÜA' nın simetriktir ve homojen kütle dağılımına sahip olduğu düşünülmektedir. Yerçekimi kuvveti ve kaldırma kuvveti merkezleri aynı dikey çizgide bulunduğu varsayılmaktadır. Bu varsayımlar altında, İSÜA için iki boyutlu havuç kovalama algoritmasının dinamik modeli [68] ve kinematiği [69] sırasıyla aşağıdaki Denklem 3.1 ve denklem 3.2 ile ifade edilebilir:

$$\begin{aligned} M \frac{d^2 p_x}{dt^2} &= C_D \frac{1}{2} \rho [v \cos(\Psi)]^2 S \\ M \frac{d^2 p_y}{dt^2} &= C_D \frac{1}{2} \rho [v \sin(\Psi)]^2 S \end{aligned} \quad (3.1)$$

Denklem 3.1, Newton'un ikinci yasasına göre türetilmiştir. Denklem 3.1' de yer alan M , İSÜA' nın kütlesini ifade eder. p_x ve p_y , İSÜA' nın x ve y koordinatlarındaki konumunu temsil eder. p_x ve p_y terimlerinin ikinci türevleri ivme anlamına gelir. C_D , pervane açısının

ve Reynolds sayısının bir fonksiyonu olan sürüklenme katsayısıdır. ρ deniz suyu yoğunluğunun sembolüdür. Gerçek hız v ile temsil edilmektedir. $\cos d(\Psi)$ ve $\sin d(\Psi)$ derece cinsinden anlık rota açısının kosinüsü ve sinüsü anlamına gelir. S , pervanenin düzlem formundaki (iz düşüm) kanat alanıdır.

$$\begin{aligned} p_x^{t+1} &= p_x^t + v \cos d(\Psi) dt \\ p_y^{t+1} &= p_y^t + v \sin d(\Psi) dt \end{aligned} \quad (3.2)$$

Denklem 3.2' de bulunan p_x^{t+1} , p_x^t , p_y^{t+1} and p_y^t sembolleri sırasıyla; bir sonraki zaman adımı için İSÜA' nın x koordinatındaki konumunu, başlangıç veya mevcut zaman adımı için İSÜA' nın x koordinatındaki konumunu, bir sonraki zaman adımı için İSÜA' nın y koordinatındaki konumunu ve başlangıç veya mevcut zaman adımı için İSÜA' nın y koordinatındaki konumunu ifade eder. Denklem 3.2' deki dt terimi, sayısal benzetimin zaman adımını belirtir.

Bir önceki bölümde detaylı olarak anlatılan Şekil 2.1' deki δ , yol izleme algoritmaları için önemli bir parametredir ve deterministik değildir. Klasik havuç kovalama algoritmaları değişen seyir koşullarını ve İSÜA' nın değişen manevra veya hızını göz önünde bulundurmaksızın her zaman adımında tek ve sabit bir δ değerini kullanır. Ancak doğru ve kararlı bir navigasyon için δ parametresinin optimum değerinin İSÜA' nın hızına, konumuna, maksimum manevra kabiliyetine göre değişmesi gerekir. Bu nedenle δ parametresinin İSÜA' nın seyir koşullarına göre tüm benzersiz zaman adımlarında optimize edilmesi gerekmektedir. Bu çalışmanın temel amaçlarından biri, desen arama optimizasyon tekniğini havuç kovalama güdüm yasasına entegre ederek her bir zaman adımı için en uygun δ parametresini belirlemektir. Havuç kovalama güdüm yasasının ve desen arama optimizasyon yönteminin detaylarına geçmeden önce kararlı bir seyrin gerçekleşmesi için geliştirilen stabilite kriterinin tanımlanmasına ihtiyaç duyulmuştur. Bu çalışma kapsamında geliştirilen stabilite kriteri, sayısal benzetimde kullanılan zaman adımının ve deniz aracının hızına bağlı olarak seçilebilecek δ parametresinin alt limitinin ne olacağının belirlenmesini sağlar. Bu alt limit, optimizasyon algoritmasında kullanılarak optimum δ parametresinin daha kısa sürede ve daha tutarlı bir şekilde belirlenmesini sağlayacaktır. Ayrıca, sayısal benzetim yapılırken stabilite kriterinin belirttiği minimum δ değerinden daha küçük değerlerin seçilmesi, gerçek dışı bir seyrin gözlemlenmesine sebep olabilir. Bu çalışma,

gerçekçi bir seyrin sayısal benzetimini temin eden stabilite kriterini Denklem 3.3 ile önerilmektedir:

$$v d_t \leq \delta / \cos \alpha \quad (3.3)$$

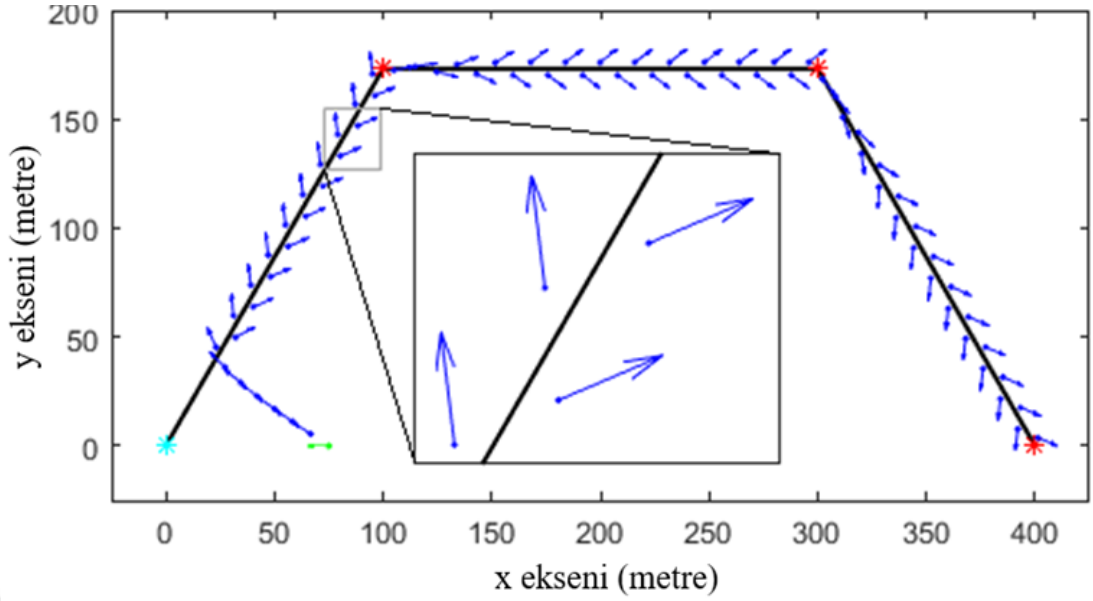
Denklem 3.3' te v deniz aracının hızını, d_t sayısal benzetimin zaman adımı büyüklüğünü ve α takip edilmesi gereken referans yol ile sanal hedef noktayı hedef yapan rota arasındaki açıdır. Denklem 3.3, her zaman adımında İSÜA' nın konum değişiminin, sanal hedef noktası ile İSÜA' nın mevcut konumu arasındaki mesafeye eşit veya bundan daha az olması gerektiğini belirtir. Bu, bir sonraki zaman adımında İSÜA' nın gerçek konumunun önceden tanımlanmış yol üzerinde veya İSÜA' nın son konumu ile önceden tanımlanmış referans yol arasında olması gerektiği anlamına gelir. Aksi takdirde İSÜA' nın navigasyonunun sayısal benzetimi gerçek dışı olabilir. Denklem 3.4, tutarlı bir navigasyonun sayısal benzetimini temin etmek için gerekli minimum δ mesafesini göstermektedir:

$$\delta \geq v d_t \cos \alpha \quad (3.4)$$

Denklem 3.4' te, sağ tarafı maksimuma çıkaran α değeri sıfırdır. Bu durumda stabilite kriteri Denklem 3.5 ile ifade edilebilir:

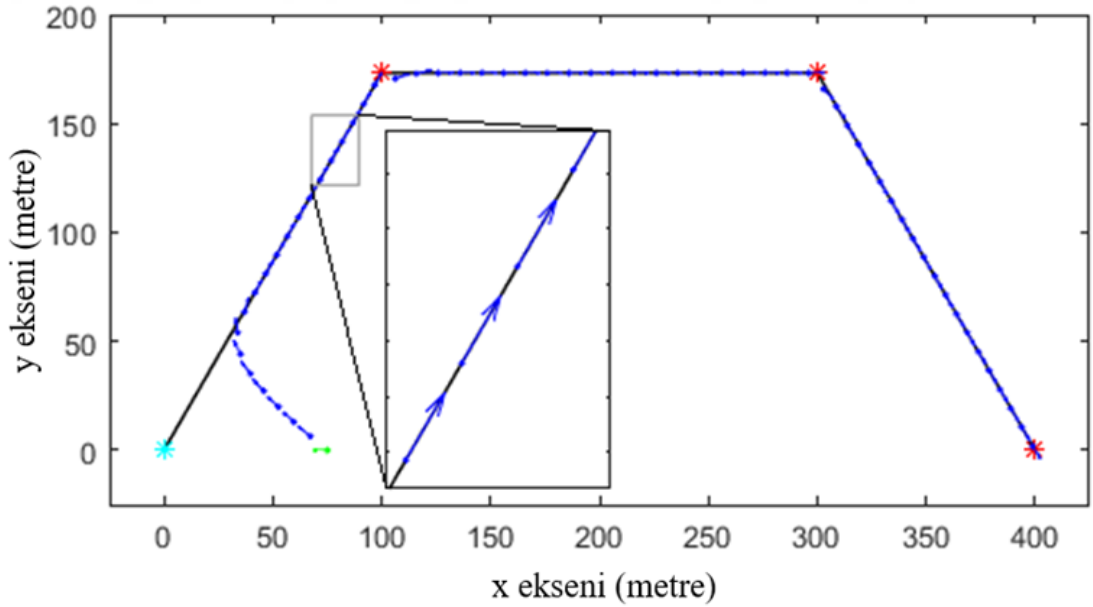
$$\delta \geq v d_t \quad (3.5)$$

İSÜA' nın hızının 20 metre/saniye ve sayısal benzetimde kullanılan zaman adım büyüklüğünün 0,5 saniye olduğu varsayılırsa, bu durumda Denklem 3.5' e göre tutarlı ve gerçeği yansıtan bir navigasyon elde edebilmek için δ değerinin 10 metreye eşit veya 10 metreden büyük olması gerekmektedir. Bu δ parametresi için alt limittir ve optimizasyon yöntemlerinde sınır koşul olarak kullanılabilir.



Şekil 3.1: $\delta = 4 \text{ metre}$ olması durumunda tutarsız bir navigasyon.

Şekil 3.1 temsili bir İSÜA' nın sayısal benzetim ortamında gerçekleştirilen navigasyonunu göstermektedir. Şekil 3.1' de yer alan kırmızı yıldızlar hedef temas noktalarını, turkuaz yıldız başlangıç noktasını, mavi oklar İSÜA' nın pruva açısını ve mavi okların başlangıç kısmında bulunan küçük mavi noktalar İSÜA' nın noktasal konumunu göstermektedir.



Şekil 3.2: $\delta = 11 \text{ metre}$ olması durumunda tutarlı ve kararlı bir navigasyon.

Sayısal benzetimler oluşturulurken, İSÜA' nın hızı 20 metre/saniye ve zaman adımı büyüklüğü 0,5 saniye olduğu varsayılmıştır. Şekil 3.1' de, İSÜA' nın güdüm yasasında yer

alan δ parametresi, 4 metre olarak yani stabilite kriteri olan 10 metreden küçük seçilmiştir. Bu durumda stabilite kriterine riayet edilmediğinden tutarsız bir seyir meydana gelmektedir ve İSÜA' fiziksel anlamı olmayan referans yolun karşılıklı iki bölgesinde konumlanmaktadır. Şekil 3.2' de ise δ parametresi, 11 metre olarak yani stabilite kriteri olan 10 metreden daha büyük seçilmiştir. Bu durumda, İSÜA tutarlı kararlı ve osilasyonsuz bir seyir gerçekleştirmekte olup önceden tanımlanmış referans yolu hassas bir şekilde takip edebilmektedir.

Yukarıda bahsi geçen kritik öneme sahip δ parametresinin önemini daha iyi açıklayabilmek için havuç kovalama güdüm yasasının temsili kodu aşağıdaki iş akışıyla özetlenmiştir:

Temsili Kod 1: Klasik havuç kovalama güdüm yasası.

- 1: $W_i \leftarrow [W_{i_x} W_{i_y}]$ {x ve y koordinatlarında başlangıç veya bir önceki temas noktasının tanımlanması}
- 2: $W_{i+1} \leftarrow [W_{i+1_x} W_{i+1_y}]$ {x ve y koordinatlarında hedef temas noktasının tanımlanması}
- 3: $p^t \leftarrow [p_x^t p_y^t]$ {Başlangıç veya bir önceki zaman adımında İSÜA' nın x ve y koordinatlarındaki konumunun tanımlanması}
- 4: v {İSÜA' nın hızının tanımlanması}
- 5: Ψ_p {Başlangıç veya bir önceki zaman adımında İSÜA' nın rota açısının belirlenmesi}
- 6: Ψ_{max} {İSÜA' nın maximum manevra kabiliyetinin açısal hız cinsinden belirlenmesi}
- 7: d_t {Sayısal benzetimde kullanılan zaman adımı büyüklüğünün tanımlanması}
- 8: δ {Kritik öneme sahip delta parametresinin belirlenmesi}
- 9: $\varphi \leftarrow atan(W_{i+1_y} - W_{i_y}, W_{i+1_x} - W_{i_x})$ {Başlangıç ve hedef temas noktalarını birleştiren doğrunun (kullanıcı tarafından belirlenen ve takip edilmesi istenilen referans yol) eğiminin açı cinsinden değerinin hesaplanması}
- 10: $\varphi_u \leftarrow atan(W_{i_y} - p_y^t, W_{i_x} - p_x^t)$ {Başlangıç temas noktası ile İSÜA' nın noktasal konumunu birleştiren doğrunun eğiminin açı cinsinden değerinin hesaplanması}
- 11: $\beta \leftarrow \varphi - \varphi_u$ {Beta açısının hesaplanması}

12: $R_u \leftarrow \sqrt{(W_{i_x} - p_x^t)^2 + (W_{i_y} - p_y^t)^2}$ {Başlangıç temas noktası ile İSÜA arasında kalan en kısa mesafenin hesaplanması}

13: $d \leftarrow R_u \times \sin(\beta)$ {Çapraz yol izleme hatasının belirlenmesi}

14: $vtp_x \leftarrow W_{i_x} + \cos(\varphi) \times \left(\sqrt{R_u^2 - d^2} + \delta \right)$ {Sanal hedef noktasının konumunun x koordinatındaki değerinin belirlenmesi}

15: $vtp_y \leftarrow W_{i_y} + \sin(\varphi) \times \left(\sqrt{R_u^2 - d^2} + \delta \right)$ {Sanal hedef noktasının konumunun y koordinatındaki değerinin belirlenmesi}

18: $\Psi_d \leftarrow \text{atan}(vtp_y - p_y^t, vtp_x - p_x^t)$ {Sanal hedef noktası hedef alınarak yakalanılması istenilen rota açısının hesaplanması}

19: $\dot{\Psi} \leftarrow (\Psi_d - \Psi_p)/dt$ {Yakalanılması istenilen rota açısını ve önceki rota açısını kullanarak İSÜA' nın dönüş hızının hesaplanması}

20: **If** $\dot{\Psi} > \dot{\Psi}_{max}$ **then** {Hesaplanan dönüş hızının İSÜA' nın maksimum dönüş hızından büyük olup olmadığının kontrol edilmesi}

21: $\Psi_d \leftarrow \Psi_p + \dot{\Psi}_{max} \times dt$ {Hesaplanan dönüş hızının İSÜA' nın maksimum dönüş hızından büyük olması durumunda yakalanılması istenilen rota açısının tekrar güncellenmesi}

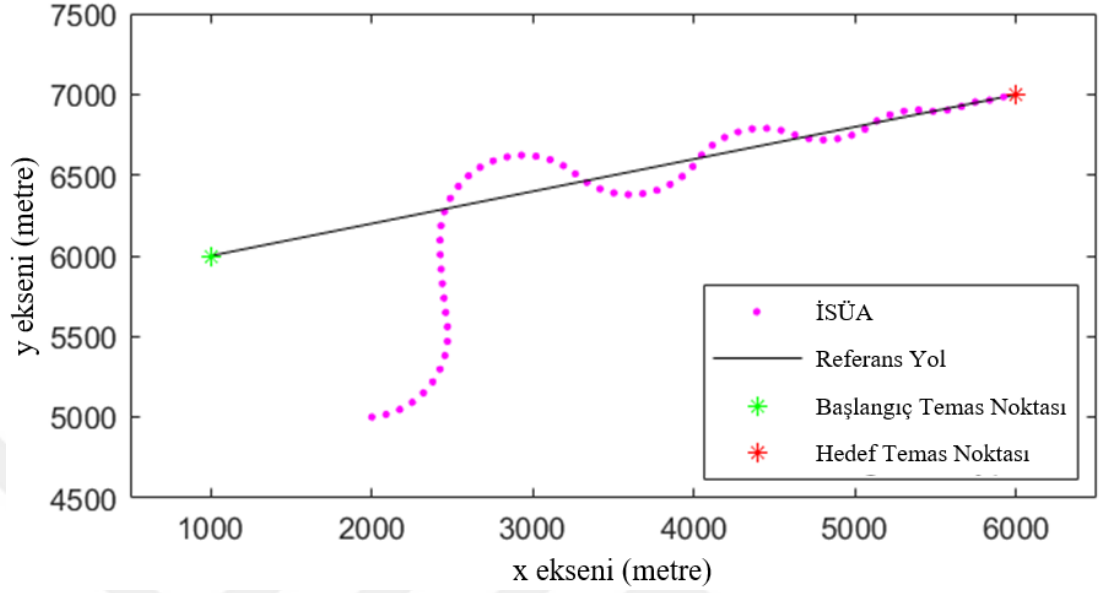
22: end if

22: $p_x^{t+1} \leftarrow p_x^t + v \times \cos d(\Psi_d) \times dt$ {Bir sonraki zaman adım için İSÜA' nın yeni konumunun x koordinatındaki değerinin bulunması}

23: $p_y^{t+1} \leftarrow p_y^t + v \times \sin d(\Psi_d) \times dt$ {Bir sonraki zaman adım için İSÜA' nın yeni konumunun y koordinatındaki değerinin bulunması}

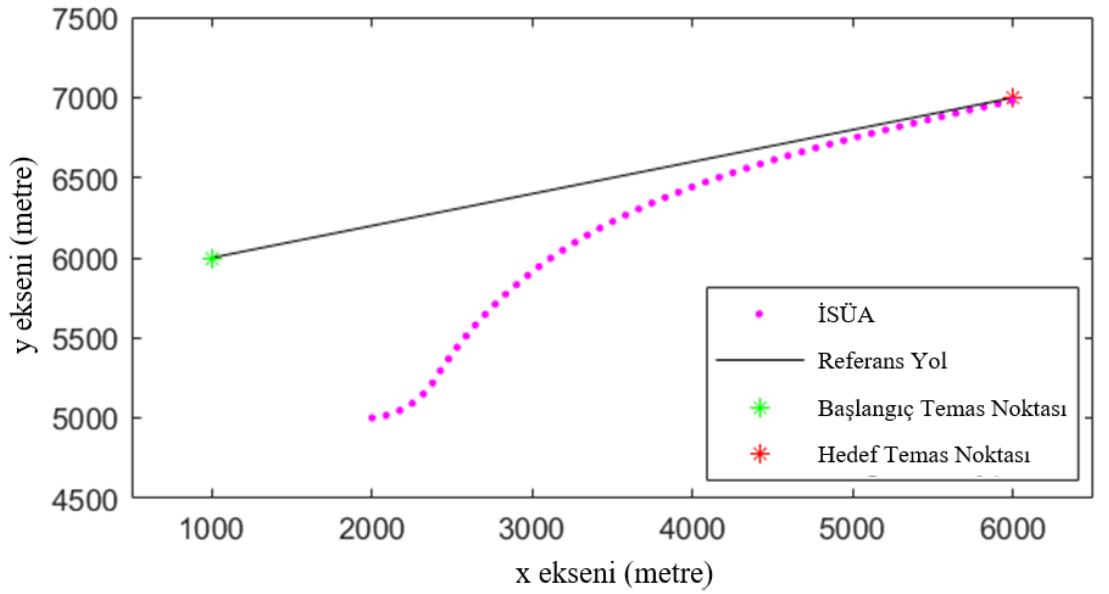
Eğer kritik δ değeri çok küçük seçilirse kararsız bir seyir (agresif navigasyon) gözlenir. İSÜA önceden tanımlanmış referans yola hızlı bir şekilde ulaşır ancak Şekil 3.3' te görüldüğü gibi yola ilk temas gerçekleştikten hemen sonra deniz aracı önceden tanımlanmış yolun karşı tarafına çıkmak zorunda kalır ve İSÜA referans yolu kararlı bir

şekilde takip edemez. Dolayısıyla çok küçük δ değeri büyük yol izleme hatalarına neden olur.



Şekil 3.3: Küçük δ değeri nedeniyle agresif navigasyon.

δ parametresinin çok büyük olduğu durumda ise kararsız bir seyir gözlemlenmese de Şekil 3.4' te olduğu gibi İSÜA önceden tanımlanmış yolu hızlı bir şekilde yakalayamaz (miskin navigasyon) ve bu durum da yine büyük yol izleme hatalarına sebep olur. Bu nedenle optimum δ parametresi istikrarlı ve doğru bir navigasyon için kritik öneme sahiptir.



Şekil 3.4: Büyük δ değeri nedeniyle miskin navigasyon.

Bu çalışmanın temel amaçlarından bir tanesi, yol izleme hatalarını en aza indirmek için desen arama optimizasyon tekniğini kullanarak havuç kovalama algoritmasında yer alan δ parametresinin optimal değerini belirlemektir.

Desen arama algoritması ilk olarak 1961 yılında Hooke ve Jeeves tarafından geliştirilmiştir [55]. Desen arama algoritması, hem doğrusal hem de doğrusal olmayan, sürekli veya süreksiz hedef fonksiyonları için oldukça etkili bir optimizasyon tekniğidir. Diğer geleneksel optimizasyon algoritmaları bazı durumlarda hedef fonksiyonun minimum değerini belirleyemezken, desen arama optimizasyon yöntemi kolay uygulanması, yüksek hesaplama performansı ve basit yapısı sayesinde çok çeşitli optimizasyon problemlerinin minimum değerini hızlı bir şekilde bulmayı başarmaktadır [70,71].

Desen arama optimizasyon yönteminin genel yapısı şu şekilde özetlenebilir: Desen arama algoritmasında optimizasyonun başlaması için bir başlangıç giriş değeri (x_0) gerekmektedir. Desen arama, ağ boyutu ve ağ genişletme faktörünün çarpımını bu başlangıç giriş değerine ekleyerek ve çıkararak yeni ağ noktaları ($x_{1,1}$ and $x_{1,2}$) oluşturur. Yeni ağ noktaları, hedef fonksiyonunda yani $f(x_{1,1})$ ve $f(x_{1,2})$ olarak değerlendirilir ve hedef fonksiyonunda elde edilen bu değerlerden küçük olanı seçilir. Örneğin $f(x_{1,1})$ değeri $f(x_0)$ ve $f(x_{1,2})$ değerinden küçükse bu yineleme başarılı yoklama olarak adlandırılır ve x_1 , $x_{1,1}$ ' e artık eşit olur. Desen arama $x_{1,1}$ ' e yaklaşır ve $x_{1,2}$ ' den uzaklaşır. İkinci yinelemede, x_1 e ağ boyutu ve ağ genişletme faktörünün çarpımı eklenip çıkarılarak yeni ağ noktaları belirlenir ($x_{2,1}$ ve $x_{2,2}$). Son yineleme, başarılı yoklama olduğu sürece ağ genişletme faktörü kullanılır. Tekrar, hedef fonksiyonunda yeni ağ noktaları hesaplanır: $f(x_{2,1})$ ve $f(x_{2,2})$. Eğer $f(x_1)$ değeri $f(x_{2,1})$ ve $f(x_{2,2})$ değerinden küçükse bu yineleme başarısız yoklama olarak adlandırılır ve x_2 , x_1 ' e eşit olur. Desen arama, üçüncü yinelemede ağ genişleme faktörü yerine ağ daralma faktörünü kullanarak ağ boyutunu azaltır. Bu işlemler yineleme durdurma kriterleri (ağ toleransı, fonksiyon toleransı, maksimum fonksiyon değerlendirmeleri ve maksimum yinelemeler) karşılanıncaya kadar devam eder. Bu araştırmada δ parametresini optimize etmek için desen arama algoritmasının varsayılan mevcut değerleri kullanılmıştır (başlangıç ağ boyutu: 100, maksimum ağ boyutu: sonsuz, ağ daralma faktörü: 0,5, ağ genişleme faktörü: 2, ağ toleransı: 1E-6, işlev toleransı: 1E-6, maksimum işlev değerlendirmeleri: 2000 ve maksimum yinelemeler: 100). Bu çalışmada, desen arama optimizasyon yöntemi entegreli havuç kovalama güdüm yasasına Akıllı

Havuç Kovalama GÜdüm Yasası (AHKGY) ismi verilmiştir ve AHKGY' nin iş akışına ilişkin temsili kod aşağıda gösterilmiştir:

Temsili Kod 2: Desen arama optimizasyon yöntemi entegreli havuç kovalama güdüm yasaı (AHKGY).

1: $\dot{\psi}_{max} \leftarrow 1$ {İSÜA' nın maksimum dönüş hızının derece/ saniye cinsinden tanımlanması}

2: $W_1 \leftarrow [1000 \ 6000]$, $W_2 \leftarrow [6000 \ 7000]$, $W_3 \leftarrow [1000 \ 7000]$ {İSÜA' nın takip etmesi istenilen referans yolu oluşturan temas noktalarının belirlenmesi}

3: $f_{cc}(\delta) \leftarrow d$ {Hedef fonksiyonunun (havuç kovalama güdüm yasaı), hedef fonksiyonun girdi parametresinin (δ değeri), ve hedef fonksiyonun çıktı parametresinin (yol izleme hatası, d) tanımlanması}

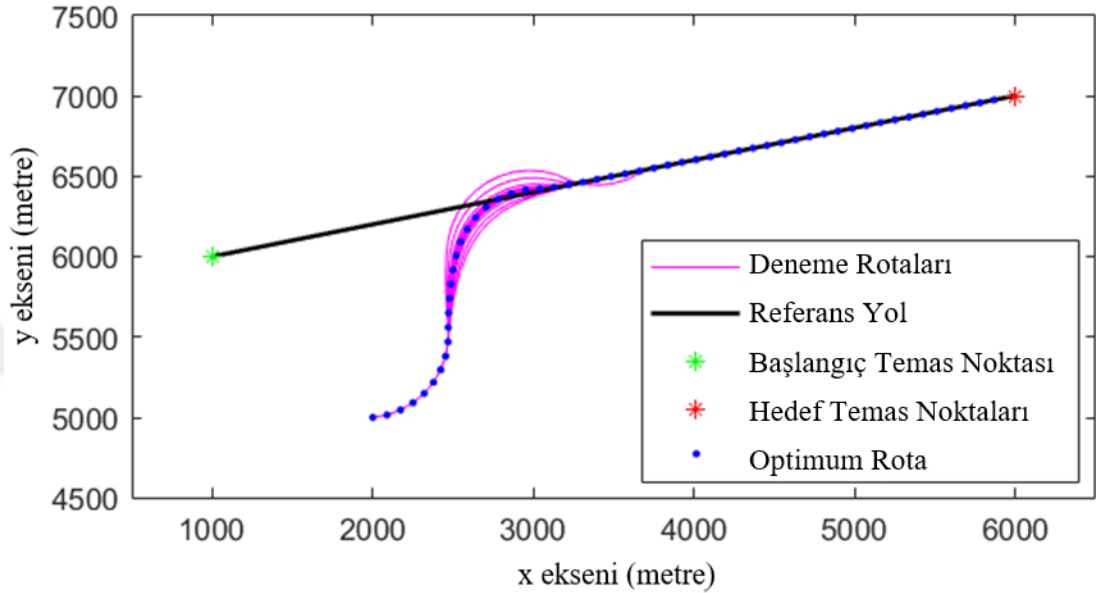
4: δ_{opt1} {İlk görev için en uygun δ değeri desen arama optimizasyon yöntemi ile bulunması (ilk görev için önceden tanımlanmış referans yol, başlangıç noktası W_1 ve birinci hedef temas noktası olan W_2 ' yi birleştiren doğrudur). İSÜA, birinci hedef temas noktasına (W_2) dokunmalıdır. Havuç kovalama güdüm yasaı bu ön koşula göre çalışmaktadır.}

5: Mp_1 {Minimum yol izleme hatasını netice veren en uygun δ değerlerini bulmak için desen arama algoritmasında çok sayıda ağ noktaları (δ değerleri) oluşturur. Şekil 3.5' te desen arama algoritması tarafından gönderilen muhtemel δ değerlerinin hedef fonksiyon olan havuç kovalama güdüm yasaında netice verdiği arama sonuçları (lila rotalar) ve bu aramanın neticesinde elde edilen en ideal rota (mavi noktaların oluşturduğu rota) gösterilmektedir.}

6: δ_{opt2} {İkinci görev için en uygun δ değeri desen arama optimizasyon yöntemi ile bulunması (ikinci görev için önceden tanımlanmış referans yol, başlangıç noktası W_2 ve ikinci hedef temas noktası olan W_3 ' ü birleştiren doğrudur). İSÜA, ikinci hedef temas noktasına (W_3) dokunmalıdır. Havuç kovalama güdüm yasaı bu ön koşula göre çalışmaktadır.}

7: Mp_2 {İkinci görevde de desen arama algoritmasında oluşturulan her bir δ değeri havuç kovalama güdüm yasaında değerlendirilir. Desen arama algoritması, hedef fonksiyondan aldığı verileri göz önünde bulundurarak optimum δ değerine yaklaşan yeni

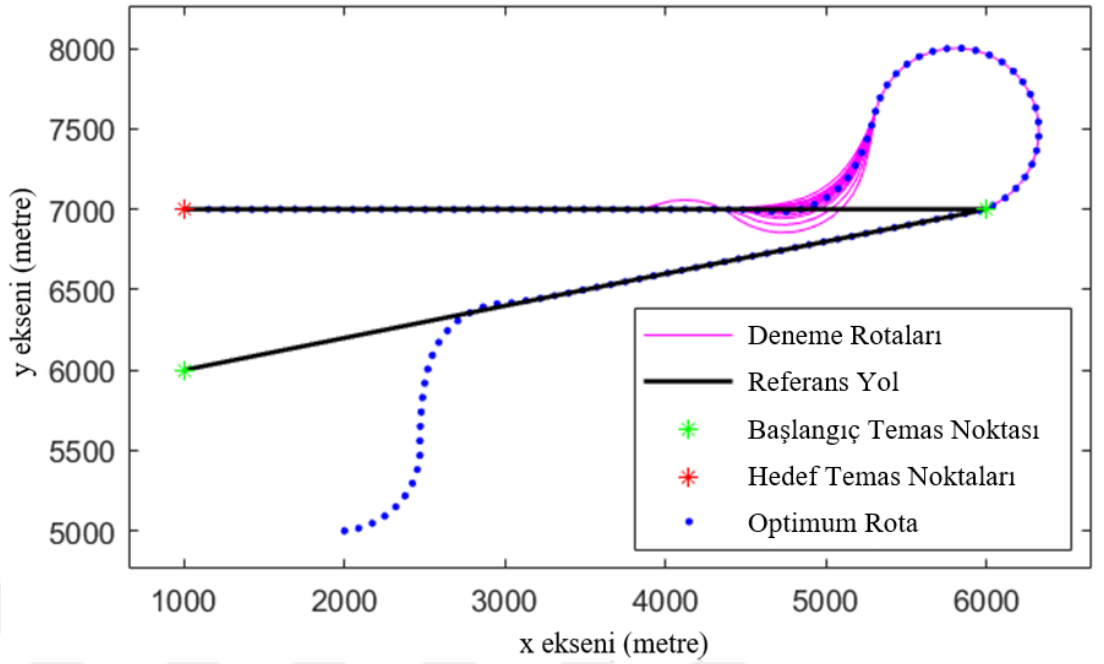
değerler türetir. Türetilen bu değerlerin de hedef fonksiyonunda değerlendirilmesiyle en uygun rotayı veren δ değeri yakalanır. Şekil 3.6 deneme rotalarını (lila rotalar) ve bu aramanın neticesinde elde edilen (en az yol izleme hatasını netice veren) rotayı (mavi noktaların oluşturduğu rota) göstermektedir.}



Şekil 3.5: İlk görev için optimal δ değerinin aranması.

47. yinelemenin sonunda ilk görev (Şekil 3.5) için optimal δ değeri 165.6569 metre, optimal δ değerinin neticesi olan minimum yol izleme hatası ise 1.3017e+04 metre olarak hesaplanmıştır. Şekil 3.6'daki, birinci görevden ikinci göreve geçiş sırasında oluşan büyük daire, İSÜA'nın maksimum dönüş hızının küçük olması nedeniyle (saniyede bir derece) oluşmaktadır. Bu çalışma kapsamında hazırlanan sayısal benzetimde deniz araçlarının hızı ve maksimum dönüş hızı gibi parametreler değiştirilebilir. Yüksek maksimum dönüş hızı seçilirse İSÜA hızlı bir şekilde dönebilir ve önceden tanımlanmış referans yolu hızlı bir şekilde yakalayabilir. Maksimum dönüş hızı İSÜA'nın fiziksel durumuna bağlıdır. Ticari gemilerde genellikle maksimum dönüş hızı saniyede bir dereceye eşit veya daha azdır.

51. yinelemenin sonunda ikinci görev (Şekil 3.6) için optimal δ değeri 172.1507 metre, minimum yol izleme hatası 1.7735e+04 metre olarak hesaplanmıştır. Bu temsili seyirde, optimal δ değerleri ve en makul rotaların oluşturulması, İSÜA'nın konumuna ve önceden tanımlanmış referans yolun şekline bağlıdır. Bu sayısal benzetimde birinci ve ikinci görevler için optimum δ değerleri bu iki değişken nedeniyle birbirinden farklı olarak tesbit edilmiştir.



Şekil 3.6: İkinci görev için optimal δ değerinin aranması.

İSÜA' nın hızı, maksimum manevra hızı ve başlangıç rota açısı gibi diğer parametreler de en iyi navigasyon rotasının bulunmasını doğrudan etkiler. Önerilen akıllı güdüm yasası (AHKGY), çoklu temas noktaları listesine sahip (W1, W2, W3, W4...) birden fazla görevi yerine getirmesi gereken deniz araçlarında, referans yola göre farklı ilk konumda bulunan deniz araçlarında, farklı başlangıç rota açılarında sahip deniz araçlarında, farklı manevra kabiliyetine sahip gemilerde ve muhtelif maksimum hızlara sahip deniz araçlarında kolaylıkla kullanılabilir. Bu çalışma kapsamında geliştirilen akıllı güdüm yasası, desen arama optimizasyon tekniğini kullanarak her koşul için en uygun rotayı araştırır ve belirler. Bu çalışmayı özgün kılan en önemli katkılardan biri budur.

3.2. Benzetimli Tavlama Optimizasyon Yönteminin Doğrusal Olmayan Güdüm Yasasına Uyarlanması

Doğrusal olmayan güdüm yasası da havuç kovalama güdüm yasası gibi, geometrik yaklaşıma dayalı yol izleme algoritmalarının bir alt dalıdır. Doğrusal olmayan güdüm yasası, merkezi İSÜA' nın mevcut noktasal konumu olan bir daire oluşturur. Doğrunun (önceden tanımlanmış referans yol) ve dairenin kesişiminden iki nokta ortaya çıkar. Bu iki noktadan hedef temas noktaya yakın olanı sanal hedef noktası olarak seçilir. İSÜA rotasını bu sanal noktaya çevirir ve bu sanal hedef noktaya ulaşmak için ilerler.

İSÜA' nın sabit bir hızda hareket ettiği ve rüzgar, dalga veya diğer dış kaynaklar gibi hiçbir dış kuvvetin olmadığı varsayılmaktadır. İSÜA için iki boyutlu doğrusal olmayan güdüm yasasının kinematiği, bu koşullar altında, havuç kovalama güdüm yasasında kullanılan Denklem 3.2 ile ifade edilebilir. Doğrusal olmayan güdüm yasasının tutarlı ve kararlı bir şekilde çalışabilmesi için ilk olarak oluşturulan dairelerin yarıçapının uzunluğu, İSÜA ile önceden tanımlanmış referans yol arasında kalan minimum mesafeden daha uzun veya eşit olması gerekmektedir. Aksi halde, türetilen çember ile referans yol kesişmeyecek ve sanal hedef noktaları oluşturulamayacaktır. Dolayısıyla güdüm yasası çalışmayacak ve İSÜA' nın rotası saptanamayacaktır. İkinci olarak, bu çalışma kapsamında doğrusal olmayan güdüm yasası için geliştirilen ve aşağıdaki Denklem 3.6 ile ifade edilen stabilite kriterinin sağlanması gerekmektedir. Denklem 3.6, tutarlı bir navigasyonun sayısal benzetimini temin etmek için güdüm yasası tarafından oluşturulan dairelerin minimum yarıçapını ifade etmektedir:

$$r \geq v d_t \quad (3.6)$$

Aşağıdaki prosedürler doğrusal olmayan güdüm yasasının temsili kodunu adım adım özetlemektedir:

Adım 1: İSÜA' nın x ve y koordinatlarındaki başlangıç konumunun tanımlanması; $[p_x^t \ p_y^t]$.

Adım 2: İSÜA' nın hızının, başlangıç rota açısının ve maksimum dönüş hızının tanımlanması; v , Ψ_i ve $\dot{\Psi}_{max}$.

Adım 3: x ve y koordinatlarındaki başlangıç ve hedef temas noktalarını kullanarak önceden tanımlanmış referans yolun belirlenmesi; $[W_{1x} \ W_{1y}]$ ve $[W_{2x} \ W_{2y}]$.

Adım 4: Sanal hedef noktayı oluşturmak için kullanılan dairenin yarıçapının (anahtar parametre) belirlenmesi; r .

Adım 5: Sayısal benzetimin zaman adımı boyutunun tanımlanması; dt .

Adım 6: Aşağıdaki Denklem 3.7' yi kullanarak başlangıç ve hedef temas noktaları ile önceden tanımlanmış referans yolun denkleminin bulunması.

$$y - W_{1y} = \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} (x - W_{1x}) \quad (3.7)$$

Denklem 3.8, önceden tanımlanmış referans yolun denklemini basitleştirmek için Denklem 3.7' nin yeniden düzenlenmesini gösterir.

$$y = \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} x - \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} W_{1x} + W_{1y} \quad (3.8)$$

Adım 7: Aşağıdaki Denklem 3.9' u kullanarak yol izleme hatasının bulunması; d .

$$d = \frac{\left| \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} p_x^t - p_y^t - \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} W_{1x} + W_{1y} \right|}{\sqrt{\left(\frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} \right)^2 + (-1)^2}} \quad (3.9)$$

Adım 8: Önceden tanımlanmış referans yol ile r yarıçaplı dairenin kesişim noktalarının Denklem 3.10' dan 3.18' e kadar olan hesaplamalar yapılarak bulunması.

$$L_x = W_{2x} - W_{1x} \quad (3.10)$$

$$L_y = W_{2y} - W_{1y} \quad (3.11)$$

$$L_r = \sqrt{L_x^2 + L_y^2} \quad (3.12)$$

$$\begin{aligned} L_a &= (W_{1x} - p_x^t) (W_{2y} - p_y^t) \\ L_b &= (W_{2x} - p_x^t) (W_{1y} - p_y^t) \\ L &= L_a - L_b \end{aligned} \quad (3.13)$$

$$x_1 = \frac{L L_y + \text{sgn}(L_y) L_x \sqrt{r^2 L_r^2 - L^2}}{L_r^2} \quad (3.14)$$

$$x_2 = \frac{L L_y - \text{sgn}(L_y) L_x \sqrt{r^2 L_r^2 - L^2}}{L_r^2} \quad (3.15)$$

$$y_1 = \frac{-L L_x + |L_y| \sqrt{r^2 L_r^2 - L^2}}{L_r^2} \quad (3.16)$$

$$y_2 = \frac{-L L_x - |L_y| \sqrt{r^2 L_r^2 - L^2}}{L_r^2} \quad (3.17)$$

Denklem 3.14 ve Denklem 3.15’ te bulunan işaret fonksiyonlarının tanımı, Denklem 3.18 ile gösterilmiştir.

$$\text{sgn}(L_y) = \begin{cases} -1, & L_y < 0 \\ 1, & L_y \geq 0 \end{cases} \quad (3.18)$$

Denklem 3.10’ dan 3.18’ e kadar olan hesaplamaların yapılmasının neticesinde q_1 ve q_2 olmak üzere iki adet kesişim noktası elde edilir; $q_1=[x_1 \ y_1]$ ve $q_2=[x_2 \ y_2]$.

Adım 9: Elde edilen her iki kesişme noktasının hedef temas noktaya olan mesafelerinin Denklem 3.19 ve Denklem 3.20’ yi kullanarak ayrı ayrı ölçülmesi.

$$d_1 = \sqrt{(W_{2x} - x_1)^2 + (W_{2y} - y_1)^2} \quad (3.19)$$

$$d_2 = \sqrt{(W_{2x} - x_2)^2 + (W_{2y} - y_2)^2} \quad (3.20)$$

Adım 10: Sanal hedef noktasının (VTP) kısa olan mesafe seçilerek Denklem 3.21 ile belirlenmesi.

$$\text{VTP} = \begin{cases} q_1, & d_1 < d_2 \\ q_2, & d_1 \geq d_2 \end{cases} \quad (3.21)$$

Adım 11: İSÜA’ nın mevcut noktasal konumunu ve bir önceki adımda belirlenen sanal hedef noktasını kullanarak Denklem 3.22 ile İSÜA’ nın ulaşması istenilen rota açısının hesaplanması; Ψ_d .

$$\Psi_d = \text{atan} \left(\frac{\text{VTP}_y - p_y^t}{\text{VTP}_x - p_x^t} \right) \quad (3.22)$$

Adım 12: İSÜA' nın ulaşması istenilen rota açısını, önceki adımda hesaplanan mevcut rota açısını ve sayısal benzetimde yer alan zaman adımı boyutunu kullanarak İSÜA' nın dönüş hızının Denklem 3.23 ile hesaplanması; $\dot{\Psi}$.

$$\dot{\Psi} = (\Psi_d - \Psi_p)/dt \quad (3.23)$$

Adım 13: Dönüş hızın kontrol edilmesi: İSÜA' nın bir önceki adımda hesaplanan dönüş hızı ($\dot{\Psi}$), İSÜA' nın maksimum dönüş hızından ($\dot{\Psi}_{max}$) büyükse, İSÜA' nın ulaşması istenilen rota açısı aşağıdaki Denklem 3.24 ile güncellenir.

$$\Psi_d = \dot{\Psi}_{max} dt + \Psi_p \quad (3.24)$$

Adım 14: Sayısal benzetimde, bir sonraki zaman adımı için İSÜA' nın yeni konumunun Denklem 3.25 ile belirlenmesi; p^{t+1} .

$$\begin{aligned} p_x^{t+1} &= p_x^t + \cos(\Psi_d) v dt \\ p_y^{t+1} &= p_y^t + \sin(\Psi_d) v dt \end{aligned} \quad (3.25)$$

Güdüm yasasının belirlediği dairenin yarıçapı (r), yol izleme hatasından (d) küçükse, daire önceden tanımlanmış yolla kesişemez. Bu durumda sanal hedef noktası belirlenemez ve İSÜA önceden tanımlanan referans yolu izleyemez. Bu nedenle, doğrusal olmayan güdüm yasasında, dairenin yarıçapı en azından yol izleme hatasının uzunluğu kadar olmalıdır. Dairelerin yarıçapının çok büyük olması veya çok küçük olması, havuç kovalama güdüm yasasında olduğu gibi miskin veya saldırgan bir seyrin gözlemlenmesine sebep olur.

İSÜA' nın kararlı, doğru ve hassas bir otonom seyir yapabilmesi için gerekli ve yeterli bir optimum yarıçap, koşuldan koşula değişmektedir. Örneğin, optimum yarıçap, İSÜA' nın önceden tanımlanmış yola göre farklı konumları için aynı değildir. Optimum değer aynı zamanda önceden tanımlanmış referans yolların birbirine göre konumlarına, İSÜA' nın ilk konumuna, hızına, rota açısına, maksimum dönüş hızına da bağlıdır. Özellikle çok görevli operasyonlar için tüm seyir koşulları için deneme yanılma yöntemini kullanarak optimum yarıçapı ayarlamak neredeyse imkansızdır. Bu çalışma, tüm navigasyon koşulları için en uygun yarıçapı otomatik olarak bulabilen benzetilmiş tavlama optimizasyon tekniği ile entegre edilmiş doğrusal olmayan güdüm yasasının kullanılmasını önermektedir.

1983 yılında Kirkpatrick ve arkadaşları [60] tarafından literatüre giren benzetilmiş tavlama algoritması, küresel optimizasyon problemlerini çözmek için en etkili sezgisel tekniklerden biridir [72]. İlgilenilen fonksiyonun sürekli-süreksiz veya doğrusal-doğrusal olmayan özelliklerine bakılmaksızın herhangi bir hedef fonksiyonu için kolaylıkla kullanılabilir [73].

Benzetilmiş tavlama yöntemi, gerçek hayattaki fiziksel tavlama işleminden esinlenilerek bulunmuştur. Fiziksel tavlama, malzemeyi istenen yapıya dönüştürmek için metalin tavlama sıcaklığına kadar ısıtılması ve daha sonra kademeli olarak soğutulması işlemidir. Yüksek sıcaklıkta malzemenin yapısal şeklini değiştirmek nasıl kolaysa, optimizasyon yöntemi de başlangıçta yüksek katsayılı, çok geniş bir aralıkta hedef fonksiyonunun (doğrusal olmayan güdüm yasası) küresel minimum değerini (yol izleme hatası) arayarak yerel minimum engellerinden kurtulur. Optimizasyon başlangıcında yüksek olarak belirlenen bu katsayı, belirli koşullar altında kademeli olarak azaltılarak nihai minimum noktanın küresel minimum alanında tam olarak belirlenmesine olanak sağlar (fiziksel tavlama sıcaklığının kademeli olarak düşürülmesi gibi). Desen arama optimizasyonu, bu fiziksel olay sayesinde hedef fonksiyonunun küresel minimum noktasını bulabilir. Bu çalışmada benzetilmiş tavlama optimizasyon yöntemi entegreli doğrusal olmayan güdüm yasası, akıllı daire tabanlı doğrusal olmayan güdüm yasası (ADTDÖGY) olarak adlandırılmıştır. ADTDÖGY' nin sözde kodu, önerilen yeni algoritmayı net bir şekilde açıklamak için aşağıdaki adımlarla gösterilmektedir:

Adım 1: İSÜA' nın x ve y koordinatlarındaki başlangıç konumunun tanımlanması;
 $[p_x^t \ p_y^t] = [2500 \ 1000]$.

Adım 2: İSÜA' nın hızının, başlangıç rota açısının ve maksimum dönüş hızının tanımlanması.

$$v = 9 \text{ metre/saniye}$$

$$\psi_i = 90 \text{ derece}$$

$$\dot{\psi}_{max} = 1 \text{ derece/saniye}$$

Adım 3: x ve y koordinatlarındaki başlangıç ve hedef temas noktalarını kullanarak önceden tanımlanmış referans yolların belirlenmesi.

$$\begin{bmatrix} W_{1x} & W_{1y} \end{bmatrix} = [3500 \ 1000]$$

$$\begin{bmatrix} W_{2x} & W_{2y} \end{bmatrix} = [1000 \ 6000]$$

$$\begin{bmatrix} W_{3x} & W_{3y} \end{bmatrix} = [1000 \ 1000]$$

Adım 4: Sayısal benzetimin zaman adım boyutunun belirlenmesi.

$$d_t = 10 \text{ saniye}$$

Adım 5: Denklem 3.26' yı kullanarak başlangıç noktası (W_{1x}, W_{1y}) ve birinci hedef temas noktası (W_{2x}, W_{2y}) ile birinci referans yolun denkleminin bulunması.

$$y - 1000 = \frac{6000 - 1000}{1000 - 3500} (x - 3500) \quad (3.26)$$

Denklem 3.27, birinci referans yolun denklemini basitleştirmek için Denklem 3.26' nın yeniden düzenlenmesini gösterir.

$$y = -2x + 8000 \quad (3.27)$$

Adım 6: Bir sonraki zaman adımı (t+1) için benzetilmiş tavlama optimizasyon tekniğini kullanarak optimum daire yarıçapının bulunması.

Adım 6.1: Benzetilmiş tavlama algoritması için gerekli parametrelerin tanımlanması.

-Hedef fonksiyon, f : doğrusal olmayan güdüm yasası.

-Hedef fonksiyonunun girdi parametresi, r_i : doğrusal olmayan güdüm yasasına gönderilen dairelerin yarıçapı.

-Hedef fonksiyonunun çıktı parametresi, $f(r_i)$: yol izleme hataları.

-Optimal yarıçapı bulmak için arama alanının seçilmesi (Dairelerin yarıçapının alt ve üst sınırları): Minimum alt sınır, mevcut zaman adımında hesaplanan yol izleme hatası kadar ve maksimum üst sınır ise mevcut zaman adımında hesaplanan yol izleme hatasının iki katı olarak varsayılmıştır.

$$[lb, ub] = [d, 2d]$$

-Benzetilmiş tavlamanın (T) sabit değerin tanımlanması: Varsayılan değer olarak T=100 olarak seçilmiştir.

-Sabit değer azaltma faktörünün (c) belirlenmesi:

c=0.8.

-Maksimum yineleme sayısı: k=100. Optimum yarıçapı bulmak için yapılan yineleme sayısı 100'den fazla ise döngü sona erer ve optimizasyon durdurulur.

-Sonlandırma faktörü (maksimum izin verilen hata): $\varepsilon = 0.01$. $\varepsilon > |r_2 - r_1|$ ise döngü sona erer ve optimizasyon durdurulur.

Adım 6.2: Benzetilmiş tavlama optimizasyonunda herbir yineleme (i^{th}) için Denklem 3.9' u kullanarak yol izleme hatasının (d) Denklem 3.28 ile ifade edilmesi.

$$d = \frac{\left| \frac{6000 - 1000}{1000 - 3500} 2500 - 1000 - \frac{6000 - 1000}{1000 - 3500} 3500 + 1000 \right|}{\sqrt{\left(\frac{6000 - 1000}{1000 - 3500} \right)^2 + (-1)^2}} \quad (3.28)$$

$$= 894,42 \text{ metre}$$

Benzetilmiş tavlama ilk yineleme için başlangıç yarıçapı ($i=1$) alt ve üst sınırların aritmetik ortalaması olarak varsayılmıştır:

$r_1=r_1=(lb+ub)/2= 1341,63 \text{ metre}$.

Adım 6.3: Birinci referans yol ile r_1 yarıçaplı dairenin (birinci yineleme için $r_1=1341,63 \text{ metre}$) kesişim noktalarının Denklem 3.29' dan 3.37' e kadar olan hesaplamalar yapılarak bulunması.

$$L_x = W_{2x} - W_{1x} = 1000 - 3500 = -2500 \quad (3.29)$$

$$L_y = W_{2y} - W_{1y} = 6000 - 1000 = 5000 \quad (3.30)$$

$$L_r = \sqrt{L_x^2 + L_y^2} = \sqrt{(-2500)^2 + 5000^2} = 5590,17 \quad (3.31)$$

$$\begin{aligned} L_a &= (3500 - 2500) (6000 - 1000) = 5 \cdot 10^4 \\ L_b &= (1000 - 2500) (1000 - 1000) = 0 \\ L &= L_a - L_b = 5 \cdot 10^4 - 0 = 5 \cdot 10^4 \end{aligned} \quad (3.32)$$

$$x_1 = \frac{(2 \cdot 10^7) 5000 - 2500 \sqrt{1341.63^2 5590.17^2 - (5 \cdot 10^4)^2}}{5590.17^2} = 2600,02 \quad (3.33)$$

$$x_2 = \frac{(2 \cdot 10^7) 5000 + 2500 \sqrt{1341.63^2 5590.17^2 - (5 \cdot 10^4)^2}}{5590.17^2} = 3799,98 \quad (3.34)$$

$$y_1 = \frac{(2 \cdot 10^7) 2500 + 5000 \sqrt{1341.63^2 5590.17^2 - (5 \cdot 10^4)^2}}{5590.17^2} = 2799,96 \quad (3.35)$$

$$y_2 = \frac{(2 \cdot 10^7) 2500 - 5000 \sqrt{1341.63^2 5590.17^2 - (5 \cdot 10^4)^2}}{5590.17^2} = 400,04 \quad (3.36)$$

Denklem 3.29' dan 3.36' ya kadar olan hesaplamaların yapılmasının neticesinde q_1 ve q_2 olmak üzere iki adet kesişim noktası elde edilir; $q_1=[x_1 \ y_1]$ and $q_2=[x_2 \ y_2]$.

Adım 6.4: Elde edilen her iki kesişme noktasının hedef temas noktaya olan mesafelerinin Denklem 3.19 ve Denklem 3.20' yi kullanarak ayrı ayrı ölçülmesi ve Denklem 3.37 ve 3.38 ile gösterilmesi.

$$d_1 = \sqrt{(1000 - 2600.02)^2 + (6000 - 2799.96)^2} = 3577.75 \quad (3.37)$$

$$d_2 = \sqrt{(1000 - 3799.98)^2 + (6000 - 400.04)^2} = 6260.95 \quad (3.38)$$

Adım 6.5: d_1 mesafesi d_2 mesafesinden daha küçük olduğu için sanal hedef nokta (VTP) olarak birinci kesişim noktasının seçilmesi ve seçilen sanal hedef noktasının Denklem 3.39 ile gösterilmesi.

$$VTP = q_1 = [x_1, y_1] = [2600,02 \ 2799,96] \quad (3.39)$$

Adım 6.6: İSÜA' nın mevcut noktasal konumunu ve bir önceki adımda belirlenen sanal hedef noktasını kullanarak Denklem 3.40 ile İSÜA' nın ulaşması istenilen rota açısının hesaplanması; Ψ_d .

$$\Psi_d = \text{atan}\left(\frac{VTP_y - p_y^t}{VTP_x - p_x^t}\right) = \text{atan}\left(\frac{2799,96 - 1000}{2600,02 - 2500}\right) = 86.82 \text{ derece} \quad (3.40)$$

Adım 6.7: İSÜA' nın dönüş hızının Denklem 3.41 ile hesaplanması; $\dot{\Psi}$.

$$\dot{\Psi} = (\Psi_d - \Psi_p)/dt = \frac{|86,82 - 90|}{10} = 0,32 \text{ derece/saniye} \quad (3.41)$$

Adım 6.8: İSÜA' nın bir önceki adımda hesaplanan dönüş hızı ($\dot{\Psi} = 0,32$), İSÜA' nın maksimum dönüş hızından ($\dot{\Psi}_{max} = 1$) küçük olduğu için, İSÜA' nın ulaşması istenilen rota açısında değişiklik yapılmaz.

Adım 6.9: Sayısal benzetimde, bir sonraki zaman adımı için İSÜA' nın yeni konumunun Denklem 3.42 ile belirlenmesi; p^{t+1} .

$$\begin{aligned} p_x^{t+1} &= p_x^t + \cos(\Psi_d) v dt = 2500 + \cos(86,82) 9 10 = 2500,5 \\ p_y^{t+1} &= p_y^t + \sin(\Psi_d) v dt = 1000 + \sin(86,82) 9 10 = 1089,9 \end{aligned} \quad (3.42)$$

Adım 6.10: En uygun yarıçap benzetilmiş tavlama yöntemi ile bulununcaya kadar döngünün devam etmesi: Yeni konumu belirlenen İSÜA' nın mevcut konumunun netice verdiği yol izleme hatasının hesaplanması için tekrar Adım 6.2' ye gidilir. Bu durumda, r_1 yarıçap ($r_1=1341,63$ metre) kullanılması durumunda ortaya çıkan çapraz izleme hatası ($d_1=853,77$ metre) hesaplanır. Bu sonuç, benzetilmiş tavlama algoritmasındaki ilk yinelemenin sonudur.

$$f(r_1)=d_1$$

$$f(1341,63)=853,77.$$

Adım 6.10' dan sonra, benzetilmiş tavlama optimizasyon yöntemi, aşağıdaki alt adımları kullanarak daha küçük yol izleme hatası veren yeni olası yarıçaplar üretir:

Adım 6.10.1: Optimizasyon yönteminde kullanılan yarıçap değerinin alt ve üst sınırları arasında rastgele dağılım uygulayarak r_1 yarıçapına yakın bir sonraki yarıçap (ikinci yineleme için benzetilmiş tavlama algoritması r_2 yarıçapını 1156,65 metre olarak belirlenmiştir) bulunur ve ardından yeni yol izleme hatasını (d_2), yeni yarıçapı (r_2) kullanarak hesaplamak için Adım 6.3'e geri dönülür. Gerekli işlemler yapılarak bulunan yeni değerler geçici bir hafızada kaydedilir.

$$f(r_2)=d_2.$$

$$f(1156,65)=848,96.$$

Adım 6.10.2: İkinci yarıçap ve ikinci yol izleme hatasını netice veren iç enerjinin (ΔE) kontrol edilmesi.

Adım 6.10.2.1: Denklem 3.43 ile ΔE hesaplanır. Eğer ΔE sıfırdan küçükse ikinci yarıçap yeni yarıçap olarak kabul edilir.

$$\Delta E = f(r_2) - f(r_1) = 848.96 - 853.77 = -4.81 \quad (3.43)$$

ΔE sıfırdan küçüktür. Bu nedenle Adım 6.10.3' e gidilir.

Adım 6.10.2.2: Eğer ΔE Denklem 3.44' te olduğu gibi sıfırdan küçük değilse ($r_2=1546,32$ ve $d_2=859,63$ olduğunu varsayalım), bu yinelemeye kötü hamle denir ancak r_2 doğrudan elenmez. Metropol kriteri bu kötü noktanın (r_2) kabul edilip edilmeyeceğine karar vermek için kullanılır:

$$\Delta E = f(r_2) - f(r_1) = 859.63 - 853.77 = 5.86 \quad (3.44)$$

Bu kötü noktayı (r_2) kabul etme olasılığı (p) aşağıdaki Denklem 3.45 ile bulunur ("T" sabit değeri Adım 6.1' de 100 olarak seçilmişti):

$$p = e^{\frac{-\Delta E}{T}} = e^{\frac{-5,86}{100}} = 0,94 \quad (3.45)$$

Kötü hamle için olasılık hesaplandıktan sonra, 0 ile 1 rakamları arasından eşit dağılımlı rastgele bir sayı (u) oluşturulur. Rastgele oluşturulan bu sayının (u değerinin) 0,72'ye eşit olduğunu varsayalım. Kötü noktayı kabul etme olasılığı ($p=0,94$), rastgele seçilen sayıdan ($u=0,72$) daha büyüktür. Bu nedenle, bu kötü nokta (r_2) kabul edilir ve Adım 6.10.3' e gidilir. Aksi takdirde yani u değeri p değerinden büyük olsaydı, bu olası değer (r_2)

reddedilirdi ve yeni daire yarıçapı oluşturmak için Adım 6.10.1' e geri dönülürdü. Benzetilmiş tavlama optimizasyon yöntemi Adım 6.10.2.1 sayesinde yerel minimum engellerden kurtulur ve küresel alanda bulunan değerler arasından en iyi minimum değeri aramaya çalışır. Bu yüzden, benzetilmiş tavlama algoritması küresel bir optimizasyon tekniğidir.

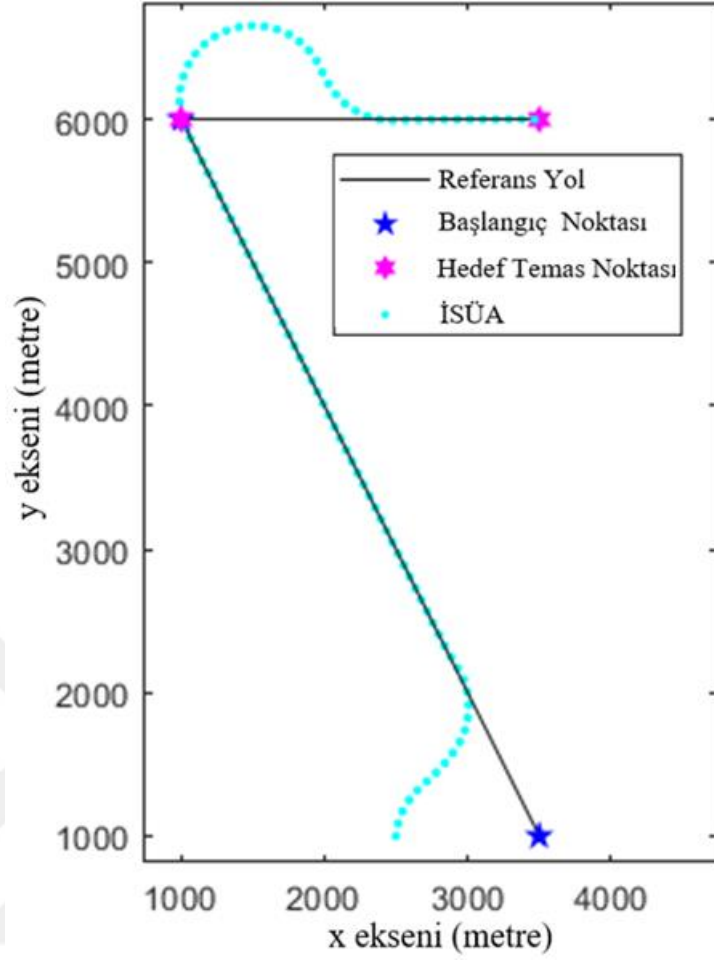
Adım 6.10.3: Maksimum izin verilen hatanın (ε) Denklem 3.46 ile kontrol edilmesi: Eğer ε , 0,01' den küçükse (Bu değer Adım 6.1' de belirlenmişti.) yineleme durdurulur. Aksi takdirde Adım 6.10.4' e gidilir.

$$\varepsilon = |r_2 - r_1| = |1156,65 - 1341,63| = 184,9800 \quad (3.46)$$

Adım 6.10.4: Bu yineleme iyi bir hamledir çünkü iç enerji sıfırdan düşüktür. Bu sebeple, sabit değer Adım 6.1' de yer alan sabit değer azaltma faktörü ($c=0,8$) ile düşürülür ($T=80$).

Buraya kadar olan işlemler ikinci yinelemenin sonudur. Dolayısıyla mümkün olan en iyi daire yarıçapını bulmak için Adım 6.10.1'e geri dönülür. Bu döngü İSÜA, ilk hedef noktasına ulaşmıyaya kadar devam eder. Birinci görevden sonra ikinci hedef temas noktası için de aynı işlemler uygulanır. İSÜA' nın rota açısı, benzetilmiş tavlama algoritması entegreli doğrusal olmayan güdüm yasası (ADTDOGY) sayesinde en az rota hatasıyla sonuçlanan yarıçapların belirlenmesiyle hesaplanır.

İç enerjinin (ΔE) artması, daha kötü noktayı kabul etme olasılığının (p) azalmasına neden olur. Bu, daha büyük çapraz izleme hatalarına sebep olan daha kötü yarıçap değerlerinin türetilmesini önler. Optimizasyon başlangıcındaki büyük sabit değer ($T=100$) (fiziksel tavlama da olduğu gibi, sıcaklık yüksek olduğunda malzemenin fiziksel yapısını değiştirmek kolaydır) global minimum değer için daha geniş bir aralıkta aranmasına olanak tanır ve algoritmanın yerel minimum engellerden kurtulmasını sağlar. Her yinelemede benzetilmiş tavlama sabiti sayısının (T) azaltılması (fiziksel tavlama da sıcaklığın kademeli olarak azaltılması gibi), benzetilmiş tavlama optimizasyon yönteminin global optimizasyon değerinin bulunduğu alana odaklanmasını ve global minimumu hassas bir şekilde belirlemesini sağlar.



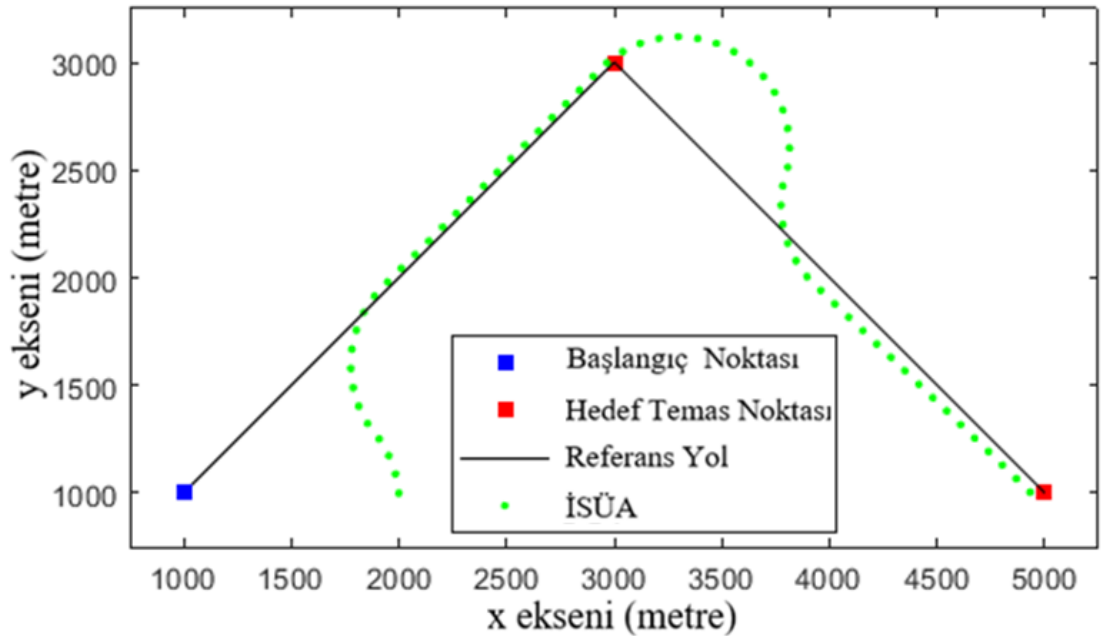
Şekil 3.7: ADTDOGY algoritmasının ürettiği optimum dairelerin netice verdiği kararlı bir navigasyon.

Şekil 3.7, önerilen ADTDOGY algoritması kullanılarak İSÜA' nın kararlı ve optimum navigasyonunu göstermektedir. Bu çok görevli sayısal benzetimde İSÜA pembe renkli hedef temas noktalarına ulaşır ve değmesi gerekmektedir. İkinci olarak, İSÜA' nın maksimum dönüş hızı saniyede 1 derecedir. İSÜA bu iki ön koşul altında hareket eder. Şekil 2.11' de ADTDOGY algoritması, hem birinci hem de ikinci referans yollarda minimum çapraz izleme hatasına yol açan benzersiz daireler oluşturur. Böylece İSÜA, insan müdahalesi olmadan çok görevli operasyona otomatik olarak uyum sağlar ve önceden tanımlanmış referans yollara hızlı ve istikrarlı bir şekilde ulaşır. ADTDOGY algoritması, bir önceki bölümde yer alan Şekil 2.3 ve Şekil 2.4' de gösterilen geleneksel doğrusal olmayan güdüm yasasıyla karşılaştırıldığında toplam yol izleme hatalarını önemli ölçüde azaltır.

3.3. Vekil Model Optimizasyon Yönteminin Vektör Alanı Güdüm Yasasına Uyarlanması

Vektör alanı güdüm yarasası, kullanıcı tarafından tanımlanan dört adet parametreye ihtiyaç duyar. Bu parametrelerden ilki, sınır bölgesinin uzunluğunu belirtir. İSÜA sınır bölgesine girmeden önce referans yolu yakalayabilmek için keskin bir rota açısı ile ilerler. İkinci parametre, sınır bölgesi dışındaki İSÜA' nın rota açısının ne kadar keskin olacağını belirler. Üçüncü ve dördüncü parametreler ise İSÜA' nın sınır bölgesindeki rota açısını ayarlar. Bu son iki parametre referans yolu yakalayabilmek ve kararlı bir seyir yapabilmek için kullanılır. İSÜA, vektör alanı güdüm yarasası sayesinde birbirinden bağımsız bu dört parametre ile hem referans yolu hızlı bir şekilde yakalar hem de sınır bölgesinde yumuşak manevra yaparak referans yolu kararlı bir şekilde takip eder ve nihai hedef noktasına ulaşır [74-80].

Literatür taraması bölümünde detaylı olarak anlatılan Denklem 2.2' de bulunan, ρ negatif ve k değeri tam sayı olmayan pozitif bir gerçık sayı olması durumunda hesaplanan rota açısının sayısal sonucu sanal sayı içerecektir $((-0.8)^{1.17} = -0.663 - 0.392i$ gibi). Karmaşık sayının sanal kısmı, yol izleme algoritmasının sayısal benzetimde oluşturduğu navigasyonun doğruluğunu azaltmaktadır.

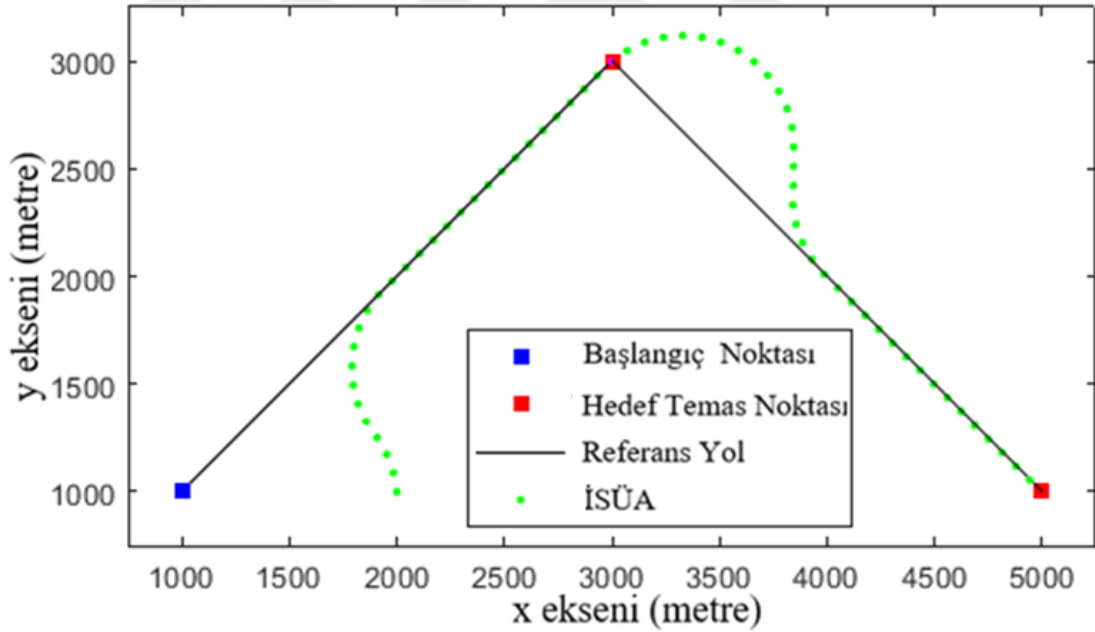


Şekil 3.8: Klasik vektör alanı güdüm yarasası.

Şekil 3.8' de İSÜA, karmaşık sayının sanal kısmı nedeniyle önceden tanımlanmış referans yolu tam olarak hassas bir şekilde takip edememektedir. Toplam yol izleme hatası klasik vektör alanı algoritması kullanılarak 9703,1 metre olarak hesaplanmıştır. Klasik vektör alanı güdüm kanunu, sanal kısmı ortadan kaldıracak şekilde aşağıdaki Denklem 3.47' de olduğu gibi modifiye edilebilir:

$$\Psi_d = \varphi - \rho \varphi_e \left[\left(\frac{d}{d_B} \right)^k + \left(\frac{k v}{\alpha d_B^k} \right) d^{k-1} \sin d(\Psi_p) \right] \quad (3.47)$$

İSÜA' nın ulaşması istenilen rota açısının Denklem 3.47 kullanılarak hesaplanması hiçbir zaman sanal sayı içermez. Bu sayede İSÜA önceden tanımlanmış yolu daha doğru ve hassas bir şekilde takip eder. Toplam yol izleme hatası, modifiye edilmiş vektör alanı güdüm yasası kullanılarak 8799,6 metre olarak hesaplanmıştır. Önerilen modifiye edilmiş vektör alanı güdüm yasası, klasik vektör alanı algoritmasına göre yüzde 10,3 daha az yol izleme hatasını netice vermektedir.



Şekil 3.9: Modifiye edilmiş vektör alanı güdüm yasası.

Şekil 3.9, modifiye edilmiş vektör alanı güdüm yasasının sayısal sonuçlarını göstermektedir. Şekil 3.8' ye göre İSÜA, referans yola ulaştığında önceden tanımlanmış yolu tam olarak hassas bir şekilde takip etmektedir. Bu yaklaşım, bu çalışmayı özgün kılan

katkılarından birisidir. Vektör alanı güdüm yasasını bir adım daha değiştirerek aşağıdaki Denklem 3.48 ile güçlendirilmiş vektör alanı güdüm yasası elde edilebilir.

$$\Psi_d = \varphi - \rho \varphi_e \left(\frac{d}{d_B} \right)^k - \rho v \frac{\varphi_e}{\alpha} \frac{d}{d_B} \quad (3.48)$$

Denklem 3.48 ile önerilen güdüm yasasına, vekil model optimizasyon yöntemi uygulanarak parametre optimizasyonunu sağlayan bir algoritma geliştirilecektir.

Vekil model optimizasyon yöntemi, genetik algoritma gibi diğer optimizasyon yöntemleri ile kıyaslandığında daha az hesaplama maliyetini gerektirmektedir [81]. Vektör alanı güdüm yasası dört adet parametrenin optimize edilmesine ihtiyaç duyar. Vektör alanı güdüm yasası, diğer güdüm yasalarına göre çoklu parametreye sahiptir ve değerlendirmesi uzun zaman alır. Bu sebeple, bu gibi hedef fonksiyonlarla uyumlu olan vekil model optimizasyon yöntemi tercih edilmiştir. Bu çalışmada vekil model optimizasyon yönteminin vektör alanı güdüm yasasına uyarlanması öneren bu model, akıllı vektör tabanlı güdüm yasası (AVTGY) olarak adlandırılmıştır.

Vekil model optimizasyon yönteminin prensibi oldukça basittir. Vekil model optimizasyon tekniği, hedef fonksiyonunun sonucunu küresel olarak en aza indiren girdi değerini bulmak için kullanılır. Bu çalışmada hedef fonksiyonu, vektör alanı güdüm yasasıdır. Vekil model yöntemi, optimizasyonun başlangıcında hedef fonksiyonunun üst ve alt sınırları içindeki başlangıç noktalarını rastgele oluşturur. Başlangıç noktalarının sayısı kullanıcı tarafından kontrol edilebilir. Vekil model optimizasyon yönteminin çalışma prensibini daha iyi ifade edebilmek için örnek olarak üç tane başlangıç noktasının (değerlendirilen noktalar: x_1 , x_2 ve x_3) olduğunu varsayalım. Başlangıç noktaları hedef fonksiyonunda değerlendirilir: $f(x_1)$, $f(x_2)$ ve $f(x_3)$. Model fonksiyon olan $s(x)$, değerlendirilen noktaları ve hedef fonksiyonundaki değerleri kullanılarak herhangi bir enterpolasyon yöntemiyle türetilir. Bu aşamada kullanıcı Gauss işlemi, eğri enterpolasyonu ve polinom enterpolasyonu gibi herhangi bir enterpolasyon yöntemini seçebilir. Hedef fonksiyonun değerlendirme sonuçlarından en küçüğünün $f(x_2)$ olduğunu varsayalım. Hedef fonksiyonda en küçük sonuca sahip olan bu değer için neden olduğu nokta seçilir (x_2) ve buna görev noktası adı verilir. Vekil model optimizasyon yöntemi, görev noktası olarak seçilen bu noktaya yakın örnek noktaları rastgele oluşturur. Üç örnek noktanın (s_1 , s_2 ve s_3) oluşturulduğunu varsayalım. Normalde örnek noktalarının sayısı üçten çok daha fazladır. Anlatımı

basitleştirmek için bu sayı üç olarak seçilmiştir. Örnek noktaların sayısı kullanıcı tarafından kontrol edilebilir. Oluşturulan örnek noktalar liyakat ismi verilen fonksiyonda değerlendirilir ve ardından hedef fonksiyonunda değerlendirilmek üzere en iyi örnek nokta seçilir. Liyakat fonksiyonunun formülü $m(x)$, aşağıdaki Denklem 3.49' de gösterilmiştir:

$$m(x) = w \frac{s(x) - s_{min}}{s_{max} - s_{min}} + (1 - w) \frac{d_{max} - d(x)}{d_{max} - d_{min}} \quad (3.49)$$

Liyakat fonksiyonunda yer alan d_{max} , seçilen herhangi iki örnek nokta arasındaki muhtemel en büyük mesafedir. d_{min} ise seçilen herhangi iki numune noktası arasındaki muhtemel en küçük mesafedir. $d(x)$, x noktasındaki örnek noktanın diğer örnek noktalara olan en küçük mesafesidir. s_{max} , vekil fonksiyondaki örnek noktaların değerlendirilmesiyle elde edilen değerlerin en büyüğüdür. s_{min} , vekil fonksiyondaki örnek noktaların değerlendirilmesiyle elde edilen değerlerin en küçüğüdür. $s(x)$, vekil fonksiyonda x noktasında bulunan örnek noktanın değerlendirme değeridir. Ağırlık değeri (w), 0 ile 1 arasında bir değer olmak koşuluyla kullanıcı tarafından seçilir. w 'nin büyük bir değer olması, vekil değerlerin etkisini arttırarak, arama işleminde yer alan vekil değerlerin daha hassas küçültülmesini sağlar. Küçük bir w değeri, değerlendirilen noktalardan uzaktaki noktalara öncelik vererek aramayı yeni alanlara yönlendirir. Büyük bir w değeri algoritmanın, minimum değerlerin bulunduğu alana tam olarak odaklanmasına olanak tanır ve doğru ve hassas bir minimum değer elde edilmesini sağlar. w 'nin küçük değeri ise, algoritmanın yerel minimum engellerden kurtulmasını ve küresel minimumu bulmasını sağlar. Liyakat fonksiyonunda her numune noktası değerlendirildikten sonra liyakat fonksiyonunu en küçük yapan numune noktası seçilir ve bu numune noktası değerlendirilmek üzere hedef fonksiyonuna gönderilir. Bu örnek nokta artık değerlendirilmiş bir nokta olarak adlandırılır. Güncellenen değerlendirilen noktalar arasından yeni bir görevli nokta seçilir ve yukarıda açıklanan prosedürlerin aynıları, vekil model algoritmasının durdurma kriteri karşılanıncaya kadar devam eder. Bu optimizasyon yönteminde en yaygın kullanılan durdurma kriteri kullanıcı tarafından belirlenen maksimum yineleme sayısıdır.

Vekil model optimizasyon yönteminin vektör alanı güdüm yasasına uyarlanmasını anlatan iş akışı aşağıda özetlenmiştir:

Adım 1: Girdi parametrelerin belirlenmesi ve başlangıç koşulların tanımlanması

Temas noktalarının listesi: $W_1 = [1000 \ 1000]$, $W_2 = [3000 \ 3000]$ and $W_3 = [5000 \ 1000]$

İSÜA' nın x-y koordinatlarındaki başlangıç konumu, $p^t = [2000 \ 1000]$

İSÜA' nın hızı, $v = 9 \text{ metre/saniye}$

İSÜA' nın başlangıç rota açısı, $\Psi_p = 90 \text{ derece}$

İSÜA' nın maximum dönüş hızı, $\dot{\Psi}_{max} = 1 \text{ derece/saniye}$

Sayısal benzetimin zaman adımı büyüklüğü, $d_t = 10 \text{ saniye}$

Hedef fonksiyon: vektör alanı güdüm yasası

Hedef fonksiyonunda optimize edilmesi istenilen girdi parametreleri: φ_e, d_B, k ve α

Hedef fonksiyonun çıktı parametresi: toplam yol izleme hatası, d

Vekil model optimizasyon yönteminin durdurma kriteri: Hedef fonksiyonun maksimum değerlendirme sayısı 200 olarak belirlenmiştir.

Adım 2: Birinci görev (İlk referans yolun başlangıç noktası W_1 ' dir ve hedef temas noktası W_2 ' dir.) için optimum φ_e, d_B, k ve α değerlerinin bulunması. Bu adımda, vekil model optimizasyon algoritması en uygun değerleri bulmak için her yenilemede farklı kombinasyonlarda φ_e, d_B, k ve α değerlerini deneme amacıyla vektör alanı güdüm yasasına gönderir. 200 yinelenin sonunda durdurma kriterine ulaşıldığı için algoritma sona erer ve optimal değerler belirlenmiş olur.

NOT: İSÜA hedef temas noktasına (W_2) dokunmalıdır. Vektör alanı güdüm yasası bu ön koşula göre çalışır.

Adım 2.1: Başlangıç noktası ve hedef temas noktasını birleştiren doğrunun eğimini netice veren açının hesaplanması aşağıdaki Denklem 3.50 ile hesaplanması, φ

$$\begin{aligned} \varphi &= \text{atan} \left(W_{2y} - W_{1y}, W_{2x} - W_{1x} \right) \\ &= \text{atan}(3000 - 1000, 3000 - 1000) \end{aligned} \quad (3.50)$$

Adım 2.2: Yol izleme hatasının Denklem 2.51 kullanılarak hesaplanması, d

$$d = \frac{\left| \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} \times p_x^t - p_y^t - \frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} \times W_{1x} + W_{1y} \right|}{\sqrt{\left(\frac{W_{2y} - W_{1y}}{W_{2x} - W_{1x}} \right)^2 + (-1)^2}} \quad (3.51)$$

$$= \frac{\left| \frac{3000 - 1000}{3000 - 1000} \times 2000 - 1000 - \frac{3000 - 1000}{3000 - 1000} \times 1000 + 1000 \right|}{\sqrt{\left(\frac{3000 - 1000}{3000 - 1000} \right)^2 + (-1)^2}}$$

Adım 2.3: İSÜA' nın önceden tanımlanmış referans yolun hangi cephesinde kaldığının Denklem 3.52 ile bulunması, ρ

$$\begin{aligned} \rho &= \text{sign} \left\{ \left([W_{2x} \ W_{2y}] - [W_{1x} \ W_{1y}] \right) \times \left([p_x^t \ p_y^t] - [W_{1x} \ W_{1y}] \right) \right\} \\ &= \text{sign} \{ ([3000 \ 3000] - [1000 \ 1000]) \\ &\quad \times ([2000 \ 1000] - [1000 \ 1000]) \} \end{aligned} \quad (3.52)$$

Adım 2.4: Güçlendirilmiş vektör alanı güdüm yasasını kullanarak İSÜA' nın ulaşması istenilen rota açısının Denklem 3.53 ile hesaplanması, Ψ_d

$$\Psi_d = \begin{cases} \varphi - \varphi_e \rho, & d_B < d \\ \varphi - \rho \varphi_e \left(\frac{d}{d_B} \right)^k - \rho v \frac{\varphi_e}{\alpha} \frac{d}{d_B}, & d_B \geq d \end{cases} \quad (3.53)$$

Adım 2.5: İstenilen rota açısını ve bir önceki zaman adımında ulaşılan rota açısını kullanarak Denklem 3.54 ile İSÜA' nın dönüş hızının derece/saniye cinsinden bulunması, $\dot{\Psi}$

$$\dot{\Psi} = (\Psi_d - \Psi_p) / dt \quad (3.54)$$

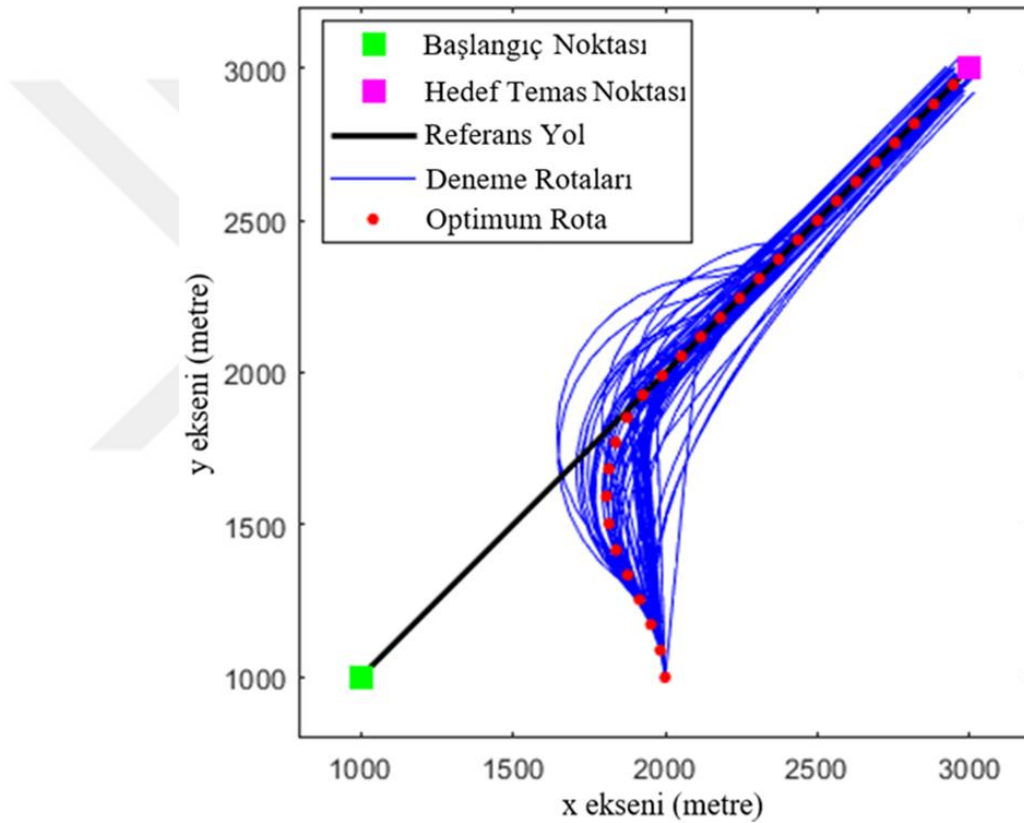
Adım 2.6: Hesaplanan dönüş hızının maximum dönüş hızından büyük olması durumunda İSÜA' nın ulaşması istenilen rota açısının Denklem 3.55 ile güncellenmesi

$$\Psi_d = \Psi_p + \dot{\Psi}_{max} dt \quad (3.55)$$

Adım 2.7: İSÜA' nın bir sonraki zaman adımında ulaşacağı mevcut konumunun Denklem 3.56 ile belirlenmesi, $p^{t+1} = [p_x^{t+1} p_y^{t+1}]$

$$\begin{aligned} p_x^{t+1} &= p_x^t + v \cos d(\Psi_d) dt \\ p_y^{t+1} &= p_y^t + v \sin d(\Psi_d) dt \end{aligned} \quad (3.56)$$

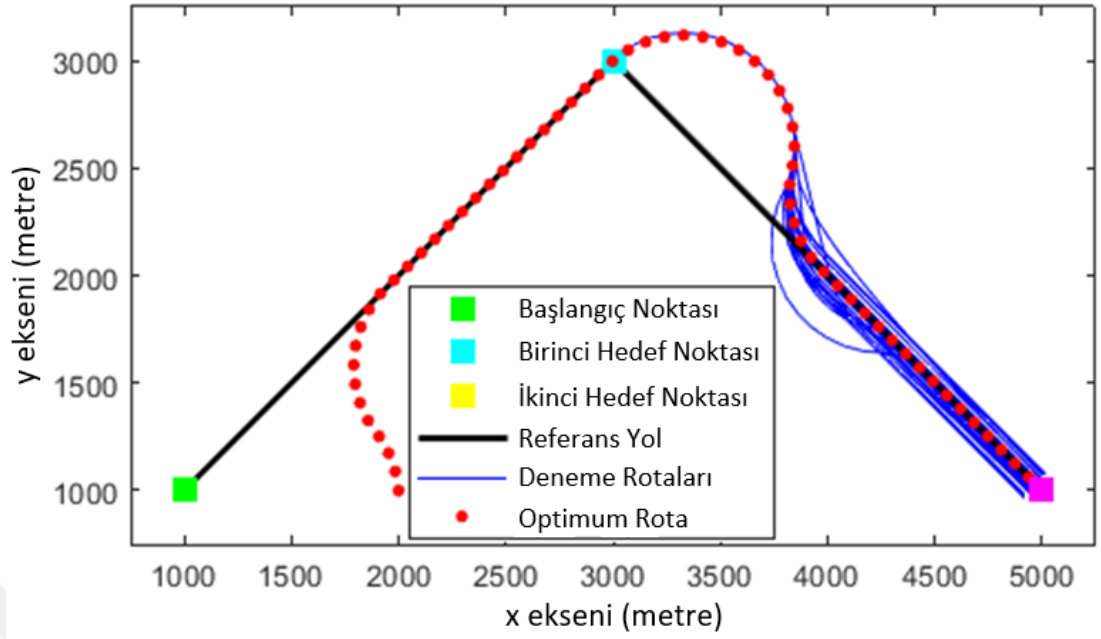
Adım 2.1' den adım 2.7' ye kadar olan tüm hesaplamalar, İSÜA ilk hedef temas noktasına ulaşana kadar devam eder. Şekil 3.10 hedef fonksiyonunda bulunması istenilen optimal φ_e, d_B, k ve α değerlerinin aranmasını göstermektedir.



Şekil 3.10: İlk görev için vektör alanı güdüm yasasında yer alan optimum φ_e, d_B, k ve α değerlerinin aranması.

200 yinelemenin sonunda en az yol izleme hatasına neden olan optimal φ_e, d_B, k ve α değerleri sırasıyla 60,1372 derece, 581,9773 metre, 2,3998 ve 8,3254 olarak belirlenmiştir.

Adım 3: İkinci görev (İkinci referans yolun başlangıç noktası W_2 ' dir ve hedef temas noktası W_3 ' tür.) için optimum φ_e, d_B, k ve α değerlerinin bulunması. Adım 2' de yer alan tüm hesaplamalar tekrar yapılarak 200. yinelemenin sonunda minimum yol izleme hatasına sebep olan değerler sırasıyla 63,14 derece, 371,44 metre, 2,19 ve 8,00 olarak bulunmuştur.



Şekil 3.11: İkinci görev için vektör alanı güdüm yasasında yer alan optimum φ_e , d_B , k ve α değerlerinin aranması.

Şekil 3.11, güçlendirilmiş vektör alanı hedef fonksiyonunda optimal rotaları ve optimal değerlerin aranmasını göstermektedir. Kırmızı renkli noktalar birinci ve ikinci görevler için belirlenen optimum rotayı göstermektedir. Akıllı vektör tabanlı güdüm yasası yöntemi kullanılarak toplam yol izleme hatası 7446,0 metre olarak hesaplanmıştır. Akıllı vektör tabanlı güdüm yasası, genetik algoritma entegreli klasik vektör alanı güdüm yasasına kıyasla yaklaşık yüzde 29' luk bir iyileşmeye göstermektedir. Yol izleme hataları ve yukarıda bahsedilen üç yöntemin optimum değerleri aşağıdaki Tablo 3.1' de özetlenmiştir.

Tablo 3.1. Akıllı vektör tabanlı güdüm yasası kullanılarak yol izleme hatalarının düşürülmesi.

		Optimum Değerler				Toplam Yol İzleme Hatası (m)	İyileşme (%)
		φ_e	d_B	k	α		
Klasik Vektör Alanı +Genetik Algoritma	1. Görev	81,88	383,56	3,96	76,80	9703,1	-
	2. Görev	69,15	193,74	3,45	84,49		
Modifiyeli Vektör Alanı +Genetik Algoritma	1. Görev	88,93	477,77	4,32	14,50	8799,6	10,27
	2. Görev	45,77	126,09	8,18	41,06		
Akıllı Vektör Tabanlı Güdüm Yasası	1. Görev	60,14	581,98	2,40	8,33	7491,4	29,52
	2. Görev	63,14	371,44	2,19	8,00		

3.4. Muhakeme Esaslı GÜDÜM YASASI

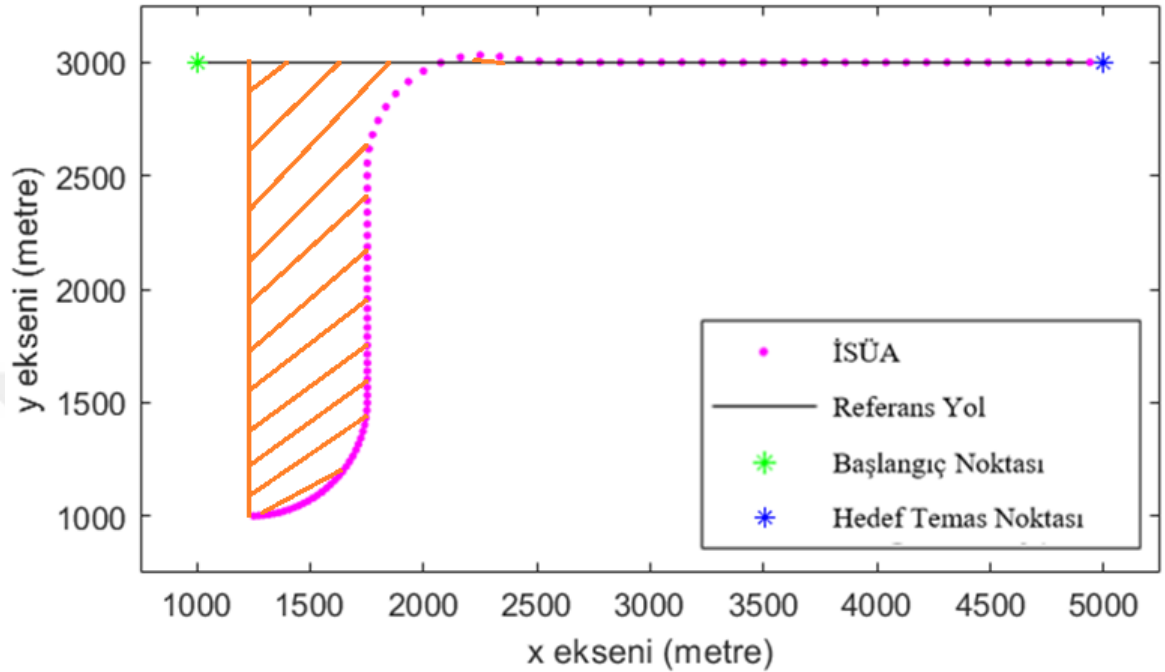
Bu çalışmada, muhakeme esaslı güdüm yarasası (MEGY) isimli özgün bir güdüm yarasası geliştirilmiştir. MEGY, İSÜA' nın referans yoldan uzaklığını ve İSÜA' nın rotası ile referans yol arasındaki sapma açısını göz önünde bulundurarak referans yol üzerinde sanal hedef noktaları oluşturur. Sanal hedef noktalarının referans yol üzerine tayin edilmesinde bu iki önemli parametrenin kullanılması yol izleme hatalarının azalmasını sağlamaktadır. İSÜA, referans yoldan çok uzak ise İSÜA' nın referans yola hızlıca ulaşmasını sağlamak için İSÜA' nın rota açısı referans yola dik olacak şekilde ayarlanmaktadır. İSÜA, kritik bölgeye gelinceye kadar rota açısı referans yola dik olarak seyrini sürdürür. İSÜA, kritik bölgeye ulaştığında ise referans yol üzerindeki sanal hedef noktaları oluşturulmaya başlar. Bu sanal hedef noktaları, İSÜA en az yol izleme hatasını netice verecek şekilde oluşturulur. Kritik bölgenin limitinin ve sanal hedef noktalarının konumunun ne olacağına algoritmaya entegre edilen optimizasyon yöntemi karar verir. Rotanın sapma açısı ve yol izleme hatası dikkate alınarak toplam hataların bir optimizasyon yöntemi ile minimize edilmesi, önerilen MEGY' nı özgün yapan ilk değerdir. Optimizasyon yönteminin girdi verileri kritik bölgenin limiti ve sanal hedef fonksiyonların konumudur. Çıktı verisi ise referans yol ile İSÜA' nın oluşturduğu seyir hattının arasında kalan bölgenin alanıdır. Bu alan bulunurken ikinci dereceden sayısal integral alma yöntemi kullanılarak sayısal hatalar en aza indirilmiştir (İntegral ile bulunan alanlar Şekil 2.16' da turuncu çizgilerle taranarak gösterilmiştir).

MEGY' yi özgün kılan ikinci değer ise, uyarlamalı zaman adımı boyutları üzerine temellendirilmiş olmasıdır. Denklem 3.57 uyarlamalı zaman adımının hesaplanmasını göstermektedir.

$$dt(j) = \max\left(\frac{dt_{max}}{4}, dt_{max} \times \left(1 - 0,5 \times \frac{|psi - phi|}{180} - 0,5 \times \frac{d(j)}{\max(d)}\right)\right) \quad (3.57)$$

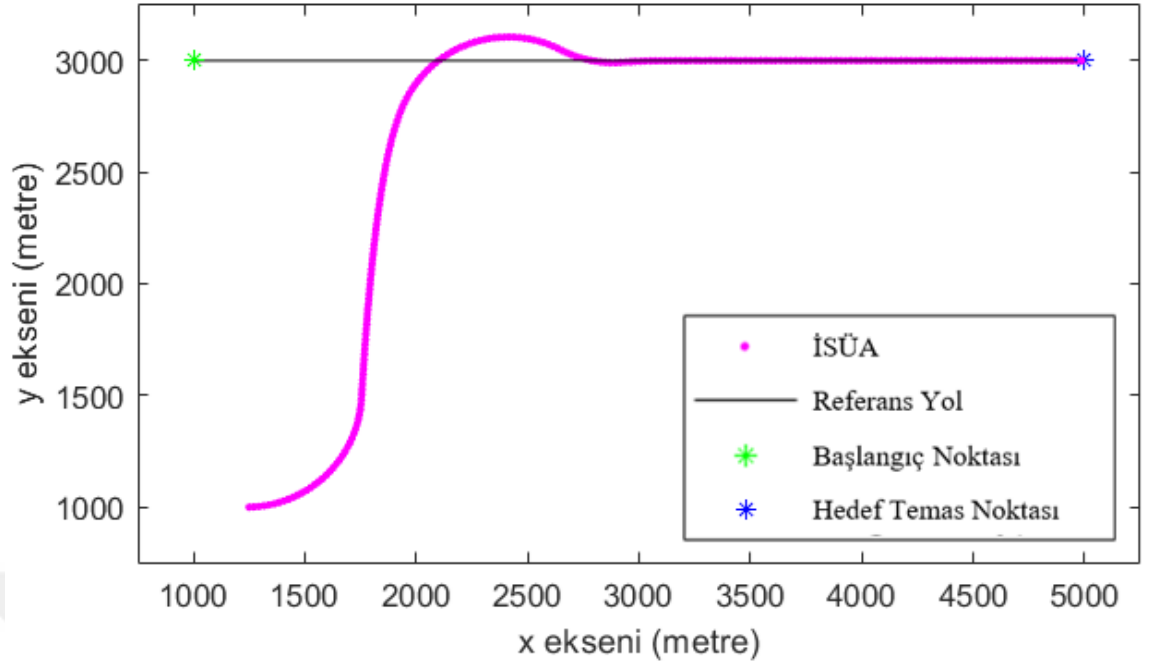
Denklem 3.57' da yer alan j zaman adımı numarasını ifade eder. dt , dt_{max} , psi , phi ve d sırasıyla zaman adımı boyutunu, sayısal benzetimde kullanılabilecek maksimum zaman adımı boyutunu, İSÜA' nın rota açısını, referans yolun eğim açısını ve yol izleme hatasının

büyükliğini sembolize eder. Denklem 3.57' a göre uyarlamalı zaman adımının alt limiti olarak kullanıcı tarafından belirlenen maksimum zaman adımının dörtte biri kullanılır. Zaman adımı büyüklüğü bu değerin altına düşemez.



Şekil 3.12: Uyarlamalı zaman adımı boyutları üzerine temellendirilmiş Muhakeme Esaslı Güzüm Yasası (MEGY).

Şekil 3.12' de olduğu gibi İSÜA seyir esnasında MEGY sayesinde, sabit bir değer yerine birbirinden farklı zaman adımlarını kullanır. İSÜA referans yola çok uzak olduğunda ve İSÜA' nın rota açısı ile referans yolun açısı arasındaki sapma yüksek iken küçük zaman adımları kullanılır. Böylece keskin manevra gibi kritik seyir esnasında daha iyi bir modelleme elde edilir. İSÜA referans yolu istikrarlı bir şekilde yakalamaya başladığı andan itibaren kademeli olarak zaman adımı boyutları artırılır ve toplam zaman adımı sayısı düşürülmüş olur. Bu durum, toplam optimizasyon süresinin kısılmasını, toplam benzetim süresinin azaltılmasını ve her zaman adımından gelecek olan toplam sayısal hataların düşürülmesini sağlar. Bu üç önemli avantaj, toplam yol izleme hatalarını ve özellikle toplam harcanan süreyi ciddi oranda azaltır. Bu iki özgün değer, MEGY' nı literatürdeki diğer algoritmalarından daha avantajlı hale getirmektedir. Şekil 3.13 sabit zaman adımları kullanan literatürdeki havuç kovalama güzüm yasasının sayısal sonuçlarını göstermektedir.



Şekil 3.13: Sabit zaman adımları kullanan havuç kovalama güdüm yasası.

Şekil 3.13’ de görüldüğü gibi klasik güdüm yasalarından biri olan havuç kovalama güdüm yasasının özellikle referans yolu yakaladıktan sonra bile sabit zaman adımı kullanması yüksek hesaplama maliyetlerine ve optimizasyon süresinin uzamasına sebep olmaktadır. Ayrıca her zaman adımından gelen sayısal integrasyon hataları da toplam sayısal hataları yükseltmektedir. Havuç kovalama algoritması referans yolu yakalamayı çalışırken tek bir parametrenin optimizasyonunu kullanır. Bu klasik güdüm yasasının bir diğer olumsuz yanındır. Yani İSÜA referans yoldan çok uzak iken sanal hedef noktasının oluşturulmasına sebep olan parametre ile İSÜA referans yola yaklaşıp kritik manevra yapacağı sırada tayin edilen sanal hedef noktasının parametresi aynıdır. Ancak, İSÜA referans yoldan çok uzak iken ve referans yola yakınken farklı seyir davranışında bulunmaktadır. Bu sebeple Şekil 3.13 olduğu gibi tek parametre kullanılması, İSÜA referans yoldan uzak iken dik bir şekilde değil de eğimli bir şekilde referans yola yaklaşırken, İSÜA referans yola yaklaştığında osilasyonlu bir manevra yapmasına neden olmuştur. Bu durum ise bu çalışmada önerilen MEGY’ na göre % 13,64 daha yüksek yol izleme hatasını netice vermektedir. Tablo 3.2 MEGY’ nin muhtelif güdüm yasalarına göre sağladığı iyileştirmeyi göstermektedir.

Tablo 3.2. Uyarlamalı zaman adımlarını kullanan MEGY' nın literatürdeki diğer güdüm yasaları ile karşılaştırılması.

Kullanılan Optimizasyon Yöntemi	Kullanılan Güdüm Yasası	Toplam Yol İzleme Hatası (m ²)	İyileşme (%)	Toplam Sayısal Benzetim Süresi (saniye)
Vekil Model Optimizasyon Yöntemi	MEGY (u)	995630,0	-	37,5
	MEGY (s)	1005100,0	0,95	145,8
	HKGY	1131400,0	13,64	130,8
	DOGY	1131426,1	13,64	134,3
	VAGY	1008200,0	1,26	119,5

Tablo 3.2' de yer alan MEGY (u), uyarlamalı zaman adımları üzerine kurgulanmış muhakeme esaslı güdüm yasasını temsil eder. MEGY (s), sabit zaman adımlarını kullanan muhakeme esaslı güdüm yasasıdır. HKGY, DOGY ve VAGY sırasıyla sabit zaman adımları kullanan havuç kovalama güdüm yasası, doğrusal olmayan güdüm yasası ve vektör alanı güdüm yasasını ifade etmektedir. Öncelikle uyarlamalı zaman adımı ve sabit zaman adımı kullanan MEGY kıyaslandığında, bu çalışma kapsamında önerilen uyarlamalı zaman adımına temellendirilmiş algoritma (37,5 saniye) sabit zaman adımı kullanan yöntem (145,8 saniye) göre yaklaşık 4 kat daha hızlı sonuç vermektedir. Ayrıca uyarlamalı zaman adımı kullanan MEGY (995630,0 m²), sabit zaman adımı güdüm yasasından (1005100,0 m²) daha az yol izleme hatasına sebep olmaktadır. Bu durum Denklem 3.57' de ifade edilen uyarlamalı zaman adımının kullanılmasının avantajını açık bir şekilde ortaya koymaktadır. Tablo 3.2' de yer alan HKGY ve DOGY tek bir parametre optimizasyonu kullandıkları için, yol izleme hatası ve toplam geçen süre açısından neredeyse yakın sonuçlar vermiştir. MEGY (u) algoritması ile HKGY ve DOGY kıyaslandığında yaklaşık %14 yol izleme hatasında iyileşme ve zaman bakımından yaklaşık 3,5 kat hızlı neticeler alınmaktadır. MEGY (u) ve VAGY karşılaştırıldığında, MEGY daha az yol izleme hatalarını netice veren sonucu yaklaşık 3 kat daha hızlı elde etmektedir. Bu karşılaştırmalar, önerilen MEGY algoritmasının avantajlı özgün değerini net bir şekilde göstermektedir.

BÖLÜM 4. ARDUİNO TABANLI BİR İNSANSIZ SU ÜSTÜ ARACININ TASARIMI

Bu çalışma kapsamında arduino programlama kartı ile çalışan bir İSÜA ve kablosuz haberleşme sağlayabilmek için uzaktan kumanda modülü tasarlanmıştır. Tasarlanan sistemler dış tasarım, iç tasarım ve kablosuz haberleşme sistemi olmak üzere üç alt başlığa ayrılarak ele alınmıştır.

4.1. İSÜA' nın Dış Tasarımı

İSÜA' nın dış tasarımı, fırçasız motorlar, fırçasız motor sürücüleri ve tekne gövdesi olmak üzere üç temel kısımdan oluşmaktadır. Tekne gövdesi üzerinde tahrik sisteminden başka, kablosuz haberleşmeyi sağlayan harici antenler ve su geçirmez ekipmanlar da yer almaktadır. Kablosuz haberleşmeyi, GY-NEO6MV2 GPS modülüne ait yassı anten ve NRF24L01 2.4 GHz kablosuz haberleşme modülüne ait harici SMA anten sağlamaktadır. Su geçirmez ekipmanlar ise teknenin bilgisayar ile haberleşmesini sağlayan konnektör, bataryanın şarj edilmesini sağlayan konnektör ve teknenin güç kaynağı ile tetiklenmesini sağlayan toggle anahtar olmak üzere üç farklı sızdırmaz parçadan oluşmaktadır.

4.1.1. Fırçasız motorlar

Fırçasız motorlar, fırçalı motorlar ile mukayese edildiğinde ciddi avantajları öne çıkmaktadır. Fırçalı motorlarda rotor üzerinde bulunan sargılara, sabit bir şekilde konumlandırılmış iletken fırçalar elektrik akımını ilettiği için sürtünmeden dolayı zamanla moturun fırçaları aşınmaktadır. Aşınan fırçalar, önce motorun performansını olumsuz yönde etkilemekte daha sonra da motorun ömrünün tükenmesine sebep olmaktadır. Bu durumda, motor üzerindeki aşınmış fırçaların yenisi ile değiştirilmesi gerekmektedir. Diğer yandan fırçasız doğru akım motorlarında elektromanyetik alanın

oluşmasını sağlayan sargılar, hareketsiz olarak motorun statoru üzerinde bulunur. Bu sebeple fırçasız motorlar, elektrik akımının tamamlanması için aşınma dezavantajına sahip fırça gibi herhangi bir iletkene ihtiyaç duymaz. Elektrik akımı stator üzerindeki sargılardan geçerken rotor üzerindeki dizili mıknatısların elektromanyetik kuvvet ile dönmesi sağlanır. Sargıların hareketsiz kalmasından ve aşınacak bir iletken olmamasından dolayı, fırçasız motorlar fırçalı motorlara göre daha uzun ömürlüdürler. Fırçalı motorların ömrünü ve performansını arttırmak için karbon fırçalar kullanılsa da fırçasız motorlar kıvılcım çıkarmama, daha az gürültü ile çalışma, yüksek performans ve yüksek tork avantajlarından dolayı kritik mekanik sistemlerde daha yaygın tercih edilmektedir.



Şekil 4.1: Yüksek devirli fırçasız motor ve pervanesi.

Şekil 4.1, A2208 ve 2600 kv teknik özelliklere sahip yüksek devir ve küçük boyut avantajlarından dolayı tercih edilen bir fırçasız moturu göstermektedir. A2208, motorun boyutlarını temsil ederken 2600 kv değeri ise volt başına devir sayısını temsil etmektedir. Örneğin, 11.1 volt gerilim ile çalıştırılan bu fırçasız motor dakikada 28860

(11.1×2600) devir yapabilmektedir. Şekil 4.1’ de yer alan fırçasız motor 2S yada 3S batarya ile yani 7.4 volt yada 11.1 volt gerilime sahip güç kaynağı ile çalıştırılabilmektedir. Fırçasız motor verimli çalışırken 4-10 amper değer aralığında akım çekmekte olup güvenli ve sağlıklı çalışabilmesi için maksimum 12 ampere kadar akım çekmesine izin verilmesi gerekmektedir.

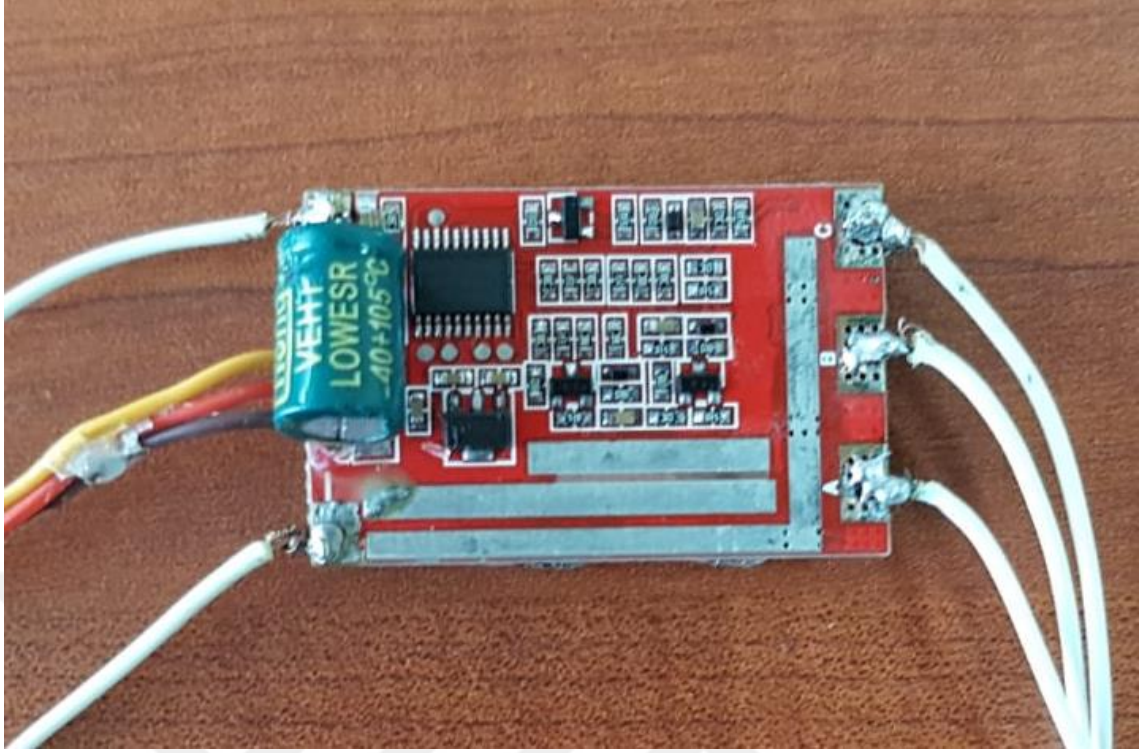
Fırçalı motorların bir diğer avantajı, galvanizli su geçirmez boya ile motor gövdesi ve sargıları boyandığından dolayı su altı uygulamalarında kullanılabilir olmasıdır. Özellikle deniz suyu, fırçalı motorların gövdesinin, şaftının ve fırçaların bulunduğu kısımların korozyona uğramasına ve paslanmasına sebep olmaktadır.

Yukarıda bahsedilen avantajlarından dolayı bu çalışmada, İSÜA’ nın tahrik sistemi olarak 2 adet fırçasız motor kullanılmıştır. Bu sayede İSÜA yüksek manevra kabiliyetine sahip olmuştur. Fırçasız motorlar yüksek akım çekebildiklerinden dolayı soğutulması gerekmektedir. Fırçasız motorların doğal olarak yani harici bir soğutucu akışkan ve devir-daim sistemi kullanmaksızın soğutulabilmesi için fırçasız motorlar su ile doğrudan temas edecek şekilde teknenin kış alt gövdesine entegre edilmiştir. Fırçasız motorların 3.17 mm çapa sahip olan millerine saat yönünde ve saat yönünün zıttında döndüğü zaman tekneyi ileri yönde itecek karşıt pervane açısına sahip aynı boyutta iki adet pervane takılarak yatay ekseninde oluşacak olan kuvvetler sönmülmüştür.

4.1.2. Fırçasız motor sürücüler

İSÜA’ na yerleştirilen fırçasız motorların denetimini yapmak için iki farklı fırçasız motor sürücüsü, teknenin kış alt gövdesine su ile temas edecek şekilde yerleştirilmiştir. Fırçasız motor sürücüler, fırçasız motorlar gibi ısındığından, sürücülerin deniz suyu ile soğutulması sağlanmıştır. Sürücülerin içine su girmemesi için gerekli yalıtımlar yapılmıştır.

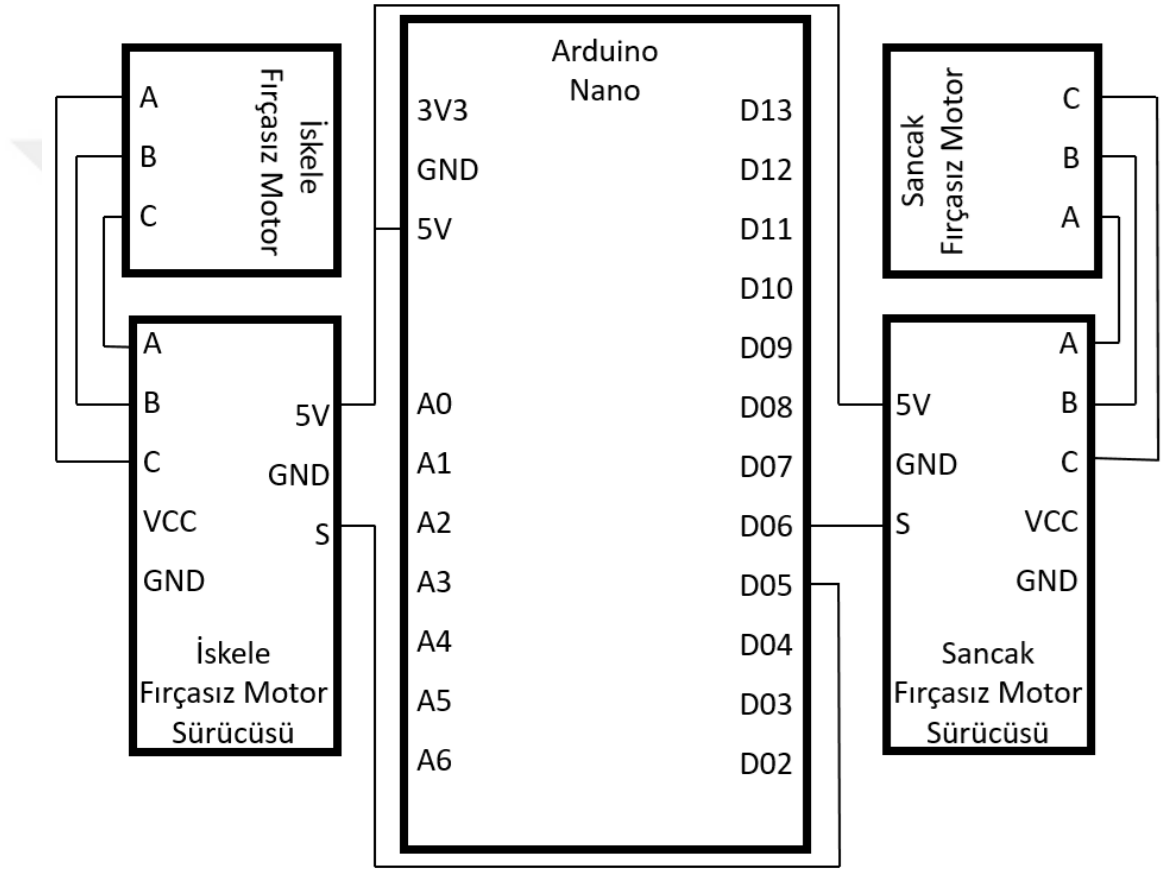
Fırçasız motor sürücüsün arduino programlama kartı ile kontrol edilmesi “servo” kütüphanesi kullanılarak gerçekleştirilebilmektedir. Servo kütüphanesini kullanarak fırçasız motorların kontrol edilmesini sağlayan örnek açık kaynaklı kod EK A kısmında yer almaktadır.



Şekil 4.2: Fırçasız motor sürücüsünün iç yapısı.

Şekil 4.2, bu çalışmada kullanılan fırçasız motor sürücüsünün iç yapısını göstermektedir. Fırçasız motor sürücüsü üzerindeki 330 μ F değerindeki kondansatör, akım ve voltaj dalgalanmaların önüne geçmek için kullanılır. Fırçasız motor sürücüsü üzerinde bulunan dokuz adet transistör, doğru zamanda doğru kabloya akımın gönderilmesini sağlar. Bu transistörler yüksek akımdan dolayı aşırı ısınırlar ve soğutulmaları gerekmektedir. Bu sebeple fırçasız motor sürücülerin su sızdırmazlığı yapılarak deniz suyu ile temas edecek şekilde teknenin dış gövdesine bağlantısı yapılmıştır. Şekil 4.2’ de yer alan fırçasız motor sürücüsü, maksimum 30 amper değerine kadar akım sağlayabilmektedir. Fırçasız motor sürücüsünün üzerinde bulunan kondansatörün uçlarında yer alan artı ve eksi kutuplar, Li-Po bataryaya bağlanarak fırçasız motor sürücüsüne enerji sağlanmaktadır. Li-Po bataryalar bu fırçasız motor sürücüsü için 2 hücreli veya 3 hücreli olarak seçilebilir. Kondansatörün yakınında sinyal (sarı renkli kablo), 5 volt (kırmızı renkli kablo) ve GND (siyah renkli kablo) kabloları yer almaktadır. Sinyal kablosu, fırçasız motorların arduino gibi bir programla kartından kontrol edilmesini sağlar. 5 volt ve GND kabloları, programla kartının beslenmesini sağlar. Programlama kartı, harici bir gerilim ile beslenmesi durumunda sinyal kablosu ile birlikte GND kablosunun da programla kartına bağlanması gerekmektedir. Fırçasız

motor sürücü üzerindeki A, B ve C ile simgelenen diğer üç kablo ise fırçasız motora bağlanır. Bu kabloların fırçasız motora bağlanma sırası önemlidir. B kablosu daima fırçasız motorun orta kablosuna bağlanmalıdır. A ve C ile simgelenen diğer iki kablo fırçasız motor üzerinde kalan diğer iki uç kabloya farklı şekillerde bağlanarak fırçasız motorun saat yönünde veya saat yönünün zıttı yönde dönmesi sağlanır. Fırçasız motor sürücülerinin arduino nano mikroişlemci ve fırçasız motorlarla bağlantısını gösteren devre şeması, aşağıdaki Şekil 4.3’ te yer almaktadır.



Şekil 4.3: Fırçasız motor sürücüsünün mikroişlemci ve fırçasız motorlar ile bağlantısının devre şeması.

4.1.3. Tekne gövdesi

Tasarımı yapılan teknenin gövdesi, üst güverte ve alt gövde olmak üzere iki parçadan oluşmaktadır. Teknenin üst güvertesinde, Şekil 4.4’ te müstakil olarak gösterilen NRF24L01 2.4 GHz kablosuz haberleşme modülüne ve GY-NEO6MV2 GPS modülüne ait sırasıyla SMA anten ve yassı anten bulunmaktadır.



Şekil 4.4: Kablosuz haberleşme için kullanılan harici antenler.

Yassı anten, üst güverteye su geçirmeyecek bir şekilde monte edilmiştir. Yassı anten dört farklı uydudan sinyallerin alınmasını ve GY-NEO6MV2 GPS modülüne alınan sinyallerin iletilmesini sağlamaktadır. Yassı anten üzerinde bulunan iletken, su ile temas ettiğinde anten yüzeyinden kalktığından dolayı bu iletken, yassı antene uç kısımlarından yapıştırılarak sabitlenmiştir. Teknenin üst güvertesinde bulunan bir diğer harici anten ise NRF24L01 2.4 GHz kablosuz haberleşme modülüne ait SMA antendir. SMA anten, teknenin kumanda ile kontrol edilmesinde veya teknenin otonom kontrolü sırasında kullanıcı bilgisayarına teknenin konum, hız, rota açısı gibi verilerin anlık olarak aktarılmasını sağlayan sinyallerin iletilmesinde görev alır. SMA antenin 1 km mesafesindeki geniş menzili, teknenin hem kumanda ile kontrolünde hem de otonom seyir sırasında verilerin iletilmesinde büyük avantaj sağlar.

Şekil 4.5' te yer alan teknenin üst güvertesinde, harici antenlerden başka arduino programlama kartının bilgisayara bağlanmasını sağlayan ve güç kaynağı olarak kullanılan 11.1 volt gerilime sahip Li-Po bataryanın şarj edilmesini sağlayan su geçirmez iki adet konnektör bulunmaktadır. Teknenin üst güvertesinde kış tarafta iskele

yönünde bulunan su geçirmez konnektör bataryanın şarj edilmesini ve sancak yönünde bulunan sızdırmaz konnektör ise teknenin bilgisayara bağlanmasını sağlar.



Şekil 4.5: Tasarlanan İSÜA' nın üst güvertesi.

İSÜA' na takılan güçlü motorlar ile keskin manevralar yapılırken teknenin stabilitesini temin edebilmek için, Şekil 4.5' te yer alan teknenin iskele ve sancak bordalarına iki adet yüzer kanat takılmıştır. Teknenin ağırlık merkezi bu iki kanadın iç kısmında kaldığı için daha dengeli bir seyir temin edilmiş olmaktadır.

Şekil 4.6, teknenin alt gövdesini göstermektedir. Şekil 4.6' da, yüzer kanatlara takılmış iki adet yüksek devirli fırçasız motor ve alt gövdenin kış tarafına yakın yerlere monte edilmiş iki adet fırçasız motor sürücüsü bulunmaktadır. Alt gövdenin kış orta kısmında ise tekneye enerjinin sağlanmasını kontrol eden su geçirmez toggle anahtar yerleştirilmiştir. Toggle anahtar, güç kaynağı ile fırçasız motor sürücüleri arasındaki bağlantı kablolarına yerleştirilmiştir. Toggle anahtar, iskele yöne baktığında güç kaynağı ile fırçasız motor sürücüleri arasındaki devre tamamlanmakta ve tekne çalışmakta, sancak yöne baktığında ise enerji kesilmektedir. Kablolarda oluşan direnç, kablounun uzunluğu arttıkça artacağından dolayı, batarya ile fırçasız motor sürücüleri arasındaki mesafe olabildiğince kısa tutulmuştur ve yüksek akıma dayanıklı kablolar kullanılmıştır.



Şekil 4.6: Tasarlanan İSÜA' nın alt gövdesi.

Teknenin tahrik sistemi için kullanılan fırçasız motorların soğutucu akış dinamiği motorun ön kısmından arka kısmına doğru dizayn edilmiş olduğundan motorlar geminin baş kısmına bakacak şekilde yerleştirilmiştir. Yani fırçasız motorlara pervane takılmasa bile motorlar teknenin baş kısmına doğru bakacak şekilde yerleştirildiğinden tekneyi ileri doğru az da olsa itebilmektedir. Bu durum motorların dönüş yönünden bağımsız olarak gerçekleşmektedir. Motorlar teknenin baş kısmına bakacak şekilde yerleştirilmek şartıyla saat yönünde de dönse saat yönünün tersine de dönse tekneyi ileri doğru itecek akışı sağlamaktadır. Bu sebeple fırçasız motorların yönü öne bakmaktadır. Aksi halde pervanenin üreteceği akış ile motorun soğutucu akış yönü zıt olacağından verimsiz bir tahrik gücü oluşacaktır.

4.2. İSÜA' nın İç Tasarımı

İSÜA' nın iç tasarımında 11.1 volt gerilime sahip Li-Po güç kaynağı, arduino nano programlama kartı, NRF24L01 2.4 GHz kablosuz haberleşme modülü, GY-NEO6MV2

GPS modülü ve QMC5883 GY-271 3 eksen dijital pusula bulunmaktadır. Şekil 4.7, İSÜA' nın iç tasarımını göstermektedir.



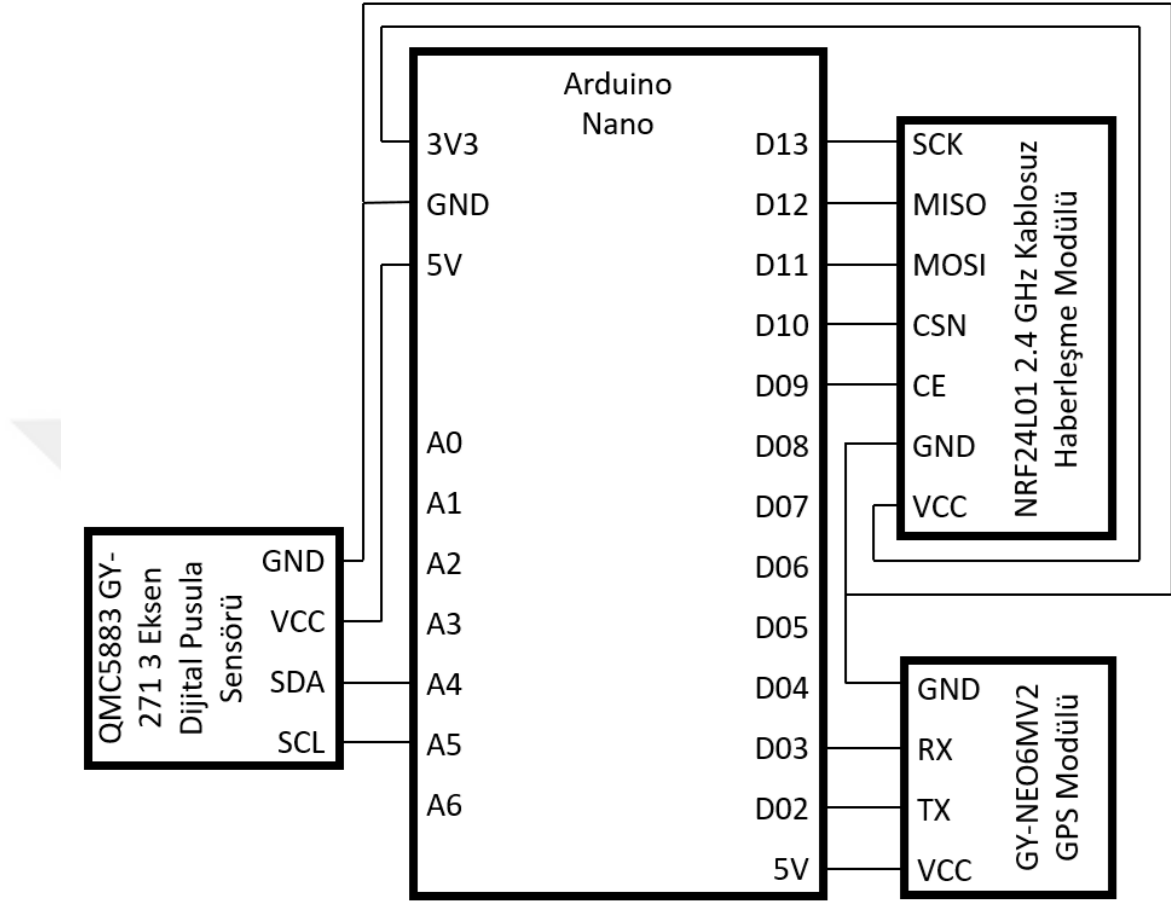
Şekil 4.7: Tasarlanan İSÜA' nın iç yapısının görünümü.

İSÜA' nın güç kaynağı olarak Şekil 4.7' de yer alan 11,1 volt gerilime sahip, 3400 mah kapasiteli bir batarya seçilmiştir. Yüksek kapasiteye sahip güç kaynağı sayesinde, İSÜA uzun süre otonom seyrini yerine getirebilmektedir. Örneğin İSÜA, 1.7 amperlik akım çekmesi durumunda yaklaşık 2 saat su üstünde kesintisiz seyir yapabilmektedir. Teknenin en büyük ağırlığını oluşturan batarya, su hattının altına gelecek şekilde yerleştirildiğinden tekne su içine gömülerek alabora olmadan dengeli bir seyir yapabilmektedir.

Teknenin su yalıtımı yeterli ve düzgün bir şekilde yapılmış olmasına rağmen, teknenin su alma ihtimaline karşı tüm elektronik modüller (arduino nano programlama kartı, NRF24L01 2.4 GHz kablosuz haberleşme modülü, GY-NEO6MV2 GPS modülü ve QMC5883 GY-271 3 eksen dijital pusula) teknenin iç üst güvertesine yerleştirilmiştir.

Şekil 4.8, teknenin iç tasarımında yer alan NRF24L01 2.4 GHz kablosuz haberleşme modülünün, GY-NEO6MV2 GPS modülünün ve QMC5883 GY-271 3 eksen dijital pusula sensörünün, arduino nano mikroişlemciye bağlantısının devre şemasını göstermektedir. Şekil 4.8' de bulunan pinlerin yerleri temsili olup, Şekil 4.7' de yer alan elektronik kartların kablo bağlantısının daha iyi ifade edilebilmesi için bu devre şeması

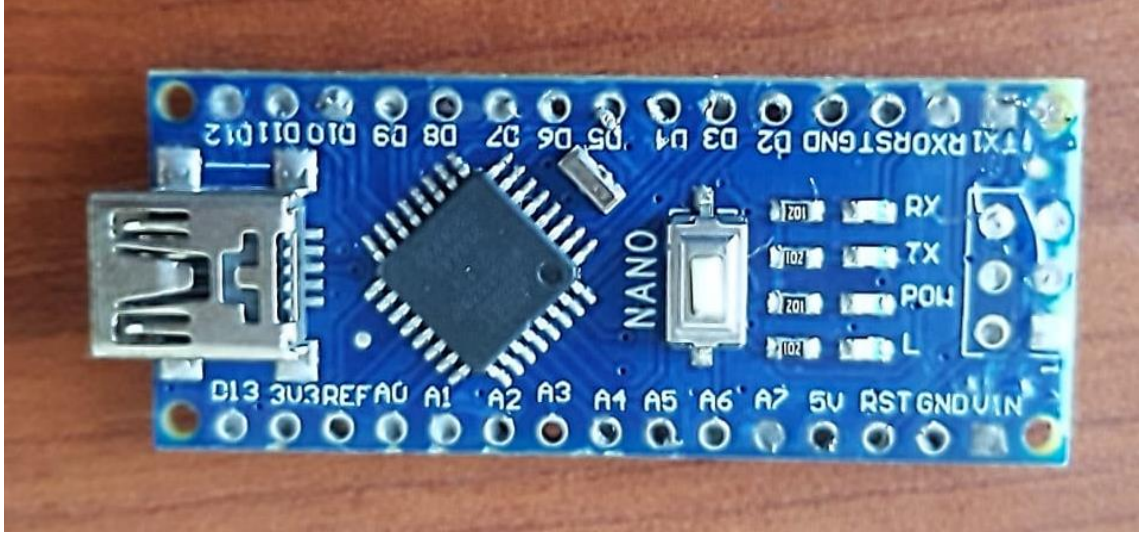
izilmiřtir. Teknenin i tasarımında yer alan elektronik kartlar, ařağıdaki alt bařlıklarda tek tek incelenerek anlatılacaktır.



řekil 4.8: İ tasarımıda yer alan elektronik kartların devre řeması.

4.2.1. Arduino nano

Arduino nano programlama kartı kk yapısı ve ucuz maliyeti sebebiyle bu alıřmada tercih edilmiřtir. Arduino nano zerinde 0' dan 13' e kadar dijital pinler bulunmaktadır. 0 ve 1 numaralı pinler RX ve TX pinleri olduklarından dolayı, arduino nano bu pinleri bilgisayar ile iletiřim kurarken kullanılır. Bu nedenle, 0 ve 1 numaralı pinler harici modller iin kullanılacaksa bilgisayar ile baėlantı sırasında bu pinlerin harici modlden ıkarılması gerekmektedir. Aksi halde bilgisayar ile veri-alıř veriři yapılırken problem yařanabilir. Arduino nano zerindeki dijital pinlerin bazıları PWM zelliėe sahip pinlerdir (rneėin řekil 4.9' da yer alan pin 5 ve pin 6 gibi).



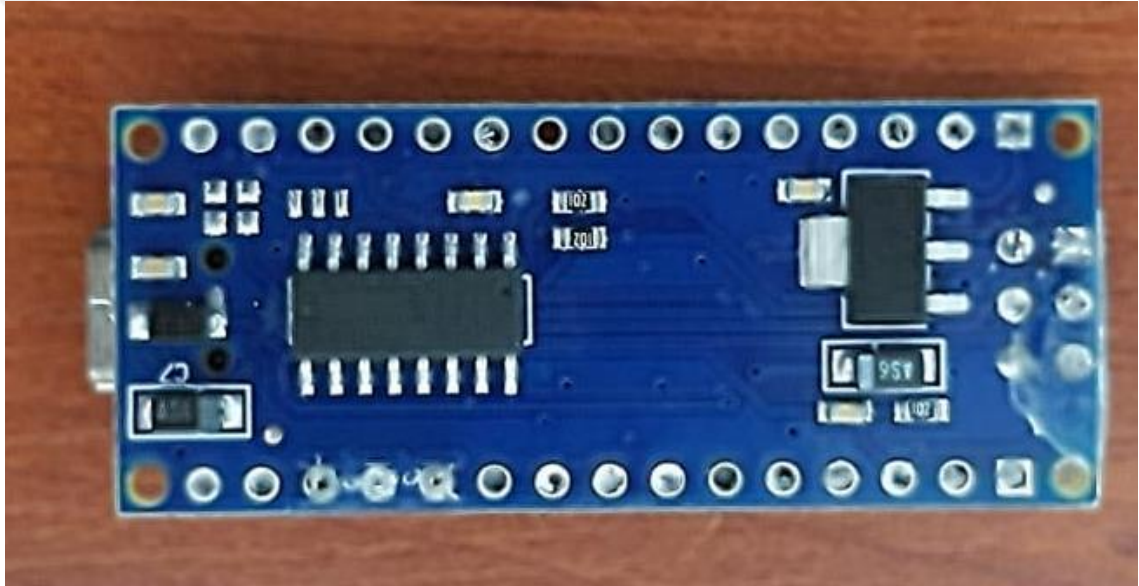
Şekil 4.9: Arduino nano kartının ön yüzü.

PWM özelliğine sahip, pinler 0 ile 255 değer aralığındaki tam sayılar ile kontrol edilerek pinlere gönderilen voltajın değeri ayarlanabilir. 255 rakamı maksimum voltaja yani 5 volta tekabül eder. PWM pinlerine 0 girilmesi ise 0 voltaja karşılık gelir. Yani bu durumda pinden akım sağlanamaz. 0 ile 255 değeri arasında kalan diğer rakamlar ise doğru orantılı olarak voltajın 5 volta kadar artmasına karşılık gelir. Bu çalışmada kullanılan fırçasız motor sürücüleri PWM pinlerinden gönderilen sinyaller ile kontrol edilmektedir. PWM özelliğine sahip olmayan diğer pinler (örneğin Şekil 4.9’ da yer alan pin 2 ve pin 4 gibi) ise ya sisteme sadece 5 volt sağlar yada sistemden akımı keser. Yani bu pinlere 1 rakamı komut olarak girildiğinde sisteme 5 volt verir, 0 rakamı girildiğinde sisteme enerji göndermez. Arduino IDE programında, PWM pinlerini kontrol etmek için “AnalogWrite” komutu kullanılırken, diğer dijital pinleri kontrol etmek için “DigitalWrite” komutu kullanılır.

Şekil 4.9’ da yer alan VIN pini, arduino kartının 7-24 volt değerleri arasında bir gerilim ile beslenmesine olanak sağlayan pozitif kutuptur. Bataryanın pozitif kutbu VIN pinine ve negatif kutbu ise GND pinine bağlanarak arduino harici bir güç kaynağı ile beslenebilir. Arduino üzerindeki voltaj düzenleyicileri sayesinde dışarıdan alınan 7 ile 24 volt arasındaki gerilim 5 volta düşürülür. Ancak, arduino kartının aşırı ısınmadan düzgün ve kararlı bir şekilde çalışabilmesi için maksimum 9 volt değerine kadar bir harici güç kaynağı kullanılması gerekmektedir. Aksi halde, arduino üzerindeki kondansatörler aşırı ısınma yapmaktadır. Arduino, sürekli 5 volt gerilim sağlayan bir harici kaynak ile beslenmek istenirse, devamlı 5 volt potansiyel fark sağlayan güç

kaynağının pozitif kutbu arduino üzerindeki 5 volt pinine ve negatif kutbu ise arduino üzerindeki herhangi bir GND pinine bağlanabilir. Arduino üzerindeki bir diğer güç pini ise Şekil 4.9’ da yer alan 3V3 pinidir. Bu pin, voltaj düzenleyiciler sayesinde kullanılacak olan sensör veya modüllere sürekli 3.3 volt gerilim sağlar.

Arduino nano üzerinde bulunan analog pinler “A” harfi ile simgelenmiştir (örneğin Şekil 4.9’ da yer alan A1 ve A5 gibi). Arduino nano toplam 7 tane analog pinine sahiptir. Bu analog pinlerden, sensörlerden gelen verilerin okunması sağlanır. A4 ve A5 analog pinleri özel pinler olup bu pinler sensörlerin SDA (Serial Data) ve SCL (Serial Clock) çıkışlarına bağlanır. SDA pini veri iletişimi için, SDA ise veri senkronizasyonu için kullanılır.



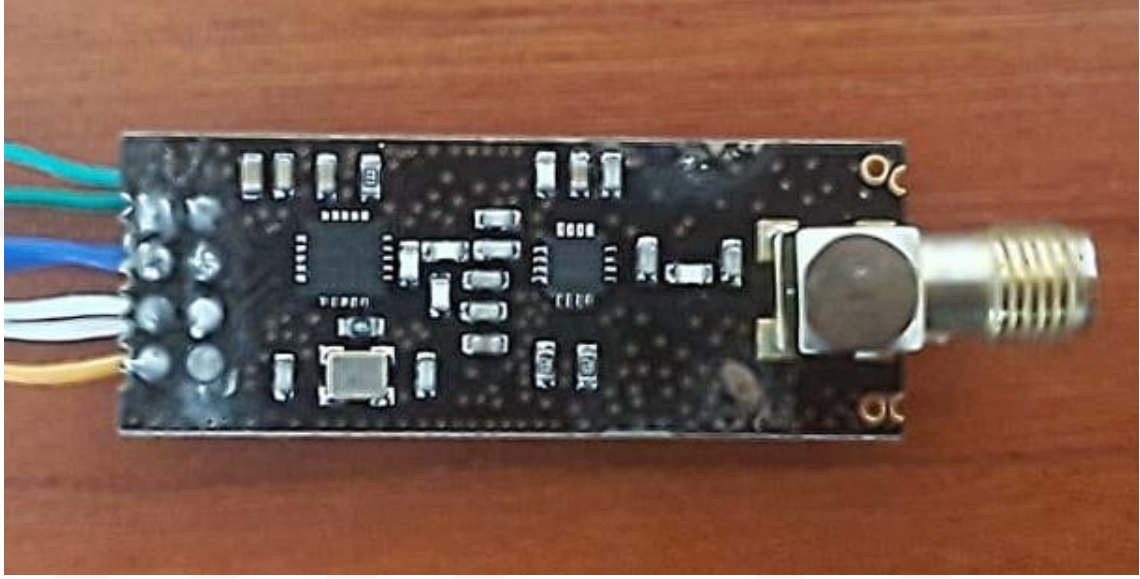
Şekil 4.10: Arduino nano kartının arka yüzü.

Şekil 4.10 arduino nano programlama kartının arka yüzünü göstermektedir. Arduinonun arka yüzünde voltaj düzenleyici ve programlama kartının bilgisayar ile haberleşmesini sağlayan CH340 entegresi bulunmaktadır. Arduino nano, mikro usb kablosu vasıtasıyla bu entegre üzerinden veri akışını sağlar.

4.2.2. NRF24L01 2.4 GHz kablosuz haberleşme modülü

Tasarımı yapılan İSÜA ile harici bir bilgisayar ile anlık olarak veri alış-verişini sağlamak için NRF24L01 2.4 GHz kablosuz haberleşme modülü kullanılmıştır. NRF24L01 2.4 GHz kablosuz haberleşme modülünün yaklaşık 1 km menzilinın olması

geniş bir alanda veri akışına olanak sağlamaktadır. NRF24L01 2.4 GHz kablosuz haberleşme modülü iki önemli görevi yerine getirebilmektedir. Bunlardan birincisi, istenildiği zaman İSÜA, bir kullanıcı tarafından kumanda ile yönlendirilebilmektedir. İkincisi ise, İSÜA önceden belirlenen görevleri uydudan ve dijital pusuladan aldığı veriler ile yerine getirirken anlık hızı, mevcut konumu, rota açısı gibi verileri anlık olarak harici bir bilgisayara gönderilmesini sağlar.

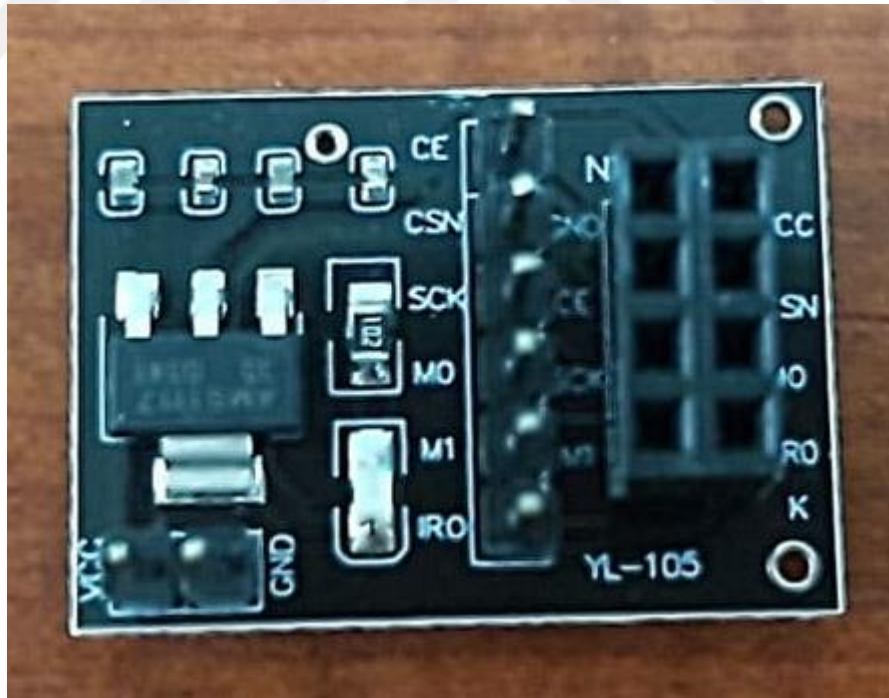


Şekil 4.11: NRF24L01 2.4 GHz kablosuz haberleşme modülü.

Şekil 4.11’ de yer alan NRF24L01 2.4 GHz kablosuz haberleşme modülü VCC (gerilimin artı kutbu), GND (gerilimin eksi kutbu), CE, CSN, SCK, MOSI, MISO ve IRQ olmak üzere sekiz adet pine sahiptir. VCC pini pozitif kutbu ifade eder. 1.9 - 3.9 volt arasında gerilim sağlayan bir güç kaynağına bağlanmalıdır. Daha yüksek gerilimlerin bu pine bağlanması NRF24L01 2.4 GHz kablosuz haberleşme modülüne zarar verir. 3.9 volttan daha yüksek veya 1.9 volttan daha düşük gerilimlerin VCC pinine bağlanması, fiziksel sorunlara neden olmasa bile kablosuz haberleşme kesintiye uğrar. VCC pini, arduinonun 3V3 çıkış pinine bağlanabilir. GND pini güç kaynağının negatif kutbunu ifade eder yani toprak pinidir. CE, Chip Enable kelimelerinin baş harfini ifade eden verinin gönderilmesini veya verinin alınmasını sağlayan aktif HIGH pinidir. Bu pin arduino nano üzerinde herhangi bir dijital pine bağlanabilir. CE pininin hangi dijital pine bağlandığı arduino nano programlanırken tanıtılmalıdır. CSN, Chip Select Not kelimelerinin baş harfini ifade eder. Bu pin açık tutulan aktif LOW pinidir. CSN pini LOW olunca modül, SPI girişlerini veri almak için kullanır. Bu pin de CE pini

gibi arduino nano üzerindeki herhangi bir dijital pine programlama sırasında tanıtılmak şartıyla bağlanabilir. SCK (Serial Clock) pini, SPI ile sağlanan saat verilerinin alınması için kullanılır ve arduino nano üzerinde dijital 13. pine bağlanmalıdır. MOSI (Master Out Slave In) pini, modülün SPI girişidir. Arduino nano üzerinde 11. dijital pine bağlanmalıdır. MISO (Master In Slave Out) pini ise modülden SPI çıkışını ifade eder. MISO pini, arduino nano üzerinde 12. dijital pine bağlanmalıdır. IRQ (Interrupt Request) pini, arduino gibi mikrodenetleyiciler tarafından harici cihazlardan gelen kesme taleplerini işlemek için kullanılır. Bu pinin arduino üzerinde herhangi bir pine bağlanma zorunluluğu yoktur. İsteğe bağlı olarak boş bırakılabilir.

NRF24L01 2.4 GHz kablosuz haberleşme modülünün güç pinleri olan VCC ve GND, arduino nano üzerindeki 3V3 ve GND pinlerine doğrudan bağlanabiliyor olsa da, arduino nano mikrodenetleyiciden gelen akımların bazı durumlarda yetersiz kalması kablosuz haberleşmede kesintiler yaşanmasına neden olmaktadır. Bu sebeple NRF24L01 2.4 GHz kablosuz haberleşme modülü, Şekil 4.12’ de yer alan adaptöre bağlanarak iletişimde yaşanan problemler ortadan kaldırılabilir.



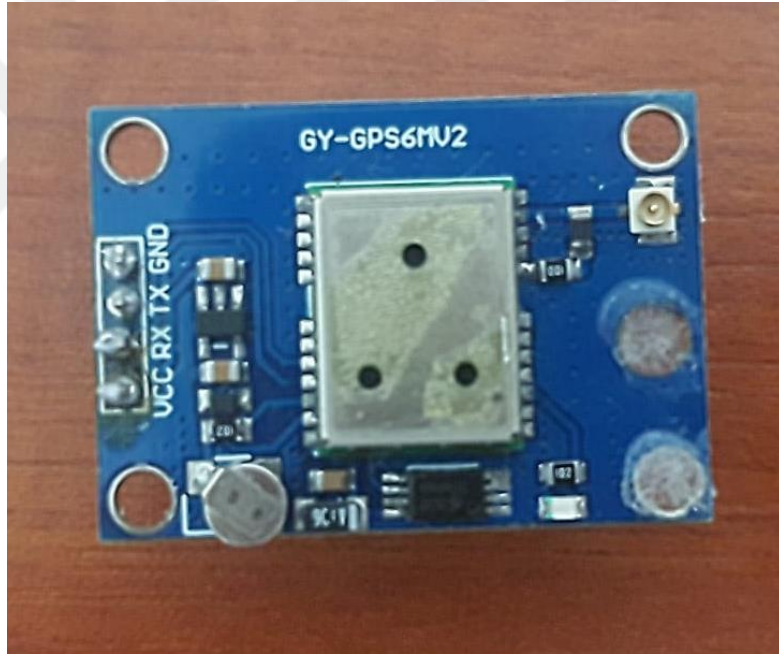
Şekil 4.12: NRF24L01 2.4 GHz kablosuz haberleşme modülü için adaptör.

NRF24L01 2.4 GHz kablosuz haberleşme modülüne özgü tasarlanan Şekil 4.12’ deki adaptör, 3.3 volt yerine 5 volt ile beslenmelidir. Adaptör üzerinde yer alan voltaj

reg lat r  5 voltu, 3.3 volta d         mod le kesintisiz akım saėlamaktadır. Bu adapt r n dar alanlarda yapılan projelerde kar ıla ılan yer problemi gibi sebeplerden dolayı kullanılmaması durumunda NRF24L01 2.4 GHz kablosuz haberle me mod l n n g   pinlerine kesintisiz ve yeterli akım elde edebilmek i in en az 10 μ F deėerinde bir kondansat r baėlanmalıdır.

4.2.3. GY-NEO6MV2 GPS mod l 

GY-NEO6MV2 GPS mod l , coėrafi koordinat sistemindeki enlem ve boylam verileri ile uydudan İ  A' nın konum bilgisinin elde edilmesini saėlar. Bu mod l sadece konum verisi deėil, zaman bilgisinin, İ  A' nın hızının ve pruvaya a ısının da belirlenmesi i in kullanılabilir.



 ekil 4.13: GY-NEO6MV2 GPS mod l .

 ekil 4.13, GY-NEO6MV2 GPS mod l n  g stermektedir. GY-NEO6MV2 GPS mod l   zerinde VCC, RX, TX ve GND olmak  zere d rt adet pin vardır. GY-NEO6MV2 GPS mod l  dahili bir voltaj reg lat r  i erdiėinden, VCC ve GND pinleri 5 volt gerilim saėlayan g   kaynaėına doėrudan baėlanabilir. RX ve TX pinleri arduino nano  zerinde yer alan herhangi bir dijital pine ( rneėin RX pin 3' e ve TX pin 2' e) baėlanabilir. RX ve TX pinlerinin hangi dijital pinlere baėlandıėı, programlama sırasında arduino mikrodenetleyiciye SoftwareSerial ss(2, 3) komutu ile tanıtılması

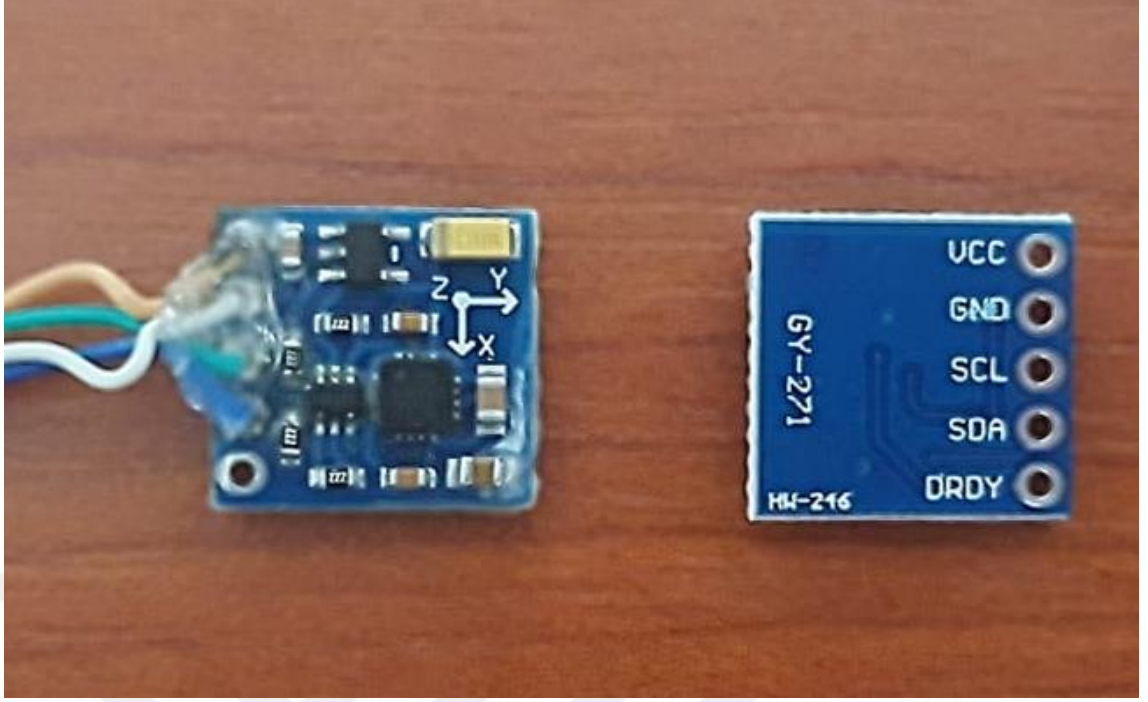
gerekmektedir. GY-NEO6MV2 GPS modülünün üzerinde bir adet şarj edilebilir pil ve bir adet led bulunmaktadır. Şarj edilebilir pil, anlık olarak modülün güç kaynağı kesildiğinde sisteme enerji sağlayarak uydudan veri akışının devamlılığını sağlar. Modül üzerindeki led, en az dört adet uydu ile haberleşme sağlandığında konum bilgisinin elde edildiğini göstermek maksadıyla yanıp sönmeye başlar. Konum bilgisinin elde edilmesi için GY-NEO6MV2 GPS modülüne ait yassı anten engel olmadan doğrudan gök yüzüne bakıyor olmalıdır. Bu sebeple, GY-NEO6MV2 GPS modülü kapalı alanlarda çalışmamaktadır. GY-NEO6MV2 GPS modülü, konumu 5 metrelik bir hata payı ile belirler. Bu çalışma kapsamında yapılan deneysel çalışmalardan elde edilen verilerde, 5 metrelik hata payı göz önünde bulundurularak değerlendirme yapılması büyük önem taşır.

4.2.4. QMC5883 GY-271 3 eksen dijital pusula

Şekil 4.14, QMC5883 GY-271 3 eksen dijital pusula sensörünü göstermektedir. Pusula sensöründe VCC, GND, SCL, SDA ve DRDY olmak üzere 5 adet pin bulunmaktadır.

QMC5883 GY-271 3 eksen dijital pusula sensörünün üzerinde voltaj regülatörü olduğu için bu sensörün VCC ve GND pinleri doğrudan 5 volt gerilim sağlayan güç kaynağına bağlanabilir. Sırasıyla, veri iletişimi ve veri senkronizasyonu için kullanılan SDA (Serial Data) ve SCL (Serial Clock) pinleri, arduino mikrodenetleyici üzerinde yer alan A4 ve A5 analog pinlerine veya doğrudan arduino üzerindeki SDA ve SCL pinlerine bağlanmalıdır.

QMC5883 GY-271 3 eksen dijital pusula, İSÜA' nın rota açısının hassas bir şekilde bulunması için kullanılan bir sensördür. GY-NEO6MV2 GPS modülü, İSÜA' nın hız ve rota açısını belirleyebiliyor olsa da, İSÜA hareketsiz durumdayken veya İSÜA yavaş hareket ediyorken GY-NEO6MV2 GPS modülü İSÜA' nın pruva açısını doğru olarak belirleyememektedir. Bu sebeple, İSÜA' nın pruva açısının hassas ve her durumda doğru bir şekilde belirlenmesi için QMC5883 GY-271 3 eksen dijital pusula sensörü kullanılmıştır.



Şekil 4.14: QMC5883 GY-271 3 eksen dijital pusula sensörü.

QMC5883 GY-271 3 eksen dijital pusula sensörü, elektromanyetik alandan etkilenme olasılığı göz önünde bulundurularak fırçasız motor, fırçasız motor sürücüler, NRF24L01 2.4 GHz kablosuz haberleşme modülü gibi diğer elektronik modüllerden mümkün olduğu kadar uzağa yani teknenin baş tarafındaki üst güverteye konumlandırılmıştır. Pusula sensörünün yönü ve nasıl konumlandırıldığı önemlidir. Bu sebeple, QMC5883 GY-271 3 eksen dijital pusula sensörü İSÜA' na, x-ekseni teknenin pruvasını gösterecek şekilde ve sensörün ön yüzü yukarıya bakacak şekilde yerleştirilmiştir.

4.3. Kablosuz Haberleşme Sistemi

Tasarlanan İSÜA' nın uzaktan kontrolünü ve veri akışını sağlayan uzaktan kumanda sistemi Şekil 4.15' te gösterilmektedir. Bu uzaktan kumanda sistemi arduino nano, NRF24L01 2.4 GHz kablosuz haberleşme modülü ve güç kaynağından oluşmaktadır. Kablosuz haberleşme sisteminin üzerinde, sancak ve iskele motorları kontrol edebilen iki adet düğme yer almaktadır. Her bir düğmenin bir bacağı arduino nano üzerindeki eksi kutba (GND) bir direnç ile birlikte ve diğer bacağı ise arduino nano kartı üzerindeki dijital bir PWM pinine bağlanmıştır. PWM pininden düşük bir voltaj gönderilerek

düğmeler aktif olmaktadır. Eksi kutba birer direnç bağlanması sebebi ise düğmelere basıldığında kısa devre olacağından karta zarar vermemek için elektriksel güç bu dirençler ile sönümlenmektedir.

Mikroişlemciye, düğmelere basıldığında bilgi verebilmek için her bir düğmenin dirençten önceki eksi kutup bağlantısına, arduino nano üzerindeki analog pinler bağlanmıştır. Analog pinlerdeki değerler her bir zaman adımında okunmaktadır. Düğmelere basıldığında, okunan değerler yükselmekte ve bu sayede tekne üzerindeki iskele veya sancak motorlara çalışmaları için komutlar verilmektedir.



Şekil 4.15: Tasarlanan İSÜA' nın kumanda ve harici bilgisayar ile haberleşme sistemi.

Kumanda, harici bilgisayara bağlı değilken çalışabilmesi için şarj edilebilir bir güç kaynağına da sahiptir. Şarj edilebilir güç kaynağı 3.7 volt Li-ion batarya tercih edilmiştir. Batarya ile arduino nano arasındaki güç bağlantısına anahtar yerleştirilerek güç kaynağının istenildiği zaman devreye girmesi ve arduino kartının bilgisayara veya bir usb vasıtasıyla şarj cihazına bağlandığında istenildiği zaman bataryanın şarj edilebilmesi sağlanmıştır. NRF24L01 2.4 GHz kablosuz haberleşme modülünün maksimum çalışma gerilimi olan 3.3 volt değerine her ne kadar bataryanın gerilimi yakın olsa da arduino nano üzerindeki 3.3 volt gerilim veren pine modül bağlanarak

daha kararlı bir kablosuz haberleşme sağlanmıştır. Arduino nano kartında, iskele ve sancak motorlara komut gönderen düğmelerden kaynaklanabilecek olası gürültüleri sönmölemek ve sürekli akım sağlayabilmek için NRF24L01 2.4 GHz kablosuz haberleşme modölünün güç pinlerine 10 µF değörinde seramik kondansatör bağlanmıştır.

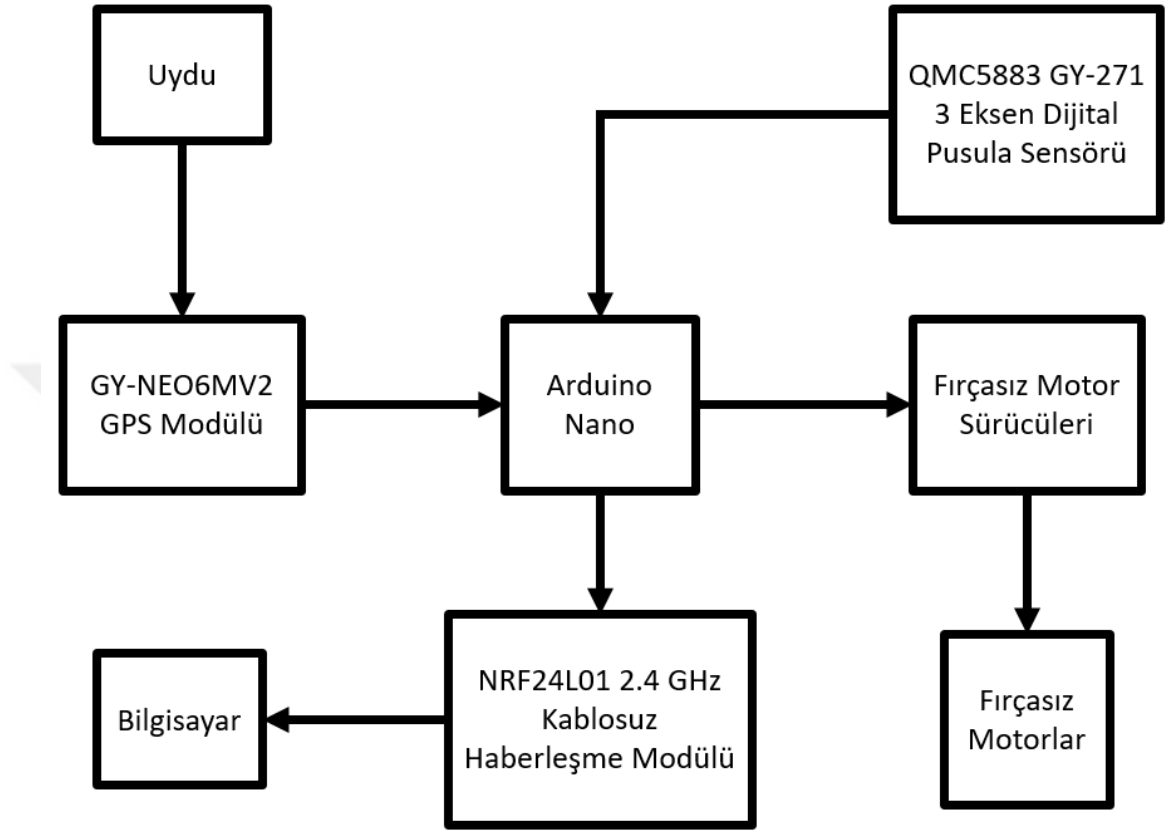
Kumanda sistemi, istenildiğı zaman tekneyi kontrol edebilen bir denetleyici olurken istenildiğı zaman da teknedenden gelen verileri harici bilgisayara ulaştıran bir haberleşme modölü olarak kullanılabilir. Şekil 4.15’ te yer alan kablosuz haberleşme sisteminin bir kumanda gibi iskele ve sancak motorlara komut verebilmesi için ve İSÜA otonom seyir yaparken İSÜA’ na ait konum, rota açısı, hız ve zaman gibi verilerin anlık olarak kullanıcı bilgisayarına kablosuz olarak gönderilmesi için gerekli açık kaynaklı kodlar EK A kısmına dahil edilmiştir.

BÖLÜM 5. İNSANSIZ SU ÜSTÜ ARACININ DENETİMİ

Bu çalışma kapsamında tasarımı yapılan İSÜA' nın çalışma prensibi temelde ikiye ayrılır. Birincisi, bir uzaktan kumandada bulunan iki adet buton ile teknenin iskele ve sancak yönlerinde bulunan fırçasız motorların kontrol edilmesidir. Kumandada bulunan sağ düğmeye basıldığında teknenin iskele tarafında bulunan fırçasız motor çalışır ve tekne sağ yönde manevra yapar. Benzer şekilde, kumanda üzerinde bulunan sol düğmeye basıldığında teknenin sancak tarafında bulunan fırçasız motor çalışarak teknenin sola dönmesi sağlanır. Her iki düğmeye birden basıldığında ise tekne tam yol ileri gider. İSÜA' nın ikinci çalışma prensibi, verilen görevleri otonom olarak dışarıdan hiçbir müdahale olmadan yerine getirmesi için kurgulanan bir sisteme dayanır.

İSÜA' nın otonom kontrolünde, GY-NEO6MV2 GPS modülü, QMC5883 GY-271 3 eksen dijital pusula ve NRF24L01 2.4 GHz kablosuz haberleşme modülü önemli rol oynar. GY-NEO6MV2 GPS modülü uydudan aldığı verilerle, İSÜA' nın konumunu, hızını, rota açısını, deniz seviyesinden yüksekliğini, zaman-tarih verilerini, verilerin kaç adet uydudan temin edildiğini tespit eder. GY-NEO6MV2 GPS modülünden bu verileri elde etmek için kullanılan açık kaynak kodu [82] EK B kısmına dahil edilmiştir. GY-NEO6MV2 GPS modülü, İSÜA' nın rota açısını doğru ve hassas bir şekilde ancak İSÜA hareket ediyorken verebilmektedir. İSÜA, hareketsiz durumdayken de denetiminin hassas bir şekilde sağlanabilmesi için QMC5883 GY-271 3 eksen dijital pusula kullanılmıştır. Bu sensör, İSÜA' nın anlık rota açısını her durumda hassas bir şekilde temin edebilmektedir. QMC5883 GY-271 3 eksen dijital pusuladan verileri sağlamak için ilgili açık kaynak kod [83] EK C kısmına yerleştirilmiştir. Son olarak, NRF24L01 2.4 GHz kablosuz haberleşme modülü de İSÜA' nın konum, hız ve rota açısı gibi verilerini, harici bir bilgisayara anlık olarak gönderir. İSÜA, verilen görevleri tamamladıktan sonra ilgili görevlerin doğru bir şekilde tamamlanıp tamamlanmadığını kontrol etmek ve İSÜA' nın gittiği rotanın grafiklerini çizdirebilmek için NRF24L01 2.4 GHz kablosuz haberleşme modülünden anlık olarak harici bilgisayara gelen bu

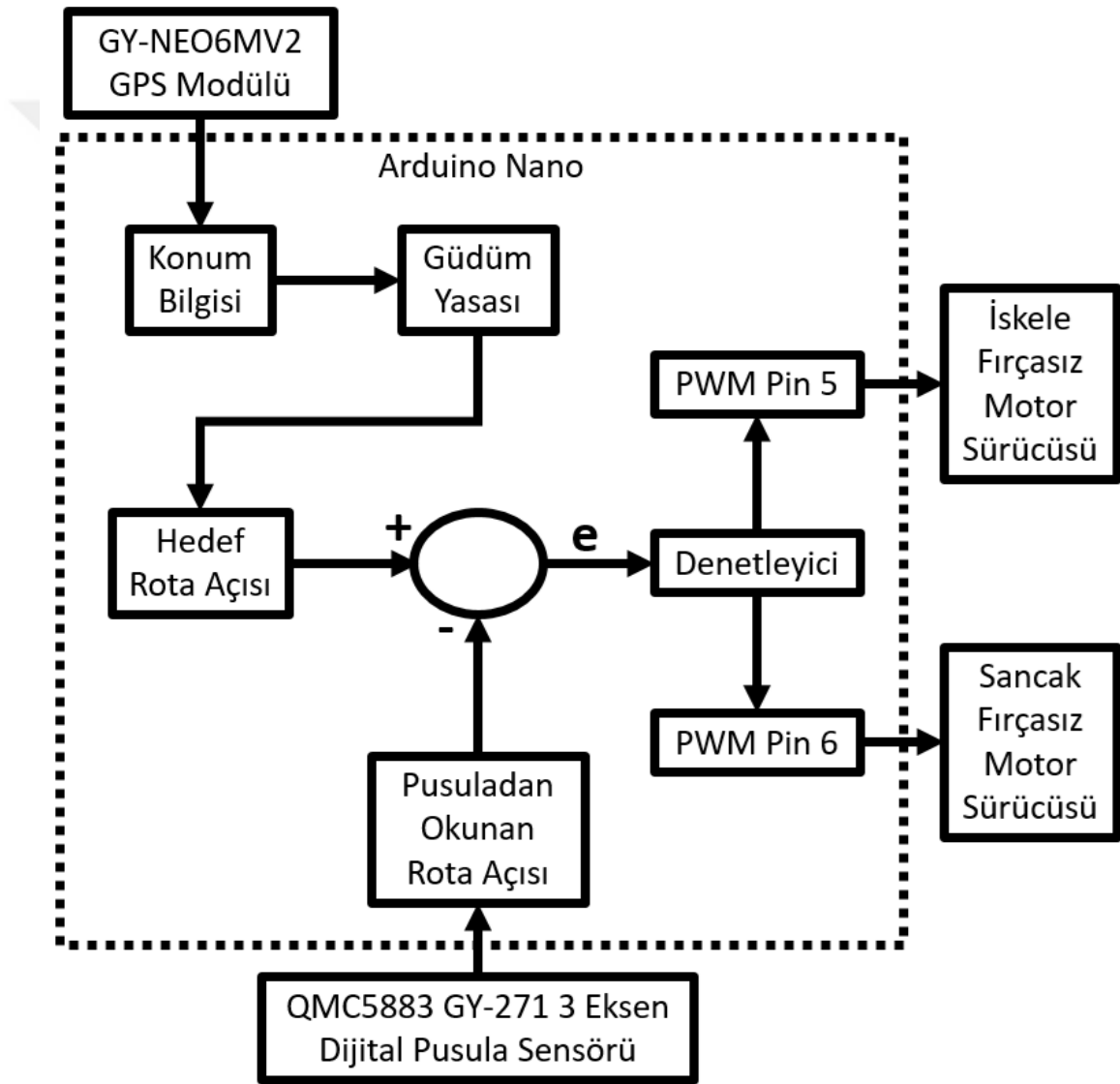
veriler kullanılmaktadır. NRF24L01 2.4 GHz kablosuz haberleşme modülünün alıcısı ve vericisi için gerekli olan açık kaynak kodları [84] sırasıyla EK D ve EK E kısımlarına konulmuştur.



Şekil 5.1: Arduino nano ve harici sistemler arasındaki veri akışı.

Şekil 5.1, arduino nano ve harici sistemler arasındaki veri akışını göstermektedir. Harici sistemler GY-NEO6MV2 GPS modülü, QMC5883 GY-271 3 eksen dijital pusula sensörü, NRF24L01 2.4 GHz kablosuz haberleşme modülü ve fırçasız motor sürücüleri olmak üzere dört gruba ayrılmıştır. GY-NEO6MV2 GPS modülünden ve QMC5883 GY-271 3 eksen dijital pusula sensöründen elde edilen veriler arduino nano mikroişlemciye girdi verisi olarak kullanılır. GY-NEO6MV2 GPS modülü harici bir yassı anten vasıtasıyla uydudan aldığı konum verilerini arduino nano mikroişlemciye gönderir. QMC5883 GY-271 3 eksen dijital pusula sensörü de elde ettiği anlık rota açısını arduino nano mikroişlemciye verir. NRF24L01 2.4 GHz kablosuz haberleşme modülü ve fırçasız motor sürücüleri, arduino nano mikroişlemcinin gönderdiği çıktı verilerini kullanırlar. NRF24L01 2.4 GHz kablosuz haberleşme modülü, arduino nano

mikroişlemciden gelen zaman bilgisi, İSÜA' nın anlık konumu, rota açısı ve hızı gibi verileri her zaman adımında kullanıcı bilgisayarına iletir. Fırçasız motor sürücüleri ise, İSÜA' nın önceden tanımlanmış referans yolu yakalayabilmesi için arduino nano mikroişlemci tarafından gönderilen sinyalleri kullanarak teknenin iskele ve sancak yönlerinde bulunan fırçasız motorlarını komuta eder. Arduino nano ve harici sistemler arasındaki veri akışı açıklandıktan sonra, İSÜA' nın otonom kontrolünde denetleyicinin rolünü daha iyi ifade edebilmek için arduino nano mikroişlemcinin iç yapısında gerçekleşen veri akışı Şekil 5.2' de gösterilmiştir.



Şekil 5.2: Arduino nano mikroişlemcinin iç yapısında gerçekleşen veri akışı.

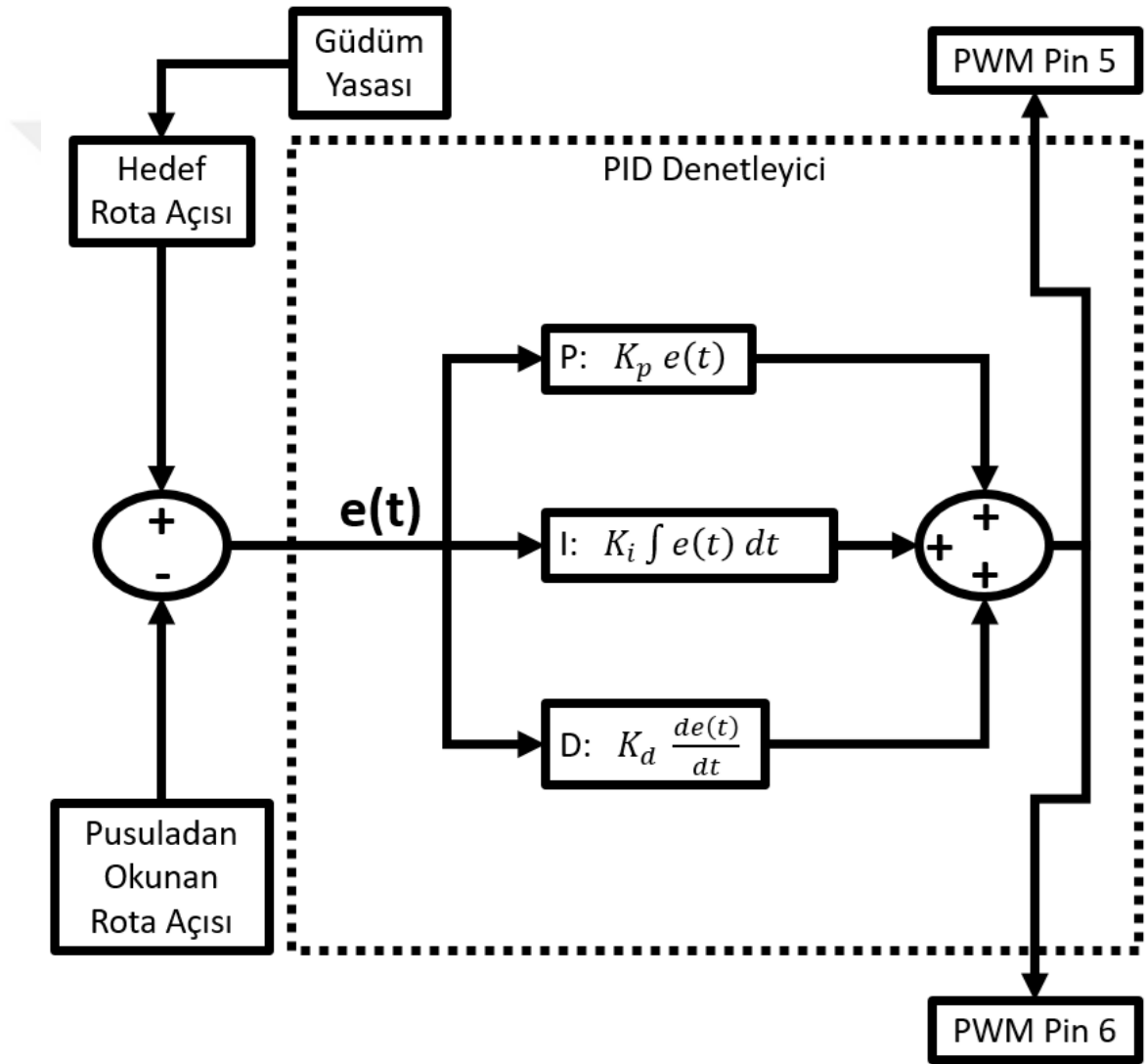
Şekil 5.2' de yer alan İSÜA' nın otonom denetiminde, öncelikle İSÜA' nın konumu GY-NEO6MV2 GPS modülünden ve anlık rota açısı QMC5883 GY-271 3 eksen dijital

pusula sensöründen belirlenir. Konum ve pusuladan okunan rota açısı verileri arduino nano mikroişlemciye gönderilir. İkinci olarak İSÜA' nın, önceden kullanıcı tarafından belirlenen referans yolu tutturması için gerekli rota açısı arduino nano içine gömülmüş olan güdüm yasası tarafından belirlenir. Yakalanması istenilen bu rota açısından, İSÜA' nın mevcut rota açısının çıkarılması ile elde edilen hata (Şekil 5.2' de "e" harfi ile gösterilen hata) arduino nano mikroişlemci içine gömülen bir denetleyici tarafından okunur ve değerlendirilir. QMC5883 GY-271 3 eksen dijital pusula sensöründen okunan İSÜA' nın mevcut rota açısı, hedef rota açısından iskele yönünde sapmış ise, denetleyici tarafından PWM 5 pinine sinyal gönderilerek iskele motor sürücüsüne komut gönderilir ve teknenin sancak yönünde manevra yapması sağlanır. Benzer şekilde, QMC5883 GY-271 3 eksen dijital pusula sensöründen okunan İSÜA' nın mevcut rota açısı, hedef rota açısından sancak yönünde sapmış ise, denetleyici tarafından PWM 6 pinine sinyal gönderilerek sancak motor sürücüsüne komut verilir ve teknenin iskele yönünde dönmesi sağlanır. Eğer İSÜA' nın QMC5883 GY-271 3 eksen dijital pusula sensöründen okunan mevcut rota açısı ile güdüm yasasından gelen hedef rota açısı arasındaki sapma 10 dereceden küçük ise bu durum, İSÜA' nın neredeyse hedef rota açısını yakaladığı anlamına gelir. Sapma açısının 10 dereceden küçük olduğunda hem iskele hem de sancak fırçasız motorlar çalıştırılarak teknenin tam yol ileri gitmesi sağlanır. Sapma açısının 180 dereceye yakın olduğu durumlarda, İSÜA' nın mevcut rota açısını en kısa sürede hangi yöndeki fırçasız motor hedef rota açısına getirebilecekse o yöndeki fırçasız motor sürücüsüne komut verilir.

Şekil 5.1 ve Şekil 5.2' de görüldüğü gibi, İSÜA' nın otonom kontrolünde navigasyon ve güdüm kadar denetleyici de önem taşımaktadır. Denetleyici, güdüm yasasından aldığı komutları ve sensörlerden elde ettiği verileri değerlendirip fırçasız motorları doğru zamanda ve gerekli devir sayısında çalıştırarak teknenin istenilen hedef rotayı kararlı bir şekilde yakalamasını sağlar. PID denetleyici, otonom sistemlerde yaygın olarak kullanılan etkili bir denetleyicidir. Bu çalışmada, klasik PID denetleyicilerin sebep olduğu sayısal hataların, yüksek mertebeden türev ve integral alma teknikleri kullanılarak azaltılması hedeflenmektedir.

5.1. Klasik PID ile İSÜA' nın Denetimi

PID denetleyici, yazılım ile donanım arasında köprü görevini üstlenen geri beslemeli bir kontrol yöntemidir. Şekil 5.3, PID denetleyicinin iç yapısını ve diğer sistemler ile olan bağlantısını göstermektedir. Şekil 5.3' te yer alan PID denetleyici, güdüm yasasından gelen hedef rota açısı ile İSÜA' nın pusuladan okunan rota açısı arasında oluşan sapma miktarını hata (Şekil 5.3' te $e(t)$ ile gösterilen değer) olarak kabul. Bu hata, zamanın bir fonksiyonudur.



Şekil 5.3: PID denetleyicinin iç yapısı.

PID denetleyici, bu hatayı Şekil 5.3' te gösterildiği gibi bir katsayı ile çarparak (P: Proportional), hatanın integralini alarak (I: Integral) ve hatanın türevini alarak (D:

Derivative) yeni bir sinyalin üretilmesini sağlar. Oluşturulan bu sinyal, arduino nano mikroişlemcide değerlendirildikten sonra ilgili PWM Pin 5 veya PWM Pin 6 vasıtasıyla fırçasız motor sürücülere ulaştırılır. Fırçasız motor sürücülere de sancak veya iskele fırçasız motorlara komut vererek İSÜA' nın ilerlemesini yada manevra yapmasını sağlarlar. İSÜA' nın hareketinden sonra, pusuladan okunan rota açısının her zaman adımında PID denetleyiciye geri besleme olarak iletilmesi, kapalı döngü özelliğine sahip bir denetleyicinin elde edilmesini sağlar. Böylece, hızlı ve etkili bir şekilde İSÜA' nın istenilen rotaya ulaşması sağlanabilmektedir.

İSÜA' nın denetiminin yapılmasını sağlayan PID kontrolcünün açık kaynak kodu [85] EK F' ye dahil edilmiştir. Denklem 5.1 literatürde yer alan ve çok sık kullanılan PID denetleyicinin matematiksel denklemini göstermektedir [86].

$$u(t + 1) = K_p \times e(t) + K_i \times \int_t^{t+1} e(t)dt + K_d \times \frac{de}{dt} \quad (5.1)$$

Denklem 5.1' de yer alan $u(t+1)$ değeri, bir sonraki zaman adımı için yani $t+1$ zamanında, PID denetleyici tarafından üretilen ve denetlenen sistemde aktüatör görevini üstlenen fırçasız motorlara dolaylı olarak gönderilen sinyaldir. $u(t+1)$ sinyali, bir önceki zaman adımı olan t zamanında elde edilen hataya çeşitli matematiksel işlemler uygulanarak bulunur. $u(t+1)$ sinyali, hatanın sayısal değerinin K_p ile çarpılarak, hatanın iki zaman adımı aralığında (t ve $t+1$) integral değerinin K_i ile çarpılarak ve hatanın türevinin K_d değeri ile çarpılarak toplanması ile belirlenir. Zamanın bir fonksiyonu olan hata, $e(t)$, sürekli ve devamlı bir fonksiyon olsaydı hatanın türevinin ve integralinin hesaplanması analitik olarak hesaplanması mümkün olabilirdi. Ancak, mikroişlemci ile otonom kontrolü sağlanan bu gibi gerçek fiziksel sistemlerde ayırık zaman üzerine denetim sağlandığından, sayısal türev ve sayısal integral alınması gerekmektedir.

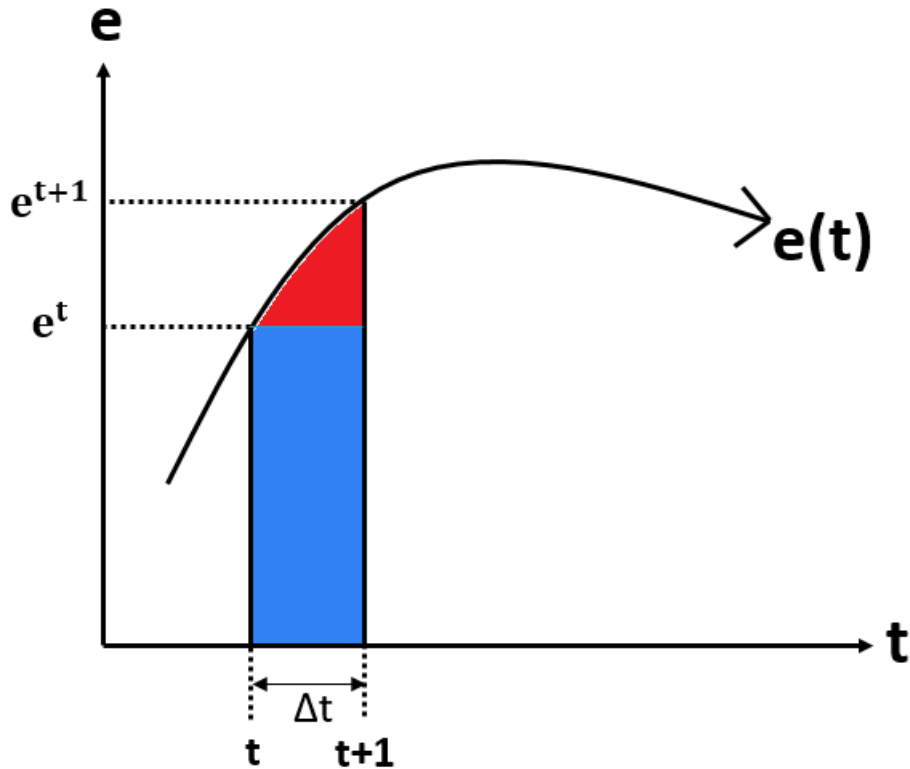
Bir sonraki zaman adımı için üretilecek olan $u(t+1)$ sinyali için bir önceki zaman adımından elde edilen hata değeri, $e(t)$, kullanıldığından sayısal hatalar meydana gelmektedir. Bu sayısal hatalar PID denetleyici için, sayısal integral ve sayısal türev alırken meydana gelen hatalardan kaynaklanmaktadır. Sayısal hataların büyük olması, aktüatörlere (bu çalışma için sancak ve iskele fırçasız motorlar) gönderilecek olan

$u(t+1)$ sinyallerin yanlış ve farklı olmasına dolayısıyla yanlış bir denetimin yapılmasına sebep olabilir.

Klasik PID denetleyicileri, birinci mertebeden sayısal türev ve sayısal integral alma tekniklerini kullanmaktadır. Bu geleneksel yöntemler büyük sayısal hatalara neden olabilmektedir. Aşağıdaki Denklem 5.2, birinci mertebeden sayısal yöntemlerle denetim sağlayan klasik PID kontrolcünün matematiksel ifadesini göstermektedir.

$$u^{t+1} = K_p \times e^t + K_i \times (e^t \times \Delta t) + K_d \times \frac{e^t - e^{t-1}}{\Delta t} \quad (5.2)$$

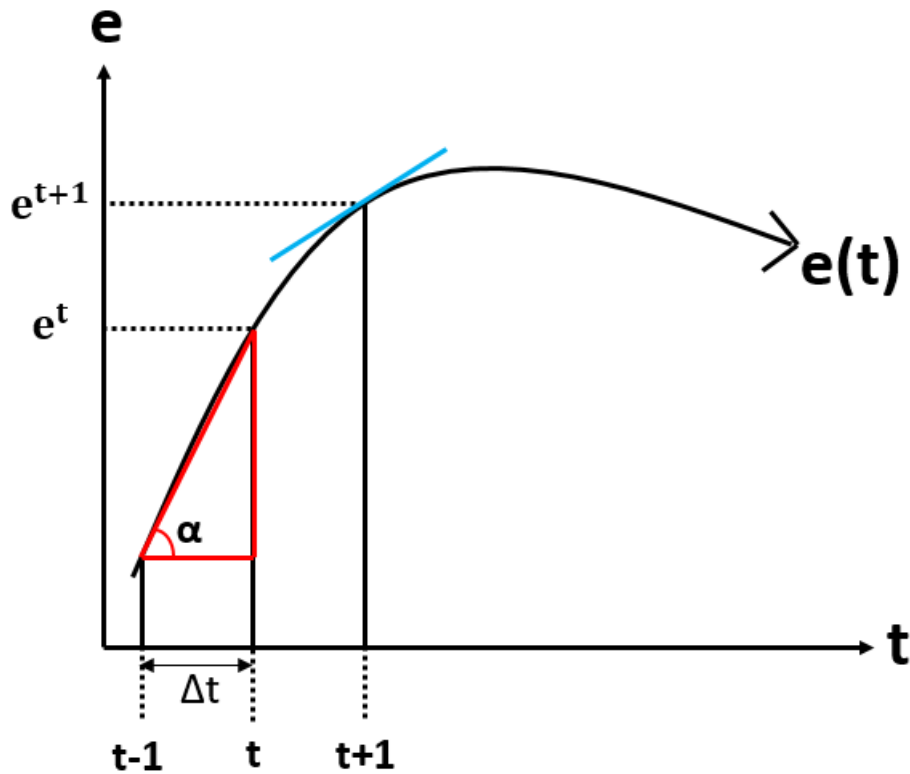
Denklem 5.2' de yer alan u^{t+1} , ayrık zamanda hesaplamalar yapıldığı için artık fonksiyon değil, sayısal bir değerdir. u^{t+1} , bir sonraki zaman adımında fırçasız motor sürücüler için gönderilecek sayısal bir değeri ifade eder. Denklem 5.2' de yer alan e^t , bir önceki zaman diliminde hesaplanmış hata değeridir. Δt ise iki zaman aralığındaki zaman adımı büyüklüğüdür. e^t değerinin Δt ile çarpılması birinci mertebeden sayısal yöntem kullanılarak hatanın integralinin bulunmasını ifade eder.



Şekil 5.4: Birinci mertebeden sayısal integral alma tekniği.

Şekil 5.4, birinci mertebeden sayısal yöntem kullanılarak hatanın integralinin alınmasını grafiksel olarak göstermektedir. Şekil 5.4’ te yer alan $e(t)$ hatanın fonksiyonunu ifade eder. e^t ve e^{t+1} değerleri, sırasıyla hatanın t ve $t+1$ zaman dilimlerindeki karşılığıdır. Grafikte yer alan mavi alan, birinci mertebeden sayısal yöntem kullanılarak hatanın integralini ifade eder. Ancak, hatanın fonksiyonu sürekli ve devamlı bir fonksiyon olduğu varsayılırsa, hatanın bu iki zaman (t ve $t+1$) aralığındaki integrali, eğrinin altında kalan alana yani kırmızı ve mavi alanların toplamına eşit olması gerekmektedir. Sonuç olarak, birinci mertebeden sayısal integral alma yöntemi Şekil 5.4’ te gösterilen kırmızı alan kadar büyüklükte bir sayısal hataya sebep olmaktadır.

Denklem 5.2’ de yer alan $(e^t - e^{t-1})/\Delta t$ işlemi, hatanın, $e(t)$, birinci mertebeden sayısal türevidir. Ancak, bir sonraki zaman adımına ait sinyalin, u^{t+1} , hesaplanması için bir önceki ve iki önceki zaman adımlarına ait hata değerlerinin (e^t ve e^{t-1}) kullanılması ile sayısal türevin hesaplanması, sayısal hatalara sebep olmaktadır.



Şekil 5.5: Birinci mertebeden sayısal türev alma tekniği.

Şekil 5.5, birinci mertebeden sayısal yöntem kullanılarak hatanın türevinin alınmasını grafiksel olarak göstermektedir. Şekil 5.5’ te yer alan kırmızı üçgene ait α açısının

tanjantı, birinci mertebeden sayısal türev alma tekniği ile eğimin bulunmasını ifade eder. Eğer hatanın fonksiyonu, $e(t)$, sürekli ve devamlı bir fonksiyon olsaydı, bir sonraki zaman adımına ait sinyalin, u^{t+1} , hesaplanması için kullanılması gereken eğim, Şekil 5.5’ te yer alan mavi doğrunun eğimi (e^{t+1} noktasında fonksiyonun eğimi) olmalıydı. Ancak, kırmızı üçgene ait α açısının tanjantı ile elde edilen eğim ve mavi doğrunun eğimi birbirinden farklıdır. Bu durum sayısal hata olarak ifade edilir. Fiziksel sistemlerde, ayrık zamanda hesaplamalar yapılma zorunluluğu, hatanın fonksiyonda, $e(t)$, sadece belirli noktaların değerlerinin bilinmesine (e^t , e^{t-1} ve e^{t-2} gibi) sebep olmaktadır. Birinci mertebeden sayısal türev alma tekniği de, bu değerlerden sadece iki değeri (e^t ve e^{t-1}) kullanarak eğimi belirlediğinden, bu yöntem büyük sayısal hatalara sebep olabilmektedir.

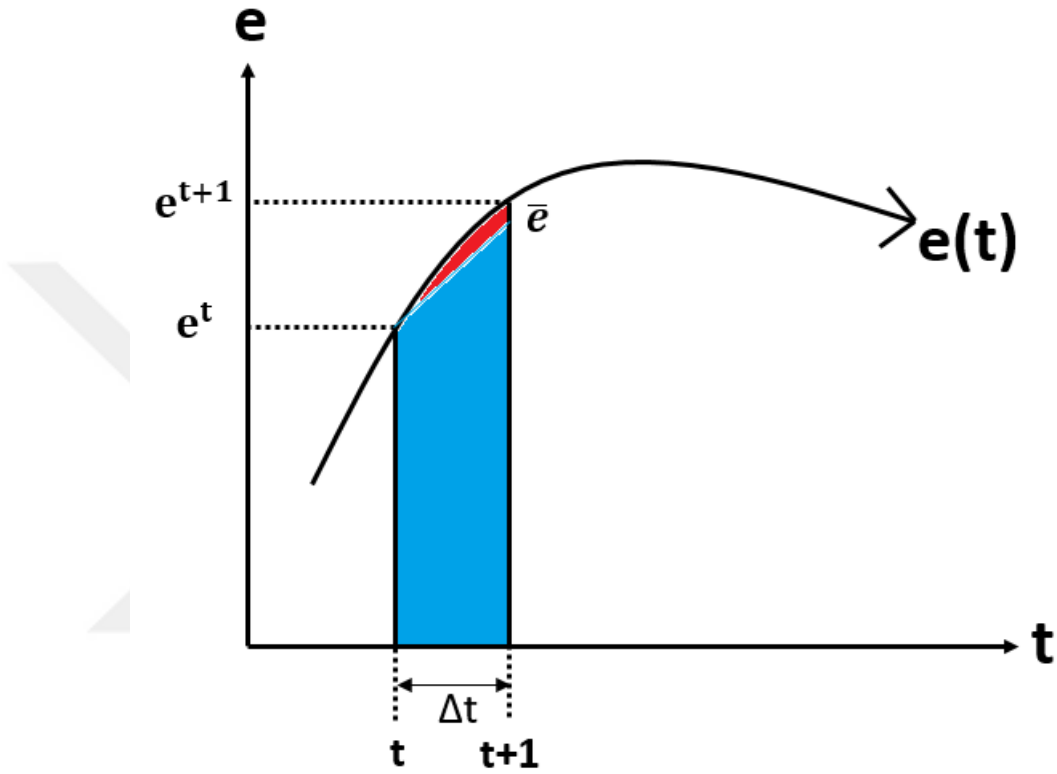
5.2. Yüksek Mertebeden Sayısal İntegral ve Türev Alma Yöntemleri Kullanarak İSÜA’ nın Denetimi

Bu çalışma, Denklem 5.2’ de yer alan PID denetimin, yüksek mertebeden türev ve yüksek mertebeden integral alınarak zenginleştirilmesini önermektedir. Sayısal türev alma işlemi için üçüncü mertebeden türev alma tekniği [87,88] tercih edilirken sayısal integral alma işlemi için ise ikinci mertebeden integral alma yöntemi [89] seçilmiştir. Bu durumda, düşük sayısal hatalara sebep olan PID denetleyici Denklem 5.3 ile ifade edilebilir.

$$\begin{aligned}
 u^{t+1} = & K_p \times e^{t+1} + K_i \times \frac{(e^{t+1} + e^t) \times \Delta t}{2} \\
 & + K_d \frac{\frac{e^{t+1} + e^t}{2} - \frac{e^{t+1} - 2e^t + e^{t-1}}{8}}{\Delta t} \\
 & - K_d \frac{\frac{e^t + e^{t-1}}{2} - \frac{e^t - 2e^{t-1} + e^{t-2}}{8}}{\Delta t}
 \end{aligned} \tag{5.3}$$

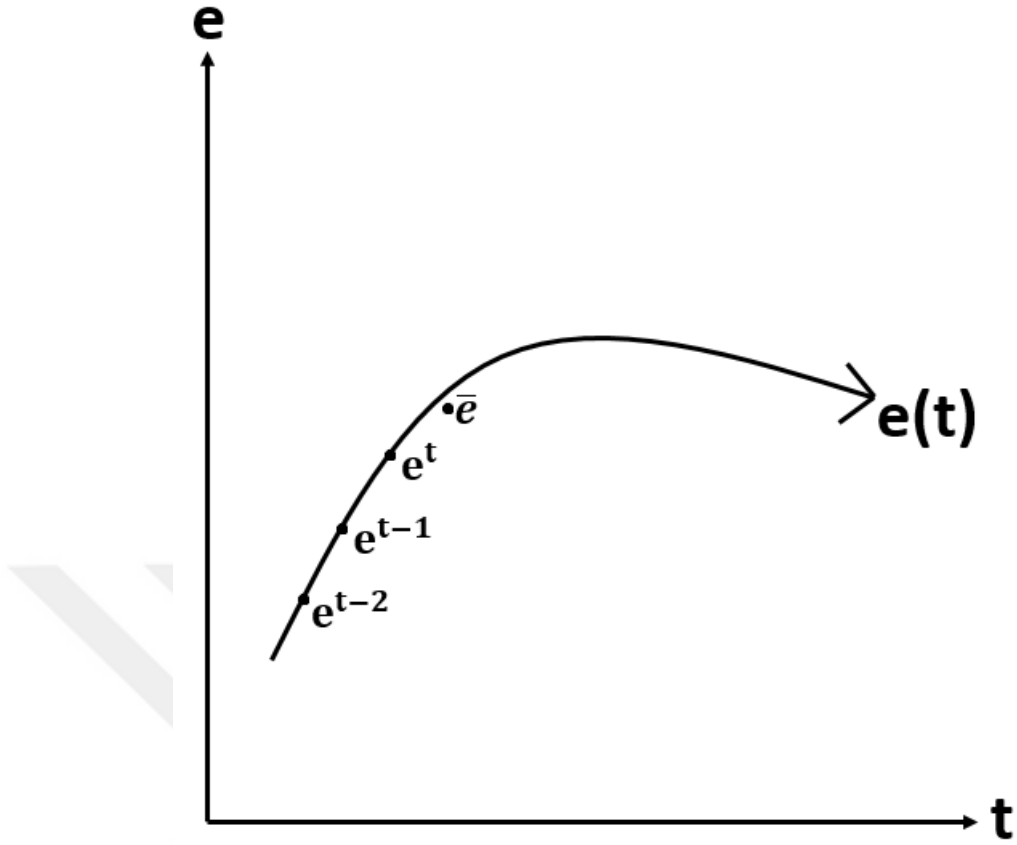
Denklem 5.3’ te yer alan u^{t+1} bir sonraki zaman adımı için fırçasız motor sürücülerine gönderilecek olan sinyaldir. Önceki zaman adımlarından elde edilen hata verileri (e^t , e^{t-1} ve e^{t-2}) kullanılarak ve bir sonraki zaman adımında oluşacak olan hata değeri (e^{t+1}) doğrusal interpolasyon gibi yöntemlerle tahmin edilerek u^{t+1} sinyali hesaplanabilir.

Hesaplanan bu sinyal ile fırçasız motorların bir sonraki zaman adımında ne kadar devirde çalışacağına karar verilir. Denklem 5.3' te bulunan e^t , e^{t-1} ve e^{t-2} sırasıyla mevcut zaman adımındaki hata, bir önceki zaman adımındaki hata ve iki önceki zaman adımındaki hatadır. Sayısal türev ve sayısal integral alınırken çoklu zaman adımlarının göz önünde bulundurulması sayısal hataları azaltmaktadır.



Şekil 5.6: Yüksek mertebeden sayısal integral alma tekniği.

Şekil 5.6, yüksek mertebeden sayısal integral alma yöntemini grafiksel olarak göstermektedir. Grafikte gösterilen mavi alan yüksek mertebeden integral alma tekniğini ifade eder ve denklem 5.3' te yer alan $(e^{t+1}+e^t)*\Delta t/2$ işlemi ile bulunan yamuğun alanını az bir hata payı ile yakalar. Grafikte yer alan $\bar{e}$ değeri, önceki zaman adımları kullanılarak $t+1$ zaman adımı için tahmin edilen hata değeridir. Şekil 5.4' te yer alan mavi dikdörtgen alan yerine, Şekil 5.6' da bulunan mavi yamuk alanın kullanılması daha düşük sayısal hataları netice verebileceğinden yüksek mertebeden sayısal yöntemlerin kullanılması avantaj sağlamaktadır. Şekil 5.6' da yer alan kırmızı alan ise yüksek mertebeden sayısal integral alma yönteminin netice olduğu sayısal hatayı ifade eder. Bu alan, Şekil 5.4' te bulunan kırmızı alandan daha az bir alana karşılık gelmektedir.



Şekil 5.7: Yüksek mertebeden sayısal türev alma yönteminde kullanılan değerler.

Denklem 5.3' te eşitliğin sol tarafında yer alan son iki işlem, yüksek mertebeden sayısal türev almayı ifade eder. Sayısal türev alma işleminde kullanılan değerler Şekil 5.7' de gösterilmiştir. e^t , e^{t-1} ve e^{t-2} değerleri önceki zaman adımlarında hesaplanan hata değerleri olup, $\bar{e}$ değeri ise mevcut verilerden çıkarım yapılarak sonraki zaman adımında meydana gelebilecek hatanın tahmin edilmesini ifade eder. Yüksek mertebeden sayısal türev alma yönteminde, birinci mertebeden türev alma tekniğinde kullanılan sadece iki değer yerine çoklu değerlerin (dört tane veri) kullanılması, hatanın eğrisinin daha iyi modellenmesini sağlamaktadır. Bu sayede bir sonraki zaman adımında eğrinin netice verdiği eğim daha iyi tahmin edilebileceğinden sayısal hatalar azaltılabilmektedir.

Denklem 5.2 ve Denklem 5.3' te yer alan K_p , K_i ve K_d değerleri sırasıyla hatanın oransal katı, integrali ve türevi için kullanıcı tarafından belirlenen katsayılardır. K_p değerinin baskın olması durumunda, İSÜA' nın istenilen rota açısı çok hızlı bir şekilde tutturulur. Yani aktüator görevini üstelenen fırçasız motorların devri motor sürücülere

gelen yüksek sinyal sayesinde aniden çok yükselir ve referans rota açısına ulaşılmış olur. Ancak bu ani yükseliş kararlı bir şekilde kontrol edilemediğinden referans açı ters yönde tekrar kaçırılır ve osilasyonlu bir davranış sergilenir. Bu osilasyonlu davranış K_d ile yani hatanın türevi alınarak bastırılır. Hatanın türevinin büyük olduğu durumda güçlü sinyaller ile hızlı cevaplar verilirken hatanın türevinin küçüldüğü yani referans değere yaklaşıldığı zaman kademeli olarak sinyaller küçültülür. Bu da istikrarlı bir denetimin yapılmasını sağlar. Referans değerın tutturulmasında kalıcı hataların ortadan kaldırılması için de K_i yani hatanın integrali kullanılır. Bu sayede QMC5883 GY-271 3 eksen dijital pusula sensöründen gelen İSÜA' nın mevcut rota açısı, güdüm yasasından gelen referans açıyla tam örtüşmüş olur.



BÖLÜM 6. BULGULAR VE TARTIŞMA

Bu bölümde, sayısal benzetimi ve tasarımı yapılan İSÜA' nın teorik ve deneysel bulguları incelenmiştir. Elde edilen sonuçlar literatürde yer alan mevcut çalışmalar ile karşılaştırılıp geliştirilen modelin avantajlarına vurgu yapılmıştır.

6.1. Sayısal Benzetimden Elde Edilen Bulgular

Bu çalışma kapsamında incelenen ve literatürde yer alan havuç kovalama güdüm yasası, doğrusal olmayan güdüm yasası [90] ve vektör alanı güdüm yasası isimli her bir yol izleme algoritmasına ayrı ayrı genetic algoritma, desen arama, benzetimli tavlama ve vektör model optimizasyon yöntemleri uygulanmıştır. Ayrıca, bu çalışmada geliştirilen muhakeme esaslı güdüm yasası isimli uyarlamalı zaman adımlarına sahip algoritmaya da her bir optimizasyon yöntemi ayrı ayrı uygulanmıştır. Elde edilen tüm sonuçlar mukayese edildiğinde geliştirilen muhakeme esaslı güdüm yasasının, sayısal benzetimin tamamlanması için toplam geçen süre ve yol izleme hataları bakımından önemli avantajlara sahip olduğu görülmüştür.

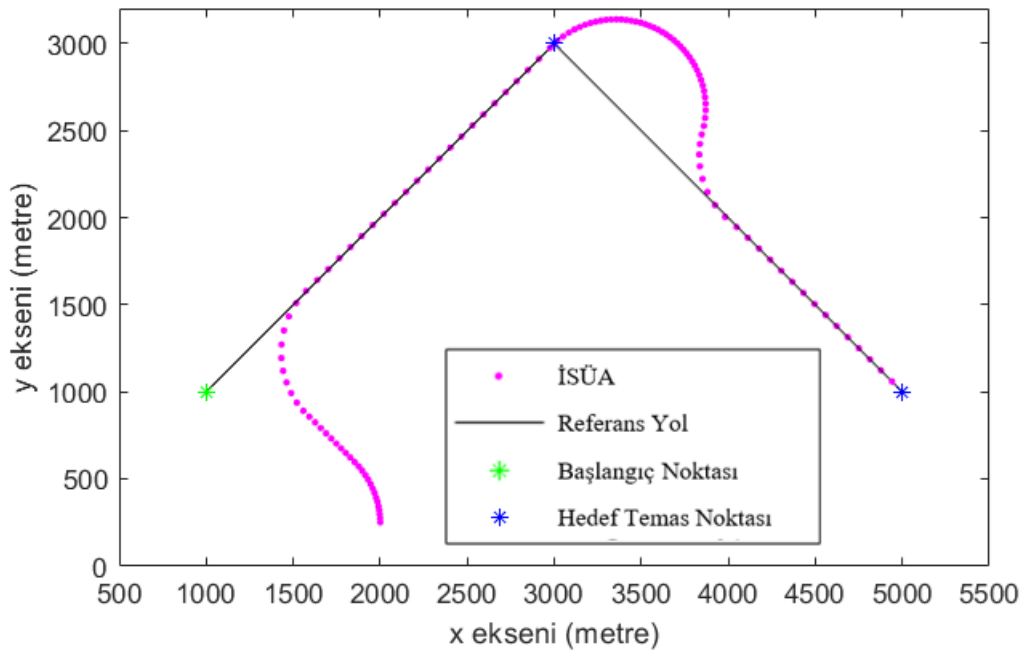
Tablo 6.1, uyarlamalı zaman adımlarına temellendirilmiş muhakeme esaslı güdüm yasasının diğer güdüm yasaları ve optimizasyon yöntemleri ile karşılaştırılmasını göstermektedir. Tablo 6.1' de yer alan sonuçlar elde edilirken İSÜA' nın maksimum dönüş hızının 1 derece/saniye, hızının 9 metre/saniye, başlangıç rota açısının 90 derece ve başlangıç konumunun sırasıyla x ve y eksenlerinde 2000 ve 250 metre olduğu varsayılmıştır. Ayrıca, sayısal benzetimlerde kullanılan zaman adımı büyüklüğünün 2.5 saniye olduğu ve uyarlamalı zaman adımı kullanan algoritmanın zaman adımı için maksimum bu değerin 4 katına çıkabileceği varsayılmıştır. İSÜA bu şartlar altında hedef temas noktaları sırasıyla [1000,1000], [3000,3000] ve [5000,1000] olan iki görevi gerçekleştirmektedir. Tablo 6.1' de yer alan MEGY (u), MEGY (s), HKGY, DOGY ve VAGY kısaltmaları sırasıyla uyarlamalı zaman adımları kullanan muhakeme esaslı

güdümlü yasa, sabit zaman adımları kullanan muhakeme esaslı güdümlü yasa, havuç kovalama güdümlü yasa, doğrusal olmayan güdümlü yasa ve vektör alanı güdümlü yasa ifade etmektedir.

Tablo 6.1. Uyarlamalı zaman adımlarını kullanan MEGY’ nın literatürdeki diğer güdümlü yasaları ve optimizasyon yöntemleri ile karşılaştırılması.

Optimizasyon Yöntemleri	Güdümlü Yasaları	Toplam Yol İzleme Hatası (m ²)	İyileşme (%)	Toplam Geçen Süre (saniye)
Genetik Algoritma	MEGY (u)	629750,0	-	78,5
	MEGY (s)	665660,0	5,70	220,1
	HKGY	753790,0	19,70	218,6
	DOGY	701880,0	11,45	187,3
	VAGY	695530,0	10,45	242,7
Desen Arama	MEGY (u)	626870,0	-	40,2
	MEGY (s)	635650,0	1,40	125,5
	HKGY	701670,0	11,93	139,8
	DOGY	701690,0	11,94	140,9
	VAGY	669200,0	6,75	162,4
Benzetimli Tavlama	MEGY (u)	654290,0	-	59,8
	MEGY (s)	675680,0	3,27	143,9
	HKGY	702140,0	7,31	160,6
	DOGY	702010,0	7,29	158,3
	VAGY	678410,0	3,69	185,6
Vekil Model	MEGY (u)	625730,0	-	45,8
	MEGY (s)	638020,0	1,96	120,9
	HKGY	702220,0	12,22	127,3
	DOGY	701780,0	12,15	121,8
	VAGY	643590,0	2,85	130,3

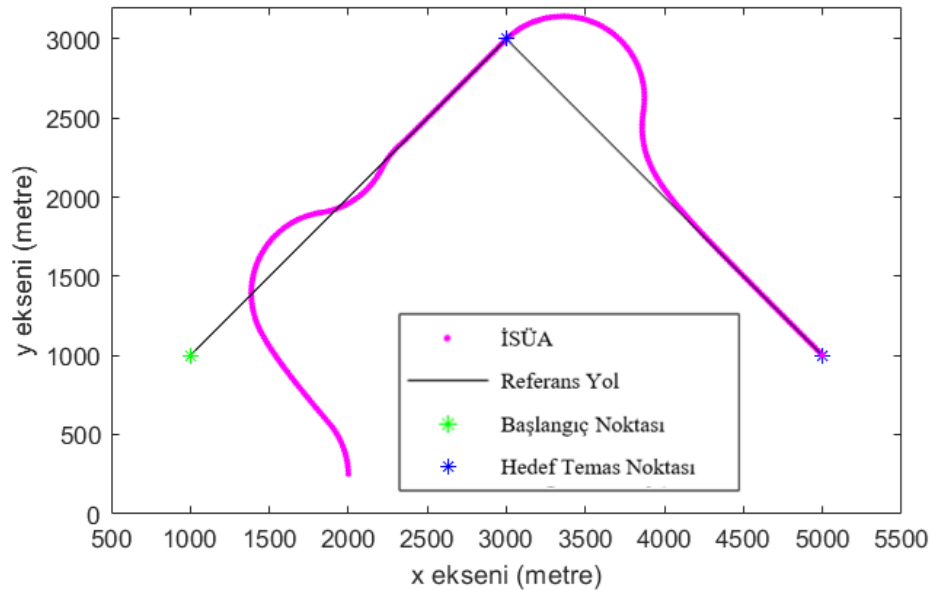
Muhtelif optimizasyon yöntemlerinin uyarlamalı zaman adımlarını kullanan muhakeme esaslı güdüm yasasına uygulanması incelendiğinde yol izleme hataları bakımından en iyi sonuç veren optimizasyon yönteminin vekil model optimizasyon yöntemi (Tablo 6.1’ de koyu renkle vurgulanan yöntem) olduğu görülmüştür ($625730,0 \text{ m}^2$). Sonuçlar, sayısal benzetimin toplam geçen süresi bakımından incelendiğinde ise en kısa sonuca ulaşan optimizasyon yönteminin desen arama optimizasyon yöntemi (Tablo 6.1’ de italik olarak gösterilen yöntem) olduğu görülmüştür (40,2). Desen arama optimizasyon yöntemi, vekil model optimizasyon yöntemine göre yaklaşık 5 saniye daha hızlı sonuç verirken yaklaşık 1000 m^2 daha fazla yol izleme hatasına sebep olmaktadır. Uyarlamalı zaman adımlarını kullanan muhakeme esaslı güdüm yasası ile sabit zaman adımlarını kullanan muhakeme esaslı güdüm yasası kıyaslandığında uyarlamalı zaman adımlarına dayalı algoritma her bir optimizasyon yönteminde daha az yol izleme hatasını netice vermiştir. Ayrıca, Tablo 6.1’ de gösterildiği gibi, uyarlamalı zaman adımlarını kullanan algoritma her bir optimizasyon yöntemi için yaklaşık 3 kat daha hızlı sayısal benzetimin tamamlanmasına neden olup optimizasyon yönteminin daha çabuk neticeye ulaşmasını sağlamıştır. Aşağıdaki Şekil 6.1, uyarlamalı zaman adımlarını kullanan muhakeme esaslı güdüm yasasının vekil model optimizasyon yöntemi kullanılarak elde edilen sayısal benzetimi göstermektedir.



Şekil 6.1: Vekil model optimizasyon yönteminin uyarlamalı zaman adımlarını kullanan muhakeme esaslı güdüm yasasına uygulanması.

Şekil 6.1’ de görüldüğü gibi geliştirilen algoritma, referans yoldan uzak olduğunda ve referans yolun eğimini netice veren açı ile İSÜA’ nın rota açısı arasındaki sapma miktarı büyük olduğunda İSÜA’ nın hareketini daha iyi modelleyebilmek için zaman adımı boyutlarını küçültülmektedir. İSÜA kritik dönüşleri tamamlamaya başladığında ve referans yolu yakalamaya başladığında zaman adımı boyutları yavaş yavaş artırılarak hem toplam geçen süre kısaltılmakta hem de her bir zaman adımından gelen sayısal hatalar azaltılmaktadır. Bu sayede güdüm yasasının sahip olduğu parametreler de daha kısa sürede optimize edilebilmektedir.

Tablo 6.1’ de yer alan havuç kovalama güdüm yasası ve doğrusal olmayan güdüm yasası, İSÜA’ nın rota açısının belirlenmesi için tek bir parametre kullandıklarından dolayı iki adet parametre kullanan muhakeme esaslı güdüm yasası ve dört adet parametre kullanan vektör alanı güdüm yasasına göre her bir optimizasyon yönteminde daha yüksek yol izleme hatası vermişlerdir. Bu iki güdüm yasası her bir optimizasyon yönteminde yaklaşık olarak benzer sonuçlara sahip olmakla birlikte sadece genetik algoritmanın havuç kovalama güdüm yasasına uyarlanmasında optimizasyon yöntemi yerel minimum noktaya takılmış ve küresel minimum noktayı netice veren optimum parametreyi belirleyememiştir. Bu sebeple, diğer sonuçlara göre yaklaşık %7 daha fazla yol izleme hatası gözlemlenmiştir. Şekil 6.2, genetik algoritmanın havuç kovalama güdüm yasasına uyarlanması ile elde edilen sayısal benzetimi göstermektedir.



Şekil 6.2: Genetik algoritmanın havuç kovalama güdüm yasasına uyarlanması neticesinde optimizasyon yönteminin yerel minimum noktaya takılması.

Şekil 6.2’ de görüldüğü gibi, İSÜA referans yolu kısa sürede yakalayabilmek için keskin manevra yaparak rotası referans yola neredeyse dik olacak şekilde ilerlemektedir. Ancak, referans yola ulaştıktan sonra karşı cepheye geçerek osilasyonlu bir seyir göstermektedir. İkinci referans yol için ise İSÜA, referans yolu daha geç yakalamaktadır. Her iki durumda da İSÜA’ nın bu davranışı yol izleme hatalarını arttırmaktadır. Navigasyonun bu şekilde olmasının sebebi, genetik algoritma optimizasyon yönteminin yerel minimum noktaya takılmış olmasıdır. Diğer optimizasyon yöntemlerinin havuç kovalama güdüm yasasına uyarlanmasında bu sorun yaşanmazken genetik algoritmanın yerel minimum noktaya takılması, genetik algoritmanın küresel minimum noktaları bulmaktaki dezavantajını göstermektedir. Bu sorunun aşılabilmesi için her bir optimizasyon yönteminde aynı ve sabit tutulan hedef fonksiyonun maksimum değerlendirme sayısı artırılmalıdır. Ancak, bu durum optimizasyon süresinin daha çok uzamasına ve toplam geçen sürenin artmasına neden olacaktır.

Tablo 6.1’ de yer alan vektör alanı güdüm yasası incelendiğinde, bu güdüm yasası çoklu parametreye sahip olduğundan yani optimize edilmesi gereken 4 adet parametre içerdiğinden farklı optimizasyon teknikleri farklı sonuçlar vermiştir. Vekil model optimizasyon yöntemi, çoklu parametre optimizasyonu için tasarlandığından, vekil model optimizasyon yönteminin vektör alanı güdüm yasasına uyarlanması en iyi neticeyi vermiştir. Bu yöntem 643590,0 metre yol izleme hatasını netice vererek uyralamalı zaman adımları kullanan muhakeme esaslı güdüm yasasına yaklaşık %3’ lük bir farkla yaklaşmıştır. Ancak, çoklu parametrenin kullanılması optimizasyon teknikleri için tek parametrelili hedef fonksiyonlarına göre daha yüksek hesaplama maliyetleri gerektirdiğinden ve parametrelerin çokluğu hedef fonksiyonun küresel minimum noktasının bulunmasını daha da zorlaştırdığından diğer optimizasyon yöntemlerine vektör alanı güdüm yasasının entegre edilmesi daha yüksek yol izleme hatalarına sebep olmuştur. Örneğin, genetik algoritmanın vektör alanı güdüm yasasına entegre edilmesi durumunda optimizasyon yöntemi yerel minimum noktaya takılmış ve vekil model optimizasyon yöntemine göre yaklaşık % 8 daha fazla yol izleme hatasını netice vermiştir. Ayrıca, genetik algoritmada toplam geçen süre, vekil model optimizasyon yönteminde geçen sürenin neredeyse iki katıdır.

Tablo 6.1’ de yer alan her bir gdm yasaı ve her bir optimizasyon yntemi incelendiğinde en az yol izleme hatasını, vekil model optimizasyon ynteminin uyarlamalı zaman adımlarını kullanan muhakeme esaslı gdm yasaına entegre edilmesiyle oluřturulan algoritma netice vermektedir.

6.2. Deneysel alıřmadan Elde Edilen Bulgular

Bu alıřma kapsamında tasarımı yapılan prototip bir İSA’ nın test srř ařağıdaki řekil 6.3’ te yer alan st aık yaklaşık 90 metre uzunluğunda bir havuzda gerekleřtirilmiřtir. İSA otonom seyrini, ierisine yerleřtirilen arduino nano programlama kartı sayesinde gerekleřtirmiřtir. İSA otonom seyrine bařlamadan nce programlama kartının iine girilen  adet noktasal konum, İSA’ nın ulařması istenilen hedef noktalarıdır.



řekil 6.3: Tasarlanan İSA’ nın test srřnn yapıldığı havuz.

İSÜA bu hedef noktalara GY-NEO6MV2 GPS modülünden aldığı konum verisi ve QMC5883 GY-271 3 eksen dijital pusuladan elde ettiği anlık rota açısı sayesinde ulaşmaktadır. İSÜA, birinci hedef temas noktasına yaklaşıp arasındaki mesafe 3 metreden kısa olduğu zaman yeni hedef noktası ikinci hedef temas noktası olmaktadır. Benzer şekilde ikinci hedef temas noktasına olan mesafesi 3 metreden kısa olduğunda üçüncü hedef temas noktası yeni hedef olmakta ve İSÜA üçüncü hedefe doğru yönelmektedir. Son olarak da üçüncü hedef temas noktasına yaklaştığında konumunu sürekli üçüncü hedef temas noktasına getirmeye zorlamakta ve dalga, rüzgar gibi bozucu kuvvetlere karşı maksimum mesafe üçüncü noktadan üç metre olacak şekilde konumunu korumaya çalışmaktadır.

Test sürüşünden elde edilen deneysel sonuçları Tablo 6.2’ de gösterilmektedir. Tablo 6.2’ de yer alan veriler, İSÜA otonom seyrini yaparken NRF24L01 2.4 GHz kablosuz haberleşme modülü sayesinde anlık olarak harici bir bilgisayara İSÜA’ dan gönderilerek elde edilmiştir.

Tablo 6.2. Tasarlanan İSÜA’ nın test sürüşü verilerinden bir kısmı.

Enlem	Boylam	Rota Açısı (Pusula)	Rota Açısı (GPS)	Hız (m/s)	Zaman	Hedef Nokta
...	...	...	...	...	...	...
41,0718	30,93116	213	131,75	0,44	8370900	0
41,0718	30,93117	132	122,87	0,34	8371000	0
41,0718	30,93117	117	75,53	0,35	8371100	0
41,0718	30,93117	105	75,53	0,13	8371200	0
41,0718	30,93117	152	157,35	0,5	8371300	0
41,07181	30,93116	24	2,88	0,62	8371600	0
41,0718	30,93116	107	130,59	0,61	8371700	0
41,0718	30,93116	194	165,5	0,69	8371800	0
41,0718	30,93116	221	207,68	0,54	8372000	0
41,0718	30,93116	108	78,84	0,95	8372100	0
...	...	...	...	...	...	...

Tablo 6.2 (devam): Tasarlanan İSÜA' nın test sürüşü verilerinden bir kısmı.

Enlem	Boylam	Rota Açısı (Pusula)	Rota Açısı (GPS)	Hız (m/s)	Zaman	Hedef Nokta
41,0719	30,93127	93	63,17	0,85	8375500	1
41,07191	30,93128	44	63,17	0,45	8375700	1
41,07191	30,93127	89	63,17	0,69	8375800	1
41,07191	30,93128	29	63,17	0,37	8375900	1
41,07192	30,93129	47	45,21	0,59	8380100	1
41,07192	30,93131	111	45,21	0,95	8380300	1
41,07192	30,93133	51	64,94	0,67	8380600	1
41,07193	30,93138	150	96,57	0,98	8381300	1
41,07194	30,93139	12	96,57	0,96	8381400	1
41,07194	30,93139	87	109,1	0,76	8381500	1
41,07194	30,9314	83	84,6	0,59	8381600	1
...	...	...	...	...	...	...
41,07199	30,93151	220	221,74	0,6	8384500	2
41,07194	30,93144	206	184,57	0,52	8390300	2
41,07194	30,93144	296	184,57	0,68	8390400	2
41,07194	30,93141	243	204,98	0,5	8390700	2
41,07194	30,93141	173	204,98	0,5	8390800	2
41,07194	30,93141	238	204,98	0,54	8390900	2
41,07193	30,93139	249	254,81	0,57	8391100	2
...	...	...	...	...	...	...

Tablo 6.2' de ilk iki sütunda GY-NEO6MV2 GPS modülünden gelen İSÜA' nın enlem ve boylam olarak anlık konum bilgileri yer almaktadır. Konum bilgileri, GY-NEO6MV2 GPS modülü en az dört tane uydudan veri almaya başladığında elde edilebilmektedir. Bulutlu havalarda modülün uyduları bulması zorlaşmaktadır. Konum verilerinin elde edilebilmesi için biraz beklenmesi gerekmektedir. GY-NEO6MV2 GPS

modülü dört ve dörtten fazla uydu ile haberleşip konum tespiti yaptığında, modül üzerinde kırmızı renkli bir led yanıp sönmektedir. Kırmızı renkli bu ışıktan, GY-NEO6MV2 GPS modülünün yeterli uyduyu bulduğu ve İSÜA' nın konumunu belirlediği anlaşılmaktadır. Bu çalışmada, biraz beklenerek kırmızı ışığın yandığından emin olduktan sonra test sürüşüne başlanmıştır.

Üçüncü sütunda, QMC5883 GY-271 3 eksen dijital pusula sensöründen gelen anlık rota açısı bulunmaktadır. QMC5883 GY-271 3 eksen dijital pusula sensöründen hassas ve doğru verilerin alınması için pusulanın kalibre olması gerekmektedir. QMC5883 GY-271 3 eksen dijital pusula sensörünün kalibrasyonu, tekne yatay ekseninde saat yönünde ve saat yönünün zıttında 360 derece döndürülerek yapılmaktadır. Kalibrasyon yapılırken, teknenin eğiminin değişmemesine dikkat edilmesi gerekmektedir. Benzer şekilde, QMC5883 GY-271 3 eksen dijital pusula sensörünün kalibrasyonu yapıldıktan sonra test sürüşüne geçilmiştir.

Dördüncü sütunda ise GY-NEO6MV2 GPS modülünden gelen rota açısı vardır. İSÜA' otonom seyrini yaparken QMC5883 GY-271 3 eksen dijital pusula sensöründen gelen anlık rota açısı referans olarak alınmıştır. Çünkü, İSÜA haraketsiz iken veya hızı yavaşken GY-NEO6MV2 GPS modülünden gelen rota açısı doğru ve hassas sonuçlar vermemektedir. Örneğin, Tablo 6.2' de üçüncü satır incelendiğinde, QMC5883 GY-271 3 eksen dijital pusula sensöründen gelen rota açısı 117 derece ve GY-NEO6MV2 GPS modülünden gelen rota açısı 75.53 derecedir. Tablo 6.2' de dördüncü satır incelendiğinde, QMC5883 GY-271 3 eksen dijital pusula sensöründen gelen rota açısı 105 derece ve GY-NEO6MV2 GPS modülünden gelen rota açısı ise yine 75.53 derecedir. İSÜA hareketli ve manevra yapıyor olmasına rağmen GY-NEO6MV2 GPS modülünden gelen rota açısının değişmemesi, GPS modülünden gelen verilerin hassas olmadığını ve birer saniyelik her zaman adımı için doğru rota açısını belirleyemediğini göstermektedir. Bu sebeple, İSÜA' nın otonom kontrolünde en önemli verilerden biri olan rota açısının belirlenmesinde QMC5883 GY-271 3 eksen dijital pusula sensörü isimli harici bir modül kullanılmak zorunda kalınmıştır.

Tablo 6.2' de beşinci ve altıncı sütunda yer alan veriler sırasıyla İSÜA' nın hızı ve uydudan gelen zaman bilgisidir. Örneğin, altıncı sütunda ilk sırada yer alan 8371200 rakamı, 08(saat):37(dakika):12(saniye):00(milisaniye) olarak zaman bilgisini ifade

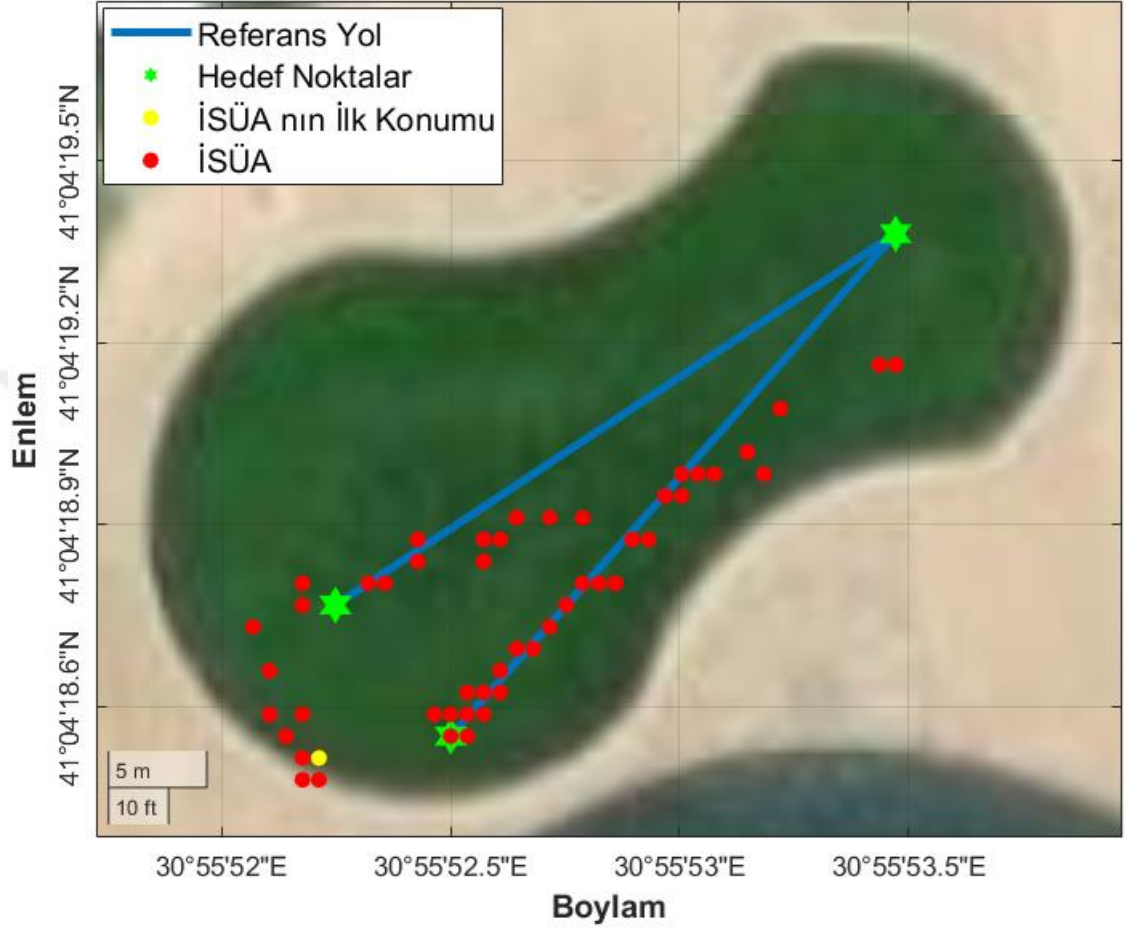
etmektedir. Zaman bilgisi, GY-NEO6MV2 GPS modülünde en hızlı elde edilen veridir. Zaman bilgisinin elde edilmesi için konum verisinde olduğu gibi dört adet uydu ile iletişime geçilme zorunluluğu yoktur. Ayrıca, modül içerisinde harici bir pil ve küçük bir hafıza entegresi olduğundan modülün mikroişlemci ile bağlantısı kesilse bile zaman bilgisi bir müddet daha modül içerisinde hesaplanmakta ve kayıt altında tutulabilmektedir. Böylece, anlık kesintilerde modül daha hızlı bir şekilde tekrar mikroişlemciye verileri aktarmaya devam edebilmektedir.

Son sütundaki rakamlar ise İSÜA kaçınıcı hedef temas noktaya gittiğini göstermektedir (0: birinci hedef, 1: ikinci hedef ve 2: üçüncü hedef). Bu rakamlar, İSÜA' ndan gelen konum verileri tekne hangi görevi yapıyorken elde edildiğini ayırt edebilmek için oluşturulmuştur. İSÜA, birinci hedef temas noktaya giderken, teknedeki kullanıcı bilgisayarına sürekli 0 rakamı gönderilmektedir. İSÜA' nın, birinci hedef temas noktaya olan mesafesi üç metreden az olduğu zaman artık 1 rakamı gönderilmeye başlanmaktadır ve bu durum İSÜA' nın yeni hedefinin ikinci hedef temas noktası olduğunu göstermektedir. Benzer şekilde İSÜA, ikinci hedef temas noktasına yaklaştığında ve mesafe üç metrenin altına düştüğünde artık İSÜA' nın yeni hedefi üçüncü hedef temas noktasıdır. Bu durumda, kullanıcı bilgisayarına 2 rakamı gönderilmeye başlar.

Tablo 6.2' de örnek olması açısından İSÜA' nın otonom seyri sırasında elde edilen sadece bazı verilere yer verilmiştir. Tüm veriler EK F' ye dahil edilmiştir.

Şekil 6.4, İSÜA' nın test sürüşü sırasında NRF24L01 2.4 GHz kablosuz haberleşme modülünden gelen anlık veriler kullanılarak oluşturulmuştur. Şekil 5.3' te yer alan sarı daire ve kırmızı daireler sırasıyla İSÜA' nın ilk konumunu ve İSÜA anlık konumlarını ifade etmektedir. Şekil 6.4' te yer alan yeşil renkli yıldızlar, İSÜA' nın ulaşması gereken hedef temas noktalarıdır. Şekil 6.4' te yer alan mavi renkli doğrular ise İSÜA' nın takip etmesi gereken referans yollardır. İSÜA' nın ilk görevi Şekil 6.4' te sol alt üst kısımda bulunan ilk yıldızla ulaşmaktır. İkinci ve üçüncü görevi ise sağ üst kısımda ve sol alt kısımda bulunan yıldızları yakalamaktır. İSÜA' nın bu hedef yıldız noktalara olan mesafesi 3 metrenin altında olduğunda mevcut görev tamamlanıp yeni hedefe doğru yönelmektedir. GY-NEO6MV2 GPS modülünün üretici firma tarafından belirlenen konum bilgisi hatası 5 metre olduğundan Şekil 6.4' te bazı sapmalar ortaya

çıkılmış olsa da İSÜA, üç hedef yıldızı sırayla yakalamaya çalışarak referans yola benzer bir seyir yapmış ve son olarak konumunu üçüncü hedef noktada durdurmaya çalışarak kullanıcı tarafından belirlenen görevleri tamamlamıştır.



Şekil 6.4: Tasarlanan İSÜA' nın test sürüşü.

BÖLÜM 7. SONUÇ VE ÖNERİLER

Bu çalışmada, literatürde yer alan havuç kovalama güdüm yasasına, doğrusal olmayan güdüm yasasına ve vektör alanı güdüm yasasına çeşitli optimizasyon yöntemleri uygulanarak İSÜA' nın muhtelif seyir şartlarına özgü (İSÜA' nın farklı hızlarına göre, İSÜA' nın farklı manevra yeteneklerine göre, İSÜA' nın takip etmesi istenilen referans yolların sayısına göre ve referans yolların konumlarına göre, İSÜA' nın referans yola uzaklığına göre) optimum parametrelerin belirlenmesi sağlanmıştır. Geliştirilen bu özgün entegre yöntem sayesinde, İSÜA en uygun rota açısını her bir zaman adımında belirleyerek yol izleme hatalarını geleneksel yöntemlere göre önemli ölçüde otomatik olarak azaltmaktadır.

Optimizasyon tekniği olarak genetik algoritma, desen arama, benzetimli tavlama ve vekil model optimizasyon yöntemleri kullanılmıştır. Bu optimizasyon yöntemlerinden desen arama ve benzetimli tavlama optimizasyon yöntemleri tek yada az parametrelili hedef fonksiyonlar için hızlı bir şekilde parametre optimizasyonu yapabildikleri gözlemlenmiştir. Ancak, desen arama ve benzetimli tavlama optimizasyon teknikleri çoklu parametreye sahip vektör alanı güdüm yasası gibi hedef fonksiyonlarda yerel minimum noktaya takılabildikleri görülmüştür. Genetik algoritma optimizasyon yönteminin iş akışı diğer optimizasyon yöntemlerine göre daha yavaş olduğu ve küresel minimum noktayı belirleyebilmesi için diğer optimizasyon tekniklerine göre daha fazla hedef fonksiyonu değerlendirmesine ihtiyaç duyduğu sonucuna varılmıştır. Optimizasyonun tamamlanması için geçen toplam sürenin genetik algoritmada daha uzun olduğu gözlemlenmiştir. Diğer yandan vekil model optimizasyon yöntemi hem çoklu parametreye sahip hedef fonksiyonlar için daha uygun bir yöntem olduğu hem de küresel minimum noktaları diğer metodlara göre daha etkili bulabildiği sonucuna varılmıştır.

Güdümlü yasaları incelendiğinde, havuç kovalama ve doğrusal olmayan güdümlü yasaları İSÜA' nın rotasının belirlenmesinde tek bir parametre kullandıklarından dolayı İSÜA' nın değişen seyir davranışlarına her durumda en az yol izleme hatasını netice verecek şekilde uyum sağlayamadıkları görülmüştür. Hangi optimizasyon yöntemi olursa olsun yüksek sayıda hedef fonksiyonun değerlendirilmesi durumunda bile çoklu parametre kullanan vektör alanı güdümlü yasası gibi algoritmaların netice verdiği minimum yol izleme hatasına, havuç kovalama ve doğrusal olmayan güdümlü yasalarının ulaşamadıkları gözlemlenmiştir. Çünkü, bu tek parametrelili güdümlü yasaları, İSÜA referans yoldan çok uzakta olduğunda (keskin manevra gereken bölgede) ve referans yola yaklaşıp kararlı davranış izlemesi gereken kritik bölgede (yumuşak ama istikrarlı manevranın gerektiği bölgede) tek sahip olduğu aynı parametreyi kullanmaktadır. Bu durumun ise yol izleme hatalarını arttırdığı sonucuna varılmıştır. Diğer yandan vektör alanı güdümlü yasası, sahip olduğu dört farklı parametre ile İSÜA' nın değişen şartlara göre davranışına daha iyi uyum sağlasa da kullandığı çok sayıdaki parametre, hem optimizasyon süresini uzatmakta hem de hedef fonksiyonun doğrusallığını düşürdüğü gözlemlenmiştir. Bu durum bazen, optimizasyon yönteminin yerel minimum noktaya takılmasına sebep olduğu görülmüştür. Bu çalışma kapsamında geliştirilen muhakeme esaslı güdümlü yasası, kullandığı iki adet parametre ile hem İSÜA' nın değişen şartlara göre davranışına etkili bir şekilde uyum sağlamak hem de çok fazla parametreyi içermediğinden optimizasyon yönteminin neticeye hızlı bir şekilde ulaşmasını kolaylaştırmaktadır. Ayrıca muhakeme esaslı güdümlü yasasına, özgün değer taşıyan uyarlamalı zaman adımları uygulanarak hedef fonksiyonun değerlendirme zamanı kısaltılmış ve her bir zaman adımından gelecek olan sayısal hatalar azaltılmıştır. Muhakeme esaslı güdümlü yasasının bahsedilen bu özelliklerinden dolayı, yol izleme hatalarının azaltılması ve uygulanan optimizasyon yönteminin daha kısa sürede tamamlanması açısından diğer yöntemlere göre ciddi avantajlarının olduğu sonucuna varılmıştır.

İSÜA' nın denetiminde birinci mertebeden sayısal türev alma ve birinci mertebeden sayısal integral alma yöntemlerinin, büyük sayısal hatalara sebep olduğu görülmüştür. Bu çalışmada, yüksek mertebeden sayısal türev ve yüksek mertebeden sayısal integral alma yöntemleri ile zenginleştirilmiş özgün bir kontrolcü tasarlanmıştır. Birinci mertebeden sayısal yöntemlere göre, yüksek mertebeden sayısal tekniklerin

kullanılmasının avantajı grafiklerle incelenmiştir. Yüksek mertebeden sayısal türev ve yüksek mertebeden sayısal integral alma yöntemlerinin daha az sayısal hataya sebep oldukları ve daha etkili yöntemler oldukları sonucuna varılmıştır.

Bu çalışma kapsamında deneysel çalışma olarak, arduino tabanlı bir İSÜA tasarlanmıştır. Tasarlanan prototip, uydudan konum bilgisi almak için GY-NEO6MV2 GPS modülünü, rota açısını belirlemek için QMC5883 GY-271 3 eksen dijital pusula sensörünü ve harici bir bilgisayar ile veri akışını sağlamak için NRF24L01 2.4 GHz kablosuz haberleşme modülünü kullanmaktadır. GY-NEO6MV2 GPS modülü, sadece konum verisi değil aynı zamanda İSÜA' nın rota açısını da belirleyebilmektedir. Ancak, İSÜA hareketsizken veya İSÜA yavaş hareket ediyorken GY-NEO6MV2 GPS modülünün, İSÜA' nın rota açısını hassas ve doğru bir şekilde belirleyemediği gözlemlenmiştir. İSÜA hareketsiz duruyorken bile rota açısının doğru ve hassas bir şekilde her an belirlenebilmesi için, QMC5883 GY-271 3 eksen dijital pusula sensörünün kullanılması gerektiği sonucuna varılmıştır. QMC5883 GY-271 3 eksen dijital pusula sensörü harici olarak İSÜA' na takıldıktan sonra, bu sensörün elektromanyetik alandan etkilendiği görülmüştür. QMC5883 GY-271 3 eksen dijital pusula sensörü, tüm elektronik kartlardan ve özellikle fırçasız motorlardan olabildiğince uzak bir yere, İSÜA' nın baş tarafta üst güvertesine, takılmıştır. Böylece, QMC5883 GY-271 3 eksen dijital pusula sensörünün hassas ve doğru bir şekilde çalışması sağlanmıştır.

Son olarak, tasarlanan İSÜA' nın seyir testi üstü açık büyük bir havuzda gerçekleştirilmiştir. Uydudan havuz üzerinde İSÜA' nın gitmesi istenilen üç hedef nokta belirlenmiştir. İSÜA, dışarıdan hiçbir müdahale olmadan hedef noktaları sırayla tek tek takip ederek verilen görevi tamamlamıştır.

KAYNAKLAR

- [1] Peng, Z., Wang, J., Wang, D., & Han, Q. L. (2020). An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Transactions on Industrial Informatics*, 17(2), 732-745.
- [2] Sun, W., & Gao, X. (2021). Deep Learning-Based Trajectory Tracking Control for Unmanned Surface Vehicle. *Mathematical Problems in Engineering*, 2021, 1-22.
- [3] Huang, Y., Wu, D., Yin, Z., & Yuan, Z. M. (2021). Design of UDE-based dynamic surface control for dynamic positioning of vessels with complex disturbances and input constraints. *Ocean Engineering*, 220, 108487.
- [4] Jimenez, J. F., & Giron-Sierra, J. M. (2020). USV based automatic deployment of booms along quayside mooring ships: Scaled experiments and simulations. *Ocean Engineering*, 207, 107438.
- [5] Kim, J. (2018). Target following and close monitoring using an unmanned surface vehicle. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11), 4233-4242.
- [6] Shao, G., Ma, Y., Malekian, R., Yan, X., & Li, Z. (2019). A novel cooperative platform design for coupled USV-UAV systems. *IEEE Transactions on Industrial Informatics*, 15(9), 4913-4922.
- [7] Qin, H., Li, C., & Sun, Y. (2020). Adaptive neural network-based fault-tolerant trajectory-tracking control of unmanned surface vessels with input saturation and error constraints. *IET Intelligent Transport Systems*, 14(5), 356-363.
- [8] Mao, Y., Chen, Q., Hou, Z., & Zheng, G. (2015, July). An improved nonlinear guidance law for unmanned aerial vehicles path following. In *2015 34th Chinese control conference (CCC)* (pp. 5271-5276). IEEE.
- [9] Niu, H., Lu, Y., Savvaris, A., & Tsourdos, A. (2016, April). Efficient path following algorithm for unmanned surface vehicle. In *OCEANS 2016-Shanghai* (pp. 1-7). IEEE.
- [10] Hernández-Morales, L., Valeriano-Medina, Y., Hernández-Santana, L., & Mesa-Suarez, E. (2020). Nonlinear guidance law algorithm applied to a small unmanned surface vehicle. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 234(3), 623-633.
- [11] Song, D., Gan, W., Yao, P., Zang, W., & Qu, X. (2023). Surface path tracking method of autonomous surface underwater vehicle based on deep reinforcement learning. *Neural Computing and Applications*, 35(8), 6225-6245.
- [12] Safwat, E., Zhang, W., Kamel, A., & Kassem, M. (2020). Robustness analysis of modified incremental nonlinear dynamic inversion for small UAVs. *Automatic Control and Computer Sciences*, 54, 128-138.
- [13] Perez-Leon, H., Acevedo, J. J., Maza, I., & Ollero, A. (2020, September). A 4D trajectory follower based on the 'Carrot chasing' algorithm for UAS within the U-

- space context. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1860-1867). IEEE.
- [14] Leonard, B. P. (1976). A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Int. J. Mech. Sci*, 16, 183-308.
 - [15] Zhang, D., Jiang, C., Liang, D., & Cheng, L. (2015). A review on TVD schemes and a refined flux-limiter for steady-state calculations. *Journal of Computational Physics*, 302, 114-154.
 - [16] Jiang, J., & Younis, R. M. (2017, February). An Efficient Fully-Implicit MFD-MUSCL Method Based on a Novel Multislope Limiting Procedure. In *SPE Reservoir Simulation Conference?* (p. D011S004R007). SPE.
 - [17] Zhang, S., Jiang, S., & Shu, C. W. (2008). Development of nonlinear weighted compact schemes with increasingly higher order accuracy. *Journal of Computational Physics*, 227(15), 7294-7321.
 - [18] Liu, Z., Zhang, Y., Yu, X., & Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, 41, 71-93.
 - [19] Zheng, Z., Sun, L., & Xie, L. (2017). Error-constrained LOS path following of a surface vessel with actuator saturation and faults. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(10), 1794-1805.
 - [20] Gong, X., Liu, L., Peng, Z., Wang, D., & Zhang, W. (2022). Resource-aware synchronized path following of multiple unmanned surface vehicles with experiments: A cooperative vector field approach. *Control Engineering Practice*, 124, 105184.
 - [21] Wang, X., Zhang, J., Zhang, D., & Shen, L. (2017). A virtual force guidance law for trajectory tracking and path following. In *Intelligent Autonomous Systems 14: Proceedings of the 14th International Conference IAS-14 14* (pp. 417-432). Springer International Publishing.
 - [22] Lee, S. H., Hur, S. W., Kwak, Y. Y., Nam, Y. H., & Kim, C. J. (2021). Ahead-time approach to carrot-chasing guidance law for an accurate trajectory-tracking control. *International Journal of Control, Automation and Systems*, 19(8), 2634-2651.
 - [23] Ashrafiuon, H., Muske, K. R., & McNinch, L. C. (2010, June). Review of nonlinear tracking and setpoint control approaches for autonomous underactuated marine vehicles. In *Proceedings of the 2010 American control conference* (pp. 5203-5211). IEEE.
 - [24] Zhang, R., Liu, Y., & Anderlini, E. (2022). Robust trajectory tracking control for unmanned surface vessels under motion constraints and environmental disturbances. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 236(2), 394-411.
 - [25] Liu, Z. (2022). Improved ELOS based path following control for underactuated surface vessels with roll constraint. *Ocean Engineering*, 245, 110348.
 - [26] Bejarano, G., Manzano, J. M., Salvador, J. R., & Limon, D. (2022). Nonlinear model predictive control-based guidance law for path following of unmanned surface vehicles. *Ocean Engineering*, 258, 111764.
 - [27] Zhang, G., Li, J., Jin, X., & Liu, C. (2021). Robust adaptive neural control for wing-sail-assisted vehicle via the multiport event-triggered approach. *IEEE Transactions on Cybernetics*, 52(12), 12916-12928.
 - [28] Zhang, W., Wang, F., & Zhai, L. (2022, August). Data-driven ESO-based LOS Guidance Law for Path Following of Unmanned Surface Vehicles With Sideslip

- Compensation. In *2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS)* (pp. 1242-1246). IEEE.
- [29] Abrougui, H., Nejim, S., Hachicha, S., Zaoui, C., & Dallagi, H. (2021). Modeling, parameter identification, guidance and control of an unmanned surface vehicle with experimental results. *Ocean Engineering*, 241, 110038.
 - [30] Wang, Z., Gong, Z., Xu, J., Wu, J., & Liu, M. (2021). Path following for unmanned combat aerial vehicles using three-dimensional nonlinear guidance. *IEEE/ASME Transactions on Mechatronics*, 27(5), 2646-2656.
 - [31] Wen, Y., Tao, W., Zhu, M., Zhou, J., & Xiao, C. (2020). Characteristic model-based path following controller design for the unmanned surface vessel. *Applied Ocean Research*, 101, 102293.
 - [32] Lekkas, A. M., & Fossen, T. I. (2014). Integral LOS path following for curved paths based on a monotone cubic Hermite spline parametrization. *IEEE Transactions on Control Systems Technology*, 22(6), 2287-2301.
 - [33] Bejarano, G., Rodríguez, D., Lemos, J. M., Vargas, M., & Ortega, M. G. (2020). MINLP-based hybrid strategy for operating mode selection of TES-backed-up refrigeration systems. *International Journal of Robust and Nonlinear Control*, 30(15), 6091-6111.
 - [34] Huang, Y., Wu, D., Li, L., & Feng, N. (2022). Event-triggered cooperative path following control of multiple underactuated unmanned surface vehicles with complex unknowns and actuator saturation. *Ocean Engineering*, 249, 110740.
 - [35] Naini, S. J. (2015). Optimal Line-of-Sight guidance law for moving targets. In *14th International Conference of Iranian Aerospace Society*.
 - [36] Sujit, P. B., Saripalli, S., & Sousa, J. B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Systems Magazine*, 34(1), 42-59.
 - [37] Miao, C. X., & Fang, J. C. (2012). An adaptive three-dimensional nonlinear path following method for a fix-wing micro aerial vehicle. *International Journal of Advanced Robotic Systems*, 9(5), 206.
 - [38] He, M., Wang, X., & Cui, N. (2022). Modified vector field and nonlinear guidance law for low-cost UAV path following. *Aircraft Engineering and Aerospace Technology*, 94(11), 20-28.
 - [39] Li, M., Guo, C., Yu, H., & Yuan, Y. (2022). Line-of-sight-based global finite-time stable path following control of unmanned surface vehicles with actuator saturation. *ISA transactions*, 125, 306-317.
 - [40] Ibrahim, A. E. B. (2022). Self-tuning Look-ahead Distance of Pure-pursuit Path-following Control for Autonomous Vehicles Using an Automated Curve Information Extraction Method. *International Journal of Intelligent Transportation Systems Research*, 20(3), 709-719.
 - [41] Perez-Leon, H., Acevedo, J. J., Millan-Romera, J. A., Castillejo-Calle, A., Maza, I., & Ollero, A. (2020). An aerial robot path follower based on the ‘carrot chasing’ algorithm. In *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 2* (pp. 37-47). Springer International Publishing.
 - [42] Jin, X., Mei, W., & Zhaolong, Y. (2020, March). Path following control for unmanned aerial vehicle based on carrot chasing algorithm and PLOS. In *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIS)* (pp. 571-576). IEEE.

- [43] Safwat, E., Zhang, W., Wu, M., Lyu, Y., & Jia, Q. (2018, October). Robust path following controller for unmanned aerial vehicle based on carrot chasing guidance law using dynamic inversion. In *2018 18th International Conference on Control, Automation and Systems (ICCAS)* (pp. 1444-1450). IEEE.
- [44] Subbarao, K., & Ahmed, M. (2014). Nonlinear guidance and control laws for three-dimensional target tracking applied to unmanned aerial vehicles. *Journal of Aerospace Engineering*, 27(3), 604-610.
- [45] Park, S., Deyst, J., & How, J. P. (2007). Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of guidance, control, and dynamics*, 30(6), 1718-1728.
- [46] Cho, N., Kim, Y., & Park, S. (2015). Three-dimensional nonlinear differential geometric path-following guidance law. *Journal of Guidance, Control, and Dynamics*, 38(12), 2366-2385.
- [47] Manzano, J. M., Salvador, J. R., Bejarano, G., & Limon, D. (2021, December). Nonlinear model predictive control applied to robust guidance of autonomous surface vehicles. In *2021 60th IEEE Conference on Decision and Control (CDC)* (pp. 5735-5740). IEEE.
- [48] Qiu, B., Wang, G., & Fan, Y. (2020). Trajectory linearization-based adaptive plos path following control for unmanned surface vehicle with unknown dynamics and rudder saturation. *Applied Sciences*, 10(10), 3538.
- [49] Yu, Y., Guo, C., & Yu, H. (2019). Finite-time PLOS-based integral sliding-mode adaptive neural path following for unmanned surface vessels with unknown dynamics and disturbances. *IEEE transactions on automation science and engineering*, 16(4), 1500-1511.
- [50] Nelson, D. R., Barber, D. B., McLain, T. W., & Beard, R. W. (2007). Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3), 519-529.
- [51] Koessler, E., & Almomani, A. (2021). Hybrid particle swarm optimization and pattern search algorithm. *Optimization and Engineering*, 22, 1539-1555.
- [52] Delahaye, D., Chaimatanan, S., & Mongeau, M. (2019). Simulated annealing: From basics to applications. *Handbook of metaheuristics*, 1-35.
- [53] Müller, J. (2017). SOCEMO: surrogate optimization of computationally expensive multiobjective problems. *INFORMS Journal on Computing*, 29(4), 581-596.
- [54] Mirjalili, S., & Mirjalili, S. (2019). Genetic algorithm. *Evolutionary Algorithms and Neural Networks: Theory and Applications*, 43-55.
- [55] Hooke, R., & Jeeves, T. A. (1961). ``Direct Search"Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)*, 8(2), 212-229.
- [56] Kandhway, K. (2022, December). Optimal Control by Pattern Search Optimization Method. In *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 224-228). IEEE.
- [57] Li, M., Cui, Y., Shi, X., Zhang, Z., Zhao, X., Zhu, X., & Gao, J. (2021). Simulated annealing-based optimal design of energy efficient ternary extractive dividing wall distillation process for separating benzene-isopropanol-water mixtures. *Chinese Journal of Chemical Engineering*, 33, 203-210.
- [58] Zhou, X., Li, S., & Feng, Y. (2020). Quantum circuit transformation based on simulated annealing and heuristic search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12), 4683-4694.

- [59] Moriguchi, K. (2020). Acceleration and enhancement of reliability of simulated annealing for optimizing thinning schedule of a forest stand. *Computers and Electronics in Agriculture*, 177, 105691.
- [60] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- [61] Tang, S., Peng, M., Xia, G., Wang, G., & Zhou, C. (2020). Optimization design for supercritical carbon dioxide compressor based on simulated annealing algorithm. *Annals of Nuclear Energy*, 140, 107107.
- [62] Ramli, A. S., Rashid, M. I. M., & Ahmad, M. A. (2020). Energy management strategy of hev based on simulated annealing. *International Journal of Integrated Engineering*, 12(2), 30-37.
- [63] Metin, S. (2021). Benzetimli tavlama algoritması ile eksik veri tamamlama. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 33(1), 295-301.
- [64] Müller, J., & Day, M. (2019). Surrogate optimization of computationally expensive black-box problems with hidden constraints. *INFORMS Journal on Computing*, 31(4), 689-702.
- [65] Koziel, S., Ciaurri, D. E., & Leifsson, L. (2011). Surrogate-based methods. *Computational optimization, methods and algorithms*, 33-59.
- [66] Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [67] Keklik, G., & Özcan, B. D. (2023). Genetik Algoritmaların İşleyişi ve Genetik Algoritma Uygulamalarında Kullanılan Operatörler. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 6(1), 1052-1066.
- [68] Setiawan, F. A., Kadir, R. E. A., Gamayanti, N., Santoso, A., Bilfaqih, Y., & Hidayat, Z. (2021, February). Dynamic modelling and controlling Unmanned Surface Vehicle. In *IOP Conference Series: Earth and Environmental Science* (Vol. 649, No. 1, p. 012056). IOP Publishing.
- [69] Rhee, I., Park, S., & Ryoo, C. K. (2010, August). A tight path following algorithm of an UAS based on PID control. In *Proceedings of SICE Annual Conference 2010* (pp. 1270-1273). IEEE.
- [70] Cardelli, E., Faba, A., Pompei, M., & Quondam Antonio, S. (2019). Pattern search approach to ferromagnetic material modelling. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 32(4), e2271.
- [71] Wahid, F., Zia, M. S., Rais, R. N. B., Aamir, M., Butt, U. M., Ali, M., ... & Khalid, O. (2020). An enhanced firefly algorithm using pattern search for solving optimization problems. *IEEE Access*, 8, 148264-148288.
- [72] Ait-Saadi, A., Meraihi, Y., Soukane, A., Ramdane-Cherif, A., & Gabis, A. B. (2022). A novel hybrid chaotic Aquila optimization algorithm with simulated annealing for unmanned aerial vehicles path planning. *Computers and Electrical Engineering*, 104, 108461.
- [73] Venkatesan, R., & Narayanswamy, R. (2003). Application of genetic algorithm and simulated annealing for optimization of extrusion die ratio and die cone angle. In *Proceedings of the 12th International Scientific Conference on Achievements in Mechanical and Material Engineering, Gliwice, Zakopane* (pp. 1001-1006).
- [74] Zhao, S., Wang, X., Lin, Z., Zhang, D., & Shen, L. (2018). Integrating vector field approach and input-to-state stability curved path following for unmanned aerial

- vehicles. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(8), 2897-2904.
- [75] Yao, W., de Marina, H. G., Lin, B., & Cao, M. (2021). Singularity-free guiding vector field for robot navigation. *IEEE Transactions on Robotics*, 37(4), 1206-1221.
 - [76] Wang, Y., Wang, X., Zhao, S., & Shen, L. (2018). Vector field based sliding mode control of curved path following for miniature unmanned aerial vehicles in winds. *Journal of Systems Science and Complexity*, 31, 302-324.
 - [77] Wang, T., Chen, Y., Liang, J., Wang, C., & Zhang, Y. (2012). Combined of vector field and linear quadratic Gaussian for the path following of a small unmanned helicopter. *IET Control Theory & Applications*, 6(17), 2696-2703.
 - [78] Sun, Y. P., & Liang, Y. C. (2022). Vector field path-following control for a small unmanned ground vehicle with Kalman filter estimation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 236(14), 1885-1899.
 - [79] Li, Z., Bachmayer, R., & Vardy, A. (2017, June). Vector field path following control for unmanned surface vehicles. In *OCEANS 2017-Aberdeen* (pp. 1-9). IEEE.
 - [80] Jung, W., Lim, S., Lee, D., & Bang, H. (2016). Unmanned aircraft vector field path following with arrival angle control. *Journal of Intelligent & Robotic Systems*, 84, 311-325.
 - [81] Dezyani, M., Dalayeli, H., Yousefi, S., & Farrokhfal, H. (2018). Two-level design optimization for compression stiffened panel. *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering*, 42, 199-211.
 - [82] <https://github.com/mikalhart/TinyGPSPlus.git>
 - [83] <https://github.com/dthain/QMC5883L.git>
 - [84] https://github.com/mericuzun/Arduino-Uno-NRF24L01-GY_NEO6MV2.git
 - [85] <https://github.com/br3ttb/Arduino-PID-Library.git>
 - [86] Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2021). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, 9, 818-827.
 - [87] Leonard, B. P. (1994). Comparison of truncation error of finite-difference and finite-volume formulations of convection terms. *Applied mathematical modelling*, 18(1), 46-50.
 - [88] Mazumder, S. (2015). *Numerical methods for partial differential equations: finite difference and finite volume methods*. Academic Press.
 - [89] Izzo, F., Runborg, O., & Tsai, R. (2023). High-order corrected trapezoidal rules for a class of singular integrals. *Advances in Computational Mathematics*, 49(4), 60.
 - [90] Unal, O., Akkas, N., & Atali, G. (2023). An Innovative Approach to Nonlinear Guidance Law for Path Following of Unmanned Surface Vehicle. *Journal of the Chinese Society of Mechanical Engineers*, 44(6), 573-581.

EKLER

EK A: İSÜA’ nın Kablosuz Haberleşme Sistemi için Gerekli Açık Kaynak Kodları.

```
// Verici
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
int mesaj[2];
RF24 verici(9,10);
const int kanal = 123;
int a2 = 0;
int a3 = 0;

void setup()
{
    Serial.begin(9600);
    verici.begin();
    verici.openWritingPipe(kanal);
    pinMode(3, OUTPUT);
}

void loop()
{
    digitalWrite(3, 150);
    a2 = analogRead(2);
    Serial.println(a2);
    a3 = analogRead(3);
    Serial.println(a3);
    if (a2 > 200) {mesaj[0] = 7;}
```

```

    if (a2 < 200) {mesaj[0] = 0;}
    if (a3 > 200) {mesaj[1] = 8;}
    if (a3 < 200) {mesaj[1] = 0;}
    verici.write(&mesaj, sizeof(mesaj));
}

----**-----

// Alici
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <Servo.h>
Servo ESC5;
Servo ESC6;
int mesaj[2];
RF24 alici(9,8);
const int kanal = 123;

void setup(){
    Serial.begin(9600);
    alici.begin();
    alici.openReadingPipe(1,kanal);
    alici.startListening();
    pinMode(4, OUTPUT);
    pinMode(7, OUTPUT);
    ESC5.attach(5,1000,2000);
    ESC6.attach(6,1000,2000);
    for (int i = 0; i <= 10; i++)
    {ESC5.write(i),delay(1000),ESC6.write(i),delay(1000);}
}

void loop(){

```

```

if (alici.available())
{
    bool done = false;
    while (!done){
        done = alici.read(&mesaj, sizeof(mesaj));
        if (mesaj[0] == 7){ESC5.write(90);}
        if (mesaj[0] != 7){ESC5.write(0);}
        if (mesaj[1] == 8){ESC6.write(90);}
        if (mesaj[1] != 8){ESC6.write(0);}
    }
}
}
}

```

---***---

// Otonom Seyir Sırasında Verilerin Kullanıcı Bilgisayarına Gönderilmesi

```

#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"

```

```

RF24 rx(9,10); //CE CSN
const int pipe = 1;
double data[7];

```

```

void setup()
{
    Serial.begin(9600);
    rx.begin();
    rx.openReadingPipe(1,pipe);
    rx.startListening();
}

```

```

void loop()
{
    if (rx.available())
    {
        bool done = false;
        while (!done)
        {
            done = rx.read(&data, sizeof(data));

```

```

        Serial.print(data[0],5);
        Serial.print(F(" "));
        Serial.print(data[1],5);
        Serial.print(F(" "));
        Serial.print(data[2]);
        Serial.print(F(" "));
        Serial.print(data[3]);
        Serial.print(F(" "));
        Serial.print(data[4]);
        Serial.print(F(" "));
        Serial.print(data[5]);
        Serial.print(F(" "));
        Serial.println(data[6]);
    }
}
}

```

---***---

```

// Fırçasız Motor Kontrolü

#include <Servo.h>

Servo ESC;

int potansiyometre;

void setup() {
    ESC.attach(5,1000,2000);
}

void loop() {
    potansiyometre = analogRead(A2);
    potansiyometre = map(potValue, 0, 1023, 0, 180);
    ESC.write(potansiyometre);
}

```

EK B: GY-NEO6MV2 GPS Modülünün Açık Kaynak Kodları.

```
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>

/*
   This sample sketch demonstrates the normal use of a TinyGPSPlus
   (TinyGPSPlus) object.

   It requires the use of SoftwareSerial, and assumes that you have a
   4800-baud serial GPS device hooked up on pins 4(rx) and 3(tx).
*/

static const int RXPin = 4, TXPin = 3;
static const uint32_t GPSBaud = 4800;

// The TinyGPSPlus object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);

void setup()
{
  Serial.begin(115200);
  ss.begin(GPSBaud);

  Serial.println(F("DeviceExample.ino"));
  Serial.println(F("A simple demonstration of TinyGPSPlus with an
  attached GPS module"));
  Serial.print(F("Testing TinyGPSPlus library v. "));
  Serial.println(TinyGPSPlus::libraryVersion());
  Serial.println(F("by Mikal Hart"));
  Serial.println();
}

void loop()
{
```

```
// This sketch displays information every time a new sentence is
correctly encoded.
```

```
while (ss.available() > 0)
{
    if (gps.encode(ss.read()))
        displayInfo();

    if (millis() > 5000 && gps.charsProcessed() < 10)
    {
        Serial.println(F("No GPS detected: check wiring."));
        while(true);
    }
}
```

```
void displayInfo()
{
    Serial.print(F("Location: "));
    if (gps.location.isValid())
    {
        Serial.print(gps.location.lat(), 6);
        Serial.print(F(", "));
        Serial.print(gps.location.lng(), 6);
    }
    else
    {
        Serial.print(F("INVALID"));
    }

    Serial.print(F("  Date/Time: "));
    if (gps.date.isValid())
    {
        Serial.print(gps.date.month());
        Serial.print(F("/"));
        Serial.print(gps.date.day());
    }
}
```

```

        Serial.print(F("/"));
        Serial.print(gps.date.year());
    }
    else
    {
        Serial.print(F("INVALID"));
    }

    Serial.print(F(" "));
    if (gps.time.isValid())
    {
        if (gps.time.hour() < 10) Serial.print(F("0"));
        Serial.print(gps.time.hour());
        Serial.print(F(":"));
        if (gps.time.minute() < 10) Serial.print(F("0"));
        Serial.print(gps.time.minute());
        Serial.print(F(":"));
        if (gps.time.second() < 10) Serial.print(F("0"));
        Serial.print(gps.time.second());
        Serial.print(F("."));
        if (gps.time.centisecond() < 10) Serial.print(F("0"));
        Serial.print(gps.time.centisecond());
    }
    else
    {
        Serial.print(F("INVALID"));
    }

    Serial.println();
}

```

EK C: QMC5883 GY-271 3 Eksen Dijital Pusulanın Açık Kaynak Kodları.

```
#include <QMC5883L.h>
#include <Wire.h>

QMC5883L compass;

void setup()
{
    Wire.begin();

    compass.init();
    compass.setSamplingRate(50);

    Serial.begin(9600);
    Serial.println("QMC5883L Compass Demo");
    Serial.println("Turn compass in all directions to calibrate....");
}

void loop()
{
    int heading = compass.readHeading();
    if(heading==0) {
        /* Still calibrating, so measure but don't print */
    } else {
        Serial.println(heading);
    }
}
```

EK D: NRF24L01 2.4 Ghz Kablosuz Haberleşme Modülünün Alıcısı için Kullanılan Açık Kaynak Kodu.

```
#include <SPI.h>
#include <RF24.h>
#include <nRF24L01.h>
#include <SoftwareSerial.h>
#include <TinyGPSPlus.h>
```

```
TinyGPSPlus gps;
// CE, CSN
RF24 radio(9, 8);
// RX, TX - TX, RX of gps
SoftwareSerial ss(2, 3);
```

```
struct dataStruct{
    // latitude, longitude and altitude are defined as double in case of an
    negative value
    double latitude;
    double longitude;
    double altitude;
    unsigned long date;
    unsigned long time;
    int satalites;
}gps_data;
```

```
void setup() {
    Serial.begin(115200);
    ss.begin(9600);

    // Setup and configure RF radio
    radio.begin();
    // Open pipe
```

```

    radio.openWritingPipe(0xF0F0F0E1LL);
    // Lowest frequency 2.4GHz
    radio.setChannel(0x00);
    // LOW power level
    // Default is MAX
    radio.setPALevel(RF24_PA_LOW);
    // Lowest bitrate
    radio.setDataRate(RF24_250KBPS);
    radio.stopListening();
    radio.powerUp();
}

void loop() {
    while (ss.available() > 0){
        if (gps.encode(ss.read())){
            get_data();
            radio.write(&gps_data, sizeof(gps_data));
        }
    }
}

void get_data(){
    if (gps.location.isValid()){
        gps_data.latitude = gps.location.lat();
        gps_data.longitude = gps.location.lng();
        gps_data.altitude = gps.altitude.meters();
    }
    else{
        gps_data.latitude = 0;
        gps_data.longitude = 0;
    }
}

```

```
if (gps.date.isValid()){
    gps_data.date = gps.date.value();
}
else{
    gps_data.date = 0;
}

if (gps.time.isValid()){
    gps_data.time = (gps.time.value());
}
else{
    gps_data.time = 0;
}

if (gps.satellites.isValid()){
    gps_data.satalites = gps.satellites.value();
}
else{
    gps_data.satalites = 0;
}
}
```

EK E: NRF24L01 2.4 Ghz Kablosuz Haberleşme Modülünün Vericisi için Kullanılan Açık Kaynak Kodu.

```
#include <SPI.h>
#include <RF24.h>
#include <nRF24L01.h>

// CE, CSN
RF24 radio(9, 8);

struct dataStruct{
    // latitude, longitude and altitude are defined as double in case of an
    // negative value
    double latitude;
    double longitude;
    double altitude;
    unsigned long date;
    unsigned long time;
    int satalites;
}gps_data;

void setup() {
    Serial.begin(115200);

    // Setup and configure RF radio
    radio.begin();
    // Open pipe
    radio.openReadingPipe(0, 0xF0F0F0F0E1LL);
    // Lowest frequency, 2.4GHz
    radio.setChannel(0x00);
    // MAX power level
    // Default is MAX
    radio.setPALevel(RF24_PA_MAX);
```

```

// Lowest bitrate
radio.setDataRate(RF24_250KBPS);
radio.startListening();
radio.powerUp();
}

void loop() {
  if (radio.available()) {
    radio.read(&gps_data, sizeof(gps_data));

    unsigned long date_value = gps_data.date;
    unsigned int date_firstTwoDigits = (date_value / 10000) % 100;
    unsigned int date_secondTwoDigits = (date_value / 100) % 100;
    unsigned int date_thirdTwoDigits = date_value % 100;

    unsigned long time_value = gps_data.time;
    unsigned int time_firstTwoDigits = (time_value / 1000000) % 100;
    unsigned int time_secondTwoDigits = (time_value / 10000) % 100;
    unsigned int time_thirdTwoDigits = (time_value / 100) % 100;

    Serial.print("Location: ");
    Serial.print(gps_data.latitude, 6);
    Serial.print(", ");
    Serial.print(gps_data.longitude, 6);
    Serial.print(", ");
    Serial.print(gps_data.altitude);
    Serial.print("M");

    Serial.print("  Date: ");
    Serial.print(date_firstTwoDigits);
    Serial.print(":");
    Serial.print(date_secondTwoDigits);
    Serial.print(":");

```

```
Serial.print(date_thirdTwoDigits);

Serial.print("  Time: ");
Serial.print(time_firstTwoDigits);
Serial.print(":");
Serial.print(time_secondTwoDigits);
Serial.print(":");
Serial.print(time_thirdTwoDigits);

Serial.print("  Satalites: ");
Serial.print(gps_data.satalites);
Serial.println();
}
}
```

EK F: PID Kontrolcü için Kullanılan Açık Kaynak Kod.

```
/*
*****
* PID Basic Example
* Reading analog input 0 to control analog PWM output 3
*****
*/

#include <PID_v1.h>

#define PIN_INPUT 0
#define PIN_OUTPUT 3

//Define Variables we'll be connecting to
double Setpoint, Input, Output;

//Specify the links and initial tuning parameters
double Kp=2, Ki=5, Kd=1;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup()
{
    //initialize the variables we're linked to
    Input = analogRead(PIN_INPUT);
    Setpoint = 100;

    //turn the PID on
    myPID.SetMode(AUTOMATIC);
}

void loop()
{
    Input = analogRead(PIN_INPUT);
    myPID.Compute();
    analogWrite(PIN_OUTPUT, Output);
}
```

EK G: Tasarlanan İSÜA’ nın Test Sürüşü Verileri.

Tablo EK G.1. Tasarlanan İSÜA’ nın test sürüşünden elde edilen veriler.

Enlem	Boylam	Rota Açısı (Pusula)	Rota Açısı (GPS)	Hız (m/s)	Zaman	Hedef Nokta
41,07181	30,93116	46	0	0,1	8370100	0
41,07181	30,93116	47	0	0,07	8370200	0
41,07181	30,93117	45	0	0,01	8370300	0
41,07181	30,93117	47	0	0,02	8370400	0
41,07181	30,93116	225	0	0,17	8370800	0
41,0718	30,93116	213	131,75	0,44	8370900	0
41,0718	30,93117	132	122,87	0,34	8371000	0
41,0718	30,93117	117	75,53	0,35	8371100	0
41,0718	30,93117	105	75,53	0,13	8371200	0
41,0718	30,93117	152	157,35	0,5	8371300	0
41,07181	30,93116	24	2,88	0,62	8371600	0
41,0718	30,93116	107	130,59	0,61	8371700	0
41,0718	30,93116	194	165,5	0,69	8371800	0
41,0718	30,93116	221	207,68	0,54	8372000	0
41,0718	30,93116	108	78,84	0,95	8372100	0
41,07182	30,93115	77	342,59	0,08	8373100	0
41,07183	30,93114	49	342,59	0,53	8373200	0
41,07183	30,93116	22	342,59	1,36	8373400	0
41,07185	30,93114	117	342,59	0,67	8373700	0
41,07187	30,93113	67	342,59	0,67	8373800	0
41,07188	30,93116	73	5,52	0,57	8374000	1
41,07189	30,93116	110	5,52	0,57	8374300	1
41,07189	30,93116	61	5,52	0,9	8374400	1

Tablo EK G.1 (devam): Tasarlanan İSÜA' nın test sürüşünden elde edilen veriler.

Enlem	Boylam	Rota Açısı (Pusula)	Rota Açısı (GPS)	Hız (m/s)	Zaman	Hedef Nokta
41,07189	30,9312	79	5,52	0,77	8374700	1
41,07189	30,93121	40	5,52	1,11	8374800	1
41,0719	30,93123	97	5,52	0,92	8375000	1
41,07191	30,93123	71	5,52	1,08	8375100	1
41,0719	30,93127	93	63,17	0,85	8375500	1
41,07191	30,93128	44	63,17	0,45	8375700	1
41,07191	30,93127	89	63,17	0,69	8375800	1
41,07191	30,93128	29	63,17	0,37	8375900	1
41,07192	30,93129	47	45,21	0,59	8380100	1
41,07192	30,93131	111	45,21	0,95	8380300	1
41,07192	30,93133	51	64,94	0,67	8380600	1
41,07193	30,93138	150	96,57	0,98	8381300	1
41,07194	30,93139	12	96,57	0,96	8381400	1
41,07194	30,93139	87	109,1	0,76	8381500	1
41,07194	30,9314	83	84,6	0,59	8381600	1
41,07195	30,93143	23	46,52	0,64	8382000	1
41,07197	30,93145	21	52,32	1,02	8382300	1
41,07199	30,93152	261	256,55	0,67	8384400	2
41,07199	30,93151	220	221,74	0,6	8384500	2
41,07194	30,93144	206	184,57	0,52	8390300	2
41,07194	30,93144	296	184,57	0,68	8390400	2
41,07194	30,93141	243	204,98	0,5	8390700	2
41,07194	30,93141	173	204,98	0,5	8390800	2
41,07194	30,93141	238	204,98	0,54	8390900	2

Tablo EK G.1 (devam): Tasarlanan İSÜA' nın test sürüşünden elde edilen veriler.

Enlem	Boylam	Rota Açısı (Pusula)	Rota Açısı (GPS)	Hız (m/s)	Zaman	Hedef Nokta
41,07193	30,93139	249	254,81	0,57	8391100	2
41,07191	30,93136	252	215,83	0,59	8391500	2
41,07191	30,93137	178	153,95	0,8	8391600	2
41,07191	30,93136	246	220,35	0,56	8391700	2
41,07191	30,93136	160	190,1	0,62	8391800	2
41,07189	30,93135	248	224,71	0,66	8392100	2
41,07189	30,93134	186	207,6	0,53	8392200	2
41,07189	30,93133	241	238,62	0,58	8392300	2
41,07189	30,93133	213	238,62	0,39	8392400	2
41,07189	30,93133	218	241,03	0,64	8392500	2
41,07188	30,93132	275	245,04	0,63	8392600	2
41,07187	30,93131	210	206,2	0,77	8392700	2
41,07187	30,93131	231	225,52	0,44	8392800	2
41,07186	30,9313	269	226,46	0,61	8392900	2
41,07186	30,9313	206	228,22	0,69	8393000	2
41,07186	30,93129	249	251,2	0,62	8393100	2
41,07186	30,93129	187	251,2	0,13	8393200	2
41,07186	30,93129	235	251,2	0,65	8393300	2
41,07185	30,93128	192	203,69	0,75	8393400	2
41,07184	30,93128	265	220,53	0,44	8393500	2
41,07184	30,93127	165	220,53	0,34	8393600	2
41,07184	30,93126	249	220,53	0,71	8393800	2
41,07183	30,93127	240	220,53	0,25	8393900	2
41,07183	30,93126	227	220,53	0,25	8394000	2

Tablo EK G.1 (devam): Tasarlanan İSÜA' nın test sürüşünden elde edilen veriler.

Enlem	Boylam	Rota Açısı (Pusula)	Rota Açısı (GPS)	Hız (m/s)	Zaman	Hedef Nokta
41,07182	30,93126	213	220,53	0,09	8394100	2
41,07182	30,93125	200	220,53	0,05	8394200	2
41,07183	30,93124	160	220,53	0,05	8394400	2
41,07183	30,93124	141	220,53	0,04	8394500	2
41,07183	30,93124	125	220,53	0,05	8394600	2
41,07183	30,93124	110	220,53	0,04	8394700	2
41,07183	30,93124	97	220,53	0,04	8394800	2
41,07183	30,93125	86	220,53	0,04	8394900	2
41,07183	30,93125	79	220,53	0,05	8395000	2
41,07183	30,93125	78	220,53	0,06	8395100	2
41,07183	30,93125	83	220,53	0,05	8395200	2
41,07183	30,93125	86	220,53	0,04	8395300	2
41,07183	30,93125	86	220,53	0,04	8395400	2
41,07183	30,93125	102	220,53	0,04	8395900	2
41,07183	30,93125	107	220,53	0,04	8400000	2
41,07183	30,93125	134	220,53	0,06	8400400	2
41,07183	30,93125	142	220,53	0,04	8400500	2
41,07183	30,93125	147	220,53	0,03	8400600	2
41,07183	30,93125	162	220,53	0,07	8401300	2
41,07183	30,93125	158	220,53	0,06	8401400	2
41,07183	30,93125	156	220,53	0,06	8401500	2