

**T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MOBİL ROBOTLARIN DAVRANIŞ PLANLAMASINDA PID, BULANIK VE
GAUSS DENETÇİLERİNİN BAŞARIMLARININ KARŞILAŞTIRILMASI**

CİHANGİR HAN KAZANCI

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

HAZİRAN 2019

**T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MOBİL ROBOTLARIN DAVRANIŞ PLANLAMASINDA PID, BULANIK VE
GAUSS DENETÇİLERİNİN BAŞARIMLARININ KARŞILAŞTIRILMASI**

CİHANGİR HAN KAZANCI

**YÜKSEK LİSANS TEZİ
BİGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

HAZİRAN 2019

Tez Başlığı: Mobil Robotların Davranış Planlamasında PID, Bulanık ve Gauss Denetçilerinin Başarımlarının Karşılaştırılması

Tezi Hazırlayan: Cihangir Han KAZANCI

Sınav Tarihi: 17.06.2019

Yukarıda adı geçen tez jürimizce değerlendirilerek Bilgisayar Mühendisliği Yazılım Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Tez Danışmanı: **Dr. Öğr. Üy. Adnan Fatih KOCAMAZ**
İnönü Üniversitesi

Tez Asil Jüri: **Prof. Dr. İbrahim TÜRKOĞLU**
Fırat Üniversitesi

Tez Asil Jüri: **Dr. Öğr. Üy. Abdullah ATEŞ**
İnönü Üniversitesi

Prof. Dr. Halil İbrahim ADIGÜZEL
Enstitü Müdürü

ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduğum “Mobil Robotların Davranış Planlamasında PID, Bulanık ve Gauss Denetçilerinin Başarımlarının Karşılaştırılması” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Cihangir Han KAZANCI

ÖZET

Yüksek Lisans Tezi

MOBİL ROBOTLARIN DAVRANIŞ PLANLAMASINDA PID, BULANIK VE GAUSS DENETÇİLERİNİN BAŞARIMLARININ KARŞILAŞTIRILMASI

Cihangir Han KAZANCI

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

64 + x sayfa

2019

Danışman: Dr. Öğr. Üy. Adnan Fatih KOCAMAZ

Gezgin robota uygulanacak olan davranış modelleme algoritmalarının belirlenmesi robotik uygulamalarında en önemli hususlardan biridir. Bu kapsamda robot hareketlerini çalışma ortamında (konfigürasyon uzayı) yapılacak görevlere ve ortam değişkenlerine göre karakterize etmek gerekmektedir. Bu tez çalışmasında, başlangıç noktasından hedef noktasına, engellerden kaçarak ulaşmaya çalışan diferansiyel tahrikli gezgin bir robota uygulanan kontrol yöntemleri ile ilgili geliştirmeler yapılmış ve gezinim algoritmaları incelenmiştir.

Mobil robot üzerindeki sensörlerden gelen veriler kullanılarak robotun çevresini algılaması ve bulunduğu ortama göre hareket etmesi sağlanmıştır. Veriler mobil robot üzerindeki sekiz adet ultrasonik mesafe sensöründen elde edilmektedir. Ultrasonik mesafe sensörleri sayesinde mobil robotun engellere göre konumu belirlenmektedir. Elde edilen bu veriler kullanılan algoritmalarda birer girdi değeri olarak kullanılmıştır. Bu girdi değerleri ile mobil robotun engellerden kaçma davranışı modellenmiştir.

Çalışmada daha önce bu konu üzerine geliştirilmiş olan PID tabanlı Sim.I.Am gezgin robot simülatörü incelenmiştir. Bu kapsamda, Sim.I.Am gezgin robot simülatör uygulamasında tespit edilen eksiklikleri gidermek ve kullanılan algoritmanın yerine farklı algoritmalar kullanılarak sonuçlar gözlemlenmiştir. Çalışmanın ilk kısmında mobil robota uygulanan PID gezinim algoritmasının optimizasyonu anlatılmıştır. Özellikle mobil robotun hareket ettiği ortama göre kullanılan algoritmalar yetersiz kalabilmektedir. Çalışmanın ilerleyen bölümlerinde diferansiyel tahrikli gezgin robot sistemlerinin gezinim denetçilerinin Bulanık ve Gauss algoritmaları ile yapılandırılması önerilmiştir.

Geliştirilen algoritmanın etkinliğini ve verimliliğini test etmek amacıyla Sim.I.Am simülasyonu, üzerinde güncellemeler yapılarak kullanılmıştır. Simülasyon ortamında bulunan engellerin konumları değiştirilerek kullanılan yeni algoritmaların kıyaslanması sağlanmıştır. Engel konumlarının değiştiği bir test ortamında simülasyon çıktıları incelendiğinde incelenen ve önerilen algoritmaların başarılı bir sonuç ürettiği gözlemlenmiştir. Tez kapsamında, önerilen algoritmalar ile elde edilen denetçilerin başarımı, daha önce yapılmış olan çalışmalar ile karşılaştırılmıştır.

ANAHTAR KELİMELEER: Diferansiyel tahrik, gezgin robot, PID algoritması, hedefe ulaşma, engelden kaçma, Bulanık algoritması, Gauss algoritması

ABSTRACT

Master's Thesis

COMPARİSON OF PERFORMANCE OF PID, FUZZY AND GAUSS CONTROLLERS İN BEHAVİOR PLANNİNG OF MOBİLE ROBOTS

Cihangir Han KAZANCI

Inönü University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

64+ x pages

2019

Supervisor: Assoc. Dr. Adnan Fatih KOCAMAZ

Identifying behavior modeling algorithms to be applied to the differential drive mobile robot is one of the most important issue in robotic applications. in this context, it is necessary to characterize the robot movements according to the tasks and environment variables in the working environment (configuration space). In this thesis, navigation algorithms which were applied to a differential drive mobile robot trying to reach target by avoiding from the obstacles were examined.

Using the data from the sensors on the mobile robot, it is ensured that the robot is able to perceive the surroundings and act according to the environment. The data are obtained from eight ultrasonic distance sensors on the mobile robot. Thanks to the ultrasonic distance sensors, the position of the mobile robot according to the obstacles is determined. These data were used as input value in the algorithms used. With these input values, the avoid behavior of the mobile robot is modeled.

In this study, previously developed PID based Sim.I.Am mobile robot simulator has been examined. In this context, Sim.I.Am mobile robot simulator application to

eliminate the deficiencies detected and the algorithm used instead of the results were observed using different algorithms.

In the first part of the study, the optimization of the PID navigation algorithm applied to the mobile robot is explained. Especially the algorithms used according to the environment where the mobile robot moves can be insufficient. In the following parts of the study, it is proposed to configure the navigation controllers of the differential driven mobile robot systems with Fuzzy and Gaussian algorithms.

Sim.I.Am simulation was used to show the effectiveness of the developed algorithm base controller. When the simulation results are examined, it is observed that the algorithms examined and proposed in a simulation test environment where the location of the obstacles change, produce a successful result. Within the scope of the thesis, the performance of the controller obtained by the proposed algorithms is compared with the previous studies.

KEYWORDS: Differential drive, mobile robot, PID algorithm, reach the target, avoid obstacle, Fuzzy algorithm, Gaussian algorithm

TEŞEKKÜR

Mobil Robotların Davranış Planlamasında PID, Bulanık ve Gauss Denetçilerinin Başarımlarının Karşılaştırılması konuları yüksek lisans tezi olarak çalışılmıştır.

Tez çalışması süresince bilgi ve deneyimlerinden yararlandığım, karşılaştığım zorluklarda beni destekleyen, tez çalışmanın planlanması ve yürütülmesinde yardımlarını esirgemeyen tez danışman hocam Dr. Öğr. Üyesi Adnan Fatih KOCAMAZ'a, çalışmalar konusunda ihtiyacım olduğunda yardımını esirgemeyen hocam Öğr. Gör. Dr. Emrah DÖNMEZ'e, ders dönemim ve tüm çalışmalarım boyunca bilgi ve tecrübeleriyle bana yol gösteren tüm hocamlarıma çok teşekkür ederim. Tüm eğitimim süresince bana destek olan aileme minnet borçlu olduğumu belirtmek isterim.



Cihangir Han KAZANCI
Yüksek Lisans Öğrencisi

İÇİNDEKİLER

Sayfa No

ÖZET.....	i
ABSTRACT.....	iii
TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
SİMGELER VE KISALTMALAR.....	vii
ŞEKİLLER LİSTESİ.....	viii
ÇİZELGE LİSTESİ.....	x
1. GİRİŞ.....	1
2. LİTERATÜRDEKİ MOBİL ROBOT KONTROL ÇALIŞMALARI..	2
3. PROJEDE KULLANILAN UYGULAMA ALTYAPISI.....	8
3.1. Robot Platformları ve Laboratuar Ortamı ve Materyaller.....	8
3.2. Yöntem ve Kinematik.....	10
4. SİMULASYON ORTAMI VE DENETÇİ TASARIMI.....	14
4.1 PID.....	15
4.1.1. PID kontroller.....	16
4.1.2. On-off kontrol.....	17
4.1.3. P (oransal) kontrol.....	18
4.1.4. PI (oransal+integral) kontrol.....	18
4.1.5. PID (oransal+integral+türev) kontrol.....	19
4.1.6. PID birim basamak performans ölçütleri.....	20
4.1.7. PID parametrelerinin hesaplanması ve denetçi tasarımı.....	22
4.1.8. Denetçi algoritmalarının karşılaştırması.....	31
4.2 Bulanık Mantık (Fuzzy).....	34
4.2.1. Bulanık lojik sistemi parametrelerinin belirlenmesi ve denetçi tasarımı.....	37
4.2.2. Giriş – çıkış üye fonksiyonlarının belirlenmesi.....	39
4.2.3. Denetçi kurallarının oluşturulması.....	43
4.2.4. Bulanık tabanlı İnocop simülasyon uygulaması.....	45
4.2.5. Denetçi algoritmalarının karşılaştırması.....	46
4.3. Gauss Denetçi Tasarımı.....	49
4.3.1. Gaussian algoritması.....	49
4.3.2. Engellerden kaçmak için Gauss denetçi kullanılması.....	50
4.3.3. Hedefe ulaşmak için gauss denetçi kullanılması.....	53
5. SONUÇ.....	58
6. KAYNAKLAR.....	61
ÖZGEÇMİŞ.....	64

SİMGELER VE KISALTMALAR

μ	Beklenen Bağılım Değerir
σ	Standart Sapma
σ^2	Varyans
θ	Açı Değeri
Δt	Zamandaki Değişim
V	Çizgisel Hız
V_r	Sağ Teker Çizgisel Hız
V_l	Sol Teker Çizgisel Hız
ω	Açısal Hız
ω_r	Sağ Teker Açısal Hız
ω_l	Sol Teker Açısal Hız
E_A	Robot ile Hedef Arasındaki Açı
E_p	Robot ile Hedef Arasındaki Mesafe
D, L	Robotun İki Tekerleği Arasındaki Mesafe, Dingil Mesafesi
K_p	Oransal Katsayı
K_i	İntegral Katsayı
K_d	Türev Katsayı
A_{max}	En Büyük Aşım
$u(t)$	Birim Basamak Yanıtı
$e(t)$	Hata Değeri
$r(t)$	Referans Değeri
$B(t)$	Geri Besleme Değeri
G	Sürekli Zaman Transfer Fonsiyonu Değeri
c	Sürekli Zaman PID denetçi Değeri
T_r	Yükselme Zamanı
T_y	Yerleşme Zamanı
Step Response	Birim Basamak Tepkisi
Time	Zaman
Amplitude	Genlik
IR	Kızıl Ötesi Sensör
IR _x	Robot ile Engel Arasındaki Mesafe

ŞEKİLLER LİSTESİ

Şekil 1. Khepera-IV robot.....	8
Şekil 2. İnocop robota ait sensör durumları	9
Şekil 3. Khepera-IV kinematik özelliği	10
Şekil 4. Gezgin robot kinematığı	12
Şekil 5. Sim.I.Am uygulaması	14
Şekil 6. Sim.I.Am hareket davranış akış şeması	15
Şekil 7. PID denetleyici	17
Şekil 8. On-Off kontrol	18
Şekil 9. P(Oransal) kontrol	19
Şekil 10. PI(Oransal + Integral) kontrol	19
Şekil 11. PID(Oransal + Integral + Türev) kontrol.....	20
Şekil 12. Birim basamak cevabı performans ölçütü parametrelerinin temsili gösterimi	21
Şekil 13. $K_p=1$, $K_i=0$, $K_d=0$ değerlerine göre PID birim basamak cevabı	24
Şekil 14. $K_p=9$, $K_i=0$, $K_d=0$ değerlerine göre PID birim basamak cevabı	24
Şekil 15. $K_p=40$, $K_i=3$, $K_d=2$ değerlerine göre PID birim basamak cevabı	25
Şekil 16. Denetçi kullanılmamış ve PID denetçiler ile kontrol edilen sistemin birim basamak tepkileri	26
Şekil 17. Denetçi kullanılmamış ve PID denetçiler ile kontrol edilen sistemin birim basamak tepkileri	26
Şekil 18. PID parametrelerinin belirlenmesinde kullanılan akış diyagramları	27
Şekil 19. Sim.I.Am robotun engele çarpma durumu	29
Şekil 20. PID Birim basamak yanıtı ($K_p=4$, $K_i=0.01$, $K_d=0.01$)	29
Şekil 21. Geliştirilen İnocop'un engelden kaçışı	30
Şekil 22. PID Birim basamak yanıtı ($K_p=6$, $K_i=0.08$, $K_d=0.09$)	30
Şekil 23. PID Birim basamak yanıtı kıyaslanması.....	31
Şekil 24. Simülasyon ortamı	32
Şekil 25. İnocop robotun izlediği yol.....	32
Şekil 26. Bulanık mantık tabanlı sıcaklık üye fonksiyonu	34
Şekil 27. Bulanık lojik sistem mimarisi	36
Şekil 28. Robot hareketinde kullanılan akış diyagramı	38
Şekil 29. 8 giriş ve 1 çıkış ile 28 kurallı Bulanık denetçiler.	39
Şekil 30. Giriş üye fonksiyonları (engele olan mesafe).....	40
Şekil 31. Çıkış üye fonksiyonları (açısal hız, ω)	40

Şekil 32. Beş girişli Trmf üye fonksiyonu	42
Şekil 33. Beş girişli Trimf üye fonksiyonu	42
Şekil 34. Beş girişli Gauss üye fonksiyonu	42
Şekil 35. Kural seti çıktıları	44
Şekil 36. Yüzer görünüm grafiği (surface)	44
Şekil 37. Simülasyon ortamı	46
Şekil 38. Bulanık tabanlı İnocop robotun izlediği yol	46
Şekil 39 Gauss eğrisi.....	51
Şekil 40. Açık – zaman değişimi	52
Şekil 41. ω , hesaplanması	52
Şekil 42. Radian – yarıçap ilişkisi	52
Şekil 43. İnocop robot ile engelin konumu	52
Şekil 44. Gauss Eğrisi	54
Şekil 45. Simülasyon ortamı	55
Şekil 46. PID tabanlı İnocop robotun izlediği yol	55
Şekil 47. Gauss tabanlı İnocop robotun izlediği yol	55
Şekil 48. Gauss tabanlı İnocop robotun izlediği yol	56
Şekil 49. Gauss tabanlı İnocop robotun izlediği yol	56

ÇİZELGE LİSTESİ

Tablo 1. PID Denetçide parametrelerin bağımsız etkileri	23
Tablo 2. PID katsayılarına göre değişen ortamlarda hedefe ulaşma süreleri	28
Tablo 3. Mobil robotların değişen ortamlara göre hedefe ulaşma süreleri.....	32
Tablo 4. Klasik – Bulanık Mantık farkları	35
Tablo 5. Açısal hız için bulanık mantık kurallar	41
Tablo 6. Giriş üye fonksiyonlarının tanımı	43
Tablo 7. Çıkış üye fonksiyonlarının tanımı	43
Tablo 8. Kural tabanı.....	45
Tablo 9. Engel tabanlı hedefe ulaşma süreleri	48
Tablo 10. Engel tabanlı hedefe ulaşma süreleri	57
Tablo 11. Engel tabanlı hedefe ulaşma süreleri	58

1. GİRİŞ

Günümüzde robotik ve kontrol bilimi alanındaki gelişmeler endüstri, hizmet, askeri, sağlık vb. farklı alanlarda hayatımızı olumlu yönde etkilemektedir. Robotik ve kontrol bilimi alanındaki bu gelişmelere paralel olarak mobil teknolojiler ve robotik sistemlerde de kayda değer ilerlemeler gözlemlenmektedir. Bugün bu teknolojiler evlerimizdeki beyaz eşyalardan cep telefonlarına, hava ve uzay sanayisinde kullanılan otonom araçlardan radyo ve uydu cihazlarına kadar pek çok değişik alanda kendilerine yer bulmaktadır.

Mobil araçlara duyulan ihtiyaç ve bu araçlar üzerine yapılan araştırmaların olumlu sonuç vermesi, gelişmeleri etkileyen en önemli faktörlerdendir. Bu kapsamda insan tarafından yapılan görevlerin mobil sistemlerin yardımıyla gerçekleştirilmesi önemli bir husustur.

Mobil robot teknolojileri alanındaki gelişmelerle birlikte farklı denetçi teorileri ortaya çıkmıştır. Kontrol teorilerinin üzerine yoğunlaştığı alanlardan biri, geliştirilecek sistemlerin istenilen özelliklerde çalışmasını sağlayan denetçilerin analizi ve tasarımıdır [1]. Geliştirilen denetçilerin kullanıldığı alanlardan biri de gezgin robot teknolojileridir. Gezgin robotlar üzerlerinde bulunan sensörler sayesinde çevrelerindeki değişimlerden anlık olarak haberdar olabilmektedirler. Otonom bir gezgin robotun en önemli parçası karar verme mekanizmasıdır. Karar verme mekanizması, ivme-ölçer, jiroskop, kodlayıcılar gibi içsel sensörlerden ve menzil sensörü, kızılötesi sensör ve termal kamera gibi dışsal sensörlerden gelen verileri işleyerek bir sonuç üretir [2]. Sensörlerden gelen verilerin doğruluğu, gerçek-zamanlı olması, elde edilen verilerin işlenmesi için gerekli işlemsel güç, karar vermeyi etkileyecek en önemli unsurlardır [3].

Bu çalışma çerçevesinde, PID, Fuzzy ve Gauss algoritmaları ayrı ayrı kullanılarak bir gezinim denetçinin tasarımı önerilmiştir. Önerilen yöntem ile geliştirilen denetçi literatürdeki mevcut yöntemlerle kıyaslanmış ve simülasyon ortamında yapılan testlerin çıktıları ile geliştirilen denetçinin etkinliği gösterilmiştir.

Tez çalışmasının ikinci bölümünde, literatürde önerilen bazı robot kontrol çalışmalarına yer almaktadır. Üçüncü bölümde sistemin genel yapısına, projede kullanılan uygulamanın altyapısına ve sistemin kinematik yapısına yer verilmiştir. Bu bölümde denetçiler tasarımında kullanılan yöntemlere de yer verilmiştir. Dördüncü bölümde kontrol programının yazılımı ve çalışmada esinlenilmeş olan Sim.I.Am

projesi ile ilgili bilgiler verilmiştir. Bu bölümün takip eden kısımlarında PID denetçi üzerinde yapılan optimizasyon çalışması detaylı olarak sunulmuştur. İlgili bölümün ilerleyen kısımlarında Bulanık tabanlı ve Gauss tabanlı denetçi tasarımları, çalışma prensipleri ve kullanılan yöntemler ayrı ayrı başlıklar halinde anlatılmıştır. Tez çalışmasının sonuç bölümünde ise yapılan çalışmalarda kullanılan yöntemler, geliştirilen denetçilerin simulasyon testleri sonucu elde edilen çıktılar ve sonraki çalışmalara rehberlik edebilecek öneriler üzerine yoğunlaşmıştır.



2. LİTERATÜRDEKİ MOBİL ROBOT KONTROL ÇALIŞMALARI

Gezgin robotun hedefe doğru ilerlerken takip edeceği yolun planlanması ve bu yolun ortamdaki engellerden kaçınarak takip edilmesi literatür çalışmalarında önemli yer tutmaktadır [4]–[6]. Bu alanda yapılan çalışmalarda alınan mesafe ve hedefe ulaşma süresi gibi parametrelerin optimizasyonu, hareket ortamında araç hareketini engelleyici veya manevrasını zorlayıcı engeller olmaması gibi farklı kriterler göz önüne alınmaktadır [7]–[9]. Sistemlerin kullanılacağı ortamlara göre literatürde birbirinden farklı yöntemler geliştirilmiş veya farklı yöntemler bir arada hibrit olarak kullanılmıştır [6].

Gezgin robot kontrolü üzerine E. Deniz vd. tarafından [10] yüksek lisans tezi olarak çalışılmıştır. Gezgin robot, bir başlangıç konumdan hareketine başlatılmış ve üzerindeki donanımlar aracılığı ile elde ettiği ölçüm sonuçlarının değerlendirilmesi ile hareketine karar vermesi sağlanmıştır. Robotun ilk olarak konumu tahmin edilmeye çalışılmış daha sonra da elde edilen bu sonuç ve duyarga ölçümleri bir arada kullanılarak ortamın haritası çıkarılmıştır. Gezgin robot kontrolü için iki farklı yöntem uygulanmıştır.

Robotun otomatik kontrolü için uygulanan ilk yöntem engelden kaçınma hareketi ile robot kontrolü olmuştur. Bu yöntemde ilk olarak robot belirlenen hedefe doğru hareketine başlatılmıştır. Robotun hareketi sırasında sensörler aracılığı ile bir engele yaklaşıldığı tespit edildiğinde, robot engelden kaçınma davranışına girmesi sağlanır. Robotun hareketi esnasında konumu sürekli olarak hesaplanmış ve sensör ölçüm sonuçları kullanılarak ortamın haritası çıkarılmaya çalışılmıştır.

Literatürde yer alan yöntemin eksikliklerini gidererek daha iyi bir denetçi oluşturulması amacıyla, engel takip etme algoritması ise ikinci yöntem olarak kullanılmıştır. Bu yöntemde robot hareketine başladıktan sonra ortamda robotun hareketini engelleyecek bir engel ile karşılaşırsa robotun konumu bu engele paralel hale getirilmiştir. Robotun, karşılaşılan bu ilk engeli ve ardından gelen engelleri takip ederek hareketine devam etmesi sağlanmaya çalışılmıştır.

Bu konuda literatürde birçok çalışma bulunmaktadır. Örneğin; Böcek algoritması üzerine M. Yüksel vd. tarafından [11] gerçekleştirilen çalışmada hedefin konumu mobil robota verilmiştir. Mobil robotun konumu bilinen hedefe “Böcek Algoritması” kullanarak ulaşması amaçlanmıştır. Bu algorithmada, robotun engele yaklaştığında

engelin etrafında dolaşarak hedefe hareket etmesi sağlanmıştır [12]. Böcek algoritması, böcek1 ve böcek2 [13]–[15] olmak üzere 2 başlık altında toplanmıştır.

Çınar vd. [16] tarafından yapılan çalışmada A* algoritması üzerine yoğunlaşmıştır. A* algortması, en kısa yolu bulmak için kullanılan sezgisel algoritmalarından birisidir. Çalışmada, yol planlaması yapılırken ortamda bulunan engellerini konumunun bilindiği varsayılmıştır. Rota planlaması için A* planlama yöntemi, hareket ortamının bilinen özelliklerine göre kurgulanmıştır [16]. Hedefe ulaşmak için harcanan maliyetin tahmin edilememesi durumu geliştirilen A* yönteminin en iyilenmiş olduğunu ifade etmektedir [17]. Yapılan bu çalışmada planlama yöntemi olarak potansiyel alan yaklaşımı yöntemi kullanılmıştır.

Potansiyel alan yaklaşımı, literatürde robotların belirlenen hedefe ulaşmaları için kullanılan yöntemlerden biridir. Klasik alan yaklaşımının temelini, gezgin robotun hareket ettiği ortamda itici ve çekici kuvvetler olduğunun varsayılması oluşturur. Hedef, çekici vektör oluşturarak robotun kendine çekilmesini sağlanmaktadır. Hareket ortamında bulunan engeller ise itici vektörler oluşturarak robotun engellerden kaçmasını amaçlamaktadır [16]. Gezgin robotun, ortamda olduğu varsayılan vektörlerin bileşkesi ile elde edilen vektör doğrultusunda hareket etmesi amaçlanmıştır. Potansiyel alan yöntemi pasif olarak yol planlama için kullanılabileceği gibi dinamik şekilde hedefe gitme ve engelden kaçınma davranışlarının modellenmesinde de kullanılmaktadır.

Klasik Potansiyel Alan yaklaşımında itici kuvvetlerin bileşkesinin, çekici kuvvetlerinin bileşkesinden büyük olması engellerin yakınındaki hedeflere ulaşamaması problemini ortaya çıkarmıştır [18]. Bu sorunu çözmek ve hedefe ulaşabilmek için hedef noktasında olduğu varsayılan çekici kuvvet vektörünün yanında minimum bir çekici kuvvet vektörünün de oluşması amaçlanmıştır.

Bir önceki çalışmaya benzer şekilde Dönmez vd. tarafından [19] yapılan çalışmada uyarlanabilir potansiyel alan yaklaşımı ile görsel tabanlı yol planlama üzerine çalışılmıştır. Potansiyel alana ait parametrelerin, robotun engellere olan mesafesi göz önüne alınarak güncellenmesi sağlanmıştır. Çalışmanın sonucunda uyarlanabilen ve uyarlanamayan potansiyel alan metotlarının güçlü yanları ortaya konulmuştur.

Yüksel vd. tarafından [11] üzerinde çalışılmış olan algoritmada, birbirine bağlı duvarların olduğu ve hareket ortamında ayrı bir duvar bütünü bulunmayan ortamlarda sağladığı başarıdan dolayı tercih edilmektedir. Bu algoritmada gezgin robot kendisine

köşe olarak belirlediği duvarı referans alarak ilerlemeye başlar. Mobil robot bu sayede duvar boyunca ilerleyerek çıkışa ulaşır.

Dönmez vd. tarafından [20] yapılan çalışmada görüntü tabanlı karar ağacı yöntemi kullanılarak yeni bir gezgin robot kontrol modeli geliştirilmiştir. Bu yöntemde hedef ile robot arasında sanal bir üçgen oluşturulmuştur. Karar ağacı yaprakları denetçisi ile sağ ve sol tekerlek hızları belirlenmiştir.

Parlaktun vd. tarafından [21] yapılan çalışmada yol planının dik doğrusal hareketlerden oluşmasını sağlayan sekiz farklı komşu-disk önceliklendirme örüntüsünü taban alan genetik algoritma önerilmiştir. Kapsama alanı, örtüşen disklerle bölünerek tüm ortamın kapsanması sağlanmıştır. Kapsamaya tabi olmayan engel oluşturan alanlar dolu disklerle gösterilir. Her diskin yarıçapı, robotun sensör menziline eşittir [21]. Yarıçap robotun fiziksel boyutundan daha büyüktür. Ortamdaki engeller gri disk ile gösterilirler. Bu model ile robot tüm disklerin ortasından geçerse, ortam tam kapsanmış olur. Böylece, algılayıcı tabanlı kapsama problemi, tüm disklerin merkezini gezen bir yol planlamaya dönüşür. Bu yaklaşımın amacı, dik doğrusal hareketleri desteklerken disklerin birden çok gezilmelerini (veya kapsanmalarını) en küçüktür.

Dirik vd. tarafından [19] görsel tabanlı kontrol sistemleri ve bulanık denetçi methodları birlikte çalışarak çarpışmasız bir yol oluşturulmaya çalışılmıştır. Çalışmada robot kontrolünü sağlamak için altı adet sensörden gelen veriler kullanılmıştır.

Saclolo vd. tarafından [22] yapılan çalışmada bulanık denetçi ile birlikte makine görme sistemi birlikte kullanılmıştır. Bir tepe kamerası kullanılarak robotun konumu ve hareketi kontrol edilmiştir.

Wang vd. tarafından [23] yapılan çalışmada mobil bir robotun performansının iyileştirilmesi için akıllı denetçiler kullanılmıştır. PID ve bulanık denetçi kullanılarak hibrit bir denetçi oluşturulmuştur. Çalışma sonucunda oluşturulan bu hibrit denetçi ile mobil robotun dinamik yanıtının iyileştirilebilir olduğu görülmüştür.

Dönmez vd. tarafından [2] yapılan çalışmada Gauss tabanlı yeni bir denetçi oluşturulması amaçlanmıştır. Gezgin robota ait tekerleklerin hızlarının genel Gauss fonksiyonu kullanılarak hesaplanması sağlanmıştır. Çalışmanın sonucunda oluşturulan yeni denetçi PID ve bulanık-PID gibi klasik denetçilerle kıyaslanmıştır.

Allagui vd. tarafından [23] yapılan çalışmada üç bulanık mantık PI denetleyicisi kullanılarak bir denetçi geliştirilmesi amaçlanmıştır. Bu denetçi hedefe ulaşmak ve

yörünge takibi amacıyla uyarlanmıştır. Her bir denetleyicinin klasik bir PI denetçi, iki giriş ve iki çıkışlı bulanık çıkarım sistemi tasarlamaya yönelik olması sağlanmıştır.

Dewivedi vd. tarafından [24] yapılan çalışmada mobil robot denetçisi için PID ve Destek Vektör Makinesiden (SVM) oluşan hibrit bir denetleyici önerilmiştir. Önerilen bu hibrit tekniğinde mobil robotun hızı PID denetleyiciyle, dönme açısı ise destek vektör makinesi ile kontrol edilmiştir. PID parametrelerinin global optimal değerlerinin bulunması için Parçacık Sürü Optimizasyonu (PSO) tekniği kullanılmıştır.

Dirik vd. tarafından [20] yapılan çalışmada birbirinden bağımsız hareket eden tekerleklerden gelen verilerin bulanık kuralları kullanılarak anlamlandırılması üzerine yeni bir görsel denetçi modeli geliştirilmiştir.

3. PROJEDE KULLANILAN UYGULAMA ALTYAPISI

Bu bölümde sistemin çalışma prensibi anlatılmıştır. Gezgin robot sisteminin tasarlanmasında kullanılan bölümler ve bu bölümlerin birbiriyle olan ilişkileri anlatılarak gezgin robot platformu verilmiştir.

Otonom araçlarda teorik olarak yeni yöntem geliştirmenin yanı sıra mevcut yöntemlerin uygulamaya aktarılması da kendi içinde zorluklar barındırmaktadır. Uygulamanın ilk aşaması geliştirilen yöntemlerin algoritma seviyesinde test edilmesini gerektirmektedir. Burada başarılı sonuçlar elde edildikten sonra, geliştirilen yöntemler otonom sistemlerin benzetiminin yapıldığı benzetim ortamlarında test edilir. Ancak bu ortamlarda yapılan testler gerçek sistemleri belli ölçüde temsil etse de yetersiz kalmaktadırlar. Bu yüzden üçüncü adım olarak ilk örnek veya gerçek sistemler üzerinde testlerin yapılması yer almaktadır. Çalışmada ilk iki adım üzerinde yoğunlaşmıştır.

Tez çalışması kapsamında Sim.I.Am gezgin robot uygulaması incelenmiş ve Sim.I.Am simülasyon ortamı kullanılmıştır. Sim.I.Am projesi, Dr. Magnus Egerstedt ve ekibi tarafından yedi haftalık kurs eğitimi olarak verilmesi amaçlanan, açık kaynaklı bir çalışmanın sonucu olarak ortaya çıkmış bir çalışmadır. Tez kapsamında eğitimin beşinci haftası sonunda geliştirilmiş olan uygulama üzerine yoğunlaşmıştır. Sim.I.Am otonom bir gezgin robot uygulamasıdır. Uygulamda kullanılan mobil robot, mesafe algılamalı IR sensörleri ve tekerlek kodlayıcıları olan diferansiyel tahrikli gezgin bir platformdur. Sim.I.Am, hem simülasyonlu hem de gerçek mobil robotlara uygulanabilen denetçi ve algoritmaların uygulanmasını ve tasarımını kolaylaştıran MATLAB ve Simulink için mobil robot simülatörüdür. Sim.I.Am gezgin robotun hareket davranış modeline “Simulasyon Ortamı Ve Denetçi Tasarımı” başlığı altında yer verilmiştir.

3.1 Robot Platformları ve Laboratuvar Ortamı ve Materyaller

Önerilen denetçi metodu çeşitli bileşenlerden oluşmaktadır. Fiziksel deney ortamı, donanım (işleme birim, robot ve sensörler), tespit ve denetçi yazılımı denetçi metodunun ana bileşenleridir. Nesne algılama, nesne izleme, parametre hesaplama ve kontrol algoritmaları ana denetçi yazılımı bileşeninin modülleridir.

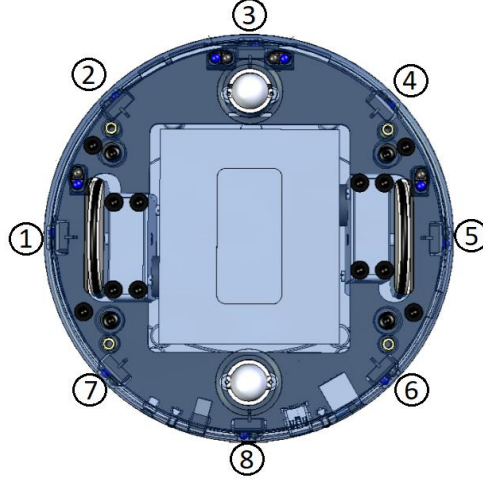
Gezgin robotlarda kullanılan hareketli ve hareketsiz teker sayısı hareket planını etkileyen önemli bir konu olmuştur. Yapılan çalışmada her tekeri ayrı hareket eden iki

tekerlekli robot kullanılmıştır. Hareket planlamasında sağladığı avantaj ve kolaylıklardan dolayı çalışmada bu diferansiyel tahrikli robot seçilmiştir. Buna göre mobil robot, bir gövde ve gövdenin merkezinden geçen düz bir hattın iki yönüne yerleştirilmiş iki tekerlekten oluşmaktadır. Bu tipteki robotlarda sağ ve sol tekerlerin açısal hızları olmak üzere iki adet denetim değişkeni bulunmaktadır. Ayrıca mobil robotun, x konumu, y konumu ve gidiş yönü θ olmak üzere üç adet durum değişkeni vardır.

Çalışmada geliştirilen denetçiler Dr. Magnus Egerstedt ve çalıştığı proje ekibi tarafından geliştirilen Matlab tabanlı bir simülasyon ortamında test edilmiştir. Matlab ortamında geliştirilen Sim.I.Am simülatör ortamı, kullanılan denetçinin etkinliğini göstermek amacıyla çalışmaya özel olarak uyarlanmıştır. Yapılan çalışma, PID denetçisi kullanan gezgin robota uygulanan PID algoritmasının geliştirilmesini, tez çalışması kapsamında önerilen ve geliştirilen Bulanık ve Gauss denetçilerinin PID algoritmasının yerine kullanılmasının denenmesini ve elde edilen çıktıların karşılaştırılmasını içermektedir. Simülatör ortamında İnocop adı verilen gezgin robot kullanılmıştır. İnocop robot Khepera-IV tipi gezgin robotun fiziksel ölçülerine ve hız limitlerine göre güncellenmiştir. Şekil 1’de Khepera-IV robotun fiziksel özellikleri gösterilmiştir [25]. Şekil 2’de İnocop robot ve üzerindeki sensör yerleri gösterilmiştir.

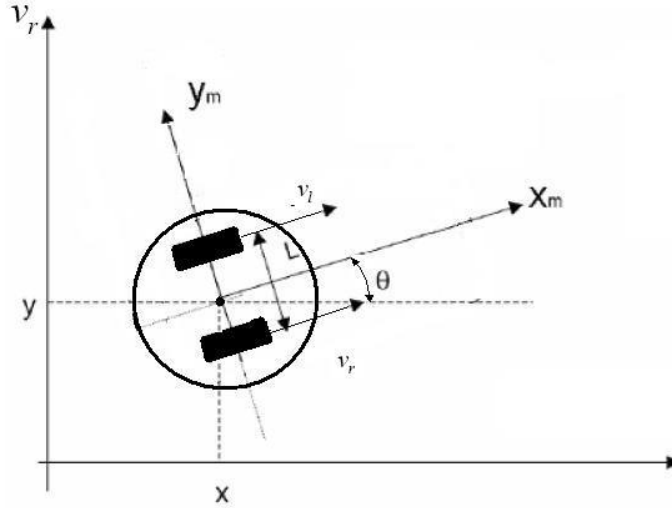


Şekil 1. Khepera-IV robot



Şekil 2. İnocop robota ait sensör durumları

Khepera-IV, iki bağımsız DC dişli motoru tarafından kontrol edilen, birbirine denk iki paralel tekerleğe sahiptir. Robotun kütle merkezi, tekerleklerin eksenine diktir (L). Mobil robotun kütle merkezi (x, y) tekerlekleri (L) birbirine bağlayan eksenin ortasında bulunur. Khepera-IV, Linux çekirdeğinde koşan 800MHz ARM Cortex-A8 işlemcisine sahiptir. 512 MB dahili RAM kapasitesi bulunmaktadır. Tekerleklerin en yüksek hızı 1 m/s, en düşük hızı ise 0.003m/s'dir. Hız birimleri, her 10ms'de tekerlek hızlarını ölçerek robotun konumunu belirler. Robot üzerinde bulunan sekiz adet kızılötesi yakınlık sensörü ve 25 cm'ye kadar ölçüm yapabilen ortam ışığı sensörleri bulunmaktadır. Çizgi takip uygulamaları ve düşmeden kaçınma için dört adet kızılötesi yere yakınlık sensörü, 25 cm – 2 m aralığını ölçebilen beş adet ultrasonik sensör, 3 eksenli ivmeölçer ve 3 eksenli jiroskop robotun dahili sensörlerindendir. Ayrıca Khepera-IV robot, 30FPS çekim yapabilen 752x480 pixel entegre bir kameraya sahiptir. Robot iletişim için iki adet USB girişine, Wifi ve Bluetooth özelliklerine sahiptir. Robotun gövde çapı 140mm, yüksekliği 58mm ve tekerlek çapı 42mm'dir. Şekil 3'de Khepera-IV robotun kinematik özellikleri gösterilmiştir [26][27].



Şekil 3. Khepera-IV kinematik özelliği

Bu robot tipinin modellenmesi ve kontrol edilmesi diğer robot tiplerine göre daha kolaydır. Çünkü iki tekerleğin açısal hızları (ω_l ve ω_r) bağımsız olarak kontrol edilebilir. İnocop robota ait beş parametre ile iki sabit değer vardır. Sabit değerler, tekerlek yarıçapı ile dingil mesafesidir. Parametreler ise tekerleklerin minimum ve maksimum hızları, encoder çözünürlüğü, robotun engele olan konumu ile robotun simülasyon ortamındaki konumundan oluşmaktadır.

3.2 Diferansiyel Tahrikli Mobil Robot Kinematığı

Diferansiyel tahrikli gezgin robot uygulamalarında gezgin robotların hedefe ulaşmasındaki etkinliğini kullanılan gezinim algoritmaları ve denetçiler belirler. Gezgin robot teknolojilerinde robotların hedefe ulaşmasında çevre en önemli faktörlerden biridir. Robotun hareket ettiği ortam, hedefe ulaşmak için kullanılacak hareket algoritmasını ve kontrolörünü belirler. Bu yüzden literatürdeki çalışmalarda değişik algoritma ve denetçiler kullanılmıştır. Algoritmalarındaki bu farklılığın temelini robotun kullanılacağı ortam koşulları oluşturmuştur.

Gezgin robotlar mevcut hareket kabiliyetleri, manevra özellikleri ve kullanılacak ortama göre kendi içlerinde ayrılmıştır. Gezgin robotların hedefe ulaşması için dinamik veya statik olmak üzere iki yöntemden biri kullanılmıştır [28]. Statik yöntemde, ortamda bulunan engellerin konumu bilinmektedir. Denetleyicide kullanılan algortimalarda engel konumları tanımlanarak bir yol planlaması oluşturulur. Gezgin robot kendisine verilen bu yolu takip ederek hedefe ulaşır. Dinamik yöntemde, gezgin robota sadece hedefin bulunduğu konum tanımlanır. Gezgin robot farklı algoritma tabanlı denetçiler kullanarak karşılaştığı engellerden kaçınır ve hedefe

ulařır. Bu tez alıřmasında hem engellerden kamak hem de hedefe ulařmak iin dinamik yntem kullanılmıřtır.

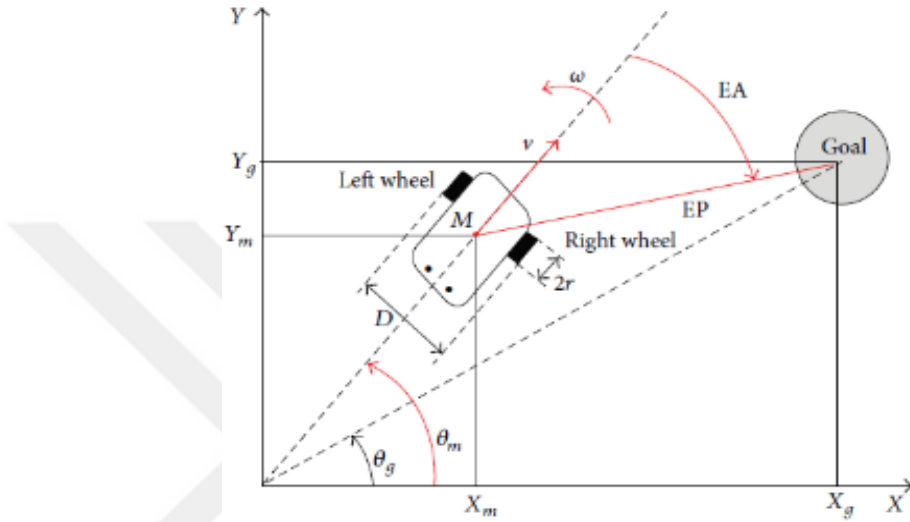
Tez kapsamında kullanılan simlatr, Sim.I.Am [29] simlatr zerinde yapılan geliřtirmeler sonucu ortaya ıkmıřtır. Bu alıřma drt fazdan oluřturulmuřtur. Birinci fazda, Sim.I.Am projesi incelenmiřtir. Robotun engel konumlarının deėiřtiėi ortama uyum saėlaması iin PID optimizasyonu yapılmıřtır. İkinci fazda, birinci fazın geliřtirilmesi zerine yoėunlařılmıřtır. Bu fazda, hedefe ilerlerken PID denetiler kullanılmaya devam edilmiřtir. Engellerden kamak iin ise Bulanık tabanlı yeni bir denetiler tasarlanmıřtır. nc fazda, engellerden kamak iin Gauss algoritması tabanlı yeni bir denetiler tasarlanması amalanmıřtır. alıřmanın son fazında, hem engellerden kamak hem de hedefe ilerlemek iin Gauss algoritmasının kullanılması zerine bir denetiler tasarlanmıřtır. Kullanılan yeni denetiler ile robotun hedefe bařarılı bir řekilde ulařması hedeflenmiřtir.

İnocop gezgin robotun hedefe ulařmasında ve engelleri ařmasında ki bařarısını etkileyen parametreler vardır. İlk parametre, gezgin robota uygulanacak olan algoritmaların belirlenmesi ve robotik sistemlerin tasarlanmasıdır. Robotun kullanılacaėı ortamın fiziksel řartları denetiler olarak kullanılacak algortimanın belirlenmesini etkileyen en nemli unsur olmuřtur. Diėer parametre, tekerleklerin maksimum ve minimum hızlarıdır. Bu parametreler sayesinde kullanılan algoritmalar ile mobil robotun manevra kabiliyeti iyileřtirilebilmektedir. nc parametre, engellere olan mesafedir. Yakınlık sensrleri ile İnocop'un engele olan mesafesi takip edilmiř ve engele yaklařıldıėında robotun engelden kaması saėlanmıřtır. Drdnc parametre, encoder znrlėdr. Bu parametre ile robotun mevcuttaki yeri ve hedefe olan mesafesi ile robotun aldıėı mesafe hesaplanmıřtır. Encoder znrlė sayesinde tekerin bir tam tur dnmesine raėmen gezgin robotun bir tam tur kadar mesafe alıp almadıėı tespit edilmektedir. zellikle kaygan zeminlerde bu tip problemlerle karřılařılmaktadır. Ayrıca tekerin yerle temasını engelleyecek bir dıř etken de bu probleme neden olabilmektedir. Bu problemin stesinden gelebilmek iin encoder znrlė olduka byk neme sahiptir.

İnocop robotun tekerlek yarıapı ve dingil mesafesi alıřmada kullanılan sabit deėerleri oluřturmuřtur. Dingil mesafesi ile İnocop'un hedefe doėru ynelme hareketi sırasında hangi tekerin ne kadar yol alacaėı hesaplanarak tekerleklerin dnř hızları hesaplanmıřtır. Bu alıřmada İnocop robot, nceki cmlelerde belirtilen parametreler

ve sabitlerle birlikte uyarlanmaya çalışılmıştır. Gezgin robotun hedefe başarılı şekilde ulaşmasını sağlamak üzerine yoğunlaşmıştır.

Diferansiyel tahrikli bir gezgin robota ait kinematik model Şekil 4'te gösterilmiştir [30]. Bu modelde EP, hedef ile robot arasındaki mesafeyi ifade etmektedir. EA ise, hedef ile robot yönü arasındaki açıdır. Bu model sayesinde robotun hedefe göre konumu belirlenmektedir.



Şekil 4. Gezgin robot kinematiki

Mobil robot kinematikiinde robotun birbirinden bağımsız hareket eden tekerleklerinin çizgisel hızlarının hesaplanması için (3.1), açısal hız ve çizgisel hızın hesaplanması için (3.2) kullanılır.

$$V_R = r\omega_R, \quad V_L = r\omega_L \quad (3.1)$$

$$\omega = \frac{V_R - V_L}{L}, \quad V = \frac{V_R + V_L}{2} \quad (3.2)$$

V_R , sağ tekerle ait çizgisel hız değeri,

V_L , sol tekerle ait çizgisel hız değeri,

V , çizgisel hız değeri,

ω , açısal hız,

ω_R , sol tekerle ait açısal hız değeri,

ω_L sağ tekerle ait açısal hız değeri,

(3.1) ve (3.2) birleştirilerek (3.3) elde edilmiştir:

$$\omega = \frac{r}{L}(\omega_R - \omega_L), \quad V = \frac{r}{2}(\omega_R + \omega_L) \quad (3.3)$$

Buna ilaveten, robot merkezinin dinamik fonksiyonu (3.4 - 3.6) ile tanımlanmıştır:

$$x = V \sin\theta \quad (3.4)$$

$$y = V \cos\theta, \quad (3.5)$$

$$\theta = \omega \quad (3.6)$$

(3.3) ve (3.4 – 3.6) fonksiyonlarına göre (3.7 -3.9) elde edilmiştir:

$$\omega_R = \frac{1}{r}V + \frac{L}{2r}\omega \quad (3.7)$$

$$\omega_L = \frac{1}{r}V - \frac{L}{2r}\omega \quad (3.8)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.9)$$

Kinematik modelleme kullanılara gezgin robota ait konum hesaplama formülleri (3.10) ve (3.11) ile verilmiştir:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \cos\theta_m & 0 \\ \sin\theta_m & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ w \end{bmatrix} \quad (3.10)$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r & \frac{1}{2}r \\ \frac{r}{D} & \frac{-r}{D} \end{bmatrix} \cdot \begin{bmatrix} V_R \\ V_L \end{bmatrix} \quad (3.11)$$

(3.10) ve (3.11) fonksiyonları birleştirilerek (3.12) elde edilmiştir:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \cos\theta_m & 0 \\ \sin\theta_m & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ w \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \frac{1}{2}r \cos\theta_m & \frac{1}{2}r \cos\theta_m \\ \frac{1}{2}r \sin\theta_m & \frac{1}{2}r \sin\theta_m \\ \frac{r}{D} & \frac{-r}{D} \end{bmatrix} \cdot \begin{bmatrix} V_R \\ V_L \end{bmatrix} \quad (3.12)$$

Gezgin robot kinematiğine göre robotun bulunduğu ortamdaki engele göre konumu aşağıdaki formüllerle hesaplanır:

$$EP = \sqrt{D_x^2 + D_y^2} \quad (3.13)$$

$$EA = \theta_m - \theta \quad (3.14)$$

$$D_x = x_g - x \quad (3.15)$$

$$D_y = y_g - y \quad (3.16)$$

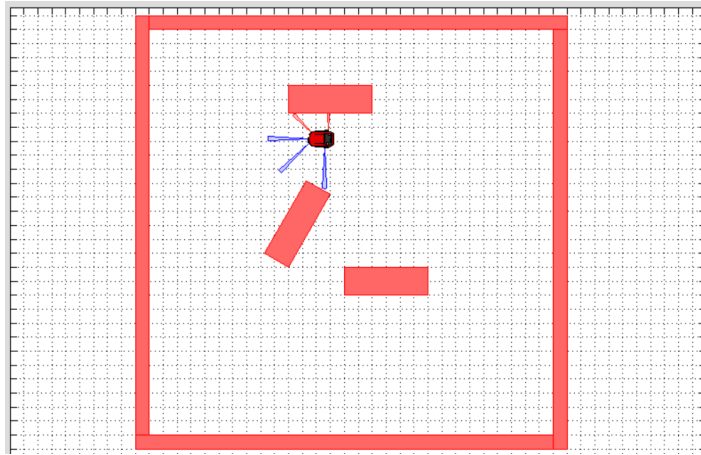
$$\theta = \arctan(D_y, D_x) \quad (3.17)$$

4. SİMULASYON ORTAMI VE DENETÇİ TASARIMI

Bu tezin temelini literatürde PID tabanlı denetçiler kullanılarak yapılmış olan Sim.I.Am [29] gezgin robot simülasyon uygulaması oluşturmuştur. Uygulama bir gezgin robot kinematiğinin oluşturulmasıyla başlayıp, denetçi tasarlanması ve tasarlanan denetçilerin geliştirilmesi şeklinde devam etmektedir. Bu çalışmada özellikle Sim.I.Am ile robotun hedefe gitme ve engelden kaçınma davranışlarını modelleme konularına yoğunlaşmıştır. Uygulamada, PID denetçi kullanan mobil robot engellerden kaçınarak hedefe ulaşabilmektedir.

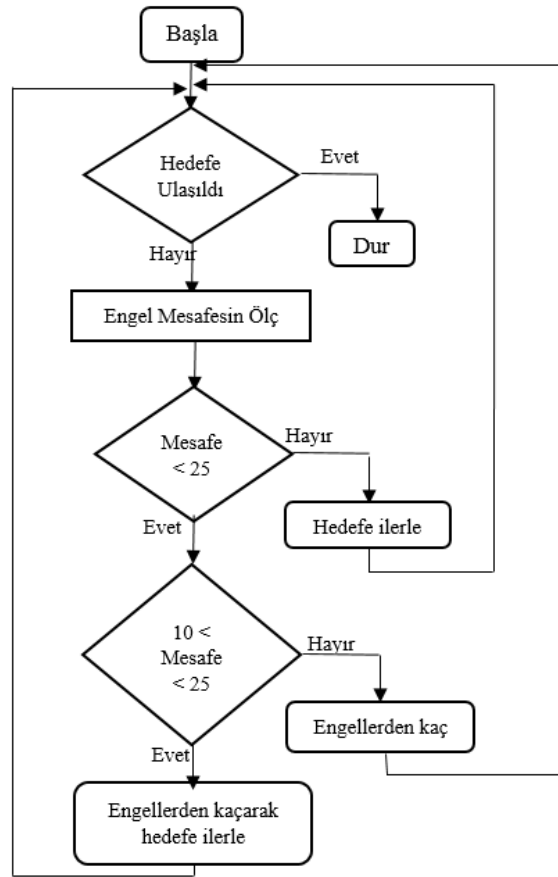
Sim.I.Am, denetçilerin otonom gezgin robotlara nasıl uygulandığını öğrenmek için kullanılan bir robotik simütatördür. Denetçi tasarımlarında, teori ile pratik arasındaki boşluğun doldurulmasına odaklanılmış kütüphane sağlar. Bu mobil robot, mesafe algılayan IR sensörleri, tekerlek kodlayıcıları olan diferansiyel tahrikli gezgin bir platformdur. Simütatör, IR sensörlerinin ve tekerlek kodlayıcılarının denetçi ünitelerinde geri besleme olarak kullanılmalarına ve mobil robotu giriş sinyalleriyle robotun sol ve sağ tekerleklerinin kontrol edilmesini sağlar. Simütatörün temel özelliği, MATLAB ortamında denetçiler tasarlanmasını ve uygulanmasını, bunların simülasyonda test edilmesini ve daha sonra MATLAB ortamın dışında kod uygulamalarına gerek kalmadan gerçek robot üzerinde çalıştırılmasını sağlamasıdır.

Sim.I.Am, hem benzetilmiş hem de gerçek mobil robotlara uygulanabilen denetçilerin ve algoritmaların uygulanmasını ve tasarımını kolaylaştıran MATLAB ve Simulink için açık kaynaklı bir mobil robot simütatördür. MATLAB uygulaması, bir robotun Şekil 5'deki ekran görüntüsünde gösterildiği gibi dağınık bir ortamda engellerden kaçarak hedefe ulaşmasını sağlayan uygulamanın bir demosudur.



Şekil 5. Sim.I.Am uygulaması

Sim.I.Am ortamında modellenen robot başlıca üç davranışı baz almıştır: hedefe ilerleme, engellerden kaçarak hedefe ilerleme ve sadece engellerden kaçma. Bu davranışların belirlenmesinde, kızılötesi mesafe sensörleri tarafından ölçülen, robotu ile engeller arasındaki mesafeler kullanılmaktadır. Robot ile engel arasındaki mesafe 25 cm.'den büyükse mobil robot hedefe doğru ilerlemektedir. Robotun gövde çapı göz önüne alındığında engel ile robot arasında yaklaşık 2 kat mesafe olduğu görülür. Bu yüzden robot herhangi bir engelden kaçma davranışına girmez. Eğer engele olan mesafe 25 – 10 cm arasındaysa engellerden kaçarak hedefe doğru ilerler. Robot ile engel arasındaki mesafe 10 cm ve daha az ise robot sadece engelden kaçmaya odaklanmaktadır. Çünkü robot ile engel arasında robot çapından daha az bir mesafe kalmıştır. Robotun engele çarpmaması için sadece engelden kaçma eğilimine girmesi beklenir. Şekil 6'te Sim.I.Am robotun hareketi esnasındaki davranış akış şeması görülmektedir.



Şekil 6. Sim.I.Am hareket davranış akış şeması

4.1 PID

Endüstriyel sistemlerde kullanılan denetçilerin pek çoğu PID ya da değiştirilmiş PID algoritmalarını kullanmaktadır. Denetçi sistemlere uygulanabilirliğinin sağladığı avantajlar açısından PID denetçiler günümüz endüstrisinde tercih edilmektedir [31].

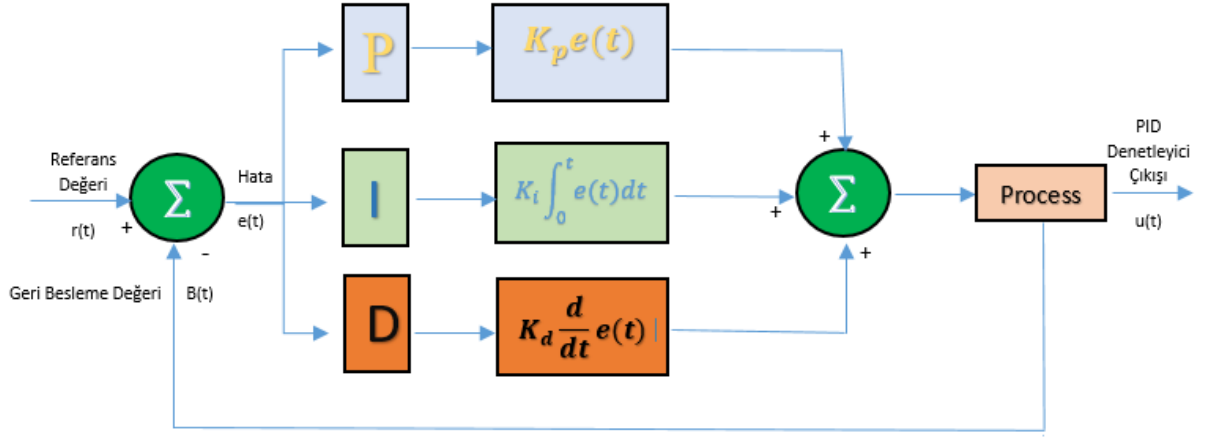
Literatürdeki robot çalışmalarında gezgin robotların engellerden kaçınarak hedefe ulaşmasında çeşitli algoritmalar kullanılmıştır. Algoritmalarındaki farklılığın temelini robotun kullanılacağı ortam olmuştur.

PID katsayıları, PID kontrolör kullanılarak tasarlanmış olan gezgin robot uygulamalarında, robotun engellerden kaçınarak hedefe ulaşmasında ki etkinliğini belirlemektedir. Bu katsayılar ne kadar doğru oluşturulursa robotun hedefe ulaşma başarısının da o kadar arttığı görülmüştür.

Çalışmanın bu kısmında, PID tabanlı denetçiler kullanılarak geliştirilen Sim.I.Am kütüphanesi incelenmiştir. Sim.I.Am simülasyonunda kullanılan denetçi katsayıları iyileştirilerek PID optimizasyonu yapılmıştır. Yapılan optimizasyonların ardından İnocop robota ait fiziksel özelliklere göre simülasyon ortamında kullanılan robotun güncellenmesi sağlanmıştır. Tez kapsamında Sim.I.Am simülatörünü geliştirmek, engel konumlarının değiştiği simülasyon ortamlarında engellerden kaçınarak hedefe ulaşma, farklı denetçiler tasarlayarak simülasyonda kullanmak üzerine yoğunlaşmıştır.

4.1.1. PID denetçiler

PID denetçiler, oransal, integral ve türev devrelerinin güçlü yanlarını tek bir denetçiler içinde birleştiren bir hibrit denetleyicidir. PID denetçileri oluşturan parametreler birer katsayı değeri ile yönetilirler [32]. Bu katsayılar(K_p , K_d , K_i) geliştirilen sistemlerin her birinde farklı değerler ile ifade edilmektedirler. Şekil 7’de bir PID denetçilere ait denetleyici yapısı verilmiştir. Bu denetleyici yapısında da verildiği gibi bir PID denetçinin iç yapısı oransal kazanç (P), integral alıcı (I) ve türev alıcı (D) denetçilerden oluşmuştur [32]. Hata fonksiyonu olarak P (oransal), hata birikimi fonksiyonu olarak I (integral) ve hatanın değişme fonksiyonu olarak da K (türevsel) fonksiyon kullanılmıştır.



Şekil 7. PID denetleyici

PID denetleyicinin çıkışı veya denetim yasası (4.1) ile ifade edilmiştir:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (4.1)$$

Bu denklemde $e(t)$ hata değerini, $u(t)$ çıkış değerini göstermektedir. Hata değeri ise (4.2) ile hesaplanmıştır:

$$e(t) = R(t) - B(t) \quad (4.2)$$

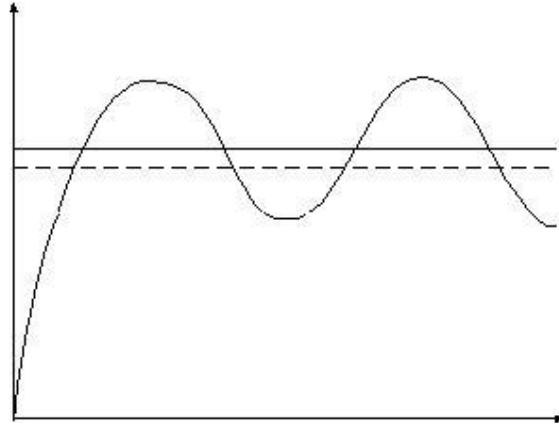
İntegral alıcı denetleyici, tasarlanan sistemde oluşan hatanın sıfırlanması için kullanılmaktadır. Türev alıcı denetleyici, tasarlanan sistemin kararlılığının cevap hızını artırılması bu sayede sistemin daha kısa sürede kararlı hale gelmesi için kullanılmaktadır [33].

Kapalı çevrim kontrol sistemleri, bir makinenin kontrolünü sağlayan karar sistemleridir. Tasarlanan sistemin istenilen değerde kalması için proses değeri ile set değeri karşılaştırılmaktadır. İstenilen bu sonucun sağlanabilmesi için Kapalı Çevrim Kontrol Sistemleri kullanılmıştır [34]. Kapalı çevrim kontrol sistemleri ile tasarlanan sistemden bir ölçüm değeri alınarak, beklenen değer ile karşılaştırılır. Elde edilen fark sistemdeki hata değerini ifade etmektedir. Kapalı çevrim kontrol sistemleriyle amaç hata değerinin sıfırlanmasıdır. Tasarlanan sistemden elde edilen değer geri besleme (Feedback) olarak sisteme tekrar dahil edilmektedir. Kapalı çevrim kontrol sistemleri sistemleri, On-Off denetçiler, oransal (p) denetçiler, oransal ve integral (PI) denetçiler, oransal, integral, türev (PID) denetçisi içermektedir.

4.1.2 On-off kontrol

Literatürde bulunan kapalı çevrim kontrol sistemleri içinde en basit olanıdır. Giriş değeri, beklenen değerden aşağıda olursa sisteme çıkış verilir. Giriş değeri, beklenen

değere ulaştığında çıkış kapatılır. Şekil 8’te On-Off kapalı çevrim denetçiler sistemine ait diagram gösterilmiştir [34].



Şekil 8. On-Off kontrol

Bu denetçiler türünde çalışan aktüatörler, sık açılıp kapatıldığı için kullanım ömürleri kısadır. Kullanım ömrünün uzatılması için On-Off denetçilere histerisiz dahil edilmiştir.

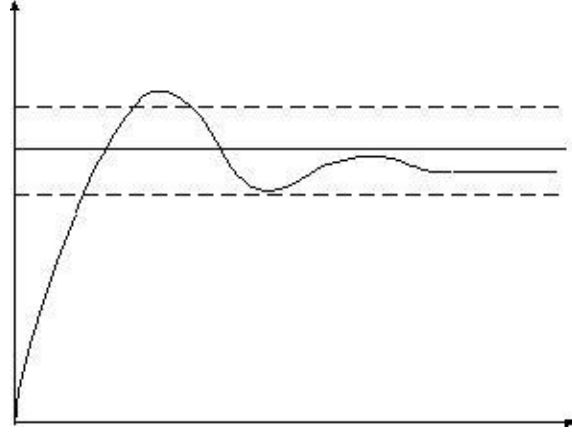
Histerisiz ile giriş değeri, istenilen değere ulaştığında çıkışın kapatılması sağlanmıştır. Giriş değeri, istenilen değer altına belirlenen histerisiz bandı kadar düştüğünde çıkışın açılması ve sistemin tekrar devreye girmesi sağlanmıştır. Bu da aktüatörün sık devreye girip çıkmasını önleyen bir sistem oluşturmuştur.

On-Off denetçilerde giriş değeri hiçbir zaman istenilen değerde sabit duramaz, beklenen değer altına ve üstüne doğru dalgalanma yapar.

On-Off denetçiler tipi, basit ve maliyeti düşük bir kontrol türüdür. Bu özelliklerinden dolayı kullanımı yaygındır. Günlük hayatta daha çok buzdolapları, ütüler, termosifon gibi endüstriyel sistemlerde kullanılmaktadır.

4.1.3 P (oransal) kontrol

P (oransal) kontrol sisteminde, On-Off kontrol sistemindeki büyük salınımları engelleme üzerine yoğunlaşmıştır. Giriş değeri, istenilen değere yaklaştıkça sistem salınımlarının oransal olarak azaltılmasına dayanır. Şekil 9’da P (oransal) denetçiler sistemine ait diagram gösterilmiştir [34].

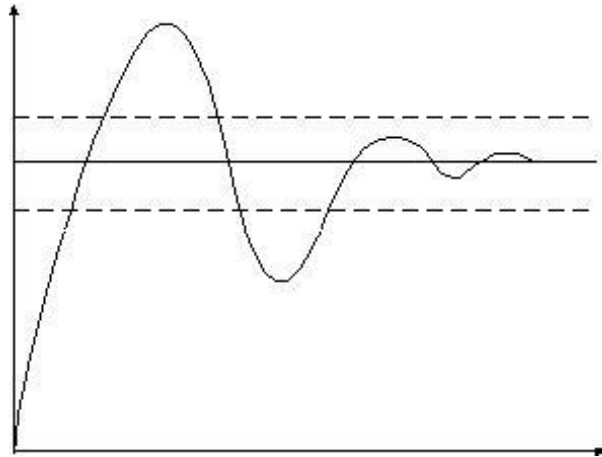


Şekil 9. P(Oransal) kontrol

P tipi kontrol sisteminde, set değerinin alt ve üst kısmında oransal bir genlik bandı tanımlanır. Oransal genlik bandı genellikle istenilen değerin yüzdelik oranı olarak ifade edilir. Giriş değeri, oransal bant içerisine girdiğinde, istenilen değerine yaklaştıkça salınımlar azaltılarak küçülen aşimlarla istenilen değere yaklaşması sağlanmıştır. Bu kontrol türünde hiçbir zaman giriş değeri ile istenilen değer çakışmaz. Bu da P tipi kontrol sistemin kendi içindeki dezavantajıdır.

4.1.4 PI (oransal+integral) kontrol

PI kontrol sistemi, oransal denetçiler türünde istenilen değere oturamayan giriş değerini, istenilen değere ulaştırmak üzerine kurgulanılmış denetçiler sistemidir. Bunun için sistemde meydana gelen hatalar integral alıcı denetleyici kullanılarak çıkışa eklenmektedir. Şekil 10'da oransal ve integral kontrol sistemine ait diagram gösterilmiştir[34].

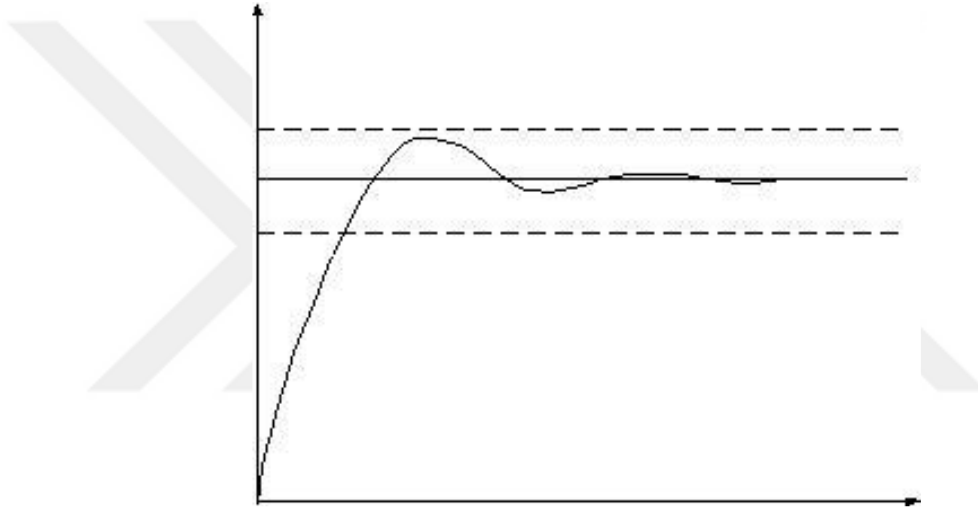


Şekil 10. PI(Oransal + Integral) kontrol

PI tipi kontrolde, proses değeri sonunda tam olarak set değerine oturur ve hata sıfırlanır. Bu kontrol sisteminde ilk anda çok yüksek aşımalar meydana gelir ve oturma süresi uzun zaman alır. Bu da PI tipi kontrol sistemin kendi içindeki dezavantajıdır.

4.1.5 PID (oransal+integral+türev) kontrol

PID denetçiler türünde, giriş değerinin kısa sürede ve mümkün olan en az aşım ile istenilen değere ulaşması amaçlanmıştır. Bunu sağlamak için tasarlanan sisteme en uygun P, I ve D katsayılarının bulunması gerekir. Bu katsayıların tespiti için modellemelerden veya deneysel sonuçlardan yararlanılabilir. Şekil 11’de oransal, integral ve türev kontrol sistemine ait diagram gösterilmiştir [34].

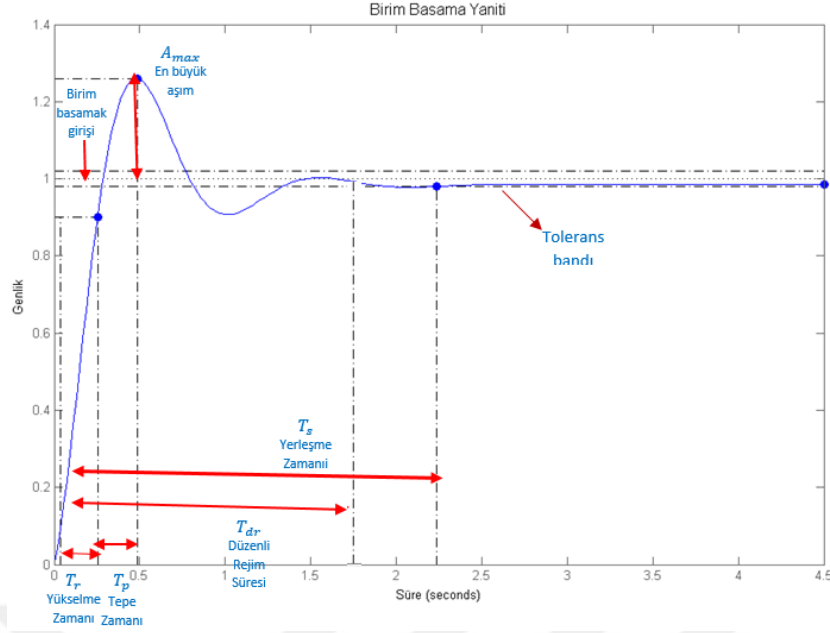


Şekil 11. PID(Oransal + Integral + Türev) kontrol

PID denetçisi oluşturan katsayılar her sistem için farklılık gösterebilmektedir. Bu farklılığın temel sebebi denetçinin uygulanacağı sistem veya kullanılacağı ortamdır. Bu yüzden PID denetçinin tasarlanması diğer denetçilere göre daha karmaşıktır. Her bir sistem için PID denetçiler yeniden tasarlanmalıdır. Bu da sistem oluşturma maliyetini arttırmaktadır.

4.1.6 PID birim basamak performans ölçütleri

Kapalı çevrim denetçiler sistemlerin, birim basamak cevabı değerlendirilmesinde genel olarak dört temel performans ölçütü üzerine yoğunlaşmıştır. Bu performans ölçütleri, yükselme zamanı, en büyük aşım, yerleşme zamanı ve birim basamak hata parametreleridir [35]. Şekil 12’de bu parametreler gösterilmiştir.



Şekil 12. Birim basamak cevabı performans ölçütü parametrelerinin temsili gösterimi

Performans tanım parametreleri açısından, salınım yapan sistemlerin birim basamak cevapları önem taşımaktadır. Bu parametreler, tepe zamanı, tepe (pik) genliği, düzenli rejim süresi ve logaritmik dekremantasyon (azalma faktörü) gibi performans tanım değerleridir.

Sistem yanıtının zamanla sifıra giden kısmı geçici hal olarak tanımlanır [36]. Bir denetçi sisteminde geçici hal yanıtının genliği ve süresi katlanabilir bir sınırın altında tutulmalıdır [27]. Doğrusal kontrol sistemlerinde geçici hal yanıtının değerlendirilmesi genellikle $u(t)$ birim basamak yanıtından yararlanılarak yapılır. Kontrol sisteminin birim basamak giriş cevabı birim basamak yanıtı olarak adlandırılır.

Maksimum aşım performans ölçütü: Tasarlanan denetçiler sistemin çıkış değerinin istenilen değere ulaşmadan önce yapılan salınımlardaki en büyük aşım miktarını ölçer. Kontrol sistemlerinde en büyük aşımın %20'den küçük olması istenir [36]. Ancak aşımın tolere edilemediği kritik kontrol uygulamalarda, aşımın mümkün olduğunca küçük olması istenir. Bir kontrol sisteminin birim basamak cevabının $(y(t), \Delta t)$ örnekleme periyodu ile örneklenmesi durumunda, ayrık domainde $y(n) = y(n \Delta t)$ ile gösterilsin. Birim basamak cevabı için maksimum aşım (4.3) ile ifade edilebilir [37]:

$$A_{max} = \max \{y(n)\} - 1, n=1,2,3,\dots,n_s \quad (4.3)$$

Yerleşme zamanı ölçütü: Tasarlanan denetçiler sisteminin çıkış değerinin istenilen değere ulaşınacaya kadar geçen süreyi ölçer [38], [39]. Kontrol

uygulamalarında en büyük aşım için yerleşme genliği genellikle $\pm 5\%$ yani $105\%-95\%$ olarak kabul görür [36]. Hızlı cevap vermesi istenilen sistemlerde yerleşme zamanı küçük seçilmelidir. Birim basamak cevabı için yerleşme zamanı (4.4) ile ifade edilebilir:

$$T_y = (n_s - n_z) \Delta t, \text{ burada } 0.95 \leq y(n) \leq 1.05 \text{ ve } n \geq n_z \quad (4.4)$$

Yükselme zamanı ölçütü: Tasarlanan denetçiler sisteminin çıkış değerinin, istenilen değerin 10% 'dan 90% 'ına ulaşmaya kadar geçen süreyi ölçer [38], [39]. Yükselme zamanı, tasarlanan sistemin verdiği cevabın sistemde meydana gelecek hızlı değişimlere uyum kabiliyetini ölçmek için kullanılır [36]. Sürekli değişen denetçiler sistemlerin işaretlerinin takibi için bu değerin küçük seçilmesi gerekmektedir. Birim basamak cevabı için yükselme zamanı (4.5) ile ifade edilebilir [37]:

$$T_r = (n_u - n_d) \Delta t, \text{ burada } 0.90 \leq y(n_u) \text{ ve } 0.10 \leq y(n_d) \quad (4.5)$$

Birim basamak cevabı ortalama karesel hata ölçütü: Tasarlanan denetçiler sisteminin çıkışının, birim basamak fonksiyonuna ne oranda değiştiğini gösteren hata ölçütüdür. Karesel hata ölçütü optimizasyon probleminde yaygın olarak kullanılır [36]. Ortalama karesel hata ölçütü, denetçiler üzerinde yapılmaya çalışılan optimizasyonlarda minimize edilmeye çalışılmaktadır. Minimize edilmeye çalışılan bu değer minimum sıfırdır. Karesel hata ölçütünün sıfıra ulaşması tasarlanan denetçiler sisteminin çıkış değerinin ideal birim basamak cevabı olduğuna işaret eder. Ancak teoride var olan bu sonuca ulaşmak pratikte tasarlanan sistemde meydana gelen gecikmelerden dolayı olanaksızlaşmaktadır. Sürekli değişkenlik gösteren kontrolör sistemlerin takibi ortalama karesel hatanın en aza indirilmesi ile sağlanabilmektedir. Ortalama karesel hata (4.6) ile ifade edilebilir [37]:

$$E_2 = \frac{1}{n_x} \sum_{n=0}^{n_x} (y(n) - r(n))^2 \quad (4.6)$$

4.1.7 PID parametrelerinin hesaplanması ve denetçiler tasarımı

PID denetçiler tasarımında P, I, D parametrelerinin belirlenmesi oldukça önemli bir konu olmuştur. Bu parametreler sistemin birim basamak cevabını oluşturmuştur. Parametrelerin doğru kurgulanması, tasarlanan denetçinin başarısını arttıracaktır. Yanlış kurgulanan parametreler, birim basamak cevabını olumsuz etkiler ve yerleşim zamanını arttırır. Bu durum başarısız bir denetçiler tasarlandığı anlamına gelir. PID parametreleri belirlenirken K_p değeri yükselme zamanı değişimleri için, K_i değeri

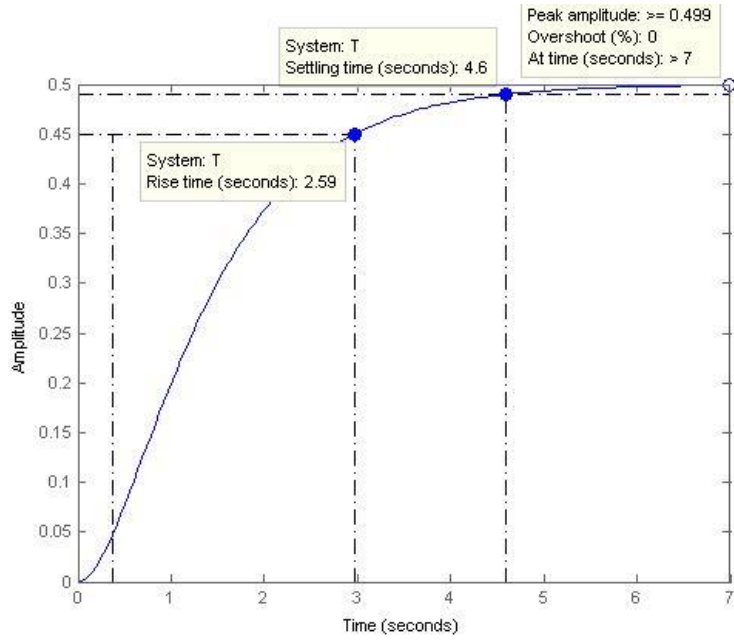
tasarlanan sistemin hatasını yok için, K_d değeri ise aşımaları düzenlemek için değiştirilmiştir.

Kapalı çevrim denetçiler sistemlerinde sistemi oluşturan denetleyicilerin etkileri (K_p , K_i , K_d) Tablo 1’de özetlenmiştir [40], [41]. Bu katsayılar dengeli bir şekilde ayarlanmayacak olursa, PID denetçinin sağlayacağı üstün özelliklerden yararlanılamaz [36], [42].

Tablo 1. PID Denetçide parametrelerin bağımsız etkileri

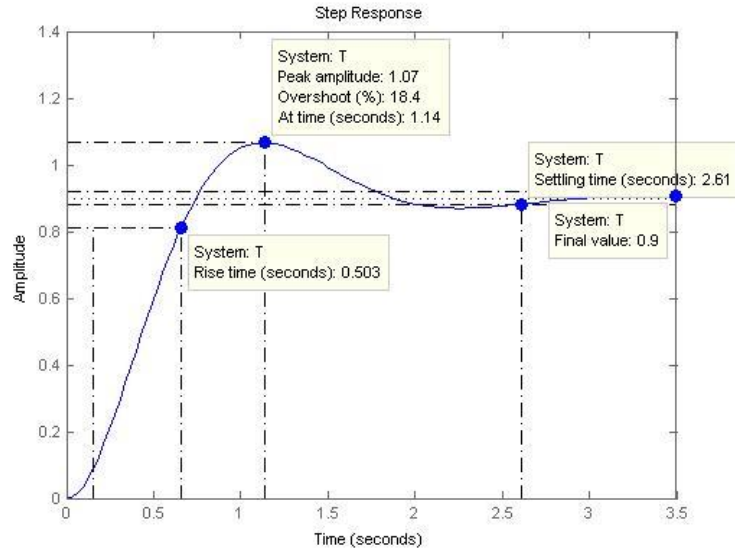
<i>Denetçi Türü</i>	<i>Kapalı Çevrim Cevabı</i>	<i>Yükseliş Zamanı</i>	<i>Aşım</i>	<i>Yerleşim Zamanı</i>	<i>Sürekli Rejim Hatası</i>
Oransal	K_p	Azalır	Artar	Küçük oranda artar	Azalır
İntegral	K_i	Küçük oranda azalır	Artar	Artar	Büyük oranda azalır
Türev	K_d	Küçük oranda değişir	Azalır	Azalır	Önemsiz değişim

Değişen P, I, D katsayılarının, birim basamak cevabına olan etkilerinin gösterilmesi amacıyla PID denetçiler tasarlanmıştır. Tasarlanan denetçilerde katsayıların belirlenmesi için Tablo 1’de gösterilen parametrelerin bağımsız etkileri göz önüne alınmıştır. Şekil 13 - Şekil 15 arasında farklı P, I, D parametreleriyle kurgulanmış PID denetçinin birim basamak cevapları gösterilmiştir. Şekil 30’da $K_p=1$, $K_i=0$, $K_d=0$ parametreleri baz alınarak bir PID denetçiler tasarlanmıştır. Bu oldukça basit bir şekilde kurgulanmış bir denetçilerdür. Burada sistemin yerleşim zamanı 4.6 sn iken yükselme zamanı 2.6 sn olmuştur. Birim basamak cevabı 0.5’te kalmıştır. Birim basamak cevabı hiçbir yerde 1’i geçemediğinden sistemde herhangi bir aşımın olmadığı gözlemlenmiştir.



Şekil 13. $K_p=1$, $K_i=0$, $K_d=0$ değerlerine göre PID birim basamak cevabı

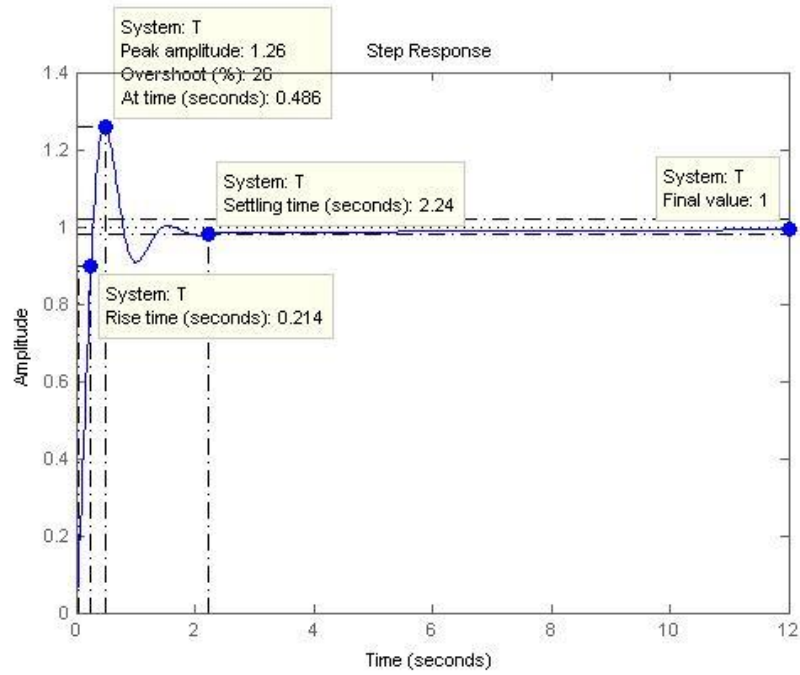
Şekil 14’de $K_p=9$, $K_i=0$, $K_d=0$ parametreleri baz alınarak bir PID denetçiler tasarlanmıştır. Bu denetçiler bir önceki tasarlanan denetçilere göre daha başarılıdır. Yükselme zamanı düşerken yerleşim zamanının da düşmesi sağlanmıştır. Yerleşim zamanındaki birim basamak cevabı 0.9 olmuştur. Bu denetçiler ile aşım %18’leri bulmuştur.



Şekil 14. $K_p=9$, $K_i=0$, $K_d=0$ değerlerine göre PID birim basamak cevabı

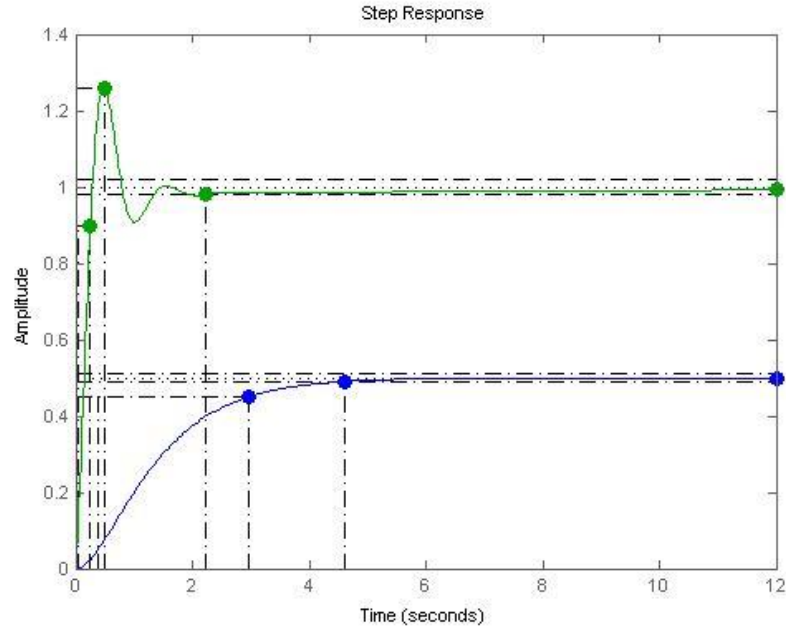
Şekil 15’de $K_p=40$, $K_i=3$, $K_d=2$ seçilerek bir denetçiler tasarlanmıştır. Diğer iki denetçilere göre daha verimli bir denetçiler oluşturulmuştur. Yerleşim zamanı 2.2 sn’lerde kalmışken en yüksek aşım %26 olmuştur. Buna ilaveten yerleşim zamanı düşmüş ve birim basamak değerinin 1 olması sağlanmıştır. Sistem tasarlanan bu

denetçiler ile kısa sürede yerleşime oturmuştur. Bu sayede PID denetçiler, bu parametrelerle başarılı bir şekilde optimize edilmiştir.



Şekil 15. $K_p=40$, $K_i=3$, $K_d=2$ değerlerine göre PID birim basamak cevabı

Şekil 16 - Şekil 17’de herhangi bir denetçiler kullanılmamış bir sistemin birim basamak cevabı ile PID ($K_p = 40$, $K_i = 3$, $K_d = 2$) ile kontrol edilen birim geri beslemeli kapalı çevrim kontrol sistemin birim basamak tepkileri birlikte gösterilmiştir. Herhangi bir denetçiler kullanılmamış olan sistemin birim fonksiyon cevabı mavi çizgi ile gösterilmiştir. PID denetçiler tabanlı sistemin birim basamak cevabı yeşil çizgi ile gösterilmiştir.



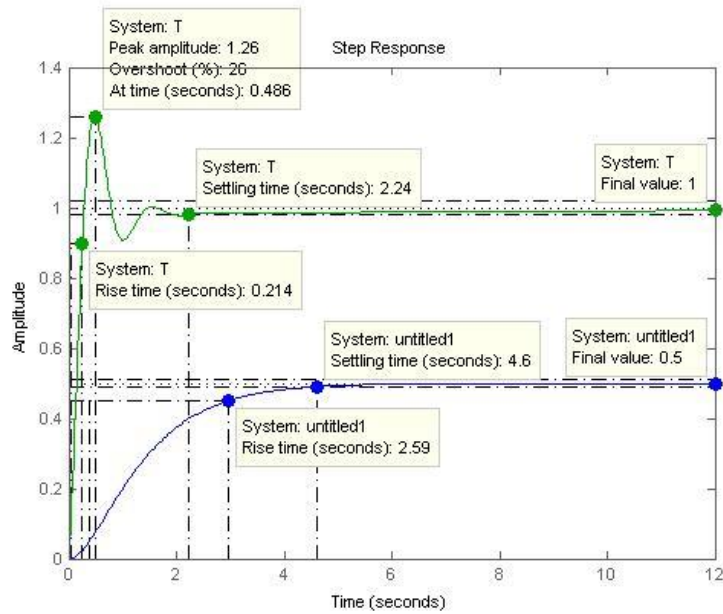
Şekil 16. Denetçi kullanılmamış ve PID denetçiler ile kontrol edilen sistemin birim basamak tepkileri

Denetçi kullanılmamış sistemin (Mavi sistemin) sürekli zaman transfer fonksiyonu (4.7) ile hesaplanmaktadır:

$$G = \frac{1}{s^2 + 3s + 1} \quad (4.7)$$

PID denetçiler ile tasarlanmış sistemin paralel formda sürekli zaman PID denetçiler değeri ise (4.8) ile hesaplanmaktadır.

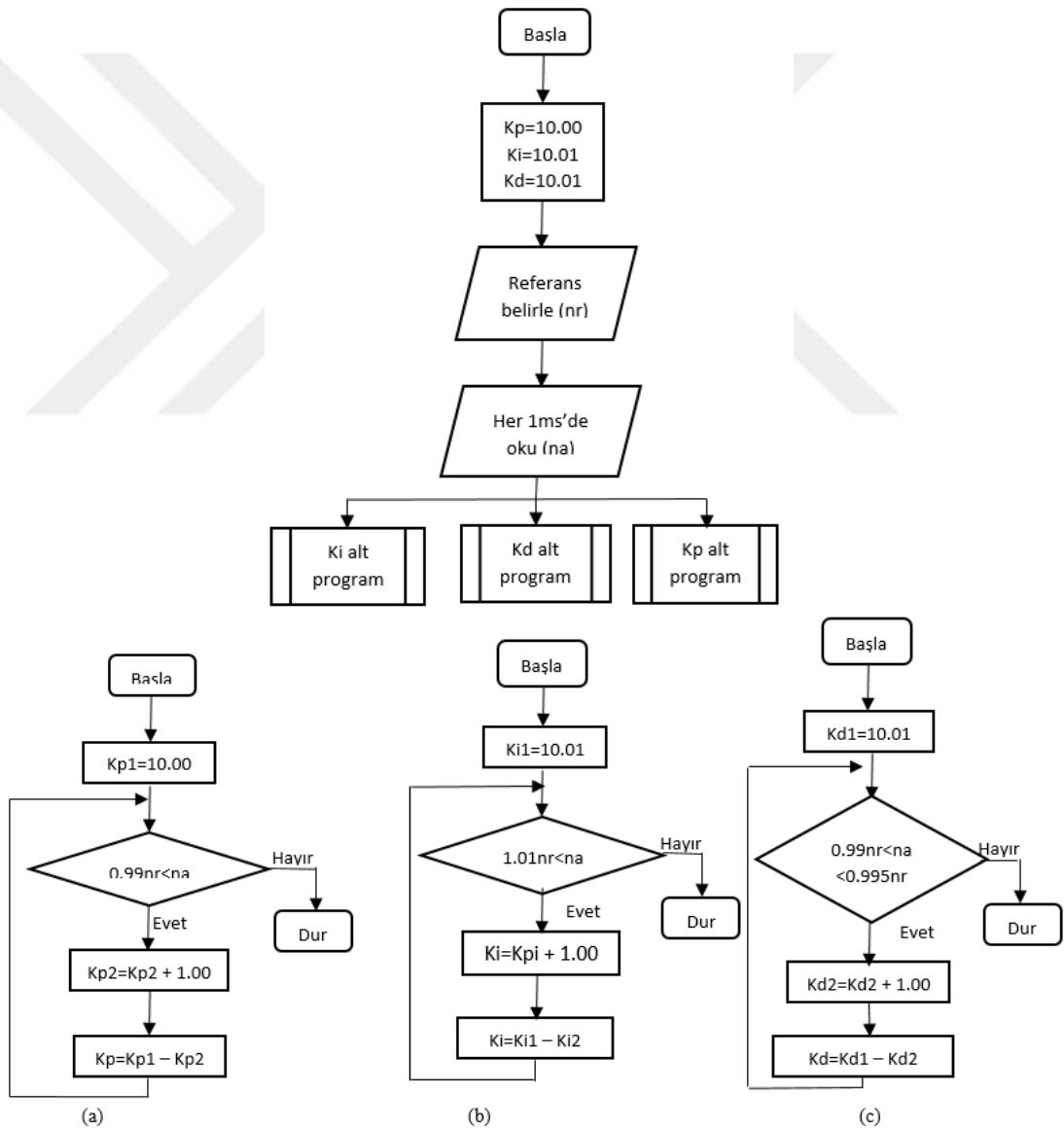
$$c = K_p + K_i \frac{1}{s} + K_d s \quad (4.8)$$



Şekil 17. Denetçi kullanılmamış ve PID denetçiler ile kontrol edilen sistemin birim basamak tepkileri

Mavi sistemin yükselme zamanı 2.59 sn iken yeşil sistemde bu değer 0.21 sn olduğu gözlemlenmiştir. Mavi system 4.6 sn’de sisteme oturmuşken bu değer yeşil sistemde 2.24 sn olduğu görülmüştür. Birim basamak cevabı mavi sistemde 0.5’te kalmışken yeşil sistemde bu değer 1’e ulaşmıştır. Şekil 17’de de görüleceği gibi önerilen yöntemle elde edilen PID tabanlı denetçi herhangi bir denetçiler kullanılmamış denetçiye göre daha iyi performans sergilediği anlaşılmıştır.

Kazanç parametrelerinin sistem üzerindeki etkilerinden faydalanılarak geliştirilen PID parametrelerinin belirlenmesi için akış diyagramları tasarlanabilir [42]. PID katsayılarını doğru şekilde belirleyip sisteme en uygun olan denetçisi tasarlamak için bir döngü kullanılmıştır. Şekil 18’te tasarlanan bir döngü gösterilmiştir.



Şekil 18. PID parametrelerinin belirlenmesinde kullanılan akış diyagramları

- Başlangıç koşulları itibariyle $K_p = 10.00$, $K_i = 10.01$, $K_d = 10.01$ olarak kabul edilmiştir.
- K_p değeri, gerçek değer referans değerinin %99'u oluncaya kadar 1 ms aralıklarla 1.00 olarak azaltılmıştır.
- K_i değeri, gerçek değer referans değerinin %99'undan küçük veya gerçek değerinin %99 ile % 101 arasında ise K_i değeri 1 ms aralıklarla 1.00 olarak azaltılmıştır.
- K_d değeri, gerçek değeri referans hızının %99 ile %99.5 arasında ise K_d değeri 1 ms aralıklarla 1.00 olarak azaltılmıştır.

Kritik durumlarda beklenen koşulların sağlanması için azaltım değerleri 1.00 yerine 0.1 veya 0.01 yapılarak hassasiyet artırılmış ve uygun katsayılar belirlenmiştir.

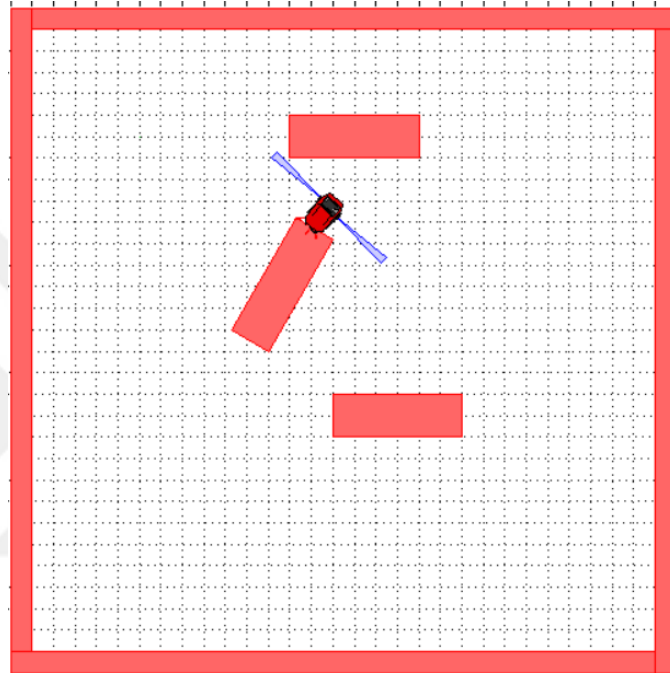
Bu akış diyagramları kullanılarak belirlenen PID katsayıları iyileştirilebilir. Bu sayede daha optimize edilmiş P, I, D katsayıları elde edilebilir. Bu katsayılarla tasarlanan denetçiler, sistemin başarısını doğrudan etkilemektedir. Gezgin robotlarda bu başarıyı, ortamda bulunan engellerden kaçarak hedefe en kısa yol ve sürede ulaşma performansı belirler.

Tablo 2'de İnocop gezgin robotu için farklı PID katsayıları ve farklı ortamlardaki sonuçlar gösterilmiştir.

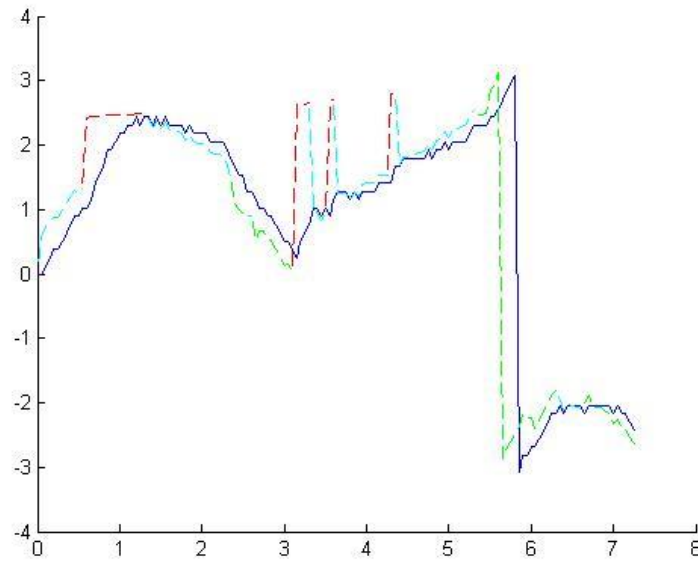
Tablo 2. PID katsayılarına göre değişen ortamlarda hedefe ulaşma süreleri

<i>Birim bas. Girişi (sn.)</i>	<i>Aşım seviyesi (sn.)</i>	<i>Aşım yüzdesi (sn.)</i>	K_p	K_i	K_d	<i>Yükselme Zamanı (sn.)</i>	<i>Hedefe ulaşma süresi (sn.)</i>	<i>Engel durumu</i>
2.35	3.08	%31.63	10	10.01	10.01	4.05	20	0
2.35	3.91	%66.47	10	8.01	8.01	3.95	20	0
2.35	3.09	%31.54	10	8.01	10.01	4.5	21	0
2.75	3.71	%40.85	8	10.01	10.01	3.65	Hedefe ulaşamadı	0
2.47	2.58	%4.26	4	0.01	0.01	1.3	12	0
2.47	2.85	%15.38	3	1.01	0.01	0.9	14	0
2.47	2.57	%4.15	3	0.01	0.01	1.6	14	0
2.47	2.51	%1.61	4	0.01	1.01	2.22	10	0
2.47	2.70	%9.31	4	1.01	0.01	0.9	13	
2.76	2.82	%2.17	4	0.01	0.01	5.16	15	1
2.76	2.83	%2.53	3	0.01	0.01	5.4	15	1
2.76	2.83	%2.53	4	1.01	0.01	4.9	14	1
2.76	2.78	%0.72	4	0.01	1.01	5.8	13	1

Sim.I.Am uygulamasında görülen eksikliklerden birinin, robotun engellerin yerinin değiştiği bir ortamda engellerden kaçmaya bir çözüm bulamaması ve engele çarpması olduğu görülmüştür. Şekil 19’da Sim.I.Am uygulamasının engeller arası mesafenin daraldığı bir ortamda engele çarpması gösterilmiştir. Şekil 20. PID Birim basamak yanıtı ($K_p=4$, $K_i=0.01$, $K_d=0.01$)’de bu ortamdaki PID birim basamak yanıtı verilmiştir.

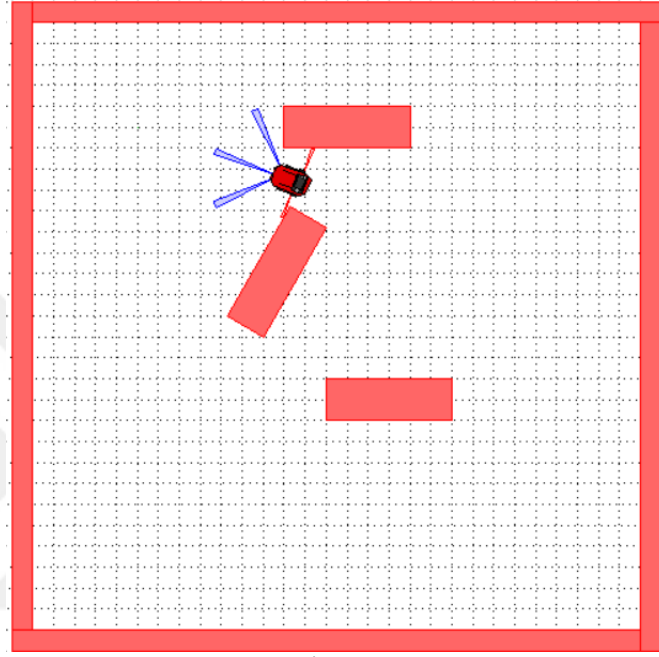


Şekil 19. Sim.I.Am robotun engele çarpma durumu

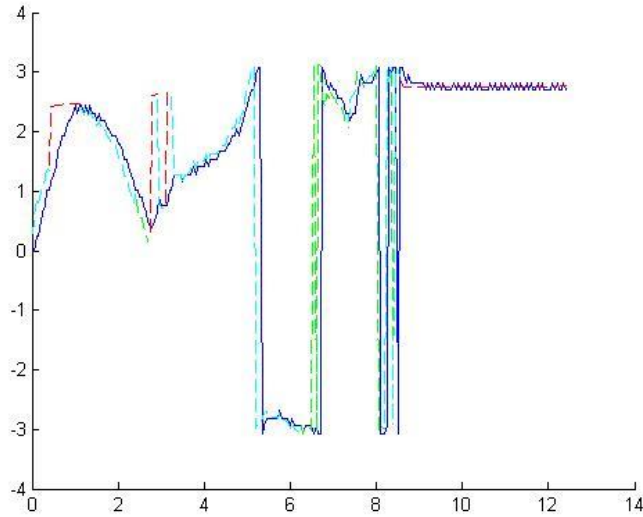


Şekil 20. PID Birim basamak yanıtı ($K_p=4$, $K_i=0.01$, $K_d=0.01$)

Bu çalışmada P, I ve D katsayıları ortam şartlarına ve İnocop'un donanım özelliklerine göre uyarlanarak bu sorun çözülüp daha optimize bir denetçiler elde edilmiştir. Şekil 21. Geliştirilen İnocop'un engelden kaçışı'de yeni tasarlanan denetçiler ile geliştirilen İnocop robotun engellerden kaçarak hedefe ilerlemesi gösterilmiştir. Şekil 22. PID Birim basamak yanıtı ($K_p=6$, $K_i=0.08$, $K_d=0.09$)'de bu ortamdaki PID birim basamak yanıtı verilmiştir.



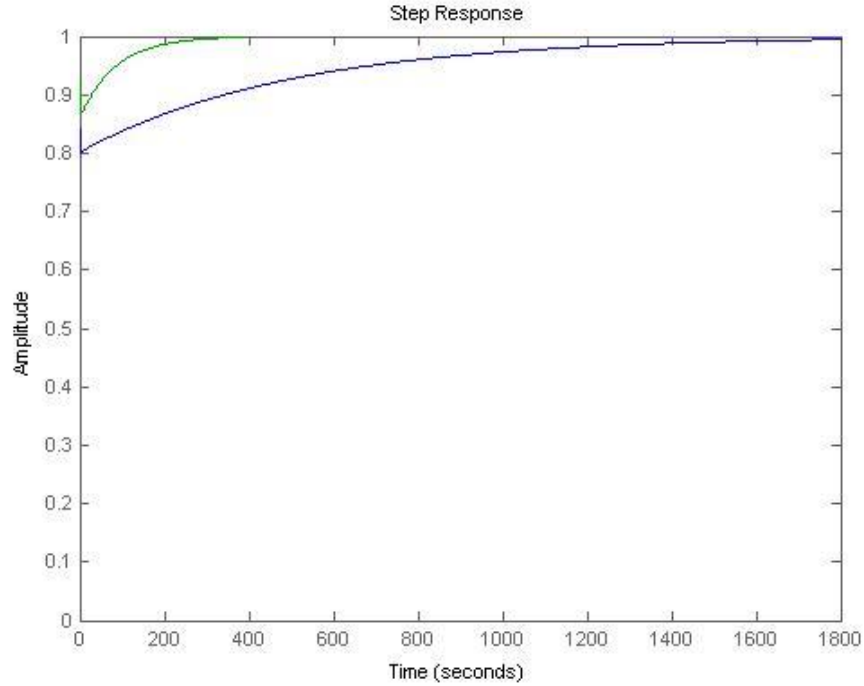
Şekil 21. Geliştirilen İnocop'un engelden kaçışı



Şekil 22. PID Birim basamak yanıtı ($K_p=6$, $K_i=0.08$, $K_d=0.09$)

Şekil 23'da Sim.I.Am ve İnocop robotta kullanılan PID kontrolör parametrelerine ait birim basamak yanıtları verilmiştir. Mavi çizgi ile belirtilen birim basamak yanıtı

Sim.I.Am’a aittir ve parametreler $K_p=4$, $K_i=0.01$, $K_d=0.01$ şeklindedir. Yeşil çizgi ile gösterilen birim basamak yanıtı ise İnocop robota aittir ve parametreler $K_p=6$, $K_i=0.08$, $K_d=0.09$ olarak seçilmiştir. Şekil 23’da görüldüğü gibi İnocop robot için tasarlanan PID denetçinin yerleşme zamanı çok daha kısa olduğu gözlemlenmiştir.



Şekil 23. PID Birim basamak yanıtı kıyaslanması

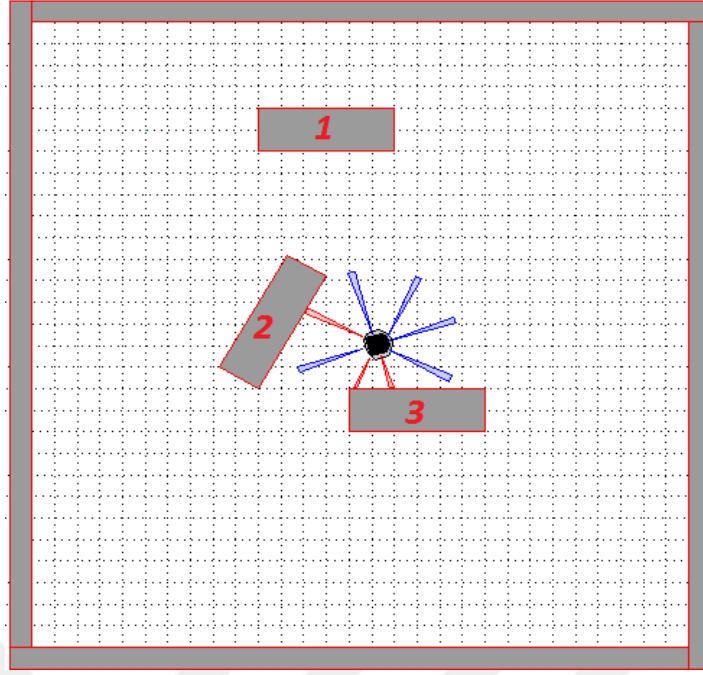
4.1.8. Denetçi algoritmalarının karşılaştırması

Tablo 3’te İnocop robotun ve orijinal Sim.I.Am robotun engelden kaçma hareketinde kullanılan algoritmalar ve ortamda bulunan engel durumları gösterilmiştir. Test için robotun (0,0) noktasından (-1,1) noktasına hareketi baz alınmıştır.

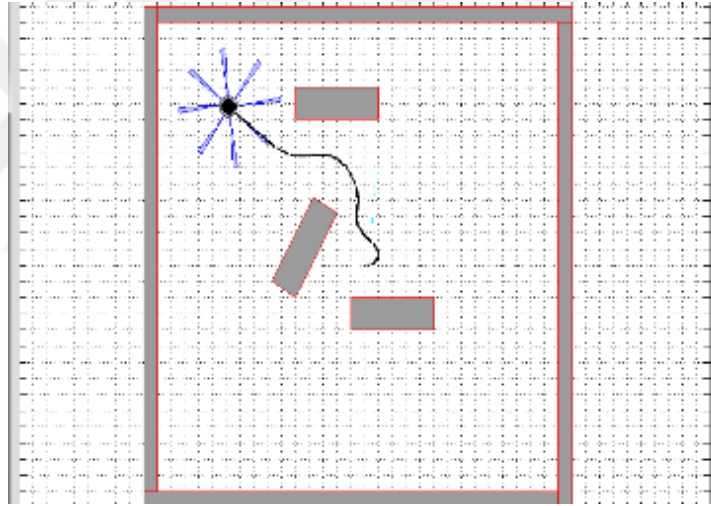
Şekil 24’de simülasyon için kullanılan engelli ortam ve engellerin konumları gösterilmiştir. Şekil 25’te robotun bu ortamda izlediği yol gösterilmiştir.

Orijinal Sim.I.Am: Hiçbir değişiklik yapılmamış Sim.I.Am robotun denetçi algoritmasıdır.

İnocop (PID): Sim.I.Am robotun Khpera IV’e uyarlanmış ve PID değerleri optimize edilmiş İnocop robotun denetçi algoritmasıdır.



Şekil 24. Simülasyon ortamı



Şekil 25. İnocop robotun izlediği yol

Tablo 3. Mobil robotların değişen ortamlara göre hedefe ulaşma süreleri

	Engelsiz (sn.)	3 nolu engel (sn.)	2 nolu engel (sn.)	1 nolu engel (sn.)	2 ve 3 nolu engeller (sn.)	1 ve 2 nolu engeller (sn.)	1 ve 3 nolu engeller (sn.)	1 ve 2 ve 3 nolu engeller (normal) (sn.)	1 ve 2 ve 3 nolu engeller (Dar) (sn.)
Orijinal Sim.I.Am (PID)	9	9	10	10	11	11	10	11	7-çarptı
İnocop (PID)	10	10	11	10	12	12	10	12	14

Tablo 3 incelendiğinde;

- Herhangi bir engelin bulunmadığı simülasyon ortamında Sim.I.Am robot hedefe İnocop robottan daha önce ulaşmıştır.
- 2 ve 3 numaralı engellerin ayrı ayrı, 2 ve 3 numaralı engellerin birlikte, 1 ve 2 numaralı engellerin birlikte, 1, 2 ve 3 numaralı engellerin birlikte bulunduğu simülasyon ortamında Sim.I.Am robot hedefe İnocop robottan daha önce ulaşmıştır.
- Sadece 1 numaralı engelin ve 1 ile 3 numaralı engellerin birlikte bulunduğu simülasyon ortamında Sim.I.Am ve İnocop robot hedefe aynı sürede ulaştığı görülmüştür.
- 1 ve 2 numaralı engellerin arasındaki mesafenin daraltıldığı simülasyon ortamında Sim.I.Am robotun şekil 19'daki gibi engelden kaçmadığı ve 2 numaralı engele çarptığı görülmüştür. İnocop üzerine tasarlanan yeni PID denetçiler ile şekil 21'de görüldüğü gibi İnocop robotun dar mesafeli engellerin arasından engellere çarpmadan geçerek hedefe ilerlemesi sağlanmıştır. Engeller arası mesafe daraltıldığında İnocop robotun hedefe ulaşma süresi mevcut ortamda hedefe ulaşma süresine göre %17 artmıştır.

İnocop, Sim.I.Am'ın daha geliştirilmiş ve optimize edilmiş bir modeli olarak ortaya çıkmıştır. Hedefe ulaşmada ortamın şartları ne olursa olsun PID denetçinin buna göre yapılandırılarak en iyi performansın alınması sağlanmıştır. Engellerin konumunun sürekli değiştiği bir ortamda PID katsayılarının belirlenmesi ve optimizasyonu zaman kaybına neden olacağı gözlemlenmiştir. Bu da maliyeti arttıracak bir dezavantaj olmuştur.

Simülasyon ortamında kullanılan test ortamındaki engellerin konumları değiştirildiğinde gezgin robotun engellere çarptığı tespit edilmiştir. Bu problem üzerine yoğunlaşarak geliştirilen İnocop robotun engellerden kaçarak hedefe başarılı bir şekilde ulaştığı gözlemlenmiştir.

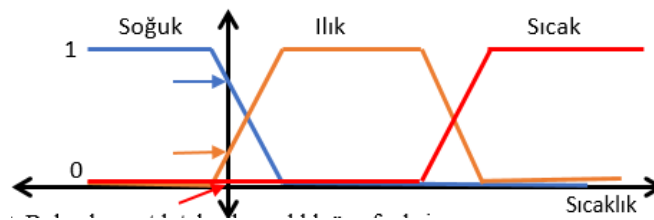
Çalışmanın bundan sonraki kısımlarında Sim.I.Am robotun yerine, geliştirilen İnocop robot kullanılmıştır. Simülasyon ortamında herhangi bir değişiklik yapılmayarak aynı simülasyon ortamı kullanılmıştır.

4.2 Bulanık Mantık (Fuzzy)

Bulanık mantık, bulanık eseme ya da puslu mantık, 1961 yılında Azeri matematikçi Lotfi A. Zadeh'in yayınladığı bir makalenin [43] sonucu oluşmuş bir mantık yapısıdır.

Bulanık mantığın temeli bulanık küme ve alt kümelerine dayanır. Klasik yaklaşımda bir varlık ya kümenin elemanıdır ya da değildir [44]. Matematiksel olarak ifade edildiğinde varlık, kümenin elemanı olduğunda "1", kümenin elemanı olmadığı zaman "0" değerini alır. Bulanık mantık ise klasik küme gösteriminin sonsuz aralıkta genişletilmesidir. Bulanık varlık kümesinde her bir varlığın üyelik derecesi vardır. Varlıkların üyelik derecesi, (0, 1) aralığında herhangi bir değer olabilir ve üyelik fonksiyonu $M(x)$ ile gösterilir.

Örnek olarak normal oda sıcaklığını 23 derece olarak kabul edersek klasik küme kuramına göre 23 derecenin üzerindeki sıcaklık derecelerini sıcak olarak kabul ederiz ve bu derecelerin sıcak kümesindeki üyelik dereceleri "1" olur. 23 altındaki sıcaklık dereceleri ise soğuktur ve sıcak kümesindeki üyelik dereceleri "0" olur. Soğuk kümesini temel aldığımızda bu değerler tersine döner. Bulanık küme yaklaşımında üyelik değerleri $[0,1]$ aralığında değerler almaktadır. Örneğin 14 derecelik sıcaklık için üyelik derecesi "0", 23 sıcaklık derecesi için üyelik değeri "0,25" olabilir. Şekil 26'te bulanık mantık tabanlı sıcaklık üye fonksiyonuna ait bir bulanık mantık fonksiyonu gösterilmiştir [44].



Şekil 26. Bulanık mantık tabanlı sıcaklık üye fonksiyonu

Klasik kümelerin aksine bulanık kümelerde elemanların üyelik dereceleri $[0, 1]$ aralığında sonsuz sayıda değişebilir[45]. Bunlar üyeliğin derecelerinin devamlı ve aralıksız bütünüyle bir kümedir [46]. Keskin kümelerdeki soğuk-sıcak, hızlı-yavaş, aydınlık-karanlık gibi ikili değişkenler, bulanık mantıkta biraz soğuk, biraz sıcak, biraz karanlık gibi esnek niteleyicilerle yumuşatılarak gerçek dünyaya benzetilir. En önemli fark, böyle bir çatıda bilginin kaynağındaki küme üyeliğinin kesin tanımlanmış önkoşullarının olmayışı ve daha çok sorunlarla rastgele değişkenlerin hazır bulunmasındadır.

Bir şeyin varlığı kendisine ait bir isimle doğar. Evrendekilerin tamamı hem/ya tek (1) hem de/ya da sonsuz hem de/ya da sonsuz eksi tektir ($\infty - 1$) [44].

Klasik mantık ile bulanık mantık arasındaki temel farklılıklar

Tablo 4'te gösterilmiştir:

Tablo 4. Klasik – Bulanık Mantık farkları

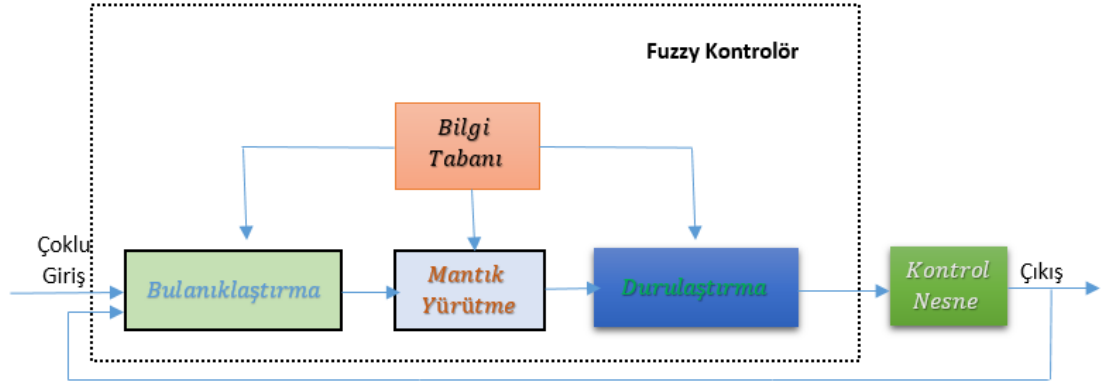
Klasik Mantık	Bulanık Mantık
A veya A Değil	A veya A Değil
Kesin	Kısmi
Hepsi veya Hiçbiri	Belirli Derecelerde
0 veya 1	0 ve 1 Arasında Süreklilik
İkili Birimler	Bulanık Birimler

Bulanık algoritması kullanılarak oluşturulan denetçiler, bilinmeyen ortamlarda otonom gezgin robotların yol planlaması ve engelden kaçarak hedefe ulaşması konusunda etkili bir denetleyici olmuştur. Bulanık denetçiler sistemleri oldukça karmaşık kural tablosuna sahiptir [47]. Giriş değerlerinin artmasına bağlı olarak bu kural tablosu eleman sayıları da exponansiyel (üssel) olarak artmaktadır.

Çalışmanın bu bölümünde PID tabanlı denetçiler kullanan İnocop robot yerine yeni tasarlanan Bulanık tabanlı denetçiler kullanan İnocop robot simülasyon ortamında kullanılmıştır. İnocop robotun engellerden kaçınması için kullanılan PID algoritması yerine Fuzzy yönlendirme algoritması kullanılmıştır. Geliştirilen bulanık tabanlı denetçiler sayesinde gezgin robotun engellerden kaçınarak hedefe başarılı bir şekilde ulaşması amaçlanmıştır.

4.2.1 Bulanık lojik sistemi parametrelerinin belirlenmesi ve denetçi tasarımı

Bulanık mantık sistemleri, temel olarak bulanıklaştırma (fuzzification), bilgi tabanı (knowledge base), bulanık muhakeme(mantık yürütme) (bulanık reasoning) ve durulaştırma (Defuzzification) kısımlarından oluşur. Bulanık lojik sistemin tipik mimarisi Şekil 27' da gösterilmiştir [48].



Şekil 27. Bulanık lojik sistem mimarisi

Bilgi tabanı: Dilsel denetçi kurallarını ve bulanık bilgi idaresini tanımlamak için gerekli olan ifadeleri sağlar. Kural tabaları sayesinde denetçi yöntemi belirlenir[49]. İlgili verileri ve denetçi kurallarını saklamak için kullanılır [26].

Bulanıklaştırma (fuzzification): Bulanık denetçinin bu ilk birimi giriş değişkenlerinin gerçek değerlerini ölçer. Ölçülen bu değeri üyelik fonksiyonlarıyla kıyaslayarak uygun olan bir etiket verir. Bu etiket sayesinde giriş değerleri dilsel bir özellik kazanmış olur[49]. Diğer bir ifade ile doğru giriş değişkenlerini bulanık değişkenler olarak adlandırılan giriş derecelerine dönüştürür.

Mantık yürütme: Bilgi tabanında var olan kurallar üzerinde mantık yürütülür ve bu kurallar çerçevesinde giriş değeriyle çıkış değeri arasında bir bağlantı sağlanır. Mantık yürütme bulanık mantık denetçinin en önemli kısmıdır [49].

Durulaştırma: Bulanıklaştırma evresinde dilsel değişkenlere dönüştürülen değerler bu kısımda kontrolörde kullanılacak gerçek sayısallaştırılmış değerlere geri dönüştürülür [49].

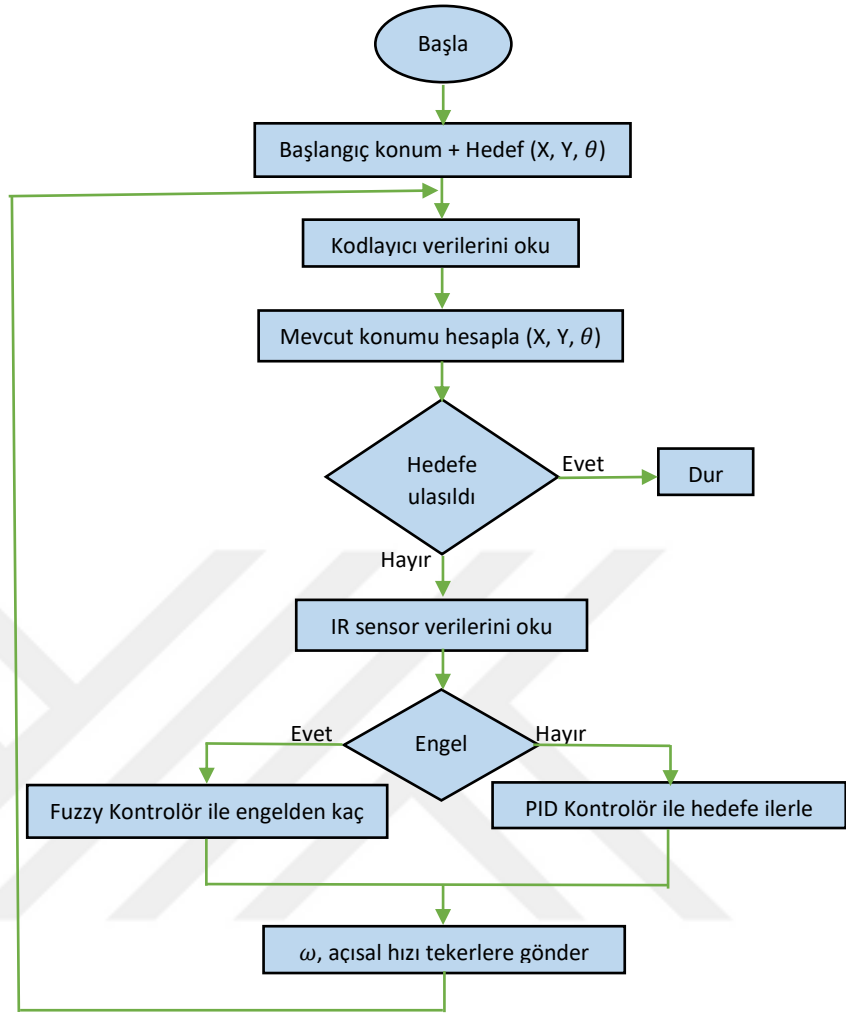
Mobil robotlar hedefe doğru hareket ederken yönlendirmeyi etkileyen üç ana etken vardır: engellerin konumu, engellerden kaçma hareketi ve dönme hareketi. Bu etkenler maliyeti etkileyen faktörlerdir[50]. Yönlendirmenin birçok görevi yerine getirilirken, robot hareket ettiği yolda engellerden kaçmalıdır. Bu nedenle, tez çalışmasının bu kısmında, İnocop gezgin robotu için engellerden kaçma davranışını sağlayan bulanık tabanlı bir denetçinin uygulanması anlatılmıştır. Bulanık tabanlı denetleyicinin tasarımı, dil değişkenlerinin, işlem durumlarının, giriş ve çıkış üye fonksiyonlarının seçimi ile başlar. Bir sonraki adım, bilgi tabanının ve mantık yürütme sürecinin seçimidir. Çıkarımdan sonra bir durulaştırma stratejisi oluşturulmalıdır.

Bulanık sistemler, insan kontrol mekanizmasını taklit ettiği için anlaşılması kolaydır. Bu taklit özelliğinden dolayı gerçekleşmesi basittir ve çok sayıda paralel

işleme izin verir [45]. Diğer sistemlere göre yazılımları basit, verimlilikleri yüksek olduğundan performans / maliyet oranı da yüksektir[51], [52].

Çalışmanın bu bölümünde yeni bir bulanık mantık denetleyicisi geliştirilmiştir. Mevcut PID denetçiler ve bulanık mantık denetçiler İnocop robotun başlangıçtan hedefe doğru ilerlemesindeki hareket denetleyicileri olarak kullanılmıştır. Hedefe ulaşmak için kullanılan PID denetçiler ve engellerden kaçmak için kullanılan bulanık mantık denetçiler robotu engellere çarpmadan hedefe ulaştırmak için birleştirilerek hibrit bir denetçiler tasarlanmıştır. Bu çalışmada robotun hareketi PID denetçiler ile başlamıştır. Eğer sensörler robotun hareketini engelleyecek herhangi bir engel tespit ederse algoritmanın bulanık mantık denetçisininü kullanacak şekilde değişmesi sağlanmıştır. PID ve bulanık mantık çıktıları açısal hız olarak tekerleklerle gönderilmiştir. Mobil robotun bu açısal hız değerlerine göre tekerlek hızlarını belirlemesi ve çıktılara göre hareket etmesi sağlanmıştır.

Şekil 28’de İnocop robotun hareketi sırasında kullanılan algoritmaların akış diyagramı gösterilmiştir. Bu akış diyagramına göre İnocop robotun öncelikle başlangıç konumunu ve hedefin konumunu belirlenmiştir. Kodlayıcıdan gelen verileri okunarak robotun mevcut konumu hesaplanmıştır. Bir sonraki adımda robotun mevcut konumuyla hedefe ulaşılıp ulaşılamadığı kontrol edilmiştir. Eğer hedefe ulaşılmadıysa hedefe doğru ilerlemek amaçlanmıştır. Robot hedefe ilerlerken IR sensörlerden gelen veriler eş zamanlı olarak işlenmiştir. Eğer hareket esnasında engellerden kaçma gereksinimi yoksa PID denetçiler kullanılarak hedefe ilerlenmiştir. Eğer engellerden kaçma gereksinimi varsa İnocop robot engelden kaçma eğilimine girmiştir. Engellerden kaçma eğilim için bulanık denetçiler kullanılmıştır.

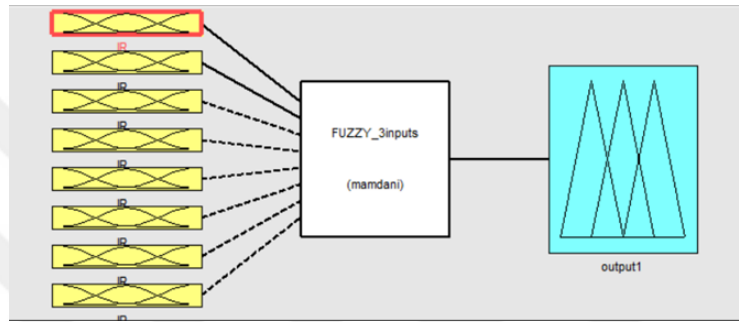


Şekil 28. Robot hareketinde kullanılan akış diyagramı

Engellerden kaçınma davranışını uygulamak ve bulanık bir denetçiler tasarlamak için robot yönlendirmesi (navigasyonu) için gerekli tüm bilgilere sahip olduğumuz varsayılmıştır. Girişler, robot ile engel arasındaki mesafeyi ölçen sensörlerden elde edilen değerlere ait dilsel değişkenlerden olmuştur. Çıktılar açısal hıza ait dil değişkenlerinden oluşmuştur. Çalışmada sekiz girdi değişkeni olarak kullanılmıştır: üç adet sensör soldaki engele uzaklığı ölçmek için, üç adet sensör sağdaki engele uzaklığı ölçmek için, bir sensör ön taraftaki engele olan uzaklığı ve bir sensör de arka taraftaki engele olan mesafeyi ölçmek için kullanılmıştır.

Her bir girdi dilsel değişkeni için, üç dilsel değer tanımlanmıştır: yakın, orta, uzak. Her bir çıktı için yedi dilsel değer tanımlanmıştır: “Geri hızlı, geri, geri yavaş, dur, ileri yavaş, ileri, ileri hızlı”. Çalışmada, engellerden kaçarak hedefe doğru ilerlemede

bulanık denetçinin uygulanması amaçlanmıştır. Bu denetçiler için, sekiz giriş ve bir çıkışlı Matlab uygulamasının kullanıcı arayüzü Şekil 29'da gösterilmiştir. Giriş ve çıkış değişkenleri için her üyelik fonksiyonunda karesel tip işlevler kullanılmıştır. Bu denetçiler özelinde 28 kural tanımlanmıştır. İnocop robotun bu kurullar çerçevesinde engellerden başarıyla kaçtığı gözlemlenmiştir. Sisteme yeni kurallar eklenmesi oldukça dikkat edilmesi gereken bir konu olmuştur. Eklenen yeni kurallar tutarsız kurallara veya daha kötüsü diğer kurallarla çakışan kurallara neden olabileceği görülmüştür. Bu durumun denetçinin başarısızlığına yol açacağı anlaşılmıştır.



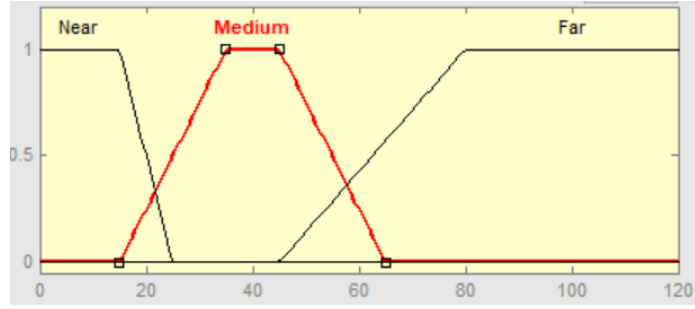
Şekil 29. 8 giriş ve 1 çıkış ile 28 kurallı Bulanık denetçiler.

4.2.2 Giriş – çıkış üye fonsiyonlarının belirlenmesi

Bulanık mantık, bilinmeyen dinamik ortamlardaki engellerden kaçmak için bir kontrol sinyali üretilmesini önermektedir. Üretilen bu kontrol sinyali bu çalışma için önerilen bulanık mantık sistemi için tekerleklerin açısal hızı (ω) olmuştur. Bulanık mantık denetçinin girişleri, robotun sol, ön ve sağ taraftaki engeller ile arasındaki mesafelerden oluşmuştur. Bu mesafeler, robotun sağ, sol, ön ve arka tarafında bulunan sekiz kızılötesi (IR) sensör ile elde edilmiştir. Denetçinin çıkışları, tekerleklerin açısal hızları olarak kullanılmıştır [53].

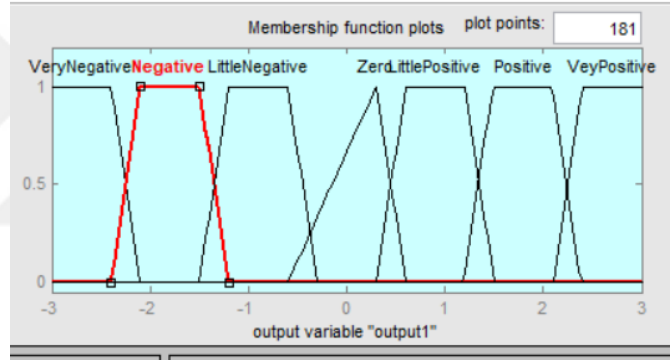
Engelden sensöre olan uzaklık için “yakın”, “orta” ve “uzak” dil değişkenleri kullanılır. Bulanık mantık denetçiler, Şekil 30’da gösterildiği gibi, her giriş için üç üyelik fonksiyonuyla gerçekleştirilmiştir.

Bu çalışmada engellerden kaçmak için bulanık algoritması kullanılmıştır. Bulanık logic sistem için giriş değeri olarak engellere olan mesafeler sensörler tarafından ölçülmüştür. Girişlere göre belirlenen kurallar sonucunda çıktılar üretilmiştir. Sensörler tarafından ölçülen engele olan mesafeler bu giriş değerlerini oluşturmuştur. Bulanık algoritmasını oluşturan giriş değerleri 0-120 aralığında seçilmiştir.



Şekil 30. Giriş üye fonksiyonları (engele olan mesafe)

Giriş değerinin karşılığı olan çıkış değerleri yedi üye fonksiyondan oluşmuştur. Bulanık algoritmasına ait çıkış değerleri -3, 3 arasında seçilmiştir. Bulanık mantık denetçiler, Şekil 31’de gösterildiği gibi, her bir çıktı için (ω) yedi üyelik fonksiyonuyla gerçekleştirilmiştir. Tekerleklerin açısal hızları için çıkış değerleri “Geri hızlı”, “geri”, “geri yavaş”, “dur”, “ileri yavaş”, “ileri”, “ileri hızlı” olarak isimlendirilmiştir.



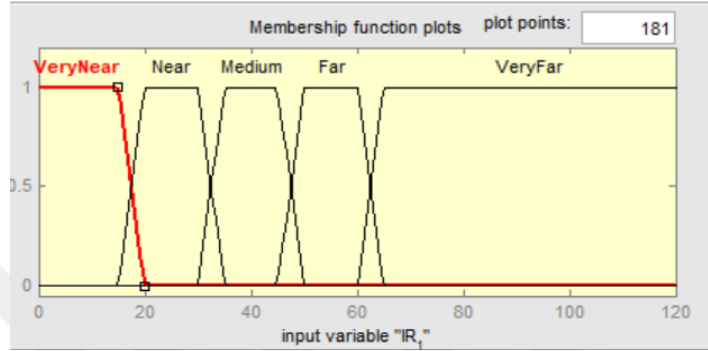
Şekil 31. Çıkış üye fonksiyonları (açısal hız, ω)

Açısal hız için yedi üyelik fonksiyonunu kullanan çıkışlar için kullanılan dil değişkenleri Tablo 5’te gösterilmiştir. Tespit edilen engele göre yirmi yedi şart oluşturulmuştur.

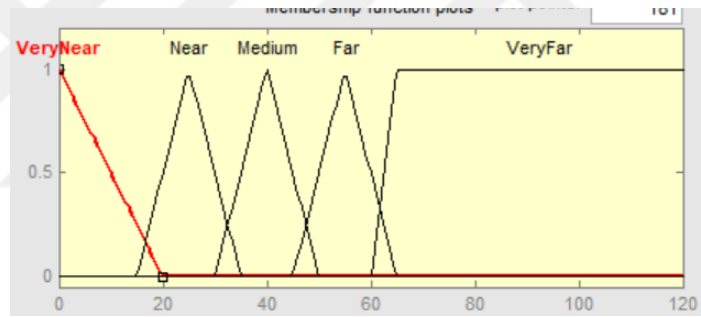
Tablo 5. Açısal hız için bulanık mantık kuralları

Giriş			Çıkış
Sol engel	Ön engel	Sağ engel	W
N	N	N	VN
N	N	M	N
N	N	F	N
N	M	N	VN
N	M	M	N
N	M	F	N
N	F	N	VN
N	F	M	N
N	F	F	N
M	N	N	N
M	N	M	VN
M	N	F	P
M	M	N	VP
M	M	M	P
M	M	F	N
M	F	N	P
M	F	M	P
M	F	F	LN
F	N	N	VP
F	N	M	LP
F	N	F	VN
F	M	N	N
F	M	M	LP
F	M	F	LP
F	F	N	P
F	F	M	LP
F	F	F	Z

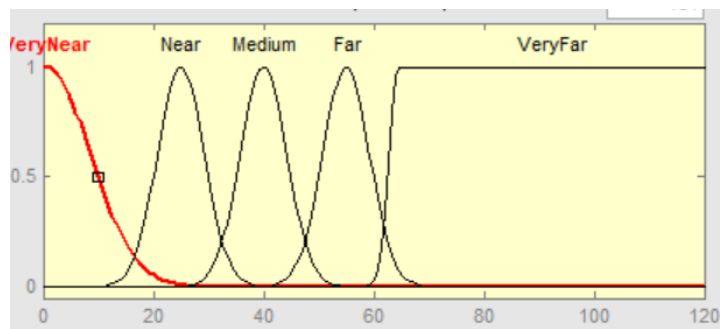
Tezin ilerleyen aşamalarında hassasiyeti arttırmak için giriş üye fonksiyon sayısı üçten beşe çıkarılmıştır (şekil 32-34). Giriş değerleri near, very near, medium, far, very far olarak isimlendirilmiştir. Çalışmada farklı tipteki üye fonksiyonlarının üreteceği çıktıları gözlemlemek için giriş üye fonksiyonları olarak bulanık trimf, gauss, tramf eğrileri ayrı ayrı kullanılmıştır. Şekil 32 - Şekil 34 arasında üye sayısı beş olan ve farklı tipteki üye fonksiyonları gösterilmiştir.



Şekil 32. Beş girişli Tramf üye fonksiyonu



Şekil 33. Beş girişli Trimf üye fonksiyonu



Şekil 34. Beş girişli Gauss üye fonksiyonu

Giriş ve çıkış üye fonksiyonları sırasıyla Tablo 6 ve Tablo 7'de gösterilmiştir. Tablolardaki değerler, x ekseninin bir tür tekil noktadaki değerlerini gösterecek şekilde seçilmiştir. Tablo 6'da yakınlık sensörleriyle alınan veriler için kullanılacak olan üyelik fonksiyonları gösterilmiştir. Örnek olarak, “-10 : 0 : 15 : 25”, giriş bulanık değişkenlerinin “yakın” olarak isimlendirilen bir üyelik fonksiyonu için tekil

noktalarda x eksenı deęerlerıdır. Tablo 7’de tekerleklerin aısal hızlarında kullanılmak üzere tasarlanan üyelık fonksiyonları gösterılmıřtır. “0.3 : 0.6 : 1.2 : 1.5” bulanık deęiřkenlerinin “ilerı yavař” bulanık bir üyelık fonksiyonu iin tekil noktalarda x eksenı deęerlerıdır.

Tablo 6. Giriř üye fonksiyonlarının tanımı

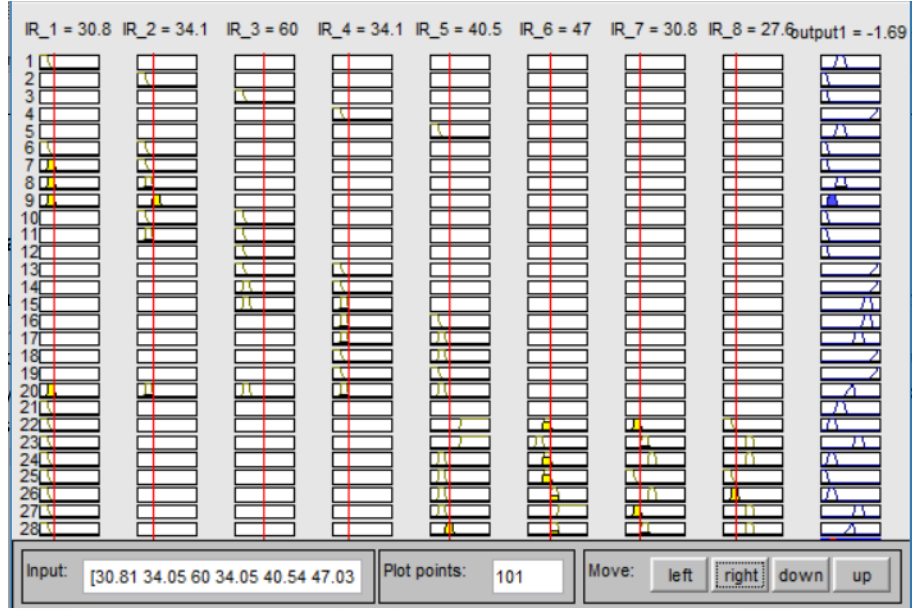
Terim	Üye Fonksiyonları Aralıęı
Yakın	-10 : 0 : 15 : 25
Orta	15 : 35 : 45 : 65
Far	45 : 80 : 120 : 150

Tablo 7. ıkıř üye fonksiyonlarının tanımı

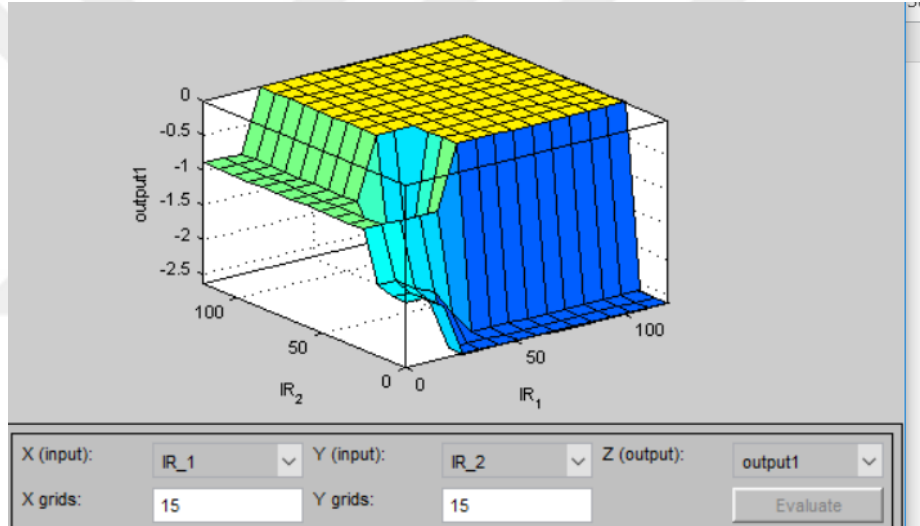
Terim	Üye Fonksiyonları Aralıęı
Geri hızlı	-4.0 : -3 : -2.5 : -2.0
Geri	-2.5 : -2.0 : -1.5 : -1.0
Geri yavař	-1.5 : -1.2 : -0.6 : -0.3
Dur	-0.6 : 0.3 : 0.3 : 0.6
İleri yavař	0.3 : 0.6 : 1.2 : 1.5
İleri	1.2 : 1.5 : 2.1 : 2.4
İleri hızlı	2.1 : 2.4 : 3 : 3.6

4.2.3 Deneti kurallarının oluřturulması

Deneti kuralları deneysel bilgi ile oluřturulabilir. Simölason ortamında kullanılan gezgin robota ait 8 sensör kullanılmıřtır. Kural tabanı, birok deney, gözlem ve operasyon deneyimine dayanan ok sayıda bulanık iliřkiden oluřur. Ayrıca, bulanık kuralların gerek sayısı birok faktöre baęlı olarak alınmalıdır. Genel ilke, öncölün tamlıęı üzerinedir. Bulanık mantık sisteminin tasarım sürecini basitleřtirmek iin daha az sayıda kural daha iyidir. Deneysel test ortamında geleneksel bulanık mantık sistemine ait yirmi sekiz kural oluřturulmuřtur. řekil 35’te bu kurallara göre oluřturulmuř kural seti ıktıları verilmiřtir. 8 sensör iin girilen deęerlere göre bu yirmi sekiz kurala göre ıktılar üretilmiřtir. řekil 36’da bu kural setlerine göre oluřan yüzey görönüm grafięi gösterilmıřtir.



Şekil 35. Kural seti çıktıları



Şekil 36. Yüzer görünüm grafiği (surface)

Bulanık denetçiler için oluşturulan kural setlerinden elde edilen yirmi adet örnek kural aşağıda gösterilmiştir. Mobil robot, sensörlerden gelen verileri işleyerek bu kural tablosundaki kurallarla kıyaslar. En uygun kuralı bularak bu kuralın sonucuna göre çıktı üretir. İnocop robotun bu eğilimde engellerden kaçması sağlanmıştır.

- R1: If IR_1=VeryNear then O=Negative;
- R2: If IR_2=VeryNear then O=VeryNegative;
- R3: If IR_3=VeryNear then O=VeryNegative;
- R4: If IR_4=VeryNear then O=VeryPositive;
- R5: If IR_5=VeryNear then O=Negative;

R6: If IR_1=VeryNear and IR_2=VeryNear then O=VeryNegative;

R7: If IR_1=Near and IR_2=Medium then O=LittleNegative;

.....

.....

R16: If IR_3=VeryNear and IR_4=VeryNear then O=VeryPositive;

R17: If IR_3=Near and IR_4=Near then O=Positive;

R18: If IR_4=Near and IR_5=Near then O=LittlePositive;

R19: If IR_4=VeryNear and IR_5=VeryNear then O=VeryPositive;

R20: If IR_1=VeryFar and IR_2=VeryFar and IR_3=Far and IR_4=Far then O=Zero;

Tablo 8’de beş girişli bulanık mantık üye fonksiyonu kullanılmıştır. Örnek olarak iki farklı sensörden elde edilen veriler giriş değeri olarak kullanılmıştır. Bu veriler ile elde edilen tablo kullanılarak daha fazla kural elde edilebilir. Tablo 8, Şekil 2’deki İnocop robotun 1 ve 2 nolu sensörlerinden gelecek olan giriş değerleri kullanılarak elde edilmiştir.

IR_1 O IR_2	VeryNear	Near	Medium	Far	VeryFar
VeryNear	Very Negative	Very Negative	Negative	Little Negative	Little Negative
Near	Very Negative	Negative	Negative	Little Negative	Little Negative
Medium	Negative	Negative	Little Negative	Little Negative	Little Negative
Far	Little Negative	Little Negative	Little Negative	Zero	Zero
VeryFar	Little Negative	Little Negative	Zero	Zero	Zero

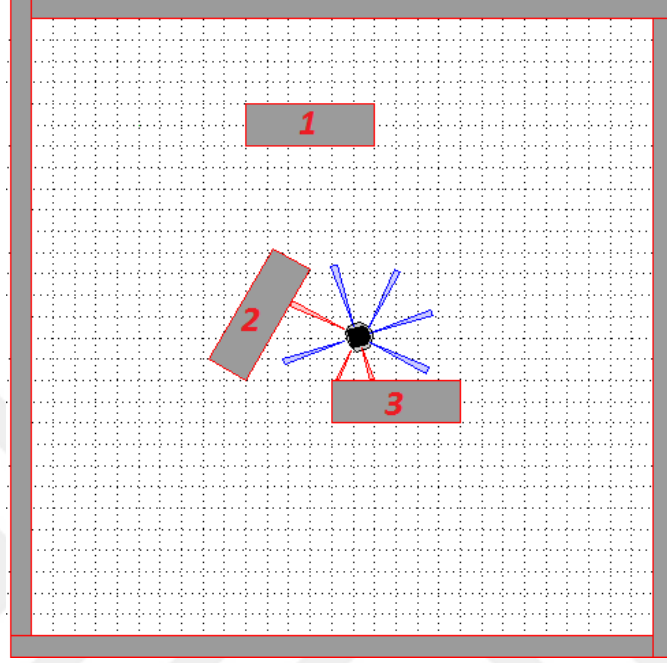
Tablo 8. Kural tabanı

4.2.4 Bulanık tabanlı İnocop simülasyon uygulaması

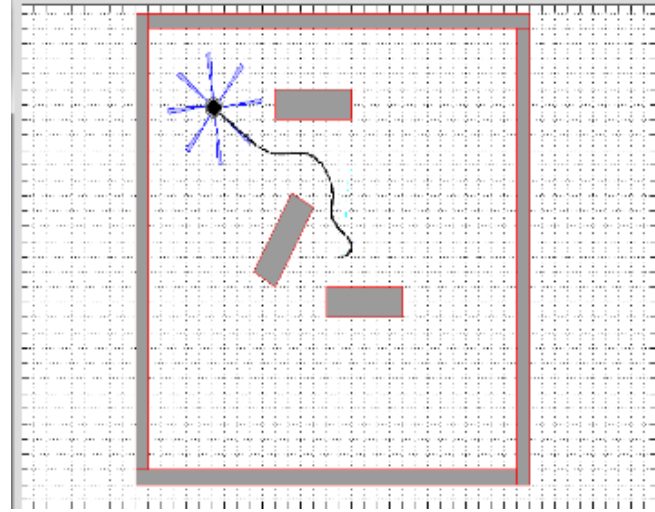
Simülasyon ortamında robotun başlangıç noktasından bitiş noktasına hareketi baz alınmıştır. Robotun başlangıç pozisyonu (0,0) ve hedefin pozisyonu (-1,1) olarak seçilmiştir.

Test için robotun başlangıç pozisyonundan (0,0) bitiş noktasına (-1,1) hareketi baz alınmıştır. Şekil 37’de robotun hareket ettiği ortam gösterilmiştir. Ortama farklı

konumlarda bulunan üç engel konumlandırılmıştır. Önerilen bulanık mantık sistemi ile İnocop robotun, engellerden güvenli bir şekilde kaçabildiği gözlemlenmiştir. Şekil 38’da İnocop robotun bu ortamda izlediği yol gösterilmektedir. İnocop robot çizgi yolu boyunca hareket ederek hedefe ulaşmıştır.



Şekil 37. Simülasyon ortamı



Şekil 38. Bulanık tabanlı İnocop robotun izlediği yol

4.2.5 Denetçi algoritmalarının karşılaştırması

Tablo 9’da çalışmanın dördüncü kısmını oluşturan PID tabanlı İnocop robotun, Sim.I.Am robotun ve farklı tipteki üye fonksiyonları kullanan bulanık tabanlı İnocop robotun ortamda bulunan engellere göre hedefe ulaşma süreleri gösterilmiştir.

Orijinal Sim.I.Am: Hiçbir deęişiklik yapılmamış Sim.I.Am robotun denetçi algoritmasıdır.

İnocop (PID): Sim.I.Am robotun Khpera IV'e uyarlanmış ve PID deęerleri optimize edilmiş İnocop robotun denetçi algoritmasıdır.

Bulanık 3 inputs Trapmf: İnocop robotun bulanık algoritması kullanılarak oluşturulan denetçi algoritmasıdır. Near, Medium, Far olmak üzere 3 üçgensel giriş ve VeryNegative, LittleNegative, Negative, Zero, LittlePositive, Positive, VeryPositive olmak üzere 7 çıkıştan oluşmaktadır.

Bulanık 5 inputs trapmf: İnocop robotun bulanık algoritması kullanılarak oluşturulan denetçi algoritmasıdır. Very Near, Near, Medium, Far, Very Far olmak üzere 5 dörtgensel giriş ve VeryNegative, LittleNegative, Negative, Zero, LittlePositive, Positive, VeryPositive olmak üzere 7 çıkıştan oluşmaktadır.

Bulanık 5 inputs Gauss: İnocop robotun bulanık algoritması kullanılarak oluşturulan denetçi algoritmasıdır. Very Near, Near, Medium, Far, Very Far olmak üzere 5 GaussTrimf giriş ve VeryNegative, LittleNegative, Negative, Zero, LittlePositive, Positive, VeryPositive olmak üzere 7 çıkıştan oluşmaktadır.

Bulanık 5 inputs Gauss2: İnocop robotun bulanık algoritması kullanılarak oluşturulan denetçi algoritmasıdır. Very Near, Near, Medium, Far, Very Far olmak üzere 5 GaussTrapmf giriş ve VeryNegative, LittleNegative, Negative, Zero, LittlePositive, Positive, VeryPositive olmak üzere 7 çıkıştan oluşmaktadır.

Tablo 9. Engel tabanlı hedefe ulaşma süreleri

	Engelsiz (sn.)	3 nolu engel (sn.)	2 nolu engel (sn.)	1 nolu engel (sn.)	2 ve 3 nolu engeller (sn.)	1 ve 2 nolu engeller (sn.)	1 ve 3 nolu engeller (sn.)	1 ve 2 ve 3 nolu engeller (normal) (sn.)	1 ve 2 ve 3 nolu engeller (Dar) (sn.)
Orijinal Sim.I.Am (PID)	9	9	10	10	11	11	10	11	7-çarptı
İnocop (PID)	10	10	11	10	12	12	10	12	14
Bulanık 3inputs Trapmf	9	9	11	10	11	11	11	12	6 çarptı
Bulanık 5 inputs trapmf	10	10	11	11	12	12	12	14	3 çarptı
Bulanık 5 inputs Gauss	10	10	11	10	11	12	10	13	3 çarptı
Bulanık 5 inputs Gauss2	10	10	11	10	11	12	11	13	4 çarptı
Bulanık 5 inputs trimf	9	9	10	10	11	12	11	13	4 çarptı

Tablo 9 incelendiğinde;

- İnocop üzerine uygulanan yeni PID değerleriyle, gezgin robotun hedefe ulaşma süresini %9 arttırdığı görülmüştür.
- Engeller arası mesafenin daraldığı simülasyon ortamında sadece PID tabanlı İnocop robotun başarılı şekilde hedefe ulaştığı görülmüştür.
- Farklı bulanık algoritmalarının kullanılmasının hedefe ulaşma süresini etkilediği görülmüştür.
- Simülasyon ortamında değişken ortamların pek çoğunda PID tabanlı İnocop robot ile bulanık tabanlı İnocop robotun hedefe ulaşma süresi benzerdir.

- Üç üye fonksiyon girişli bulanık denetçinin beş üye fonksiyon girişli bulanık denetçilere göre daha başarılı olduğu görülmüştür. Hedefe ulaşma süresinde yaklaşık %10 - %13 arası bir başarımla elde edilmiştir.
- Beş giriş ve farklı tipte üye fonksiyonlarını kullanan bulanık denetçiler kendi içlerinde başarımları değişmektedir.
- Bulanık algoritmalarında daha geniş kapsamlı kural seti oluşturulduğunda hedefe ulaşma başarısının da doğru orantılı olarak arttığı görülmüştür.
- Engellerin yerinin dinamik olarak değiştiği ortamlarda bulanık algoritması etkin olarak kullanılabilir. Çünkü bu tür ortamlarda PID kat sayılarının kurgulanması zaman kaybına neden olmaktadır. Kural setleri iyi kurgulanırsa en iyi başarımın bulanık algoritması kullanılarak elde edildiği gözlemlenmiştir.

4.3 Gauss Dağılımı Tabanlı Denetçi Tasarımı

Çalışmanın son bölümünde diferansiyel tahrikli gezgin robot sistemlerinin gezinim denetçilerinin Gauss algoritması ile yapılandırılma yöntemi önerilmiştir. Önerilen bu yöntemde açısal hız parametreleri, tasarlanan Gauss tabanlı denetçi yöntemiyle hesaplanmış ve hem engellerden kaçınmak hem de hedefe ulaşmak için mobil robota gönderilmiştir.

Bu çalışma iki fazdan oluşmuştur. Birinci fazda, robot hedefe ilerlerken PID yerine Gauss algoritması ve engellerden kaçarken mevcut PID algoritmasının kullanılması amaçlanmıştır. İkinci fazda, birinci fazın geliştirilmesi üzerine yoğunlaşmıştır. Bu fazda, engellerden kaçmak için de Gauss algoritmasının kullanılması amaçlanmıştır. İkinci fazın her iki adımında Gauss dağılımları ayrı ayrı parametrelerle oluşturulmuştur. Kullanılan yeni denetçiler ile robotun hedefe başarılı bir şekilde ulaşması hedeflenmiştir.

4.3.1 Gaussian dağılımı tabanlı hareket

Olasılık teorisinde normal (veya Gaussian veya Gauss veya Laplace – Gauss) dağılımı çok yaygın bir sürekli olasılık dağılımıdır. İstatistikte normal dağılımlar önemlidir. Normal dağılımlar, genellikle dağılımları bilinmeyen gerçek değerli rasgele değişkenleri temsil etmek için doğal ve sosyal bilimlerde kullanılmıştır. Gauss dağılımına sahip rastgele bir değişkenin normal olarak dağıtıldığı ve normal sapma olarak adlandırıldığı söylenir.

Gaussian algoritması istatistik biliminin birçok alanında kullanılmıştır. Konum gösteren ortalama (μ , aritmetik ortalama) ve ölçek gösteren varyans (σ^2 , "yayılım")tan oluşturulmuştur. Normal dağılımın olasılık yoğunluğu (4.9)'de gösterilmiştir:

$$f_G(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sigma} \varphi\left(\frac{x-\mu}{\sigma}\right), \quad x \in R \quad (4.9)$$

Burada;

μ , beklenen dağılım değeridir

σ , standart sapma ve

σ^2 , varyanstır.

Bu hesaplamada, π (~ 3.14) sabit bir parametredir. (4.10) standart normal dağılım için yoğunluk fonksiyonudur. Standart normal dağılım $\mu = 0$ ve $\sigma = 1$ parametreleri olan bir normal dağılımdır.

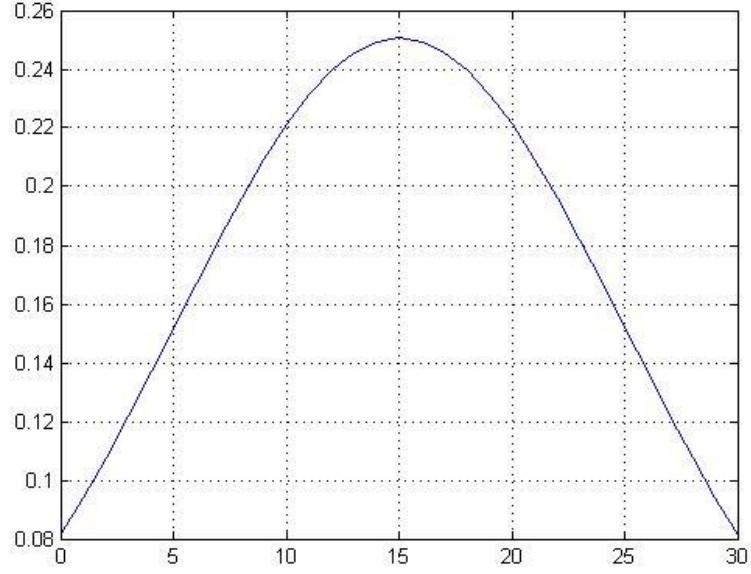
$$\varphi(x|\mu, \sigma^2) = \varphi_{0,1}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in R, \quad (4.10)$$

4.3.2 Engellerden kaçmak için Gauss dağılımı tabanlı denetçi kullanılması

Gezgin robotlar hedefe doğru ilerlerken karşılarına çıkabilecek engellere olan mesafeleri kızılötesi yakınlık sensörleri ile takip edilmiştir. Sensörlerden alınan bu mesafe belirlenen bir eşik değerinin altına düştüğünde gezgin robot engelden kaçma eğilimine başlaması sağlanmıştır.

Çalışmanın bu kısmında engel ile robot arasındaki mesafenin eşik değeri 30 olarak belirlenmiştir. Gauss girdi değeri olarak robot ile engel arasındaki mesafe girilmiştir. Bu nedenle x değeri 0:30 olacak şekilde fonksiyonun girdi değeri sağlanmıştır. Sigma 10 ve μ 15 olarak belirlenmiştir. Bu değerler yapılan deneysel çalışmalarla belirlenmiştir. Simülasyon ortamında engellerden kaçmak için en iyi değer olarak bu katsayılar belirlenmiştir. Şekil 39'da bu parametrelere göre oluşan Gauss eğrisi gösterilmiştir.

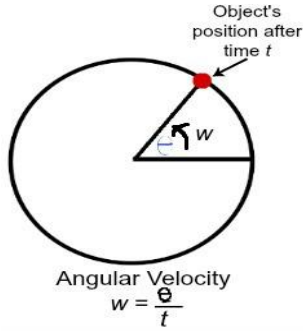
```
xmesafe=-0:30;
sigma=10;
m=15;
fx=(1/sigma*sqrt(2*pi))*exp(-(xmesafe-m).^2/(2*sigma^2));
```



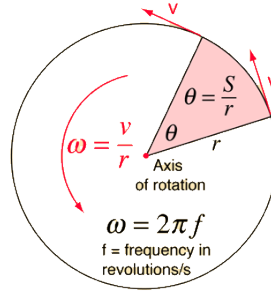
Şekil 39 Gauss eğrisi

Mesafe sensörlerinden gelen veriler içinde minimum değere sahip olan belirlenmiştir. Bu değeri engele en yakın olan sensörden elde edilmiştir. Bu sayede engelin robota göre konumu belirlenmiştir. Robotun, engelin sağ veya sol tarafta olmasına göre değerler pozitif veya negatif olarak belirlenmiştir. Gauss eğrisinden elde edilen sonuç bir sabit olarak kabul edilmiştir. Bu sabit değer robotun hareketi için açısal hıza bir kat sayı olarak kullanılmıştır. Gezgin robotun, bu açısal hıza (ω) göre hareket ederek engellerden hedeften uzaklaşmayacak şekilde kaçması sağlanmıştır.

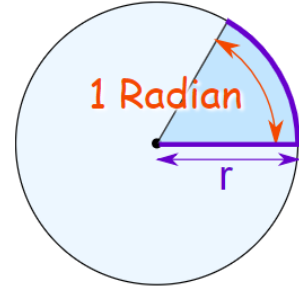
Robotun ön kısmındaki 3 numaralı sensör robotun hareket doğrultusundadır. Robotun engele en yakın olan sensörüyle robotun ön kısmındaki bu sensör arasındaki açı hesaplanmıştır (θ). Şekil 40’de görüldüğü gibi bu açının zamandaki değişimleri açısal hıza etki eder. Şekil 41’de açısal hız ve yarıçap değerleri ile tekerleklerin hızının hesaplanması gösterilmiştir. Devir sıklığı, açısal hıza etki eden diğer bir faktördür. Belirtilen iki sensörün arasındaki açının radyan olarak karşılığı hesaplanır (28). Şekil 42’te radyan ile yarıçap arasındaki ilişki gösterilmiştir.



Şekil 40. Açık – zaman değışimi



Şekil 41. ω , hesaplanması



Şekil 42. Radian – yarıçap ilişkisi

Çember çevre hesaplanması (4.11)’de gösterilmiştir.

$$S = 2\pi r \quad (4.11)$$

$$2\pi \text{ radian} = 360^\circ \quad (4.12)$$

$$1 \text{ radian} = 180/\pi \quad (4.13)$$

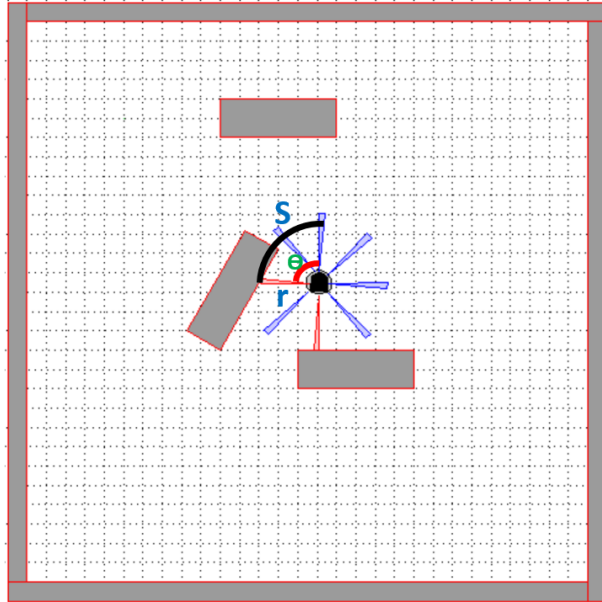
(4.13)’e göre, (4.14) ve (4.15) elde edilmiştir:

$$1 \text{ radian} = 57,2958^\circ \quad (4.14)$$

$$1^\circ = 0.017453292 \quad (4.15)$$

$$\text{radian}_\theta = \theta * \text{radian} \quad (4.16)$$

Şekil 43’de İnocop robotun engele olan pozisyonu görölmektedir. Engele olan mesafe yarıçap olacak şekilde engele en yakın sensör ile robotun ön sensörü arasında bir yay çizilmiştir. Bu yayı gören merkez açı θ olarak seçilmiştir.



Şekil 43. İnocop robot ile engelin konumu

Açık ve yarı çap kullanılarak bu yay uzunluğu hesaplanmıştır (4.17):

$$S = 2\pi r \frac{\theta}{360} \quad (4.17)$$

Robotun bu yay doğrultusunda engelden kaçması beklenmiştir. Elde edilen bu değer ile gauss eğrisinden elde edilen sabit değer kullanılarak açısal hız hesaplanmıştır (4.18):

$$\omega = \text{radian}_\theta * s / (fx(i)) \quad (4.18)$$

(4.18) ile elde edilen açısal hız robota gönderilmiş ve robotun engellerden kaçması bu bağlantı ile sağlanmıştır.

4.3.3 Hedefe ulaşmak için Gauss dağılımı tabanlı denetçi kullanılması

Gezgin robotun hedefe ulaşmasında robotun yönlendirilmesi büyük öneme sahip olmuştur. Burada önemli olan parametreyi robotun hedefe olan konumu oluşturmıştır. Bu konum robot ile hedef arasındaki mesafe ve açıdan elde edilmiştir. Gauss eğrisinde tepe değerinin 1'e eşit olmasının sağlanması amaçlanmıştır.

Buradan;

$$x = 0, \quad \sigma = \frac{1}{\sqrt{2\pi}} \quad (4.19)$$

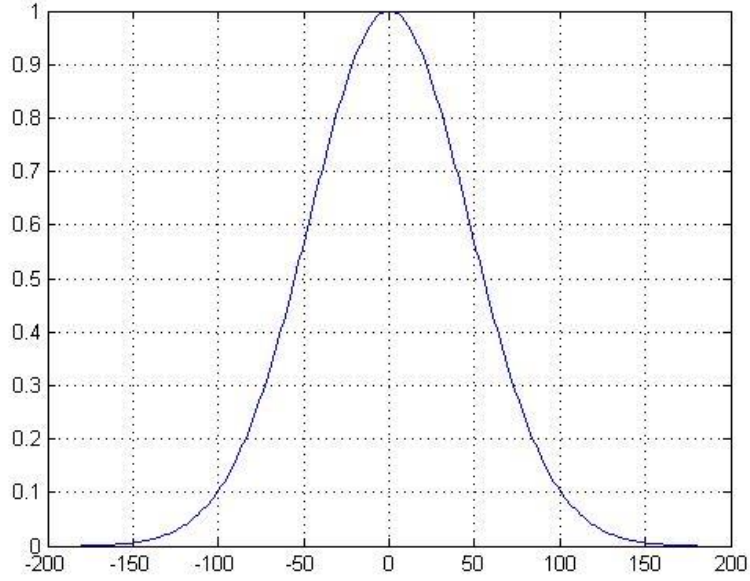
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.20)$$

(4.19) ve (4.20) kullanılarak, (4.21) elde edilmiştir:

$$f_0 = 1 \quad (4.21)$$

Bu çalışmada Gauss girdi değeri olarak robot ile hedef arasındaki açı kullanılmıştır. Bu nedenle x değeri $-\pi, \pi$ (-180,180) aralığında olacak şekilde fonksiyonun girdi değeri sağlanmıştır. $\sigma=47, \mu=0$ olarak seçilmiş ve gauss fonksiyonu bu parametrelerle hesaplanmıştır. **Error! Reference source not found.** Şekil 44'te bu parametrelerle elde edilen Gauss eğrisi gösterilmiştir. Bu parametreler ile $x=0$ iken $f(x)=1$ olması sağlanmıştır. Matlab kod bloğu aşağıdaki gibidir.

```
xmesafe=-180:180;
sigma=47;
m=0;
fx=(1/sigma*sqrt(2*pi))*exp(-(xmesafe-m).^2/(2*sigma^2));
```



Şekil 44. Gauss Eğrisi

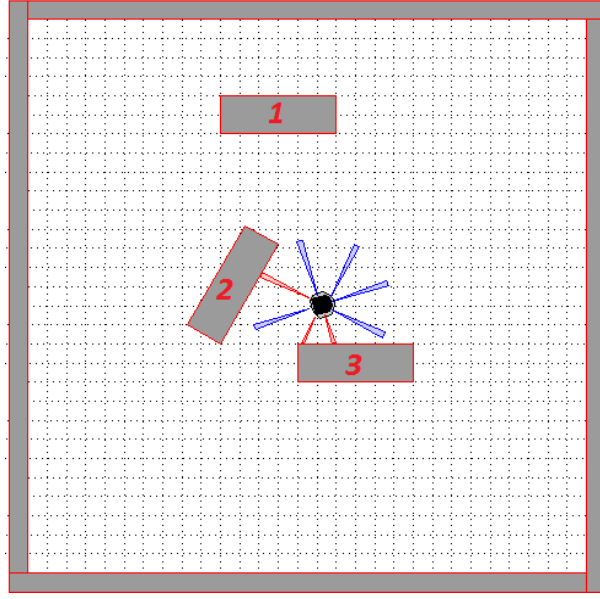
Burada, olasılık fonksiyonu (4.22) kullanılarak robotun doğrultusunun ayarlaması sağlanmıştır.

$$p(x) = 1 - f(x) \quad (4.22)$$

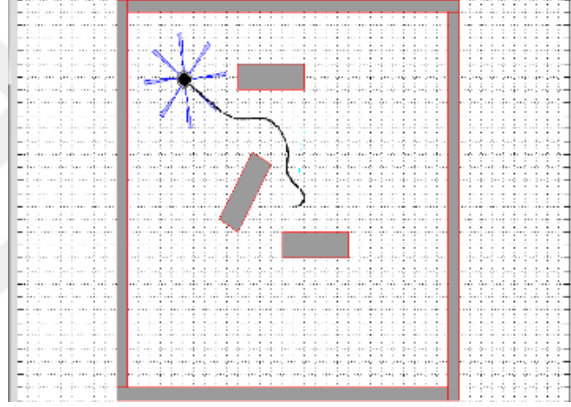
Tepe noktasında, $x=0$ ve $f(x)=1$ iken (38) kullanılarak $p(x) = 0$ elde edilmiştir. Bu durum, robot ile hedef arasında açı farklı olmadığını ifade etmektedir. Robot ile hedef aynı doğrultuda olduğu için robotun aynı doğrultuda yoluna devam etmesi sağlanmıştır.

x açı değerine göre robotun hedefe göre konumu belirlenmiştir ve gauss eğrisinden elde edilen sonuçlar olasılık fonksiyonuna sokularak elde edilen değer açısal hız olarak robota gönderilmiştir. Robot hedefe göre konumunu dikkate alarak bu açısal hız ile hedefe doğru ilerlemiştir.

Şekil 45'te simülasyon ortamı, Şekil 46'te çalışmanın dördüncü bölümünde geliştirilen İnocop robotun bu ortamda izlediği yol gösterilmiştir.

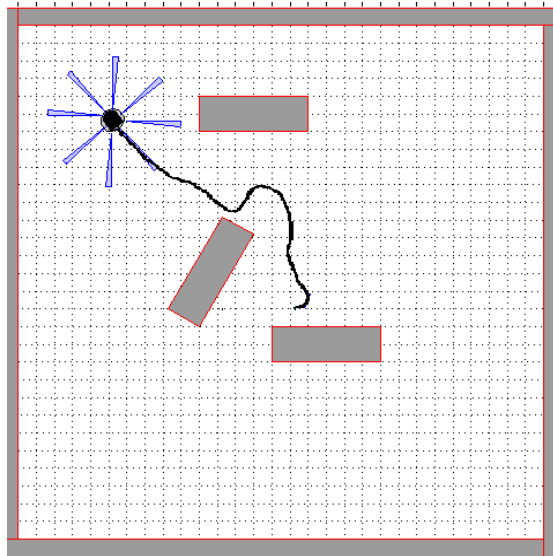


Şekil 45. Simülasyon ortamı



Şekil 46. PID tabanlı İnocop robotun izlediği yol

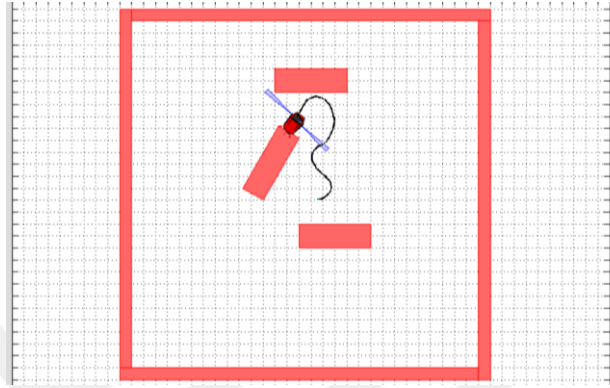
Şekil 47’da önerilen gauss denetçisinin kullanan İnocop robotun izlediği yol ve etkinliği gösterilmiştir.



Şekil 47. Gauss tabanlı İnocop robotun izlediği yol

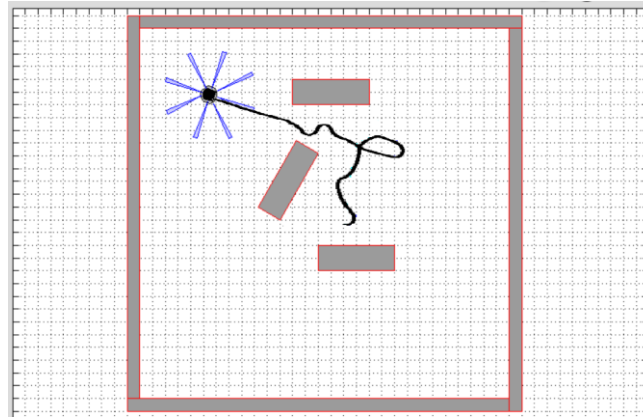
Şekil 46 ve Şekil 47 birlikte incelendiğinde Gauss tabanlı İnocop robot, PID tabanlı Sim.I.Am robottan daha fazla yol katetmesine rağmen engellerden kaçarak hedefe başarılı bir şekilde ulaştığı gözlemlenmiştir.

Sim.I.Am uygulamasında görülen eksikliklerden birinin Şekil 48’de görüldüğü gibi robotun, engellerin yeri değiştiği ortamda engellerden kaçamaması ve engele çarpması olduğu gözlemlenmiştir.



Şekil 48. Gauss tabanlı İnocop robotun izlediği yol

Şekil 49’de İnocop robotun engeller arası mesafe daraldığında izlediği yol gösterilmiştir. İnocop robot engellerden kaçmak ve doğrultusunu düzeltmek için fazla yol olsa da hedefe başarılı bir şekilde ulaşmıştır.



Şekil 49. Gauss tabanlı İnocop robotun izlediği yol

Tablo 10’da İnocop robotun ve orijinal Sim.I.Am robotun engelden kaçma hareketinde kullanılan algoritmalar ve ortamda bulunan engel durumları gösterilmiştir. Test için robotun (0,0) noktasından (-1,1) noktasına hareketi baz alınmıştır.

Tablo 10. Engel tabanlı hedefe ulaşma süreleri

	Engelsiz (sn.)	3 nolu engel (sn.)	2 nolu engel (sn.)	1 nolu engel (sn.)	2 ve 3 nolu engeller (sn.)	1 ve 2 nolu engeller (sn.)	1 ve 3 nolu engeller (sn.)	1 ve 2 ve 3 nolu engeller (normal) (sn.)	1 ve 2 ve 3 nolu engeller (Dar) (sn.)
Orijinal Sim.I.Am (PID)	9	9	10	10	11	11	10	11	7-çarptı
<i>İnocop</i> (PID)	10	10	11	10	12	12	10	12	14
Gaussian	10	10	10	10	11	12	11	13	18

Tablo 10 incelendiğinde;

- *İnocop* üzerine uygulanan yeni PID değerleriyle, gezgin robotun, engeller arası mesafelerin azaltıldığı ortamda engellerden kaçarak hedefe ulaşması sağlanmıştır. Yapılan bu değişikliğin hedefe ulaşma süresini %9 arttırdığı görülmüştür [54].
- *İnocop* üzerine PID yerine Gauss algoritması kullanıldığında robotun hedefe başarılı bir şekilde ulaştığı tespit edilmiştir. Ancak tüm engeller ortamda olduğunda robotun hedefe ulaşma süresi %18 oranında arttığı görülmüştür.
- Gauss tabanlı *İnocop* robot, Sim.I.Am robot ve PID tabanlı *İnocop* robottan daha fazla yol katetmiştir.
- Engeller arası mesafeler daraltıldığında Sim.I.Am robot hedefe ulaşamamıştır. PID tabanlı *İnocop* robot ve Gauss uygulanmış *İnocop* robot hedefe ulaşabilmiştir. PID yerine Gauss denetçiler kullanıldığında hedefe ulaşma süresinin %28 arttığı görülmüştür.

Çalışmanın bu bölümünde *İnocop* robot, Gauss denetçiler kullanılacak şekilde güncellenmiştir. Bildiğimiz kadarıyla çalışmanın bu kısmı literatürde hem hedefe ulaşmak hem de engellerden kaçmak için gauss denetçiler kullanan ilk çalışmadır.

Hedefe ulaşmada ortamın şartları ne olursa olsun Gauss denetçinin buna göre yapılandırılarak başarımın artması sağlanmıştır. Bu şekilde *İnocop*, engel konumlarının değiştiği ortamlarda kendi üzerinde hiçbir optimizasyona gerek kalmadan, ortama uyum sağlamıştır.

5. SONUÇ

İnocop, Sim.I.Am'ın daha geliştirilmiş ve optimize edilmiş bir modeli olarak ortaya çıkmıştır. Çalışmanın bu son bölümünde çalışma boyunca kullanılan tüm algoritmaların birbirine göre kıyaslaması yapılmıştır. Ortam koşullarına göre hangi denetçinin daha uygun olduğu tespit edilmiştir. Tablo 11'de çalışma boyunca kullanılan tüm denetçilerin, farklı senaryolar için yapılan uygulamalarının simülasyon ortamındaki test sonuçları bir bütün halinde gösterilmiştir ve yorumlanmıştır.

Tablo 11. Engel tabanlı hedefe ulaşma süreleri

	Engelsiz (sn.)	3 nolu engel (sn.)	2 nolu engel (sn.)	1 nolu engel (sn.)	2 ve 3 nolu engeller (sn.)	1 ve 2 nolu engeller (sn.)	1 ve 3 nolu engeller (sn.)	1 ve 2 ve 3 nolu engeller (normal) (sn.)	1 ve 2 ve 3 nolu engeller (Dar) (sn.)
Orijinal Sim.I.Am (PID)	9	9	10	10	11	11	10	11	7-çarptı
İnocop (PID)	10	10	11	10	12	12	10	12	14
Bulanık 3inputs Trapmf	9	9	11	10	11	11	11	12	6 çarptı
Bulanık 5 inputs trapmf	10	10	11	10	11	12	12	14	3 çarptı
Bulanık 5 inputs Gauss	10	10	11	11	12	12	10	13	3 çarptı
Bulanık 5 inputs Gauss2	10	10	11	10	11	12	11	13	4 çarptı
Bulanık 5 inputs trimf	9	9	11	10	11	12	11	13	4 çarptı
Gaussian	10	10	10	10	11	12	11	13	18

Tablo 11 incelendiğinde;

- *İnocop* üzerine uygulanan yeni PID değerleriyle, gezgin robotun, engeller arası mesafelerin azaltıldığı ortamda engellerden kaçarak hedefe ulaşması

sağlanmıştır. Yapılan bu değişikliğin hedefe ulaşma süresini %27 arttırdığı görülmüştür.

- Bulanık algoritmalarında daha geniş kapsamlı kural seti oluşturulduğunda hedefe ulaşma başarısının da doğru orantılı olarak arttığı görülmüştür.
- PID katsayıları iyi kurgulandığında hedefe ulaşmada en iyi başarımın elde edildiği gözlemlenmiştir.
- Ortamdaki engeller sürekli yer değiştiriyorsa PID katsayılarının optimizasyonu zaman alacağından bu tür ortamlarda bulanık veya Gaussian algoritmalarından birinin tercih edilmesi gerektiği görülmüştür. Eğer kural setleri iyi kurgulanırsa en iyi başarımın bulanık algoritması kullanılarak elde edildiği gözlemlenmiştir.

Çalışmada Sim.I.Am çalışmasında görülen engel konumunun değişti ortamda robotun engelden kaçamayarak çarpışma problemi üzerine yoğunlaşmıştır. hem geliştirilen PID denetçiler ile hem de yeni tasarlanan Gauss denetçiler ile bu problem çözüme kavuşturulmuştur.

Yapılan tez kapsamında, 1 adet uluslararası indekslerde taranan dergi yayını ve 1 adet uluslararası kongre bildirisi yapılmıştır. 3 adet uluslararası kongre bildirisi kabul edilmiştir. 1 adet uluslararası indekslerde taranan dergi yayını ile 1 adet Science Citation Index (SCI) kriterlerine uygun dergi yayını hazırlanmış ve yayımlanması için ilgili dergilere gönderilmesi planlanmıştır.

Mobil robotların yönlendirilmesi için kullanılan algoritma tabanlı denetçiler, robotların hedeflere ulaşmalarını çok yönlü bir şekilde etkilemektedir. Robot kontrol görevlerinde kullanılan algoritmaların geliştirilmeye açık olması bu tez konusunun belirlenmesinde etkili olmuştur. Bu tezde sunulan yöntemler geliştirilerek, mobil robotların dış ortamlarda kullanılması, hareketli nesnelerin bulunduğu bir ortamda robotların hareket etmesi, otonom sürüş modellerinin geliştirilmesi gibi literatürde açık olan problemlere çözüm önerileri sunulabilir.

İlerleyen çalışmalarda [54] kıyaslama tablosunda (Tablo 11) çıkan sonuçlara bir ağırlık algoritması uygulanarak kullanılan algoritmaların verimliliği hesaplanabilir. Ayrıca yeni bir ağırlıklandırma algoritması oluşturularak İnocop üzerine uygulanan algoritmaların başarı oranı ayrı ayrı hesaplanıp birbirleriyle kıyaslanabilir. PID, bulanık ve Gauss denetçiler beraber kullanılarak başarımın arttırılması hedeflenmektedir. Bu kapsamda hedefe ilerlerken Gauss denetçiler, engellerden kaçarak hedefe ilerlerken PID denetçiler, sadece engellerden kaçmak için bulanık

denetçinin kullanılması düşünülmektedir. Bu algoritmaların tek tek ve birlikte kullanılarak oluşturulan denetçinin çıktıları kıyaslama tablosuna (Tablo 11) eklenecek ve çıkan sonuçlara bir ağırlık algoritması uygulanarak kullanılan algoritmaların verimliliği hesaplanacaktır.



6. KAYNAKLAR

- [1] D. Vatansever, Fahri; ŞEN, *Genetik Algoritma Tabanlı PID Denetçi Simülatörü Tasarımı*, **Uludağ Üniversitesi Mühendislik-Mimarlık Fakültesi Derg.**, 2:18 (2013) 7–18.
- [2] E. Dönmez, A. F. Kocamaz, and M. Dirik, *A Vision-Based Real-Time Mobile Robot Controller Design Based on Gaussian Function for Indoor Environment*, **Arab. J. Sci. Eng.**, 43:12 (2018) 7127–7142.
- [3] A. Tozan, *Otonom Mobil Robot*, PhD Thesis, İstanbul Teknik Üniversitesi, 2007.
- [4] J. C. Latombe, *Robot motion planning*, **Kluwer Acad. Publ.**, (1991).
- [5] A. R. Willms and S. X. Yang, *An efficient dynamic system for real-time robot-path planning*, **IEEE Trans. Syst. Man Cybern. Part B-cybernetics**, 36 (4) (2006) 755–766.
- [6] S. . LaValle, *Planning Algorithms*, **Cambridge Univ. Pres.**, (2006).
- [7] V. B. Larin, *Motion Planning for a Wheeled Robot (Kinematic Approximation)*, **Int. Appl. Mech.**, 41 (2) (2005) 187–196.
- [8] J. P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, *Motion Planner for Nonholonomic Mobile Robots*, **IEEE Trans. Robot. Autom.**, 10 (5) (1994) 577–593.
- [9] Y. Hu and S. X. Yang, *A knowledge based genetic algorithm for path planning of a mobile robot*, **Proc. IEEE Int. Conf. Robot. Autom.**, (2004) 4350–5355.
- [10] E. Deniz, *Çoklu Robot Simülasyonu ve Kumanda Ortamı*, PhD Thesis, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2009.
- [11] M. Yüksek and S. İkizoğlu, *Bir Mobil Robotun Hedef Noktaya Erişimi Ve Toplanan Verilerin Rf Ile Transferi*, *feeb*. Elazığ: Fırat Üniversitesi, pp. 270–274, 2012.
- [12] V. Lumelsky, *Sensing, intelligence, motion : how robots and humans move in an unstructured world*, Wiley, 2006 .
- [13] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, *Obstacle avoidance of mobile robots using modified artificial potential field algorithm*, **EURASIP J. Wirel. Commun. Netw.**, 2019:1 (2019) 70.
- [14] D. Hager and Z. Dodds, *Robotic Motion Planning: Bug Algorithms*. Carnegie Mellon School of Computer Science.
- [15] R. Ronsse, *Mobile Robot Planning and Navigation*, *BEST2015-Autonomous Mobile Robots* . UCLouvain: École polytechnique de Louvain, 2015.
- [16] E. Çınar, O. Parlaktuna, A. Yazıcı, E.-E. M. Bölümü, O. Üniversitesi, and B. M. Bölümü, *Robot Navigasyonunda Potansiyel Alan Metodlarının Karşılaştırılması ve İç Ortamlarda Uygulanması*, (2007).
- [17] S. J. (Stuart J. Russell, P. Norvig, and J. Canny, *Artificial intelligence : a modern approach*, 2003 .
- [18] S. S. Ge and Y. J. Cui, *New potential functions for mobile robot path planning*, **IEEE Trans. Robot. Autom.**, 16:5 (2000) 615–620.
- [19] M. Dirik, A. F. Kocamaz, and E. Donmez, *Visual servoing based path planning for wheeled mobile robot in obstacle environments*, 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), (2017) pp.1–5.
- [20] M. Dirik, A. F. Kocamaz, and E. Donmez, *Static path planning based on visual servoing via fuzzy logic*, 2017 25th Signal Processing and Communications Applications Conference (SIU), (2017) pp.1–4.
- [21] O. Parlaktuna, A. Yazıcı, M. Özkan, and B. Mühendisliği Bölümü Eskişehir Osmangazi Üniversitesi, *Devinik İç Ortamların Tam Kapsanması Amacıyla*

- Gezgin Robot Rota Planlaması*, (2009).
- [22] R. J. C. Saclolo et al., *Machine vision system, robot hardware design and fuzzy controller design for autonomous multi-agent mobile robot platform*, 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), (2015) pp.1–6.
- [23] T.-Y. Wang and C.-D. Chang, *Hybrid Fuzzy PID Controller Design for a Mobile Robot*, 2018 IEEE International Conference on Applied System Invention (ICASI), (2018) pp.650–653.
- [24] S. K. Dwivedi, R. K. Meena, R. Upadhyay, and P. K. Padhy, *Tracking of mobile robot using hybrid controller*, 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), (2014) pp.1–4.
- [25] Anonymous, (2019). <https://www.k-team.com/khepera-iv> (on-line access on 21, May, 2019).
- [26] C. G. Rusu and I. T. Birou, *Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors*, **10th Int. Conf. Dev. Appl. Syst.**, (2010) 25–29.
- [27] Anonymus, *PID Denetçi ile Tasarım*, .
- [28] R. Siegwart and I. R. Nourbakhsh, *Autonomous Mobile Robots*, MIT Press, 2004 .
- [29] M. Egerstedt, (on-line access on).
- [30] H. Mo, Q. Tang, and L. Meng, *Behavior-Based Fuzzy Control for Mobile Robot Navigation*, **Math. Probl. Eng.**, 2013 (2013) 1–10.
- [31] Z. Güzen, *Kutup Yerleştirme Yöntemi ile Genelleştirilmiş Ayrık Pid Denetçi Tasarımı* , PhD Thesis, İstanbul Teknik Üniversitesi, 2003.
- [32] M. Fatih ÖZLÜK et al., *Matlab Gui ile DA Motor için PID Denetleyicili Arayüz Tasarımı*, **J. Adv. Technol. Sci.**, 3:3 (2013).
- [33] İ. Kocaarslan, (2019). <https://docplayer.biz.tr/29598426-Kontrol-sistemleri-prof-dr-ilhan-kocaarslan.html> (on-line access on 21, May, 2019).
- [34] O. M. Aki, *Robot Bilimi Robot Kontrol Sistemleri*, .
- [35] F. Nur Deniz et al., *SAÜ Fen Bil Der 18. Cilt, 3. Sayı, s*, (2014).
- [36] H. Develi, *Ziegler Nichols Yöntemi Ve Migo Yaklaşımı Yüksek Lisans Tezi* , PhD Thesis, istanbul Teknik Üniversitesi, .
- [37] F. Nur Deniz et al., *Kapalı çevrim PID denetçiler tasarımında birim basamak cevabı çoklu ölçüt performans haritalaması*, **SAÜ Fen Bil Der**, 18:3 (2014) 157–165.
- [38] K. Ogata, *Modern control engineering*, Prentice Hall, (2002).
- [39] A. Visioli, *Practical PID control*, Springer, 2006 .
- [40] Kiam Heong Ang, G. Chong, and Yun Li, *PID control system analysis, design, and technology*, **IEEE Trans. Control Syst. Technol.**, 13:4 (2005) 559–576.
- [41] S. Singh and M. Kaur, *Gain Scheduling of PID Controller Based on Fuzzy Systems*, **MATEC Web Conf.**, (2016).
- [42] İ. Coşkun and H. Terzioğlu, *Hız Performans Eğrisi Kullanılarak PID Parametrelerinin Gerçek Zamanda Belirlenmesi*, in *aedibus B.G. Teubneri*, 1913 .
- [43] L. A. Zadeh, *Fuzzy sets as a basis for a theory of possibility*, **Fuzzy Sets Syst.**, 100 (1999) 9–34.
- [44] Anonymous, (2019). <http://www.wikizero.biz/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEu b3JnL3dpa2kvQnVsYW7EsWtfbWFudMSxaw> (on-line access on 21, May, 2019).

- [45] Ç. Ergüven, *Bulanık Mantık Denetçi ile Klasik PID Denetçi Algoritmalarının Karşılaştırılması* , PhD Thesis, İstanbul Teknik Üniversitesi, 1999.
- [46] H. R. Özçalık, *Güneş Panellerinin Elektrik Üretiminde Verimli Kullanımı İçin Optimizasyon Yöntemleri* , PhD Thesis, Kahramanmaraş Sütçü İmam Üniversitesi, 2018.
- [47] A. Martinez, E. Tunstel, and M. Jamshidi, *Fuzzy Logic Based Collision Avoidance For a Mobile Robot*, (1994).
- [48] X. Li and B.-J. Choi, *Design of Obstacle Avoidance System for Mobile Robot using Fuzzy Logic Systems*, **Int. J. Smart Home**, 7:3 (2013).
- [49] Ş. Saime, *Bulanık Kontrol ve Uygulamaları* , PhD Thesis, İstanbul Teknik Üniversitesi, 1999.
- [50] T. Jin, *Obstacle Avoidance of Mobile Robot Based on Behavior Hierarchy by Fuzzy Logic*, **Int. J. Fuzzy Log. Intell. Syst.**, 12:3 (2012) 245–249.
- [51] B. Kasko, *No TiNeural Network and Fuzzy Systems*, **Acad. Press**, (1993).
- [52] E. Cox, *Fuzzy Fundamentals*, :Spectrum (1992) 58.
- [53] T. Takeuchi, Y. Nagai, and N. Enomoto, *Fuzzy control of a mobile robot for obstacle avoidance*, **Inf. Sci. (Ny)**, 45:2 (1988) 231–248.
- [54] C. H. Kazancı and A. F. Kocamaz, *PID Optimization on Differential Drive Mobile Robot*, 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), (2018) pp.1–6.

ÖZGEÇMİŞ

Adı: Cihangir Han

Soyadı: Kazancı

Eğitim: Yüksek Lisans

Alanı: Bilgisayar Mühendisliği

Lisans: Kocaeli Üniversitesi - 2012

Çalışma Alanları:

- Robotik
- Yapay Zeka
- Veri İşleme
- Makine Öğrenmesi

Yayınlar

C. H. Kazancı and A. F. Kocamaz, PID Optimization on Differential Drive Mobile Robot, 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), (2018) pp.1–6.