

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI

**DİFERANSİYEL SÜRÜŞ SİSTEMLİ MOBİL ROBOTLARIN ÇOK SAYIDA SABİT
VE HAREKETLİ ENGELLER ARASINDAKİ HAREKET PLANLAMASI VE HİPER
GEREĞİNDEN ÇOK SERBESTLİK DERECELİ ROBOTLARIN ÜÇ BOYUTLU
DAR ORTAMLARDA KESKİN MANEVRA KABİLİYETİNİN ARTTIRILMASI**

DOKTORA TEZİ

OKAN MİNNETOĞLU

DENİZLİ, EKİM - 2022

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI



**DİFERANSİYEL SÜRÜŞ SİSTEMLİ MOBİL ROBOTLARIN ÇOK SAYIDA SABİT
VE HAREKETLİ ENGELLER ARASINDAKİ HAREKET PLANLAMASI VE HİPER
GEREĞİNDEN ÇOK SERBESTLİK DERECELİ ROBOTLARIN ÜÇ BOYUTLU
DAR ORTAMLARDA KESKİN MANEVRA KABİLİYETİNİN ARTTIRILMASI**

DOKTORA TEZİ

OKAN MİNNETOĞLU

DENİZLİ, EKİM - 2022

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalışmalara atfedildiđine beyan ederim.

OKAN MİNNETOĐLU

ÖZET

DİFERANSİYEL SÜRÜŞ SİSTEMLİ MOBİL ROBOTLARIN ÇOK SAYIDA SABİT VE HAREKETLİ ENGELLER ARASINDAKİ HAREKET PLANLAMASI VE HİPER GEREĞİNDEN ÇOK SERBESTLİK DERECELİ ROBOTLARIN ÜÇ BOYUTLU DAR ORTAMLARDA KESKİN MANEVRA KABİLİYETİNİN ARTTIRILMASI

DOKTORA TEZİ

OKAN MİNNETOĞLU

PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: PROF. DR. ERDİNÇ ŞAHİN ÇONKUR)

DENİZLİ, EKİM - 2022

Bu tezde, hem mobil robotun hem de bu robotun üzerine yerleştirilen gereğinden/hiper-gereğinden çok serbestlik dereceli manipülatörün yol ve hareket planlaması problemleri üzerine çalışılmıştır. Çalışma iki kısımdan oluşmaktadır. Çalışmanın birinci kısmında, sabit ve hareketli engellerle dolu olan iki ve üç boyutlu hareket ortamlarında mobil robotun hareket planlamasının gerçekleştirilmesi amacıyla “Hayali Engellerle Güçlendirilmiş Güvenlik Çemberleri” (SC-IO) isminde yeni bir algoritma sunulmuştur. Öncelikle, algoritmada başlangıç noktasından hedef noktasına doğru bir global yol sadece sabit engellerin konumları dikkate alınarak potansiyel alan metodu aracılığıyla belirlenmiştir. Daha sonra, belirlenen yola güvenlik çemberleri eklenerek robotun tüm engellerden kaçınması ve güvenle hedefine ulaşabilmesi için gerekli olan hız değerleri hesaplanmıştır. Algoritmanın sınırları dâhilinde hiçbir çözümün bulunmadığı durumlarda, belirlenen yol hayali engeller yardımıyla güncellenerek hareket planlaması gerçekleştirilmiştir. Çalışmanın ikinci kısmında ise, sabit engellerle dolu ve dar alanlardan oluşan iki ve üç boyutlu hareket ortamlarında gereğinden/hiper-gereğinden çok serbestlik dereceli manipülatörün yol ve hareket planlaması için gerçek zamanlı yeni bir algoritma sunulmuştur. Öncelikle, hareket ortamında potansiyel alan metodu kullanılarak global yol elde edilmiştir. Daha sonra, ardışık iki yol noktasının oluşturduğu doğruya dik olacak şekilde ışın doğruları üretilmiştir. Son olarak, basit bir kontrol stratejisi uygulanarak belirlenen yol üzerinde hareket planlaması gerçekleştirilmiştir. Bu tezde sunulan örnek hareket simülasyonları geliştirilmiş olan her iki algoritmanın basit ve sağlam bir yapıya sahip olduklarını göstermektedir. Özellikle hiper manipülatör için üç boyutlu hareket ortamlarında geliştirilen metot literatürde bulunan metotlar ile kıyaslandığında, metodun manevra kabiliyetinin bu metotlara göre oldukça üstün olduğu bariz bir şekilde görülmektedir.

ANAHTAR KELİMELELER: Mobil robot, yol planlama, hareket planlama, yılanlı robot, hiper-gereğinden çok serbestlik dereceli manipülatör, engelden kaçınma.

ABSTRACT

THE MOTION PLANNING OF MOBILE ROBOTS WITH A DIFFERENTIAL DRIVE SYSTEM AMONG NUMEROUS STATIONARY AND MOVING OBSTACLES AND ENHANCING THE SHARP MANEUVERING ABILITY FOR HYPER REDUNDANT ROBOTS IN THREE-DIMENSIONAL TIGHT ENVIRONMENTS

PH.D THESIS

OKAN MINNETOGLU

PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE

MECHANICAL ENGINEERING

(SUPERVISOR:PROF. DR. ERDINC SAHIN CONKUR)

DENİZLİ, OCTOBER 2022

In this thesis, it has been studied on path and motion planning problems of both the mobile robot and the redundant/hyper-redundant manipulator which is placed on the mobile robot. The study consists of two parts. In the first part, a novel algorithm for mobile robot motion planning in two- and three-dimensional workspaces cluttered with stationary and moving obstacles called “Safety Circles with Imaginary Obstacles” (SC-IO) is presented. First, a global path from start point to the goal point is determined by using stationary obstacle positions via potential field method. Then, the speed values required for the robot to avoid all obstacles and reach its goal safely are calculated by placing safety circles on the global path. In cases where no solution can be found within the limits of the algorithm, the motion planning is achieved by updating the determined path with the help of imaginary obstacles. In the second part of the study, a novel real-time algorithm for the path and motion planning of the redundant/hyper-redundant manipulator in two- and three-dimensional workspaces cluttered with stationary obstacles and tight areas is presented. First, a global path is obtained by using potential field method in the workspace. Then, beam lines are generated in such a way that they will be perpendicular to the line formed by the two consecutive path points. Finally, the motion planning is achieved on the determined path by employing a simply control strategy. The exemplary motion simulations presented in this thesis show that the developed two algorithms have a simple and robust structure. Especially when the method developed for hyper-manipulator in three-dimensional workspaces is compared with the existing methods in the literature, it has been seen that the maneuverability of the method is quite superior to these methods.

KEYWORDS: Mobile robot, path planning, motion planning, snake robot, hyper-redundant manipulator, obstacle avoidance.

İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ.....	v
TABLO LİSTESİ	vii
SEMBOL LİSTESİ.....	viii
ÖNSÖZ.....	ix
1. GİRİŞ.....	1
2. LİTERATÜR ÇALIŞMASI.....	3
2.1 Mobil Robotların Yol Planlaması İle İlgili Çalışmalar	3
2.1.1 Grafik Araştırma Tabanlı Çalışmalar	3
2.1.2 Potansiyel Alan Tabanlı Çalışmalar	6
2.1.3 Hız Konisi Tabanlı Çalışmalar	9
2.1.4 Model Oluşturma Tabanlı Çalışmalar.....	11
2.1.5 Optimizasyon Tabanlı Çalışmalar	12
2.1.6 Literatürü Derleme Amacıyla Yapılan Çalışmalar	15
2.2 Manipülasyonların Yol Planlaması İle İlgili Çalışmalar	16
2.2.1 Potansiyel Alan Metodu Kullanılarak Yapılan Çalışmalar.....	16
2.2.2 Farklı Metotlar Kullanılarak Yapılan Çalışmalar	19
2.2.3 Eğri Uydurma Tabanlı Çalışmalar	22
2.2.4 Optimizasyon Tabanlı Çalışmalar	23
2.2.5 Kablo İle Tahrik Edilen Manipülasyonlar İle İlgili Çalışmalar	24
2.2.6 Literatürü Derleme Amaçlı Yapılan Çalışmalar	27
2.3 Çalışmanın Literatürdeki Yeri.....	28
3. POTANSİYEL ALAN METODU.....	30
3.1 Matematiksel İfadesi	30
3.1.1 İki Boyutlu Hareket Ortamı İçin Matematiksel İfade.....	30
3.1.2 İki Boyutlu Hareket Ortamı İçin Yol Belirleme Algoritması	33
3.1.3 Üç Boyutlu Hareket Ortamı İçin Matematiksel İfade.....	34
3.1.4 Üç Boyutlu Hareket Ortamı İçin Yol Planlaması	35
4. HAYALİ ENGELLERLE GÜÇLENDİRİLMİŞ GÜVENLİK	
ÇEMBERLERİ (SC-IO) ALGORİTMASI	37
4.1 Güvenlik Çemberi (SC) Aşaması	37
4.1.1 Tehlikeli Durumların Tespit Edilmesi	38
4.1.2 Geçerli Hızların Kontrol Edilmesi	40
4.1.3 Belli Hareketli Engellerin Dikkate Alınmaması.....	41
4.2 Hayali Engel (IO) Aşaması	42
5. GOAVOID YAZILIMI.....	45
5.1 SC-IO Algoritmasının Eklenmesi	45
5.1.1 Çalışma Alanının Oluşturulması.....	45
5.1.2 Hareketli Engellerin Eklenmesi	46
5.1.2.1 Hareketli Engel Parametrelerinin Ayarlanması	47
5.1.2.2 Engel İle Yol Kesişiminin Belirlenmesi	47
5.1.2.3 Güvenlik Çemberinin Eklenmesi	49
5.1.3 Hayali Engelin Eklenmesi	50

6. SC-IO ALGORİTMASININ SİMÜLE EDİLMESİ	51
6.1 Yüksek Boyutlu Hareket Ortamı	52
6.2 Orta Boyutlu Hareket Ortamı	55
6.3 Rastgele Parametrelili Hareket Ortamı	58
6.4 ORCA Metodu İle Karşılaştırılması	60
7. HİPER-GEREĞİNDEN ÇOK SERBESTLİK DERECELİ SERİ MANİPÜLATÖRÜN BİR MOBİL ROBOTUN ÜZERİNE YERLEŞTİRİLMESİ	65
7.1 İki Boyutlu Hareket Ortamı İçin Yol Planlama Algoritması	65
7.1.1 Global Yol Noktaları ve Işın Doğrularının Elde Edilmesi	65
7.1.2 Kritik Bölgelerin Belirlenmesi.....	67
7.1.3 Manipülatör İçin Güvenli Yolun Belirlenmesi	68
7.1.4 Kontrol Stratejisi.....	69
7.2 İki Boyutlu Hareket Ortamı İçin Simülasyonlar	70
7.3 Üç Boyutlu Hareket Ortamı İçin Yol Planlama Algoritması	74
7.3.1 Işın Çiftlerinin Elde Edilmesi	74
7.3.2 Kritik Bölgelerin Belirlenmesi ve Yön Değişimi	75
7.3.3 Işın Çifti Değişirme Noktalarının Belirlenmesi.....	76
7.4 Üç Boyutlu Hareket Ortamı İçin Simülasyonlar	77
7.5 Manevra Kabiliyetinin Matematiksel Olarak Belirlenmesi.....	87
8. SONUÇ VE ÖNERİLER	89
9. KAYNAKLAR.....	92

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Her bir hareket için belirlenen yollar.	29
Şekil 3.1: Potansiyel alanda mobil robotun hedefini bulması.	30
Şekil 3.2: Potansiyel alanın üç boyutlu görüntüsü.	31
Şekil 3.3: P noktasının iki boyutlu bir alanda gösterilmesi.	31
Şekil 3.4: Alanın aşağıdan yukarıya doğru hesaplanması.	33
Şekil 3.5: İki boyutlu hareket doğrultusunun hesaplanması.	34
Şekil 3.6: Üç boyutlu hareket doğrultusunun hesaplanması.	35
Şekil 3.7: Üç boyutlu hareket ortamında interpolasyon.	36
Şekil 4.1: Güvenlik çemberi üzerindeki pozisyonlar.	38
Şekil 4.2: Kritik yaklaşma açısının hesaplanması.	39
Şekil 4.3: İç içe geçmiş güvenlik çemberleri.	42
Şekil 4.4: Hareketli engel durumları a) Tehlikeli durum, b) Hayali engel ile güvenli durum.	43
Şekil 4.5: SC-IO algoritmasının akış diyagramı.	44
Şekil 5.1: Çalıştırma alanı oluşturma ayar paneli.	45
Şekil 5.2: Oluşturulmuş çalışma alanı.	46
Şekil 5.3: Engelin hız ve doğrultu ayarlama ekranı.	47
Şekil 5.4: Işın doğrusu ile sabit engel doğrusunun kesişmesi.	48
Şekil 5.5: Güvenli mesafe.	50
Şekil 5.6: Hayali engelin eklenmesi.	50
Şekil 6.1: Yüksek boyutlu hareket ortamı.	53
Şekil 6.2: Yüksek boyutlu hareket ortamında hareket planlaması.	54
Şekil 6.3: Orta boyutlu hareket ortamında yol planlaması a) Hayali engel (IO) aşaması aktif değilken, b) Hayali engel (IO) aşaması aktif iken.	56
Şekil 6.4: Orta boyutlu hareket ortamında hareket planlaması.	57
Şekil 6.5: Rastgele parametrelili hareket ortamı.	58
Şekil 6.6: Rastgele parametrelili hareket ortamında hareket planlaması.	60
Şekil 6.7: Yüksek boyutlu hareket ortamı için iki metodun hızlarının karşılaştırılması.	61
Şekil 6.8: Orta boyutlu hareket ortamı için ORCA metodu ile yapılan yol planlaması.	62
Şekil 6.9: ORCA metodu ile yapılan hareket planlaması.	63
Şekil 6.10: Orta boyutlu hareket ortamı için iki metodun hızlarının karşılaştırılması.	63
Şekil 6.11: Rastgele parametrelili hareket ortamında ORCA metodu yardımıyla yapılan yol planlaması.	64
Şekil 6.12: Rastgele parametrelili hareket ortamı için iki metodun hızlarının karşılaştırılması.	64
Şekil 7.1: Global yol noktaları ve temel tanımlamalar.	66
Şekil 7.2: Işın doğrusu tanımlamaları.	66
Şekil 7.3: Yol belirleme tanımlamaları.	67
Şekil 7.4: Manipülatör için belirlenen yol.	68
Şekil 7.5: Manipülatörün yol üzerine yerleştirilmesi.	69
Şekil 7.6: İki boyutlu hareket ortamı.	70

Şekil 7.7: Mobil robotun ve üzerindeki manipülatörün hareketi a) İlk sabit ve hareketli engellerden kaçınması, b) İkinci sabit ve hareketli engellerden kaçınması, c) Manipülatörün yol planlaması.....	71
Şekil 7.8: Dar bölgelerde manipülatörün hedefine güvenle ulaşması.	71
Şekil 7.9: Kritik bölgeler ve ışın doğruları.	72
Şekil 7.10: Manipülatörün uç noktasının takip etmesi gereken yol.....	72
Şekil 7.11: Hareketi boyunca manipülatörün süpürmesi gereken alanlar.	73
Şekil 7.12: Üç boyutlu hareket ortamında potansiyel alan metodu uygulaması (Engeller gösterilmemiştir).	74
Şekil 7.13: Işın çiftleri a) Bir adet ışın çifti, b) İki adet ışın çifti, c) Üç adet ışın çifti, d) Dört adet ışın çifti.	75
Şekil 7.14: Tek ışın çifti üzerindeki kritik bölgeler ve noktalar.	75
Şekil 7.15: İki ışın çifti üzerindeki kritik bölgeler ve noktalar.	76
Şekil 7.16: İlk örneğin üç boyutlu çalışma ortamı.	77
Şekil 7.17: Mobil robot üzerinde manipülatörün hareketi.	78
Şekil 7.18 : Manipülatör hareketinin farklı açılardan görüntüleri a) Hareketin başlangıcı, b) Işın çiftinin değiştirilmesi, c) Yardımcı elemanlar olmadan dar alan manevralarının yapılması, d) Hareketin bitişi. ..	80
Şekil 7.19: Işın çiftleri ve manipülatör yolunun gösterilmesi.....	81
Şekil 7.20: Kritik bölgeler.	82
Şekil 7.21: Üç boyutlu hareket ortamında dar alan manevralarının yapılması.	82
Şekil 7.22: İkinci örneğin üç boyutlu hareket ortamı.	83
Şekil 7.23: Manipülatörün mobil robot üzerinde taşınması.	84
Şekil 7.24: Manipülatörün dar bölgelerden manevra yaparak kaçınması.....	84
Şekil 7.25: Manipülatörün dar alandan güvenle geçmesi.	85
Şekil 7.26: Manipülatörün hedef noktasına ulaşması.	85
Şekil 7.27: Manipülatörün ikinci örnekteki alanda hareketi (Engeller gösterilmemiştir).	86
Şekil 7.28: Karşılaştırmalı değerlendirme için kullanılan hareket ortamı.	88

TABLO LİSTESİ

Sayfa

Tablo 6.1: İlk iki hareket ortamı için parametre değerleri.	52
Tablo 6.2: Yüksek boyutlu hareket ortamındaki engellerin bilgileri.	53
Tablo 6.3: Yüksek boyutlu hareket ortamında bulunan hareketli engellerin doğrultu ve güvenlik çemberi yarıçap değerleri.....	55
Tablo 6.4: Orta boyutlu hareket ortamındaki engellerin bilgileri.	55
Tablo 6.5: Orta boyutlu hareket ortamındaki hareketli engellerin doğrultu ve güvenlik çemberi yarıçap değerleri.	57
Tablo 6.6: Rastgele parametrelili hareket ortamındaki engellerin bilgileri.....	59
Tablo 6.7: Rastgele parametrelili hareket ortamındaki hareketli engellerin parametrelerinin değerleri.	60

SEMBOL LİSTESİ

θ	:	İki Boyutlu Alanda Robotun Hareket Etme Açısı
β	:	Üç Boyutlu Alanda Robotun Hareket Etme Açısı
n	:	İterasyon Sayısı
D	:	Hareketli Engel İle Mobil Robot Arasındaki Güvenli Mesafe
α_{cr}	:	Hareketli Engelin Kritik Yaklaşma Açısı
r	:	Güvenlik Çemberinin Yarıçapı
α	:	Hareketli Engelin Mevcut Yaklaşma Açısı
v_{max}	:	Mobil Robotun Maksimum Hızı
v_{ref}	:	Mobil Robotun Referans Hızı
t_{O1}	:	Hareketli Engelin O1 Noktasına Ulaşma Süresi
t_{P2}	:	Hareketli Engelin P2 Noktasına Ulaşma Süresi
t_{O2}	:	Hareketli Engelin O2 Noktasına Ulaşma Süresi
t_{P1}	:	Hareketli Engelin P1 Noktasına Ulaşma Süresi
d_p	:	Belirlenen Yolun İki Noktası Arasındaki Mesafe

ÖNSÖZ

Doktora tez çalışmam boyunca bana yaptığı rehberlikten dolayı danışman hocam Sayın Prof. Dr. Erdiñ Şahin ÇONKUR'a teşekkür eder, öğrencisi olmaktan dolayı mutluluk duyduğumu belirtmek isterim. Tez izleme komitesi üyeleri olan değerli hocalarım Sayın Doç. Dr. Yasin YILMAZ ve Sayın Doç. Dr. Ersin DEMİR'e çalışmam boyunca yaptıkları katkı ve önerileri için minnetlerimi sunarım. Ayrıca bilimsel çalışmalarına fikirleriyle destek olan değerli hocam Sayın Doç. Dr. Veysel ALKAN'a bu teze yaptığı katkılardan dolayı teşekkür ederim. Son olarak, gösterdikleri anlayış, sabır ve desteklerinden dolayı sevgili aileme ve her zaman bana destek olan arkadaşlarıma teşekkür ederim.

1. GİRİŞ

Bu tezde yapılan çalışmalar iki kısımdan oluşmaktadır. Birinci kısım da mobil robotların yol ve hareket planlama problemleri üzerine çalışma yapılırken, ikinci kısımda ise gereğinden/hiper-gereğinden çok serbestlik dereceli manipülatörlerin yol ve hareket planlama problemleri ele alınmıştır.

Mobil robotlar fiziksel bir noktaya sabitlenmemiş vaziyette karada, denizde ve havada olmak üzere tanımlanmış bir çevrede hareket ederek istenilen fonksiyonları yerine getirebilen robotlardır ve kısaca gezici robotlar olarak tanımlanmaktadır. Mobil robotlar temelde insanlar için zor, tehlikeli ve sıkıcı olan görevleri yerine getirmek amacıyla tasarlanmışlardır. Elektronik ve yazılımın gelişmesi ile gün geçtikçe kullanım alanları artmaktadır. Günümüzde mobil robotlar genellikle askeri, bilimsel araştırma ve endüstriyel üretimde insanlara yardımcı olmak amacıyla kullanılmaktadırlar. Askeri alanda mayın tespiti için kullanılan robotlar ve insansız hava araçları mobil robotlara örnek olarak verilebilmektedirler. Gerek deniz altında gerekse uzayda araştırma yaparak veri toplayan mobil robotlar da bilim insanlarına yardımcı olmaktadır. Endüstriyel üretim alanlarında da mobil robotlar aktif rol oynamaktadırlar.

Bu tezin birinci kısmında mobil robotların sabit ve hareketli engellerle dolu hareket ortamlarında, mevcut engellere çarpmadan belirlenmiş olan hareket başlangıç noktasından hedef noktasına doğru güvenle ulaşabilmesi için yol ve hareket planlama algoritması geliştirilmiştir. Bu algorithmada her engel için oluşturulan güvenlik çemberi yardımıyla robotun güvenli hızları belirlenerek hareket planlaması, gerektiğinde hayali engeller eklenerek ise yol planlaması gerçekleştirilmiştir.

Gereğinden/hiper-gereğinden çok serbestlik dereceli manipülatörler fiziksel bir noktaya sabitlenmiş, mafsallarla birbirine bağlanmış uzuvlardan (robot kolları) oluşan seri robotlardır. Yerde sürünme hareketi yapmamaktadır sadece havada çalışmaktadır. Ulaşılması zor, tehlikeli ve dar alanlarda istenilen görevleri yerine getirebilmektedirler.

Bu tezin ikinci kısmında, bu tür manipölatörlerin dar alanlardan gerekli manevraları yaparak belirli bir hareket başlangıç noktasından hedef noktasına sabit engellere çarpmadan ulaşabilmesini sağlamak amacıyla hem iki hem de üç boyutlu hareket ortamları için yol ve hareket planlama algoritması geliştirilmiştir. Bu algorithmada belirlenen yol noktalarından hem iki boyutlu hem de üç boyutlu hareket ortamlarında ışın doğrusu çizilerek, dar alanların ve bu alanlardan kaçınabilmek için gerekli olan manevraların yönleri tespit edilmiştir ve bu bilgiler kullanılarak yol ve hareket planlamaları gerçekleştirilmiştir.

Bu tezin amacı, mobil robotun üzerine konumlandırılmış bir gereğinden/hiper-gereğinden çok serbestlik dereceli manipölatörün ilgili görevi tamamlayabilmesi için, görevin gerçekleştirileceği hareket başlangıç noktasına bir diferansiyel sürüş sistemine sahip mobil robot üzerinde sabit ve hareketli engellerin bulunduğu bir alandan engellere çarpmadan ulaştırılması ve manipölatör için geliştirilmiş olan yol ve hareket algoritması ile iki ve üç boyutlu alanda sabit engeller ve dar alanlarla dolu hareket alanında hedefine ulaşabilmesini sağlamaktır. Bu amaç doğrultusunda, bu tezde, belli görevlerin yerine getirilmesinde günümüzde yaygın olarak kullanılan hem mobil robotlar, hem gereğinden/hiper gereğinden çok serbestlik dereceli manipölatörler incelenerek birlikte hareket etmeleri gerektiği durumlar da dikkate alınarak yol ve hareket planlama algoritmaları geliştirilmiştir.

2. LİTERATÜR ÇALIŞMASI

2.1 Mobil Robotların Yol Planlaması İle İlgili Çalışmalar

Mobil robotların yol planlaması problemi robotik alanında önemli araştırma alanlarından birisidir. Bu problem, belli bir başlangıç ve hedef noktasına sahip ve engellerle dolu olan bir alanda robotun güvenli bir şekilde hedefine ulaşması için takip etmesi gereken güvenli yolun belirlenmesi olarak tanımlanabilmektedir. Yapılan literatür taramasından mobil robotların yol planlaması üzerine bir çok akademik çalışmanın son yıllarda araştırmacılar tarafından sıkça makalelere konu edildiği görülmüştür. Ancak hareketli engellerin bulunduğu ortamların araştırıldığı az sayıda çalışma bulunmaktadır. Amaç, sabit ve hareketli engellerin birlikte bulunduğu ortamlar için bu çalışmayı yaparak literatürdeki bu boşluğu doldurmaktır. Yapılan literatür taraması aşağıda sunulmuştur.

2.1.1 Grafik Araştırma Tabanlı Çalışmalar

Hart ve diğ. (1968) yaptıkları çalışmada; sadece sabit engellerin bulunduğu ortamlarda yol planlamasının yapılması amacıyla A^* algoritmasını geliştirmişlerdir. Bu algorithmada hareket alanı düğümlere ayrılarak her bir düğüme ulaşılabilmesi ve o düğümden hedefe varılabilmesi için kat edilmesi gereken toplam mesafe değerleri dikkate alınarak en kısa yol belirlenmiştir. Bu algoritmanın temel olarak alındığı farklı algoritmalar birçok araştırmacı tarafından literatüre kazandırılmıştır. Bunlardan bazıları aşağıda verilmiştir.

Aine ve diğ. (2015) çalışmalarında; MHA^* (Multi-Heuristic A^*) algoritmasını geliştirmişlerdir. Bu algorithmada, A^* algoritmasında Hart ve diğ. (1968) hesaplanan katedilmesi gereken mesafe değerlerine ilave olarak sezgisel bir fonksiyon yardımıyla yol planlamasını gerçekleştirmişlerdir.

Zhang ve diğ. (2016) yaptıkları çalışmada; yol planlamasının yapılması için gerekli hesaplama işlemlerini azaltmak üzerine çalışmışlardır. Bu kapsamda bulanık

mantık (fuzzy logic) kullanılarak hareket alanındaki trafik miktarı tahmin edilmiştir ve A* algoritması ile hesaplama zamanları kıyaslanmıştır.

Kang ve diğ. (2016) çalışmalarında; Modified A* algoritmasını sunmuşlardır. Bu algorithmada sadece katedilmesi gereken mesafeleri dikkate alan A* algoritmasının, dönüş açı değerlerini de dikkate alacak şekilde kapsamı arttırılmıştır ve modüler kara taşımacılığında kullanılacak şekilde optimize edilmiştir.

Le ve diğ. (2018) yaptıkları çalışmada; hTetro isimindeki temizleme robotu için yol planlama algoritması geliştirmişlerdir. A* tabanlı olarak geliştirilen bu algoritmanın dar alanları da dikkate alması sağlanarak kapsama alanının arttırılması hedeflenmiştir.

Song ve diğ. (2019^a) çalışmalarında; insansız deniz araçları için Smoothed A* algoritmasını tanıtmışlardır. A* algoritmasının temel olarak alındığı bu algoritma ile A* algoritmasının kapsamı, kontrol edilebilen serbestlik derecesi toplam serbestlik derecesinden küçük olan (holonomik olmayan) deniz araçlarını da kapsayacak şekilde arttırılmıştır ve bu algoritma Springer isimli insansız deniz aracında kullanılarak deneysel sonuçlar rapor edilmiştir.

Sharma ve Dubay (2013) yaptıkları çalışmada; Anytime A* algoritmasını sunmuşlardır. Bu algorithmada, A* algoritmasında hesaplama için kullanılan hafıza ve zaman aralığı optimize edilerek hesaplama zamanının azaltılması amaçlanmıştır.

Likhachev ve diğ. (2004) çalışmalarında; ARA* (Anytime Repairing A*) algoritmasını tanıtmışlardır. Bu çalışmada yol planlaması yapılırken araştırma zamanı ayarlanarak optimum sonuç elde edildiği iddia edilmiştir. Algoritmalarını, Anytime A* algoritması Sharma ve Dubay (2013) ile karşılaştırarak bir manipülatör ve mobil robot için sonuçlarını rapor etmişlerdir.

Stern ve diğ. (2014) yaptıkları çalışmada; mobil robot yol planlamasında da kullanılabilen PTS (Potential Search) ve APTS/ANA* (Anytime Potential Search/Anytime Non-Parametric A*) olmak üzere iki adet algoritma sunmuşlardır. Bu algoritmaların kullanıcı değerlerine bağlı olmadığı ve çok hızlı bir şekilde sonuca ulaştığı belirtilmiştir.

Wang ve diğ. (2022) çalışmalarında; A* algoritmasını temel olarak alan EBS-A* isimli yol planlama algoritmasını geliştirmişlerdir. Mobil robot yol planlama problemini çözmek amacıyla geliştirdikleri algoritmada, A* algoritmasından farklı olarak belirlenen yolun engellerden daha uzak olmasını sağlamayı amaçlamışlardır. Anlık hesaplamalar ile belirlenen yoldaki keskin dönüşlerin de bu sunulan yeni algoritma ile azaldığını iddia etmişlerdir. A* algoritması ile geliştirdikleri algoritmalarını karşılaştırarak sonuçlarını raporlamışlardır.

Zhang ve diğ. (2022) yaptıkları çalışmada; dört tekerlekli bir mobil robotun engebeli arazi üzerinde engelden kaçınarak hedefine ulaşabilmesi için geliştirdikleri yol planlama algoritmasını geliştirmişlerdir. A* algoritması ile yol planlamasını yaparken, Q-Öğrenme (Q-Learning) algoritması ile de engellerden kaçınmasını sağlamışlardır. Algoritmalarını hem bilgisayar ortamında simüle edip, hem de gerçek bir engebeli arazi üzerinde dört tekerlekli bir mobil robot üzerinde uygulamışlardır. A* algoritması ile geliştirdikleri algoritmayı karşılaştırıp sonuçlarını rapor halinde sunmuşlardır.

Stentz (1993) çalışmasında; Dynamic A* (D*) algoritmasını sunmuştur. Hareketli engellerin bulunduğu ortamlarda yol planlamasının yapılabilmesi için A* algoritması iyileştirilmiştir. Bu algoritmada katedilmesi gereken mesafeler bir listede kaydedilmiş ve hareketli engelin her hareketinde bu listenin güncellenmesi ile yol planlaması gerçekleştirilmiştir. Değerler devamlı olarak güncellendiği için yol planlaması bu algoritma ile uzun sürmektedir. Bu eksikliği gidermek amacıyla Focused D* algoritmasını tanıtmıştır (Stentz 1995). Bu algoritma ile yol planlama süresinin optimize edilmesi hedeflenmiştir.

Koenig ve Likhachev (2002) çalışmalarında; D* Lite algoritmasını geliştirmişlerdir. Focused D* algoritması Stentz (1995) ile benzer sonuçlar verdiği fakat algoritma olarak farklılıklar içerdiği belirtilmiştir. Algoritmanın avantajları ve deneysel sonuçları rapor olarak verilmiştir.

Przybylski ve Putz (2017) yaptıkları çalışmada: D* Extra Lite algoritmasını tanıtmışlardır. Bu algoritmanın yol planlaması yapılan alandaki değişiklik yapılacak olan kısımlardaki hesaplamalar konusunda D* Lite algoritmasından Koenig ve

Likhachev (2002) 1,08-1,94 kat daha hızlı olduğu belirtilmiştir. Deneysel olarak iki algoritma karşılaştırılarak sonuçları rapor edilmiştir.

Likhachev ve diğ. (2005) çalışmalarında: AD* (Anytime Dynamic A*) algoritmasını sunmuşlardır. Bu algoritma ile belli bir zaman aralığında hesaplamalar yapılarak ve daha önce yapılan hesaplamalar dikkate alınarak sonuçların düzeltilmesi ve D* algoritmasının yol planlama süresinin azaltılması hedeflenmiştir.

Ferguson ve Stentz (2006) yaptıkları çalışmada; Field D* algoritmasını tanıtmışlardır. Bu algorithmada katedilmesi gereken mesafeler lineer interpolasyon yardımıyla hesaplanarak D* algoritmasının ihtiyaç duyduğu hafıza ve işlem süresinin azaltılması amaçlanmıştır.

Sun ve diğ. (2010) çalışmalarında; D* Lite algoritmasına Koenig ve Likhachev (2002) eklemeler yaparak Moving Target D* Lite algoritmasını sunmuşlardır. Bu algoritma ile hedef noktasının devamlı değiştiği ortamlar için de yol planlamasının yapılabilmesi amaçlanmıştır.

Oral ve Polat (2016) yaptıkları çalışmada; MOD* Lite (Multiobjective D* Lite) algoritmasını tanıtmışlardır. Sadece yol uzunluğunun optimize edilerek yol planlamasının gerçekleştirildiği D* Lite algoritması iyileştirilerek farklı parametreleri de optimize edebilecek kapasiteye ulaştırılmıştır.

Przybylski (2018) çalışmasında; D* Extra Lite Przybylski ve Putz (2017) ile ARA* Likhachev ve diğ. (2004) algoritmalarını birleştirerek geliştirdiği AD*-Cut (Anytime D*-Cut) isimli algoritmasını tanıtmıştır. AD* algoritması Likhachev ve diğ. (2005) ile karşılaştırarak hesaplamaları daha hızlı gerçekleştirdiğini ve daha kısa yollar elde ettiğini deney sonuçlarını rapor ederek belirtmiştir.

2.1.2 Potansiyel Alan Tabanlı Çalışmalar

Khatib (1986) çalışmasında; sabit engellerden oluşan bir çevrede mobil robotun yol planlamasının yapılabilmesi için matematiksel bir modele sahip olan potansiyel alan metodunu sunmuştur. Bu matematiksel modele göre hareket alanında

bulunan engellerin robota itme kuvveti, ulaşmak istenen hedefin ise çekme kuvveti uyguladığı belirtilmiştir.

Ge ve Cui (2002) yaptıkları çalışmada; Potansiyel alan metodundaki Khatib (1986) engellerin robota itme, hedefin de çekme kuvveti uygulaması durumunu yeniden modelleyerek, hareketli engellerin ve hareketli hedefin bulunduğu dinamik ortamlarda da robotun yol planlamasının yapılabilmesi için yeni bir konsept tanıtmışlardır.

Poty ve diğ. (2004) çalışmalarında; kısmi potansiyel alan metodunu tanıtmışlardır. Bu metotta, çevrede bulunan hareketli engellerin tehlikeli durumları dikkate alınarak kısmi olarak potansiyel alan hesaplanmıştır.

Ko ve Lee (1996) yaptıkları çalışmada; robotun engele çarpma olasılığını belirleyen Kaçınabilme Ölçütü (Avoidability Measure) adında yeni bir konsept sunmuşlardır. Engel ile robot arasındaki mesafeye ve engelin robota göre bağlı hızına bağlı olan bir hayali uzaklık fonksiyonu tanımlanmıştır. Tanımlanan yeni konsept ve hayali uzaklık fonksiyonu dikkate alınarak potansiyel alan metodu kullanılarak robotun yol planlaması gerçekleştirilmiştir.

Qixin ve diğ. (2006) çalışmalarında; yol planlama problemini hem engellerin hem de hedef noktasının hareket ettiği robotik futbol senaryosu için incelemişlerdir. Robot, engele belirli bir mesafede yaklaştığında, buna bağlı olarak bir tehdit fonksiyonu belirlenerek potansiyel alan metodu yardımıyla yol planlaması yapılmıştır.

Li ve diğ. (2017) yaptıkları çalışmada; potansiyel alan metodunu kullanarak üç tekerlekli çok yönlü hareket edebilen bir mobil robotun engellerden kaçınabilmesi için gerekli olan yol planlamasını gerçekleştirmişlerdir. Robotun kinematik ve dinamik modelleri dikkate alınarak robotun hareketinin kontrolü üzerine çalışılmıştır.

Raja ve Dutta (2017) çalışmalarında; A^* ve potansiyel alan metodunu birlikte kullanarak, hareketli engellerin bulunduğu engebeli arazilerde mobil robotun yol planlamasını gerçekleştirmişlerdir. Robotun üzerine monte edilmiş lazer ışıkları ile arazinin haritası çıkarıldıktan sonra A^* algoritması kullanılmıştır. Daha sonra sensör verileri kullanılarak tespit edilen hareketli engeller için daha önce elde edilen yol potansiyel alan metodu ile modifiye edilmiştir.

Zhang (2018^a) yaptığı çalışmada; kaotik potansiyel alan metodunu (CAPF) tanıtmıştır. Bu metotta, potansiyel alan metodunun engellerin itme ve hedefin çekme miktarını hesaplayan fonksiyonlarını modifiye ederek ve bu bilgileri kaotik optimizasyon algoritması ile optimize ederek sabit engellerin bulunduğu ortamlar için mobil robot yol planlamasını gerçekleştirmiştir.

Zhu ve diğ. (2020) çalışmalarında; yol planlama problemini çözebilmek amacıyla potansiyel alan metodu ile karınca kolonisi optimizasyonu algoritmalarını birleştirerek yeni bir yol planlama algoritması sunmuşlardır. Algoritmalarında, karınca kolonisi optimizasyonu algoritmasının planladığı yolu daha da geliştirmek için potansiyel alan metodunu kullanmışlardır. Bilgisayar ortamında üç boyutlu bir hareket alanında insansız hava aracı için yol planlamasını simüle ederek, sonuçlarını rapor olarak bildirmişlerdir.

Chen ve diğ. (2021) yaptıkları çalışmada; potansiyel alan metodu Khatib (1986) ve A* algoritmasını Hart ve diğ. (1968) birleştirerek mobil robotların engellerden kaçınarak belirli bir başlangıç noktasından belirli bir hedef noktasına güvenle ulaşabilmesi için bir yol planlama algoritması geliştirmişlerdir. Potansiyel alan metodu ile hedefe doğru giden en kısa yolu hesaplarken, A* algoritması ile engellerden kaçınmayı sağlamışlardır. Algoritmalarını bilgisayar ortamında simüle ederek, sonuçlarını paylaşmışlardır.

Montiel ve diğ. (2015) çalışmalarında; potansiyel alan metodu ile bakteri gelişim algoritmasını birleştirerek bakteriyel potansiyel alan metodu (BPF) isminde yeni bir yol planlama algoritmasını tanıtmışlardır. Sabit ve hareketli engellerin bulunduğu ortamlarda simülasyonlar gerçekleştirilmiştir ve sonuçlar üç adet potansiyel alan metodu türevi ile karşılaştırılmıştır.

Orozco-Rosas ve diğ. (2019) yaptıkları çalışmada; biyolojiden esinlenerek mobil robot yol planlaması problemine bir çözüm getirebilmek amacıyla zar tabanlı potansiyel alan metodunu (memEAPF) sunmuşlardır. Bu yeni metotlarını kullanarak statik ve dinamik ortamlarda simülasyonlar yapmışlardır ve sonuçlarını potansiyel alan metodunun türevi olan üç farklı metot ile karşılaştırmışlardır.

2.1.3 Hız Konisi Tabanlı Çalışmalar

Fiorini ve Shiller (1998) yaptıkları çalışmada; VO (Velocity Obstacles) adında bir metot tanıtmışlardır. Bu metotta, robotun engeller ile çarpıştığı ve çarpışmadığı hız değerleri hız konisi kullanılarak belirlenmiştir. Bu bilgiler doğrultusunda robotun hareketinde attığı her adımda engellerden kaçınmak için gerekli olan hız değeri ve manevra belirlenmiştir. Bu şekilde belirlenen hareketin salınımlı olduğu gözlenmiştir.

Van Den Berg ve diğ. (2008) çalışmalarında; robotun gerçek zamanlı yol planlamasını gerçekleştirebilmek amacıyla RVO (Reciprocal Velocity Obstacle) metodunu sunmuşlardır. Bu metot ile belirlenen yolda VO metodu Fiorini ve Shiller (1998) ile belirlenen yoldaki salınımlı olma probleminin giderildiği belirtilmiştir.

Van Den Berg ve diğ. (2011) yaptıkları çalışmada; RVO metodunun Van Den Berg ve diğ. (2008) kapsamını genişleterek ORCA (Optimal Reciprocal Collision Avoidance) metodunu tanıtmışlardır. Bu metot ile robot yol planlama problemini düşük boyutlu bir lineer probleme indirgemişlerdir ve sonuç olarak daha az salınımlı ve kesintisiz bir yol elde ettiklerini rapor etmişlerdir.

Snape ve diğ. (2011) çalışmalarında; salınımlı harekete sebep olan etmenlerin oluşumunu önlemek amacıyla VO ve RVO metotlarını birleştirerek HRVO (Hybrid Reciprocal Velocity Obstacle) metodunu sunmuşlardır. Bu metotta, güvenli yolun belirlenebilmesi amacıyla robotun mevcut pozisyon ve hızları dikkate alınmıştır.

Liu ve diğ. (2018) yaptıkları çalışmada; VO metodunu temel alan bir algoritma tanıtmışlardır. Bu algoritma ile robot ile engeller arasında oluşabilecek olası çarpışmaları tahmin etmeye çalışarak yol planlamasını gerçekleştirmişlerdir.

Gyenes ve Szadeczky-Kardoss (2018) çalışmalarında; VO metodunu temel alan SVO (Safe Velocity Obstacle) metodunu sunmuşlardır. Bu metot ile engellerin ve robotun mevcut konum ve hız bilgileri yardımıyla, gerekli olan kaçınma manevralarını planlayarak robotun yol planlamasını gerçekleştirmişlerdir.

Owen ve Montano (2005) yaptıkları çalışmada; robotun sınırlarını dikkate alarak hareket alanının dinamiğini modelleyebilmek amacıyla DVS (Dynamic Velocity Space) metodunu tanıtmışlardır. Metotlarında çarpışma zamanını

hesaplamak ve herhangi bir çarpışmayı önlemek için tahmini varış zamanı bilgisini kullanmışlardır.

Seder ve Petrovic (2007) çalışmalarında; hem dönme hem de öteleme hızları olan dairesel bir yörüngeyi takip eden robotlar için DWA (Dynamic Window Approach) isimli yeni bir yöntem sunmuşlardır. Bu yöntemde, kabul edilebilir ve ulaşılabilir hızlar, hız uzayından seçilmiştir. Kabul edilebilir hızlar robotun engellerle çarpışmamasını sağlamakta iken, ulaşılabilir hızlar ise robotun dinamik ve kinematik kısıtlama gereksinimlerini karşılamaktadır.

Fraichard ve Asama (2004) yaptıkları çalışmada; robotun engelle çarpışıp çarpışmayacağını belirten IC (Inevitable Collision) durumunu belirleyen bir yöntem tanıtmışlardır. Bu yönteme göre, robotun kaçınılmaz bir çarpışma (IC) durumunda olduğunda gelecekte bir engelle çarpışacağını, aksi takdirde ise çarpışmasız bir yol sağlayan robot için en az bir kontrol girişinin mevcut olacağını belirtmişlerdir.

Alonso-Mora ve diğ. (2018) çalışmalarında; RVO metodunu Van Den Berg ve diğ. (2008) temel alarak geliştirdikleri epsilon CCA metodunu sunmuşlardır. Bu metodun çok sayıda robotun veya insanların bulunduğu ortamlarda hareketli engellerden kaçınarak yol planlamasının gerçekleştirilmesinde etkili olduğunu belirtmişlerdir. Algoritmalarını mevcut olan türevleri ile karşılaştırarak sonuçları rapor etmişlerdir.

Kufoalor ve diğ. (2018) çalışmalarında; RVO metodunun eksikliklerini giderebilmek amacıyla yeni bir metod geliştirmişlerdir. Bu metodlarını otonom deniz aracının hareketli engellerden kaçınabilmesi için gerekli yol planlamasını yapılabilmesi için kullanmışlardır. Geliştirdikleri metodun performansını, VO ve RVO metodlarının performansları ile karşılaştırarak sonuçları rapor halinde sunmuşlardır.

Xu ve diğ. (2020) çalışmalarında; hız konisi tabanlı metodların, mobil robotun ulaşabileceği maksimum hızı aşan hareketli engellerden kaçınmayı tam olarak sağlamadıklarını belirterek, bu problemi çözmek amacıyla MVO (Maximum-Speed Aware Velocity Obstacle) algoritmalarını geliştirmişlerdir. Algoritmalarında, mobil robotun kinematik kısıtlamalarını dikkate alarak hareketin belli bir alanında sensörler ile ölçümler yapmışlardır. Hem bilgisayar ortamında hem de fiziksel olarak

algoritmalarını uygulamışlardır. ORCA metodu Van Den Berg ve diğ. (2011) ile algoritmalarını karşılaştırarak sonuçlarını rapor halinde sunmuşlardır.

Gu ve Huang (2021) yaptıkları çalışmada; VO (Velocity Obstacles) metodunun Fiorini ve Shiller (1998) uygulanmasında robot üzerindeki sensörün ölçüm sonuçlarındaki hatalardan kaynaklanan engel ile çarpışma problemini incelemişlerdir. Bu problemin çözümü amacıyla AVO (Adaptive Velocity Control) algoritmalarını tanıtmışlardır. Algoritmalarında, engelin hız ve mesafesine göre hız konisinin yarıçap değerini devamlı olarak güncellemişlerdir. Bu sayede, sensörlerin ölçümünden kaynaklanan hataların giderildiğini ifade ederek, algoritmalarını VO algoritması ile karşılaştırarak sonuçlarını paylaşmışlardır.

Gyenes ve Szadeczky-Kardoss (2022) çalışmalarında; Mobil robotların hareketli engellerden kaçınmak için VO metodu ile hiçbir güvenli hız değerinin bulunamadığı durumların olduğundan bahsetmişlerdir. Bu tür durumlarda robotun ani durmasının gerektiğini ve engeller ile çarpışmanın kaçınılmaz olduğunu belirtmişlerdir. Çarpışmalarda robot ile engelin daha az hasar alması için, bu tür durumlarda robotun hızını belirleyen CVO (Collidable Velocity Obstacles) algoritmalarını sunmuşlardır. Bilgisayar ortamında algoritmalarını simüle ederek sonuçlarını raporlamışlardır.

2.1.4 Model Oluşturma Tabanlı Çalışmalar

Yang ve Meng (2000) yaptıkları çalışmada; mobil robotların gerçek zamanlı hareket planlamasına biyolojik olarak ilham veren bir yaklaşım olan bir sinir ağı modeli (neural network model) sunmuşlardır. Bu modelin topolojik olarak organize edilmiş bir sinir ağındaki nöronlardan oluştuğunu belirtmişlerdir.

Zelinsky (1997) çalışmasında; bir uzaklık dönüşüm modeli (distance transform model) geliştirmiştir. Bu model, robot üzerinde herhangi iki kontrol noktasını seçerek her kontrol noktası için robot çalışma alanının bir yol dönüşümünü oluşturmaktadır.

Willms ve Yang (2006) yaptıkları çalışmada; sinir ağı modeli Yang ve Meng (2000) ve uzaklık dönüşüm modelinden Zelinsky (1997) ilham alarak robotların

gerçek zamanlı çarpışmalarının önlenmesi için gerekli yol planlamasını sağlayabilmek amacıyla kullanılabilir bir algoritma geliştirmişlerdir. Bu algoritma ile hareketli engellerin olduğu çalışma alanlarında bile hareketli hedefe ulaşılabilirdi belirtilmiştir. Daha sonra yaptıkları çalışma ile mevcut algoritmalarına güvenlik sınırlarını dâhil ederek kapsamını genişletmişlerdir (Willms ve Yang 2008). Her iki algoritmada da robotun hareketli hedefi yakalayabilmesi için, hedefin hızının robotun hızından düşük olması gerektiği belirtilmiştir.

Yuan ve diğ. (2019) çalışmalarında; mobil robotun yol planlama problemine bir çözüm getirebilmek amacıyla GRU-RNN (Gated Recurrent Unit-Recurrent Neural Network) sinir ağı modelini tanıtmışlardır. Robotun hareketini kontrol etmek ve böylece engeller ile çarpışmadan kaçınma davranışını sağlayabilmek amacıyla bu modele yeni bir kontrol stratejisi çıkışı oluşturabilmek için sensör girişine sahip derin bir sinir ağı kullanmışlardır.

Mester (2010) yaptığı çalışmada; mobil robotun sabit engellerden oluşan ve eğimli bir çalışma alanında güvenli yol planlamasının yapılabilmesi için bir bulanık mantık kontrol modeli sunmuştur. Bu modeli, sensörlerle donatılmış Khepera isimli minyatür bir mobil robota uygulayarak deneysel sonuçlarını raporlamıştır.

Zhu ve diğ. (2019) çalışmalarında; dinamik hareket alanını, doğrusal olmayan programlama ve optimizasyon sorunlarını ele alabilmek amacıyla mobil robotlar için yeni bir hareket planlama yöntemini tanıtmışlardır. Doğrusal olmayan programlamanın performansını geliştirmek amacıyla bulanık mantık yönteminin sonuç bulma yeteneğinden ve sinir ağının öğrenme yeteneğinden faydalanarak RFNN (Recurrent Fuzzy Neural Network) modelini sunmuşlardır. Genişletilmiş Kalman filtresi (EKF) kullanılarak sinir ağının eğitilmesi sağlanmıştır.

2.1.5 Optimizasyon Tabanlı Çalışmalar

Zafar ve diğ. (2010) yaptıkları çalışmada; hareketli engellerin olduğu ortamlarda yol planlamasının gerçekleştirilebilmesi için karınca kolonilerinden yararlanan SAAS (Simulated Ant Agent System) isimli bir optimizasyon tekniği sunmuşlardır. Bu teknik kullanılarak dinamik kısıtlamalar altında güvenli ve kısa bir

yolun planlandığını gösterebilmek amacıyla çalışmalarını bir mayın tarlası üzerinde simüle ederek test etmişlerdir.

Zeng ve diğ. (2016) çalışmalarında; en uygun yolu bulabilmek için karınca koloni optimizasyon algoritması Zafar ve diğ. (2010) tabanlı yeni bir algoritma tanıtmışlardır. Bu algoritmalarında mevcut karınca koloni optimizasyon algoritmalarının hesaplama adımlarının teker teker uygulanması, yerel optimuma düşmesi ve değeri yakınsama problemlerinin giderildiğini belirtmişlerdir.

Li ve diğ. (2020) yaptıkları çalışmada; otomatik olarak yönlendirilen araçlar için yol planlamasını gerçekleştiren Quantum karınca kolonisi optimizasyon algoritmasını sunmuşlardır. Bu algortmada Quantum teorisi ve karınca kolonisi optimizasyon algoritmaları birleştirilerek her iki algoritmanın faydalı kısımlarından yararlanılmıştır. Geliştirdikleri algoritmayı diğer benzer üç metot ile karşılaştırıp, sonuçlarını raporlamışlardır.

Liu ve diğ. (2017) çalışmalarında; karınca kolonisi optimizasyon algoritmasının gelişmiş bir varyasyonunu tanıtmışlardır. Bu algoritma ile mevcut karınca kolonisi optimizasyon algoritmalarında bulunan değeri hızlı yakınsayamama problemini ele almışlardır. Gerçekleştirdikleri simülasyonlar ile mobil robotun yol planlamasının etkili bir şekilde yapıldığını ve değerlerin diğer metotlara nazaran daha hızlı yakınsandığını belirtmişlerdir.

Yang ve diğ. (2019) yaptıkları çalışmada; robot yol planlaması problemine bir çözüm getirmek amacıyla DL-ACO (Double Layer-Ant Colony Optimization) adı verilen çift katmanlı karınca kolonisi optimizasyon algoritmasını sunmuşlardır. Planlanan yolun salınımlı olmasını engellemek amacıyla B-Spline eğrileri kullanılmıştır. Gerçekleştirilen simülasyon sonuçları ile diğer yol planlama algoritmalarının simülasyon sonuçları karşılaştırılmıştır.

Hou ve diğ. (2022) çalışmalarında; gelişmiş bir karınca kolonisi algoritması ile mobil robotların yol planlamasına çözüm getirmeyi amaçlamışlardır. Algoritmalarında, gerçek hayattaki karıncaların doğada birbirleri ile iletişim sağlamalarından esinlenerek, karınca kolonisi algoritmasına iletişim mekanizması

eklemişlerdir. Algoritmalarını, karınca kolonisi optimizasyon algoritması ile karşılaştırarak, sonuçlarını rapor halinde sunmuşlardır.

Castillo ve diğ. (2007) yaptıkları çalışmada; bir holonomik robotun iki boyutlu bir ızgara ile temsil edilen düz bir harita üzerinde engellerden kaçınarak yol planlamasının yapılabilmesi için kullanılacak bir genetik algoritmasını tanıtmışlardır. Genetik algoritmayı (GA) uzunluk ve güvenlik kriterlerine göre optimize ederek MOGA isimli çok amaçlı bir genetik algoritma geliştirmişlerdir. GA ve MOGA algoritmalarının performans ölçümlerini ve simülasyon sonuçlarını raporlamışlardır.

Lamini ve diğ. (2018) çalışmalarında; hareketli engellerin bulunmadığı ortamda Genetik Algoritmanın kullanılması ile gerçekleştirilen yol planlamasına yeni bir geçiş operatörü sunmuşlardır. Bu operatörün, değerleri erken yakınsamayı önlediğinden ve daha optimum yollar bulunmasına sebep olduğundan bahsedilmiştir. Ayrıca yolun mesafesi, güvenliği ve hareket sırasında harcanacak enerji miktarı dikkate alınarak bir uyum fonksiyonu da geliştirilmiştir. Çalışmada sunulan operatör ve uyum fonksiyonu, farklı hareket ortamlarına uygulanmış ve literatürdeki üç farklı çalışma ile karşılaştırılmıştır.

Tuncer ve Yildirim (2012) yaptıkları çalışmada; genetik algoritma için yeni bir mutasyon operatörü önermişlerdir ve bu operatörü hareketli engellerin bulunduğu ortamlardaki mobil robotların yol planlama problemine uygulamışlardır. Bu operatörün diğer operatörlere nazaran daha hızlı değerleri yakınsadığı ve daha optimum yollar belirlediğinden bahsedilmiştir. Literatürdeki diğer çalışmalarla kıyaslanarak sonuçlar raporlanmıştır.

Mac ve diğ. (2017) çalışmalarında; dinamik bir ortamda hareket eden mobil robotlar için yeni bir hiyerarşik global yol planlama algoritması tanıtmışlardır. Bu algoritmalarında robotların fiziksel sınırlarını ve verimliliklerini dikkate alarak yol uzunluğu ve yol düzgünlüğü bakımından en uygun yolun planlanması amaçlanmıştır. Bu hususta, PSO (Parçacık Sürü Optimizasyonu) algoritması geliştirilmiştir ve simülasyonlar gerçekleştirilerek sunulmuştur.

Song ve diğ. (2019^b) yaptıkları çalışmada; kinematik kısıtlamaları olan mobil robotlar için yol planlamasını gerçekleştirebilmek amacıyla modifiye edilmiş bir parçacık sürü optimizasyon (MPSO) algoritmasını tanıtmışlardır. Bu algoritma ile yol planlamasında sık karşılaşılan değerleri yakınsama problemine bir çözüm getirmeyi amaçlamışlardır. Algoritmalarını, literatürdeki birbirinden farklı beş adet parçacık sürü optimizasyonu algoritması ile karşılaştırıp, simülasyon sonuçlarını raporlamışlardır.

2.1.6 Literatürü Derleme Amacıyla Yapılan Çalışmalar

Leena ve Saju (2014) yaptıkları çalışmada; mobil robotların hareketli engellerin bulunduğu ortamlarda engellerden kaçınabilmesi için gerekli olan yol planlama tekniklerini ve bu çalışma alanındaki mevcut gelişmeleri ele almışlardır.

Mac ve diğ. (2016) çalışmalarında; yol planlama metotlarını klasik ve sezgisel metotlar olmak üzere sınıflandırmışlardır. Bu alanda yapılan çalışmaları inceleyerek, avantajlarını ve dezavantajlarını belirtmişlerdir. Çalışmalarında sezgisel metotların incelemesine odaklanmışlardır.

Mohanan ve Salgoankar (2018) yaptıkları çalışmada; robot yol planlaması araştırma alanına yapılan katkıları incelemişlerdir. 1985 ile 2015 yılları arasında yapılan çalışmaların klasik yaklaşımlardan sezgisel yaklaşımlara doğru evrildiğini belirtmişlerdir. Mevcut metotlardan ICS-AVOID Fraichard ve Asama (2004) metodunun en güvenilir sonuç veren metot olduğunu ifade etmişlerdir.

Zhang ve diğ. (2018^b) çalışmalarında; mobil robotların yol planlaması için literatüre kazandırılan çalışmaları modelleme, optimizasyon kriterleri ve çözüm algoritmaları bakımından incelemişlerdir. Genetik algoritma, parçacık sürü optimizasyonu, potansiyel alan ve karınca kolonisi optimizasyonu metotlarının bu problemin çözümünde en çok kullanılan metotlar olduğunu belirtmişlerdir.

Sanchez-Ibanez ve diğ. (2021) çalışmalarında; mobil robot yol planlama algoritmalarını sınıflandırarak, otonom kara ve deniz araçlarındaki uygulamalarını incelemişlerdir. Hareket alanını modellemek için literatürde mevcut olan metotları

derleyerek, mobil robotların hareket kabiliyetini ve dinamiğini yol planlama açısından ele almışlardır. Her bir yol planlama algoritma kategorisini analiz ederek, uygulanabilirliği hakkında bilgi vermişlerdir.

Wang ve diğ. (2021) yaptıkları çalışmada; mobil robotların engellerden kaçınarak belirli bir başlangıç noktasından belirli bir hedef noktasına doğru güvenle ulaşabilmesini sağlamak amacıyla geliştirilen yol planlama metotlarını inceleyerek derlemişlerdir. Bu metotların avantaj ve dezavantajlarını inceleyerek tablo halinde sunmuşlardır. Farklı metotların birleştirilmesi ile elde edilen metotların yol planlama probleminin çözümünde umut vadeden metotlar olduğunu belirtmişlerdir.

Ngwenya ve diğ. (2022) çalışmalarında; mobil robot navigasyonu için literatürde mevcut olan çalışmaları inceleyerek derlemişlerdir. Her bir metodun farklı hareket ortamlarında farklı engeller için performanslarını belirleyerek tablo halinde sunmuşlardır. Ayrıca, sanal engel kullanımının çalışmalarda artış eğiliminde olduğundan bahsederek, sanal engellerin, robotların üzerinde bulunan sensörlerin ölçüm sonuçlarına yaptıkları etkileri değerlendirmişlerdir.

2.2 Manipülörlerin Yol Planlaması İle İlgili Çalışmalar

Gereğinden çok serbestlik dereceli manipülörler eklem değişkenleri için sonsuz sayıda çözüme sahip manipülörler olarak tanımlanmıştır (Conkur ve Buckingham 1997^a). Bu manipülörlerin endüstriyel olarak uygulanması ve kontrol edilmesinde ciddi problemler olmasına rağmen Chiacchio ve diğ. (1991), bu manipülörler engellerden kaçınarak güvenli bir yolda hareket ederek Nakamura (1991), inceleme, bakım ve onarım işlemlerini yerine getirebilmektedir (Ma ve diğ. 1994). Yol planlaması ile ilgili yapılan çalışmaların literatür özeti aşağıda sınıflandırılarak verilmiştir.

2.2.1 Potansiyel Alan Metodu Kullanılarak Yapılan Çalışmalar

Graham ve Buckingham (1993) yaptıkları çalışmada; potansiyel alan metodunu temel alan gereğinden çok serbestlik derecesine sahip manipülörler için

yol planlama algoritması geliřtirmişlerdir. Engellerin kapladığı ızgara noktalarının deęerlerine yüksek, hedef noktasının kapladığı ızgara deęerine ise düşük deęer vererek potansiyel alan hesaplamalarını gerekleřtirmişlerdir. Dört farklı iki boyutlu hareket ortamı için algoritmalarını manipülatör üzerinde uygulayarak sonuçlarını rapor olarak sunmuşlardır.

Mclean ve Cameron (1993) alışmalarında; gereęinden çok serbestlik derecesine sahip manipülatörlerin yol planlamasının yapılabilmesi amacıyla potansiyel alan metodunu kullanarak bir yol planlama algoritması geliřtirmişlerdir. Algoritmalarını hem iki hem de üç boyutlu hareket ortamlarında 7 serbestlik derecesine sahip bir manipülatör üzerinde bilgisayar ortamında simüle ederek sonuçlarını rapor etmişlerdir.

Wang ve dię. (2018) yaptıkları alışmada; potansiyel alan metodunu modifiye ederek hiper-gereęinden çok serbestlik dereceli manipülatörler için daha kapsamlı bir yol planlama algoritması geliřtirmişlerdir. Algoritmalarında, manipülatör uzuvlarının uç noktalarının duruş açılarını dikkate alarak potansiyel alan metodunun denklemini optimize etmişlerdir. 9 serbestlik derecesine sahip bir manipülatör üzerinde algoritmalarını uygulayarak sonuçlarını rapor halinde sunmuşlardır.

Lin ve Chuang (2010) alışmalarında; üç boyutlu hareket ortamlarında hiper-gereęinden çok serbestlik dereceli manipülatörlerin yol planlama problemini özmek amacıyla potansiyel alan metodunu temel alan bir yol planlama algoritması geliřtirmişlerdir. Bilgisayar ortamında algoritmanın elde ettiği yol noktalarına manipülatörün uzuvlarını yerleřtirerek, algoritmalarını simüle etmişlerdir ve sonuçlarını raporlamışlardır.

Fahimi ve dię. (2002) yaptıkları alışmada; potansiyel alan metodunu temel alan bir yol planlama algoritmasını üç boyutlu hareket ortamlarında hareket eden hiper-gereęinden çok serbestlik dereceli manipülatörler için geliřtirmişlerdir. Ayrıca, manipülatörlerin kinematik kontrolünü iyileřtirebilmek amacıyla geliřtirdikleri algoritmayı eğri uydurma algoritması ile birleřtirmişlerdir. Bilgisayar ortamında farklı uzuv sayılarına sahip manipülatörler üzerinde algoritmalarını uygulayarak sonuçlarını rapor etmişlerdir.

Conkur ve Buckingham (1997^b) yaptıkları çalışmada; manevra alanını maksimize edecek şekilde potansiyel alan metodu ile hiper-gereğinden çok serbestlik dereceli manipülatörler için yol planlamasını gerçekleştirmişlerdir.

Conkur ve diğ. (2005) yaptıkları çalışmada; engellerle çarpışmanın gerçekleşebileceği yol noktalarını belirlemek için ışın doğrusu analizi algoritmalarını tanıtmışlardır. Potansiyel alan metodu ile hiper-gereğinden çok serbestlik dereceli manipülatörler için yol planlamasını gerçekleştirmişlerdir. Bilgisayar simülasyonlarını gerçekleştirip, sonuçlarını rapor halinde sunmuşlardır.

Conkur (2005) yaptığı çalışmada; potansiyel alan metodunu kullanarak güvenli yolu belirlemiştir. Daha sonra engellerin çarpışmasının olası olduğu orta noktaları belirleyerek hiper-gereğinden çok serbestlik dereceli manipülatörler için yol planlamasını gerçekleştirmiştir. Bilgisayar ortamında oluşturduğu hareket alanları için algoritmasını simüle ederek, sonuçlarını raporlamıştır.

Bulut ve Conkur (2021) çalışmalarında; ışın doğrusu analizi algoritması Conkur ve diğ. (2005) ile orta nokta belirleme algoritmasını Conkur (2005) birleştirerek geliştirdikleri yol planlama algoritmalarını sunmuşlardır. Birleştirilen iki metotta da kullanıldığı gibi yol noktalarının belirlenmesinde potansiyel alan metodunu kullanmışlardır. Manipülatör uzuvlarının dar alanda manevra yaparken teğet olacak şekilde konumlanması gerektiği belirtilerek, iki boyutlu hareket ortamları için simülasyonlar gerçekleştirilmiştir ve sonuçları paylaşılmıştır.

Mclean ve Cameron (1997) yaptıkları çalışmada; iki ve üç boyutlu hareket ortamları için potansiyel alan metodunu temel alan bir metot geliştirmişlerdir. Metotlarında, gereğinden çok serbestlik dereceli manipülatörlerin sabit engellerden kaçınarak hedefine ulaşabilmesi için gerekli yol planlamasını hayali yayları (virtual springs) kullanarak oluşturdukları bir matematiksel model yardımıyla gerçekleştirmişlerdir. Bilgisayar ortamında hareket simülasyonunu gerçekleştirerek, sonuçları rapor etmişlerdir.

Liang ve Liu (1999) çalışmalarında; gereğinden çok serbestlik dereceli manipülatörlerin sabit engellerle dolu hareket ortamlarında yol planlamasını yapabilmek amacıyla bir yöntem geliştirmişlerdir. Potansiyel alan metodunu

kullanarak manipülatör uzuvlarının engellere çarpmadığı durumları tespit ederek, geliştirdikleri bir elastik model yardımıyla tüm uzuvların pozisyonlarını belirlemişlerdir. Manipülatörün hareket simülasyonlarını bilgisayar ortamında gerçekleştirerek, elde ettikleri sonuçları bildirmişlerdir.

2.2.2 Farklı Metotlar Kullanılarak Yapılan Çalışmalar

Seereeram ve Wen (1995) çalışmalarında; gereğinden çok serbestlik dereceli manipülatörlerin yol planlama problemini sonlu zamanlı lineer olmayan kontrol problemine indirgeyerek, Newton-Raphson metodu ile çözümünü gerçekleştirmişlerdir. Çalışmalarını 9 serbestlik dereceli bir manipülatör üzerinde uygulamışlardır.

Schilling ve diğ. (1995) yaptıkları çalışmada; hiper-gereğinden çok serbestlik dereceli manipülatörün belirlenen yolu takip edebilmesi için bir metot geliştirmişlerdir. Engellerden kaçınabilmek amacıyla belirlenen yolu, manipülatörün uzuvlarının yol noktalarına teğet olacak şekilde takip etmesini sağlamışlardır.

Takahashi ve Schilling (1989) çalışmalarında; çokgen engellerden oluşan hareket ortamları için yol planlama algoritması geliştirmişlerdir. Geliştirilen algoritmada, engellerden kaçınmak için gerekli yolun belirlenmesinde Voronoi diyagramlarından yararlanmışlardır.

Motahari ve diğ. (2012) yaptıkları çalışmada; hiper-gereğinden çok serbestlik dereceli manipülatörlerin engellerden kaçınabilmesi için güvenli yolunun belirlenmesi problemine çözüm getirebilmek amacıyla bir metot geliştirmişlerdir. Metotlarında, engellerle çarpışma noktalarını belirleyerek, çarpışma yaşanmayacak olan durumlar için manipülatörün tüm uzuvlarını yol noktalarına yerleştirmişlerdir. İki adet problemi nümerik olarak çözerek, metotlarını Genetik Algoritma (GA) ile karşılaştırmışlardır.

Machmudah ve diğ. (2021) çalışmalarında; geometrik analiz yardımıyla hiper-gereğinden çok serbestlik dereceli manipülatörler için yol planlama algoritması geliştirmişlerdir. Algoritmalarında Genetik Algoritma (GA) ve WOA (Whale Optimization Algorithm) isimli algoritmaları kullanarak, hesaplamaları güvenli yol

noktası elde edilinceye kadar tekrar etmek suretiyle yol planlamasını gerçekleştirmişlerdir.

Ananthanarayanan ve Ordonez (2017) çalışmalarında; hiper-gereğinden çok serbestlik dereceli manipülatörlerin yol planlaması problemini incelemişlerdir. Gerçek zamanlı olarak engellerin konumlarını dikkate alarak en kısa yolun elde edilebilmesi için bir teknik geliştirmişlerdir. Bilgisayar ortamında tekniklerini 6 ve 9 serbestlik derecesine sahip ayrı iki adet manipülatör üzerinde uygulayarak sonuçları rapor olarak bildirmişlerdir.

Reznik ve Lumelsky (1994) yaptıkları çalışmada; robotun üzerine yerleştirilmiş olan yakınındaki engelleri tespit edebilen sensörler yardımıyla hiper-gereğinden çok serbestlik dereceli manipülatörün belli bir başlangıç noktasından belirlenmiş olan bir hedef noktasına doğru yol planlamasını iki ve üç boyutlu hareket ortamları için gerçekleştirmişlerdir. Geliştirdikleri metodu bilgisayar ortamında iki boyutlu hareket ortamları için 20 uzuvlu, üç boyutlu hareket ortamı için ise 15 uzuvlu bir manipülatör kullanmak suretiyle simüle ederek, sonuçlarını raporlamışlardır.

Islam ve diğ. (2008) yaptıkları çalışmada; hiper-gereğinden çok serbestlik dereceli manipülatörler için yol planlaması yapılırken hesaplamaların daha hızlı ve daha az hafızaya ihtiyaç duymasını sağlamak amacıyla bir yol planlama algoritması sunmuşlardır. Algoritmaları kullanıldığında hesaplama süresinin ve hafıza ihtiyacının manipülatör uzuv sayısı ve engel sayısı ile doğrusal olarak arttığını ifade ederek, farklı değerler için sonuçları tablo halinde vermişlerdir.

Jamali ve diğ. (2014) çalışmalarında; hiper-gereğinden çok serbestlik dereceli robotların engellerden kaçınarak belli bir hedef noktasına güvenle ulaşabilmesi için gerekli olan yolun belirlenmesi için bir algoritma geliştirmişlerdir. İki aşamadan oluşan bu algortmada, ilk aşamada engel konumları dikkate alınarak manipülatörün ters kinematik hesapları yapılmıştır. Son aşamada ise manipülatör uzuvları elde edilen yola yerleştirilerek yol planlaması tamamlanmıştır. İki boyutlu hareket ortamlarında 10 adet uzva sahip bir manipülatör kullanılarak hareket simülasyonları gerçekleştirilmiş ve sonuçları rapor edilmiştir.

Tappe ve diğ. (2015) çalışmalarında; elektromanyetik enerjiyle hareketi sağlanan hiper-gereğinden çok serbestlik dereceli manipülatörler için kontrol sistemi geliştirmişlerdir. En son manipülatör uzvu, diğer uzuvlar tarafından takip edilerek manipülatör belirlenen yola yerleştirilmiştir.

Lin ve diğ. (2022) çalışmalarında; boru hatlarının güvenli bir şekilde incelenebilmesi için hiper-gereğinden çok serbestlik dereceli manipülatörlerde kullanılabilir bir yol planlama algoritması geliştirmişlerdir. Yol planlamasını gerçekleştirdikten sonra yılanı robotun sürünerek hareket etmesi için kontrol sistemi üzerine de çalışmışlardır. Algoritmalarını 16 ve 24 serbestlik derecesine sahip iki ayrı manipülatör üzerinde bilgisayar ortamında simüle ederek sonuçları raporlamışlardır.

Ma ve Konno (1997) yaptıkları çalışmada; hiper-gereğinden çok serbestlik dereceli manipülatörlerin sabit engellerden kaçınarak belirli bir noktadan belirli bir noktaya yük taşıyabilmesi amacıyla yeni bir teknik geliştirmişlerdir. Manipülatör uzuvlarının pozisyonlarını analiz ederek engellerden kaçınma şeması oluşturmuşlardır. Tekniklerini, bilgisayar ortamında uygulayıp simüle ettikten sonra sonuçlarını rapor halinde sunmuşlardır.

Burhanuddin ve diğ. (2013) çalışmalarında; manipülatör uzuvları arasındaki ve uzuvlar ile engeller arasındaki en kısa mesafeyi belirleyerek gereğinden çok serbestlik dereceli manipülatörlerin yol planlamasını gerçekleştiren bir algoritma sunmuşlardır. Algoritmalarını iki boyutlu hareket ortamları için bilgisayar üzerinde uygulayarak simülasyonlarını gerçekleştirmişlerdir ve sonuçlarını raporlamışlardır.

De Maeyer ve diğ. (2018) yaptıkları çalışmada; gereğinden çok serbestlik dereceli robotların dar alanlarda hareket edebilmesi amacıyla örnekleme tabanlı bir yol planlama algoritması geliştirmişlerdir. Algoritmalarını literatürdeki üç adet metotla karşılaştırarak bilgisayar ortamında simülasyonlarını gerçekleştirmişlerdir. Elde ettikleri sonuçları rapor halinde sunmuşlardır.

2.2.3 Eğri Uydurma Tabanlı Çalışmalar

Conkur (2003) çalışmasında; çok yüksek serbestlik derecesine sahip manipülatörler için yol planlama algoritması geliştirmiştir. Algoritmasında, manipülatör uzuvlarını birbirinden ayırmak suretiyle, her bir uzuv uç noktasının belirlenen eğri noktalarına devamlı teğet olacak şekilde takip etmesini sağlamıştır.

Chirikjian ve Burdick (1990) çalışmalarında; hiper-gereğinden çok serbestlik dereceli manipülatörlerin engellerle dolu hareket ortamlarında güvenli bir şekilde hedef noktasına ulaşabilmesi için geometrik tabanlı bir yol planlama algoritması geliştirmişlerdir. Algoritmaları ile omurga eğrisi (backbone curve) denklemi elde ederek yol noktalarını belirlemişlerdir. Manipülatörün hareketini bilgisayar ortamında simüle ederek sonuçlarını raporlamışlardır.

Choset ve Henning (1999) çalışmalarında; yılanı robotların hareket planlaması problemini incelemişlerdir. Robotun en son uzvunun, diğer uzuvlar tarafından takip edilmesi prensibinin (follow-the-leader) geçerli olduğu algoritmalarını tanıtmışlardır. Yol planlaması yapılırken eğri denklemi elde ederek, robotu o noktalara yerleştirmişlerdir. Sonuçlarını simüle ettikten sonra rapor olarak sunmuşlardır.

Miao ve diğ. (2014) yaptıkları çalışmada; yılanı robotların kıvrılma eğrilerini incelemişlerdir. Modülleri Stewart platformundan oluşan bir yılanı robotu çeşitli kıvrılma eğrileri üzerine yerleştirmişlerdir. Farklı modül sayısı için performans değerleri tablolar halinde verilerek, bilgisayar ortamında simülasyonları gerçekleştirmişlerdir. Bu çalışmada, robottaki modülleri oluşturan pnömatik silindirlerin ağır olması sebebiyle modüllerin hareketinin kısıtlı olduğu ve bu durumun da yılanı robotun yol üzerine yerleşimini zayıf hale getirdiği tespit edilmiştir.

Xu ve diğ. (2017) yaptıkları çalışmada; uzay istasyonunda görev yapabilecek hiper-gereğinden çok serbestlik dereceli manipülatörler için hareket planlamasını gerçekleştirmişlerdir. Manipülatörün uzuvlarının hareket açıları şekilsel olarak bir fonksiyon ile ifade edilerek robotun hareket planlaması tamamlanmıştır. Üç boyutlu uzay istasyonu hareket alanı bilgisayar ortamında simüle edilerek, geliştirilen metot

12 ve 20 serbestlik derecesine sahip robotlar üzerinde denenmiştir ve elde edilen sonuçlar rapor olarak paylaşılmıştır.

Duan ve diğ. (2022) çalışmalarında; A* Hart ve diğ. (1968) yol planlama algoritmasını B-Spline eğrileri ile birleştirerek hiper-gereğinden çok serbestlik dereceli manipülatörlerin engellerden kaçınarak belirli bir başlangıç noktasından belirli bir hedef noktasına güvenli bir şekilde ulaşması için takip etmesi gereken yolu planlamak amacıyla bir algoritma geliştirmişlerdir. Algoritmalarında hareket alanının kenarlarını ve engel konumlarını dikkate alarak kontrol noktaları oluşturmak suretiyle yol planlamasını gerçekleştirmişlerdir. Algoritmalarını bilgisayar ortamında uygulayarak, sonuçlarını rapor etmişlerdir. Ayrıca manipülatör tasarımlarını paylaşmışlardır.

2.2.4 Optimizasyon Tabanlı Çalışmalar

Xidias ve Aspragathos (2014) çalışmalarında; dar alanlara sahip üç boyutlu hareket ortamlarında hiper-gereğinden çok serbestlik dereceli manipülatörler için geliştirmiş oldukları yol planlama algoritmalarını sunmuşlardır. Algoritmaları ile yol planlama problemini optimizasyon problemine indirgeyerek Genetik Algoritma (GA) ile bu problemi çözmüşlerdir. Bilgisayar ortamında simülasyonları gerçekleştirerek sonuçları raporlamışlardır.

Xidias (2018) yaptığı çalışmada; sadece sabit engellerden oluşan üç boyutlu hareket ortamlarında hiper-gereğinden çok serbestlik derecesine sahip manipülatörlerin engellerden kaçınarak hedefine güvenli bir şekilde ulaşabilmesi için gerekli olan yol planlamasını gerçekleştiren bir algoritma geliştirmişlerdir. Yol planlama problemi, optimizasyon problemine indirgenerek genetik algoritma yardımıyla çözüm gerçekleştirilmiştir.

Ayten ve diğ. (2017) yaptıkları çalışmada; gereğinden/hiper-gereğinden çok serbest dereceli endüstriyel manipülatörlerin minimum enerji harcayarak yol planlamasının gerçekleştirilebilmesi için geliştirmiş oldukları optimizasyon tekniğini sunmuşlardır. Ayrıca, hayali uzuv konsepti ile uzuvların yerlerini değiştirerek, planlanan yol üzerinde fiziksel olarak imkânsız olabilecek yerleştirmelerin önüne

geçildiğini bildirmişlerdir. 8 adet uzva sahip bir hiper-gereğinden çok serbestlik dereceli bir manipülatörde bilgisayar simülasyonlarını gerçekleştirerek, sonuçlarını rapor etmişlerdir.

Azariadis ve Aspragathos (2005) çalışmalarında; hem mobil robot hem de gereğinden çok serbestlik dereceli manipülatörlerin yol planlamasında kullanılacak bir metot geliştirmişlerdir. Metotlarında, hareket ortamını matematiksel olarak ifade ederek yasaklı bölgeler belirlemişlerdir. Yasaklı bölgelerden kaçınacak şekilde yol planlamasını gerçekleştirmişlerdir. Metotlarını bilgisayar ortamında iki boyutlu hareket ortamı için simüle ederek, sonuçlarını rapor halinde sunmuşlardır.

Chu ve diğ. (2018) çalışmalarında; uzay istasyonlarının dış alanında engellerden kaçınarak güvenli bir şekilde hareket edebilmesi için hiper-gereğinden çok serbestlik dereceli manipülatörler için geliştirdikleri yol planlama algoritmasını tanıtmışlardır. Algoritmalarında, öncelikle manipülatör uzuv noktalarının uç noktalarının yerleştirilmesi gereken yol noktalarını enterpolasyon ile hesaplamışlardır. Daha sonra, manipülatör uzuvlarının uç noktaları ile mafsal bağlantılarının dönme açıları arasındaki ilişkileri tanımlamışlardır. Son olarak engellerle çarpışma noktalarını belirleyerek yol planlamasını tamamlamışlardır. Algoritmalarını bilgisayar ortamında simüle ederek raporlamışlardır.

Collins ve Shen (2016) yaptıkları çalışmada; üç boyutlu hareket ortamlarında hiper-gereğinden çok serbestlik dereceli manipülatörlerin ters kinematik hesaplarını ve yol planlamasını gerçekleştirmişlerdir. PASO ismini verdikleri optimizasyon tabanlı metotlarını bilgisayar ortamında engellerle dolu üç boyutlu hareket ortamında maksimum 120 uzva sahip manipülatörler için simüle ederek sonuçlarını paylaşmışlardır.

2.2.5 Kablo İle Tahrik Edilen Manipülatörler İle İlgili Çalışmalar

Wei ve diğ. (2021) yaptıkları çalışmada; RRT (Rapidly-Exploring Random Tree) algoritması tabanlı Sp-RRT isimli algoritmalarını hiper-gereğinden çok serbestlik dereceli manipülatörün en son uzvunun hareketinin takip edilmesi prensibi

(follow-the-leader) üzerine geliřtirmişlerdir. Kendilerinin imal ettiđi bir manipülatör üzerinde algoritmalarını deneyerek sonuçlarını raporlamışlardır.

Zheng ve diđ. (2021) yaptıkları çalışmada; kablolarla hareketi sağlanan 12 uzuvlu hiper-geređinden çok serbestlik derecesine sahip bir manipülatörün tasarımını gerçekleřtirmişlerdir. Ayrıca, manipülatörün kapalı-döngü kontrol sistemini geliřtirerek, kinematik hesapları üzerine de çalışmalar yapmışlardır. Sonuçları rapor halinde sunmuşlardır.

Tang ve diđ. (2019) çalışmalarında; geliřtirmiş oldukları iki aşamalı bir hareket planlama algoritmasını kablo ile hareketi sağlanan hiper-geređinden çok serbestlik derecesine sahip manipülatörler için sunmuşlardır. Algoritmanın ilk aşamasında manipülatörün kinematik modeli oluşturulurken son aşamasında belli olan bir yolun takip edilebilmesi için uzuv uç noktalarının konum, hız ve ivme profillerini planlamışlardır. Kablo uzunluklarının ayarlanması suretiyle robotun hareketini sağlamışlardır. Algoritmalarını, 24 serbestlik derecesine sahip bir manipülatör üzerinde uygulayarak, sonuçlarını raporlamışlardır.

Tang ve diđ. (2018) yaptıkları çalışmada; hiper-geređinden çok serbestlik derecesine sahip yılanlı robotların yol planlaması için iki aşamalı bir metot geliřtirmişlerdir. Metotlarının ilk aşamasında, yılanlı robotun engellerden kaçınması için gerekli olan yolu yılanlı robotun kıvrımlı hareketini dikkate alarak bir eğri olarak belirlemişlerdir. Son aşamada ise engel konumlarını dikkate alarak güvenli yolu oluşturan eğriyi modifiye ederek yol planlamasını gerçekleřtirmişlerdir. Metotlarını, hem bilgisayar ortamında hem de imal ettikleri 25 serbestlik derecesine sahip bir yılanlı robot üzerinde uygulamışlardır. Elde ettikleri sonuçları rapor halinde sunmuşlardır.

Jia ve diđ. (2022) çalışmalarında; hiper-geređinden çok serbestlik dereceli manipülatörler için gerekli olan yol planlamasını gerçekleřtirebilmek amacıyla RRT (Rapidly-Exploring Random Tree) algoritmasını temel alarak bu algoritmanın yeni bir varyantını sunmuşlardır. Algoritmalarında, manipülatör uzuvlarının maksimum dönme açılarını dikkate alarak, belirlenen yol ile manipülatör parametreleri arasındaki ilişkileri elde etmişlerdir. 17 serbestlik derecesine sahip, kablolar ile hareketi sağlanan

bir manipülatör üzerinde sundukları algoritmayı uygulayarak, sonuçlarını rapor etmişlerdir.

Qin ve diğ. (2022) yaptıkları çalışmada; kablolar ile hareketi sağlanan hiper-gereğinden çok serbestlik dereceli manipülatörlerin kontrolü ve tahrik sistemini basitleştirerek bir yol planlama algoritması geliştirmişlerdir. Algoritmalarında, manipülatör uzuv uzunluklarını dikkate alarak bir kontrol prensibi oluşturmuşlardır. Bu kontrol prensibine bağlı kalarak elde edilen yol eğrisi takip edilmiştir. Algoritmalarını denemek amacıyla toplamda 2300 mm uzunluğa sahip 10 uzuvlu bir manipülatör tasarlamışlardır ve deneme sonucunda çıkan sonuçları paylaşmışlardır.

Canali ve diğ. (2022) yaptıkları çalışmada; ulaşılması zor, tehlikeli ve dar olan hareket alanlarında denetleme ve bakım faaliyetlerini gerçekleştirebilecek 15 serbestlik derecesine sahip kablo ile hareketi sağlanan bir hiper-gereğinden çok serbestlik dereceli manipülatör tasarlayıp prototipini imal etmişlerdir. Ürettikleri prototip manipülatörün performans testlerini gerçekleştirerek sonuçlarını raporlamışlardır.

Xue ve Fan (2022) çalışmalarında; su altında tehlikeli ve dar bölgelerde belirli bir başlangıç noktasından belirli bir hedef noktasına güvenli bir şekilde ulaşabilen kablo ile hareketi sağlanan yılanı bir robotun tasarımını gerçekleştirmişlerdir. Bu robot için yol planlama algoritması geliştirerek, bu algoritmalarını bilgisayar ortamında simüle etmişlerdir. Elde ettikleri sonuçları paylaşmışlardır.

Tang ve diğ. (2017) yaptıkları çalışmada; kablolar aracılığı ile hareketi sağlanan bir hiper-gereğinden çok serbestlik dereceli manipülatörün tasarımını gerçekleştirmişlerdir. Manipülatörün her bir uzvunun 3 adet kablo ile hareket edebileceği şekilde bir tasarım yapmışlardır. Tasarımlarını imal ederek 12 uzuvlu, 24 serbestlik derecesine sahip bir prototip elde etmişlerdir. Prototipleri ile manipülatörün uç noktasına yerleştirilen bir kalem yardımıyla çeşitli geometrik şekillerin çizimlerini gerçekleştirmişlerdir.

Xie ve diğ. (2019) çalışmalarında; yılanı robotların yol planlamasını gerçekleştirebilmek için geometrik bir yaklaşımda bulunmuşlardır. Yılanı robotun son uzvunun, diğer uzuvlar tarafından takip edildiği (follow-the-leader) bir metot

geliştirmişlerdir. Kablo ile hareket eden bir yılanlı robotun tasarımını yapıp, prototipini imal etmişlerdir. Metotlarını, prototip yılanlı robot üzerinde uygulayarak sonuçlarını rapor etmişlerdir.

2.2.6 Literatürü Derleme Amaçlı Yapılan Çalışmalar

Liljebäck ve diğ. (2012) yaptıkları çalışmada; yılanlı robotların modellenmesi, hareket mekanizmalarının geliştirilmesi ve kontrol sistemi tasarımı üzerine yapılan çalışmaları incelemişlerdir. Literatürde yapılan çalışmaların çoğunun yılanlı robotun aşırı düz hareket ortamları için gerçekleştirildiğini ve o dönemde son yapılan çalışmalarda daha zorlu gerçek hayattaki hareket ortamlarının ele alınması yönünde bir eğilimin olduğunu belirtmişlerdir.

Qian ve diğ. (2018) çalışmalarında; kablolar ile hareketi sağlanan paralel robotların tarihçesinden ve uygulama alanlarından bahsetmişlerdir. Tasarımları, performans analizleri ve kontrol teorileri ile ilgili literatürde mevcut olan çalışmaları incelemişlerdir. Esnek kabloların kullanıldığı robotların, rijit uzuvlardan oluşan manipülatörlere nazaran endüstriyel uygulamalarda daha az kullanıldığı sonucuna varmışlardır.

Du ve diğ. (2020) yaptıkları çalışmada; hiper-gereğinden çok serbestlik dereceli manipülatörler için mevcut çalışmaları incelemişlerdir. İlgili manipülatörün tanımları, çeşitleri, kinematik metotları, hareket alanının hesaplanması ve engellerden kaçınması için gerekli olan yol planlaması ile ilgili çalışmaları sınıflandırmışlardır.

Tho ve Thinh (2022) çalışmalarında; kablolar aracılığıyla hareketi sağlanan paralel robotların mekanik tasarımı, hareket alanının analizi, kablo gerginliklerinin ayarlanması ve kabloların sarkma miktarlarının belirlenmesi alanlarında literatürde mevcut olan çalışmaları sınıflandırarak özetlemişlerdir. Ayrıca bu tür robotların tasarım süreci ve analizi için gerekli olan matematiksel modeller de genel bir bilgi olarak verilmiştir.

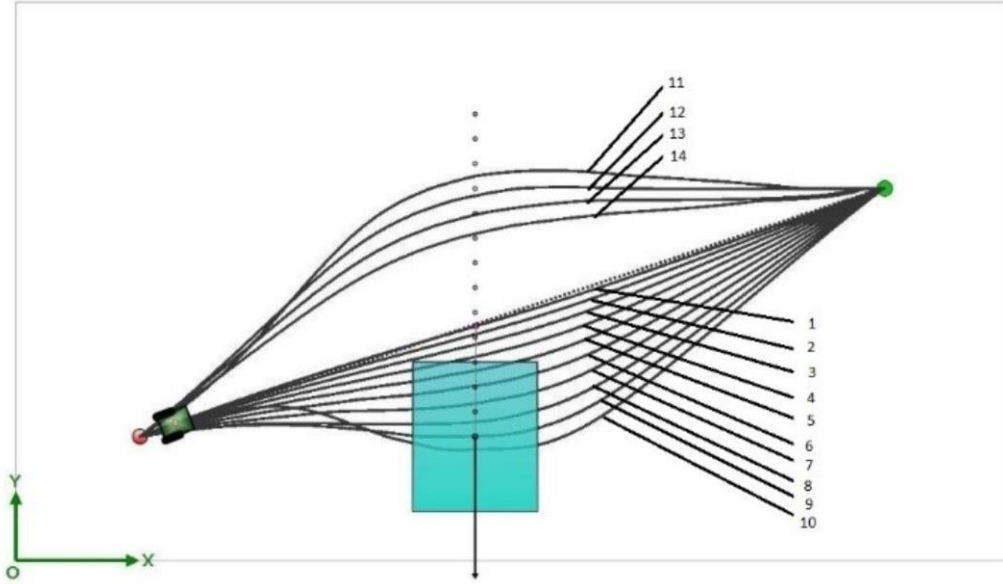
2.3 Çalışmanın Literatürdeki Yeri

Mobil robotların hareket planlamasının yapılacağı alan hem sabit hem de hareketli engellerden oluşabilmektedir. Literatürde sadece sabit engellerin bulunduğu ortamlar için yol planlaması yapan çalışmalar da mevcuttur. Fakat pratik olarak mobil robotların kullanılacağı alanlar da hareketli engeller de bulunmaktadır. Bu sebeple hareketli engellerin bulunduğu dinamik ortamlar için mobil robot hareket planlaması araştırma alanı çok önem arz etmektedir.

Dinamik ortamlar için geliştirilen yol planlama algoritmalarında genellikle yol, hareketli engelin her bir hareketi için yeniden hesaplanarak güncellenmektedir. Bu durum takip edilmesi gereken yolun uzamasına dolayısıyla da optimum olmamasına sebebiyet vermektedir. Bu durumu açıklayabilmek için Şekil 2.1’de yukarıdan aşağıya doğru hareket eden bir adet hareketli engel içeren bir çalışma alanı örnek olarak incelenmiştir. Görüldüğü üzere hareketli engelin her bir hareketinde yol yeniden hesaplanmıştır ve toplam 14 adet yol belirlenmiştir. 1 den 10 a kadar olan yollar engel aşağıya doğru hareket ederken belirlenen yollardır. Ancak, hareketli engel çalışma alanının alt kısmına yaklaştığında, yol planlama algoritması artık bu kısımda yol belirleyememektedir. Bu kısım yerine boş olan üst taraftan güvenli 11 nolu yolu belirlemektedir. 1 ile 10 arasındaki yol değişimi nispeten daha kabul edilebilir olsa da 10 dan 11 nolu yola doğru değişim kabul edilebilir bir değişim olmamaktadır. Bu değişim ile birlikte robotun aniden yolunu değiştirmesi gerekmektedir. Yol değişimi yerine en kısa yol olan başlangıç noktasından hedef noktasına doğru belirlenen bir doğru üzerinde mobil robotun hızının ayarlanarak yol planlaması yerine hareket planlaması yapılması daha optimum bir sonuç elde edilmesine sebep olmaktadır. Bu şekilde yeniden hesap yapılmayarak yolun tek hesapla belirlenmesi de sağlanmaktadır.

Bu tez çalışmasının birinci kısmında dinamik ortamlarda mobil robotun hareket planlamasının yapılabilmesi amacıyla SC-IO (Hayali engellerle güçlendirilmiş güvenlik çemberleri) isminde bir algoritma geliştirilmiştir. Bu algorithmada sabit engellerin konumları dikkate alınarak yol planlaması gerçekleştirilip, hareketli engellerin yol ile kesiştiği noktaların bilgisi kullanılarak hesap yapılmaktadır. Dolayısıyla bu tez çalışmasında geliştirilen hareket planlama algoritması sabit engelli ortamlarda güvenli yol planlaması yapabilen herhangi bir metot ile birlikte

çalışabilmektedir. Bu tezdeki hesaplamalarda güvenli yol planlaması için potansiyel alan metodu kullanılarak, hareketli engellerin hareket doğrultuları da dikkate alınarak güvenlik çemberleri ile güvenli hız değerleri hesaplanmıştır. Ayrıca robota tam karşıdan yaklaşan hareketli engellerin olduğu durumlar için de hesaplanan yolun hayali sabit engellerle güncellenmesine olanak sağlanmıştır.



Şekil 2.1: Her bir hareket için belirlenen yollar.

Bu tez çalışmasının ikinci kısmında mobil robotun üzerine hiper-gereğinden çok serbestlik dereceli manipülatör yerleştirilerek, bu seri manipülatörün dar alanlardan kaçınması için gerekli olan manevraları yapabilmesi amacıyla iki ve üç boyutlu hareket ortamları için yol ve hareket planlama algoritmaları geliştirilmiştir. Bu hususta, tez çalışmasının birinci kısmında geliştirilen SC-IO algoritması kullanılarak manipülatör güvenli bir şekilde mobil robot üzerinde taşınmıştır. Daha sonra manipülatör için önce iki boyutlu hareket ortamlarında daha sonra da üç boyutlu hareket ortamlarında potansiyel alan metodu ve yeni geliştirilen ışın konsepti kullanılarak yol ve hareket planlama algoritmaları geliştirilmiştir.

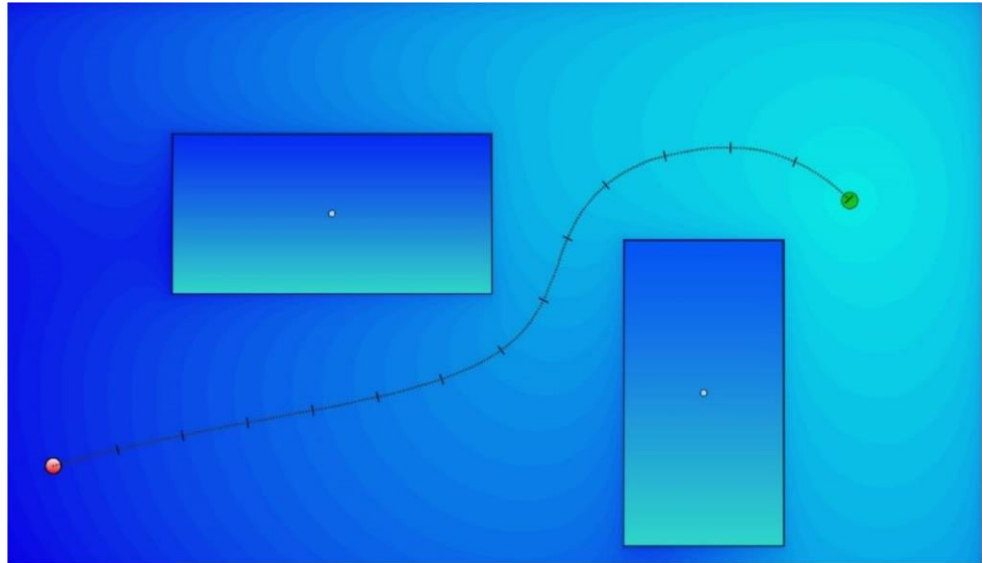
3. POTANSİYEL ALAN METODU

3.1 Matematiksel İfadesi

3.1.1 İki Boyutlu Hareket Ortamı İçin Matematiksel İfade

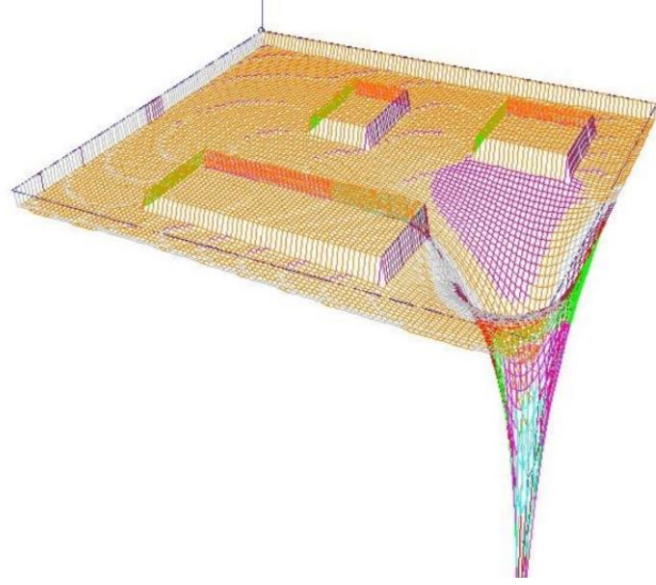
Potansiyel alan metodunda engeller mobil robot üzerinde itme, hedef ise çekme etkisi yaratmaktadır (Khatib 1986). Bu iki etkinin sonucunda mobil robot engellerden kaçınarak hedefine ulaşabilmektedir (Şekil 3.1).

Potansiyel alan denklemini tüm çalışma alanına uygulandıktan sonra hedef çukurda kalmaktadır ve engeller de tepe etkisi yaratmaktadır. Bunun sonucunda hedefe doğru bir eğim oluşmaktadır (Şekil 3.2).

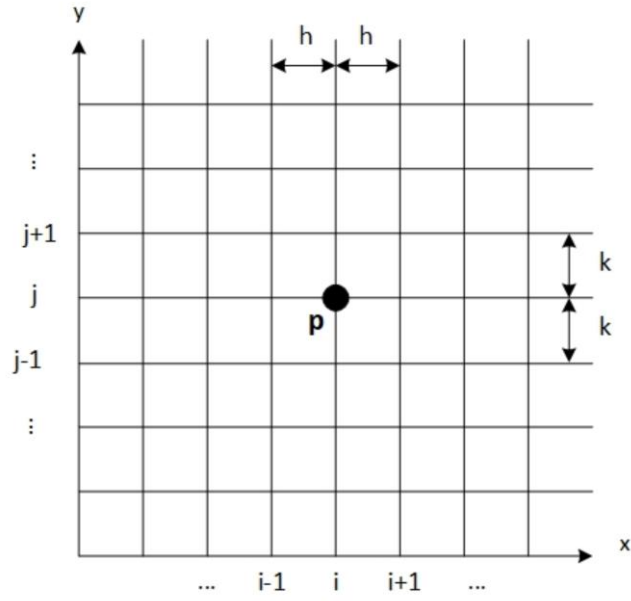


Şekil 3.1: Potansiyel alanda mobil robotun hedefini bulması.

Potansiyel alan metodunun matematiksel ifadesi sonlu farklar yönteminden yararlanılarak elde edilmiştir. İkinci mertebeden kısmi diferansiyel denklemler sonlu farklar metodu ile çözülmek istendiğinde, çözüm uzayı bir ızgara sistemiyle ifade edilerek çözüm yapılmıştır. Genellikle ızgara sistemi olarak Şekil 3.3'te görüldüğü gibi dikdörtgensel bir ızgara sistemi kullanılmaktadır.



Şekil 3.2: Potansiyel alanın üç boyutlu görüntüsü.



Şekil 3.3: P noktasının iki boyutlu bir alanda gösterilmesi.

$$\frac{\partial^2 U_{i,j}}{\partial x^2} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} \quad (3.1)$$

$$\frac{\partial^2 U_{i,j}}{\partial y^2} = \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2} \quad (3.2)$$

P noktasındaki ikinci mertebeden merkezi farkları x ve y 'ye göre (3.1) ve (3.2)'de verilmiştir.

İki boyutlu Laplace denklemi (3.3) eşitliği ile verilmiştir.

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad (3.3)$$

(3.1) ve (3.2) denklemleri, (3.3) denkleminde yerine yazıldığında (3.4) eşitliği elde edilmiştir ve $h=k$ durumu için bu eşitliğin sadeleştirilmiş hali (3.5)'te verilmiştir.

$$\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{k^2} = 0 \quad (3.4)$$

$$U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{i,j} = 0 \quad (3.5)$$

Herhangi bir ızgara değerinin hesaplanması (3.6)'da verilmiştir.

$$U_{i,j} = \frac{U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}}{4} \quad (3.6)$$

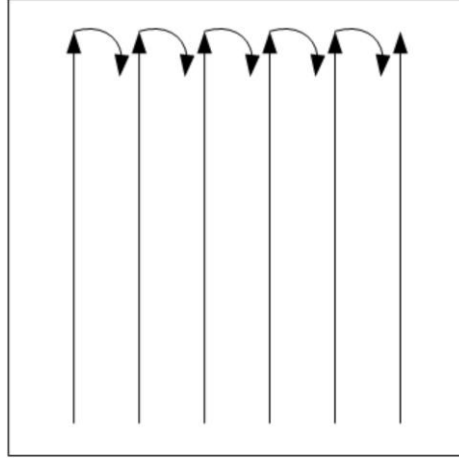
(3.6) denklemi potansiyel alan denklemdir. Eğer alandaki sınır değerleri biliniyorsa bu denklem Dirichlet sınır koşulları altındaki potansiyel alan denklemdir.

Bu tez çalışmasında, potansiyel alan hesaplamalarında çalışma alanını çevreleyen ızgara noktalarına ve engelleri temsil eden ızgara noktalarına sıfır değeri, hedef noktasına ise çukurda kalması için çok küçük bir değer olan -2^{124} değeri verilmiştir. En dıştaki ızgara noktaları, engellere ait ızgara noktaları ve hedefe ait ızgara noktası dışındaki ızgara noktaları potansiyel alan denklemi kullanılarak hesaplanmaktadır. Bir iterasyon yapıldığında bu bahsedilen ızgara noktaları bir defa hesaplanmaktadır ve bu hesaplamalar her bir ızgara noktası için iterasyon sayısı kadar tekrar etmektedir.

(3.7) denklemindeki m iterasyon sayısını göstermektedir. Şekil 3.4'te alanın aşağıdan yukarıya doğru hesaplanma şekli gösterilmektedir. Hesaplama yönü yukarıdan aşağıya, sağdan sola veya soldan sağa şeklinde de yapılabilmektedir. Eğer

alanın indis deęerleri (0,0)'dan bařlıyorsa (i, j) indisi (1, 1)'den bařlamaktadır. Alanın en byk indis deęerleri (n, n) ise (n-1, n-1) indisli ızgara noktasına kadar hesaplama yapılmaktadır.

$$U_{i,j}^{m+1} = \frac{1}{4}(U_{i-1,j}^m + U_{i+1,j}^m + U_{i,j-1}^m + U_{i,j+1}^m) \quad (3.7)$$



řekil 3.4: Alanın ařađıdan yukarıya doęru hesaplanması.

3.1.2 İki Boyutlu Hareket Ortamı İin Yol Belirleme Algoritması

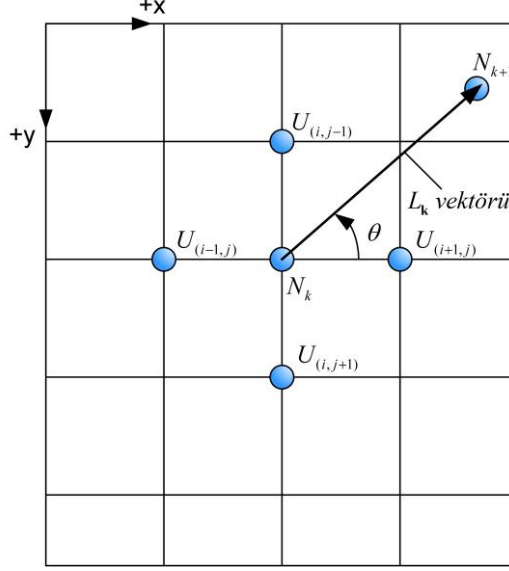
Potansiyel alan denkleminin hareket ortamına uygulanarak elde edilen ızgara deęerlerine gre gvenli yolu belirlemek iin yol belirleme algoritması geliřtirilmiřtir. Bu algoritma gvenli yolu noktalar halinde tespit etmektedir. Robotun hareket doęrultusunun belirlenmesi iin ızgara deęerlerindeki řekil 3.5'te gsterilen θ aı deęeri (3.8) eřitlięi ile hesaplanmıřtır.

$$\theta = a \tan 2 \left(\frac{U_{i,j-1} - U_{i,j+1}}{U_{i-1,j} - U_{i+1,j}} \right) \quad (3.8)$$

řekil 3.5'te grldę gibi N_k 'dan N_{k+1} 'e doęru izilen bir L_k vektr yardımıyla yolu oluřturan noktalar belirlenmektedir. Iızgara noktalarının arasında kalan kısımlardaki hesaplar interpolasyon yardımıyla gerekleřtirilmektedir. Bu hesaplar robotun hedef noktasına ulařmasına kadar tekrar etmektedir. Hesaplama

tamamlanınca belirlenen yolu oluşturan noktaların koordinatları bir dizi de kaydedilmektedir. L_k vektörünün x ve y bileşenleri (3.9) eşitliği ile verilmiştir.

$$L_{kx} = \cos(\theta)|L_k|, L_{ky} = \sin(\theta)|L_k| \quad (3.9)$$



Şekil 3.5: İki boyutlu hareket doğrultusunun hesaplanması.

3.1.3 Üç Boyutlu Hareket Ortamı İçin Matematiksel İfade

Üç boyutlu hareket ortamı için potansiyel alan hesabı iki boyutlu hareket ortamındaki hesap ile benzerlik göstermektedir. L , hareket ortamının kenar uzunluğunu temsil eden küpün uzunluğu ve N , bu uzunluktaki bölümlerin sayısıdır. Potansiyel alanının ızgara değerleri (3.10) denklemi ile verilmiştir.

$$dx = dy = dz = h = L / N$$

$$u_{i,j,k} = u(x_i, x_j, x_k) : i, j, k \in [0, N]$$

$$u_{i,j,k} = \frac{u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k} + u_{i,j,k-1} + u_{i,j,k+1}}{6}$$

$$\frac{\partial^2}{\partial x^2} u(x_i, y_j, z_k) \approx \frac{u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}}{h^2} \quad (3.10)$$

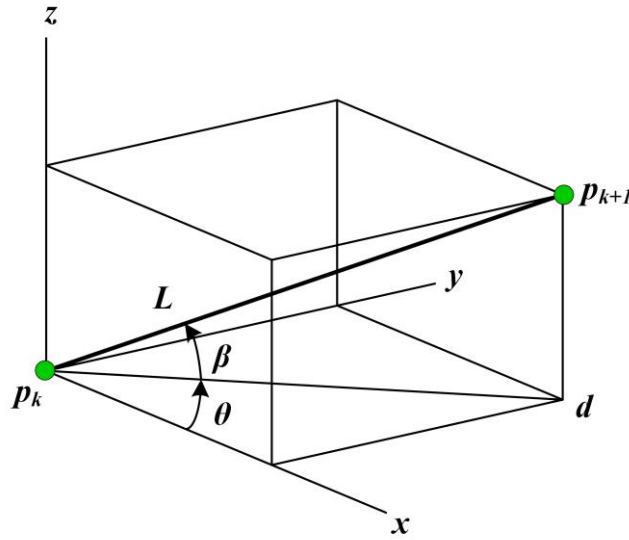
Gauss-Seidel metodu kullanılarak (3.11) eşitliği elde edilmiştir.

$$u_{i,j,k}^{m+1} = \frac{1}{6} (u_{i-1,j,k}^m + u_{i+1,j,k}^m + u_{i,j-1,k}^m + u_{i,j+1,k}^m + u_{i,j,k-1}^m + u_{i,j,k+1}^m) \quad (3.11)$$

i, j ve k sırasıyla x, y ve z doğrultularındaki ızgara konumlarını, m ise iterasyon numarasını göstermektedir.

3.1.4 Üç Boyutlu Hareket Ortamı İçin Yol Planlaması

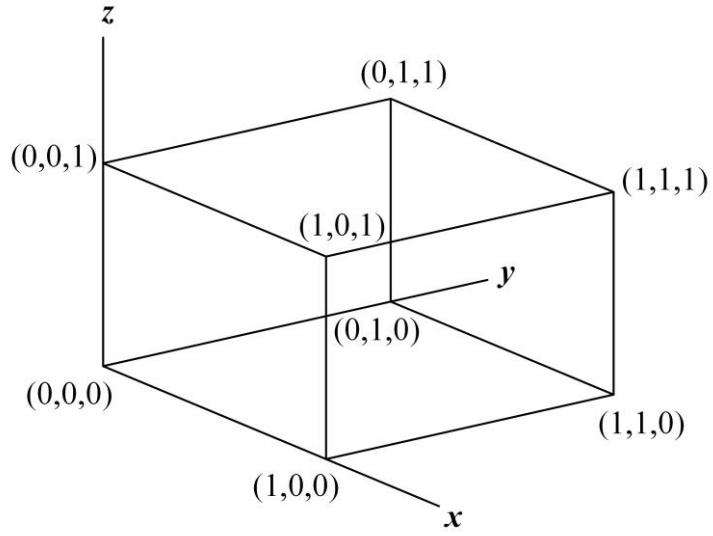
Üç boyutlu hareket ortamında, robotun hareket doğrultusunu belirlemek Bölüm 3.1.2’de detaylıca açıklanan iki boyutlu hareket ortamı için yapılan hesaplamalar ile benzerlik göstermektedir. θ açısına ek olarak Şekil 3.6’dan görüleceği üzere $p_k d$ ve $p_{k+1} d$ doğrularının z doğrultusundaki farkı olan doğrunun β açısının da hesaplanması gerekmektedir. Böylelikle, p_k noktasından p_{k+1} noktasına doğru L vektörü elde edilerek robotun hareket doğrultusu belirlenmektedir.



Şekil 3.6: Üç boyutlu hareket doğrultusunun hesaplanması.

Izgara noktaları arasındaki herhangi bir noktanın potansiyel alan değeri (3.12) eşitliğinde verilen üç boyutlu doğrusal interpolasyon formülü yardımıyla kolaylıkla hesaplanabilmektedir. Böylece yumuşatılmış yol, ilgili formüldeki V değerleri kullanılarak Şekil 3.7’de gösterildiği gibi elde edilebilmektedir.

$$\begin{aligned}
V_{xyz} = & V_{000}(1-x)(1-y)(1-z) + V_{100}x(1-y)(1-z) + V_{010}(1-x)y \\
& (1-z) + V_{001}(1-x)(1-y)z + V_{101}x(1-y)z + V_{011}(1-x)yz + V_{110}xy \\
& (1-z) + V_{111}xyz
\end{aligned}
\tag{3.12}$$



Şekil 3.7: Üç boyutlu hareket ortamında interpolasyon.

4. HAYALİ ENGELLERLE GÜÇLENDİRİLMİŞ GÜVENLİK ÇEMBERLERİ (SC-IO) ALGORİTMASI

“Hayali Engellerle Güçlendirilmiş Güvenlik Çemberleri” (Safety Circles with Imaginary Obstacles) algoritmasının amacı sabit ve hareketli engellerin bulunduğu ortamlarda mobil robotun güvenli bir şekilde hedefine ulaşmasını sağlamaktır. SC-IO algoritmasındaki varsayımlar aşağıda listelenmiştir.

- Robot ızgaralardan oluşan iki boyutlu ortamlarda hareket etmektedir.
- Sabit ve hareketli engellerin boyutları keyfidir, şekilleri ise dikdörtgensel, eliptik ve çokgenseldir. Ayrıca konumları da bilinmektedir.
- Hareketli engellerin konumları, hareket doğrultuları ve hızları önceden bilinmektedir. Bir doğru boyunca hareket etmektedirler.
- Mobil robot dikdörtgen şeklindedir.

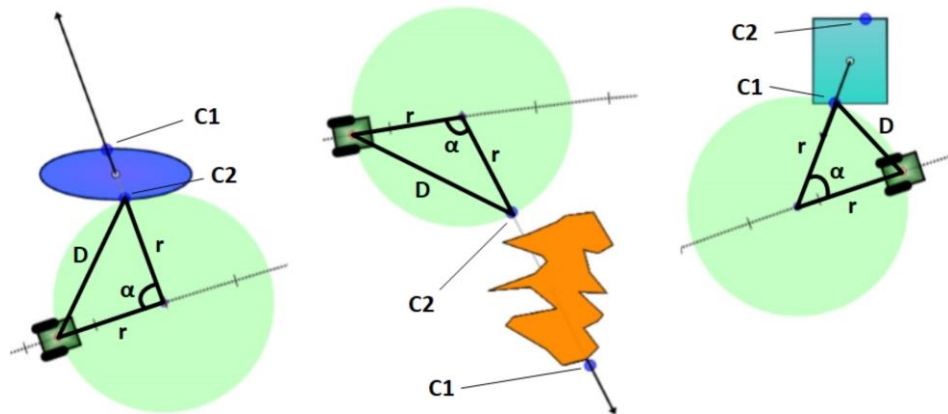
Geliştirilen algoritma başlangıçta belirlenmiş olan yol bilgisine ihtiyaç duymaktadır. Bu nedenle algoritma herhangi bir yol planlama algoritması ile birlikte çalışabilmektedir. Algoritma, Güvenlik Çemberi (SC) ve Hayali Engel (IO) olmak üzere iki aşamadan oluşmaktadır.

4.1 Güvenlik Çemberi (SC) Aşaması

Algoritmanın bu aşamasında, hareketli engeller başlangıç noktasına olan uzaklıklarına göre numaralandırılmaktadırlar. Sabit engellerin konumları dikkate alınarak belirli bir iterasyon sayısı (n) kullanılarak yol planlaması yapıldıktan sonra bu hareketli engellerin hareket doğrultularında bir ışın doğrusu çizilerek yol ile kesiştikleri noktalar belirlenmektedir. Bu noktalar merkez noktaları olacak şekilde her bir hareketli engel için güvenlik çemberleri yerleştirilmektedir. Şekil 4.1’de görüldüğü gibi bu çember üzerinde P1, P2, O1 ve O2 olmak üzere dört adet pozisyon bulunmaktadır. P1 pozisyonu mobil robotun güvenlik çemberinin sınırlarının içine girmeye başladığı pozisyon iken P2 pozisyonu mobil robotun güvenlik çemberinin sınırlarını terk ettiği pozisyon olarak tanımlanmıştır. O1 ve O2 pozisyonları için benzer

Hareketli engellerin C1 ve C2 temas noktalarının belirlenmesinde geometrik şekilleri dikkate alınmaktadır. Dikdörtgenel engeller için engelin hareket doğrultusu boyunca çizilen ışın doğrusunun kesiştiği kenar noktaları olarak belirlenmektedir. Eliptik engeller de ise kenar çizgisi olmadığı için kesişen yay noktaları temas noktaları olarak belirlenmektedir. Çokgenel engeller de ise çokgeni çevreleyen dikdörtgenin kenar noktaları dikkate alınarak temas noktaları belirlenmektedir. Temas noktalarının konumları dikkate alınarak bu algorithmada iki durum güvenli olarak kabul edilmiştir. Engelin C1 temas noktasının O1 pozisyonunda iken robotun P2 pozisyonunda olduğu (O1P2 durumu) ve engelin C2 temas noktasının O2 pozisyonunda iken robotun P1 pozisyonunda olduğu (O2P1 durumu) güvenli durumlar olarak belirlenmiştir. Başka bir deyişle, hareketli engel güvenlik çemberine girerken robotun çemberi terkettiği (O1P2 durumu) ve hareketli engel çemberi terk ederken robotun çembere girdiği durum (O2P1 durumu) güvenli durumlar olarak tanımlanmıştır. Ancak, bu durumlar güvenlik çemberinin boyutu Şekil 4.2’de görüldüğü gibi mobil robot ile hareketli engel arasında güvenli bir mesafe (D) oluşturacak kadar olduğunda güvenli olabilmektedir. Bu güvenli durumu sağlayabilmek için hareketli engelin kritik yaklaşma açısı α_{cr} algorithmada hesaplanmıştır. Güvenlik çemberinin yarıçapı (r) ve hareketli engel ile mobil robotun merkezleri arasındaki mesafe (D) keyfi olarak belirlenen parametrelerdir. Mobil robotun ve hareketli engelin boyutları dikkate alınarak α_{cr} Kosinüs teoremi ile (4.1) eşitliği kullanılarak hesaplanmıştır.

$$\alpha_{cr} = \arccos\left(1 - \frac{D^2}{2r^2}\right) \quad (4.1)$$



Şekil 4.2: Kritik yaklaşma açısının hesaplanması.

Her bir hareketli engelin konumu, hareket doğrultusu ve yol ile kesiştiği nokta bilinmektedir. Dolayısıyla bu engellerin yaklaşma açıları hesaplanabilmektedir. Algoritmada her bir hareketli engelin yaklaşma açısı (α) hesaplanmıştır ve kritik yaklaşma açısına (α_{cr}) göre kıyaslanmıştır. Bu açı değerinin α_{cr} değerinden büyük olması durumunda hareketli engel ile mobil robot arasında güvenli mesafenin sağlanabildiği sonucu çıkmaktadır. Dolayısıyla mobil robot ile hareketli engelin birbiri ile çarpışması önlenmektedir. Tam tersi durumda ise hareketli engel ile mobil robot arasında güvenli bir mesafe olmamaktadır ve bu durum birbiri ile çarpışmalarına neden olmaktadır. Sonuç olarak algoritmanın bu adımında, hareketli engeller yukarıda belirtilen kritere göre güvenli ya da tehlikeli olarak tespit edilmektedirler. Tehlikeli olarak tespit edilen hareketli engeller Bölüm 4.2’de detaylıca anlatılacak olan Hayali Engel (IO) aşaması ile güvenli hale getirilmektedirler.

4.1.2 Geçerli Hızların Kontrol Edilmesi

Algoritmanın bu adımında mobil robotun her bir hareketli engeli geçmesi için sahip olması gereken hız değerleri ayrı ayrı hesaplanmıştır. Bu hız değerleri mobil robotun fiziksel olarak ulaşabileceği hız limitleri ile sınırlı olmaktadır. Algoritmada sıfır değeri ile maksimum ulaşabileceği hız değeri (v_{max}) arasında değer alması sağlanmıştır. Ancak, düşük değerdeki hızlar ile hareket etmesini engellemek için referans hız değeri (v_{ref}) tanımlanmıştır ve mobil robotun mümkün olduğunca v_{ref} ve v_{max} hız aralığında hareket etmesi amaçlanmıştır.

Algoritma tarafından öncelikle tüm engeller için güvenli tek bir hız değerinin olup olmadığı kontrol edilmiştir. Bunun için v_{ref} ve v_{max} arasındaki hız değerleri ile O1P2 veya O2P1 güvenli durumlarının sağlanıp sağlanmadığı kontrol edilmiştir. Birden fazla geçerli tek hız değeri tespit edildiğinde en yüksek hız değeri güvenli hız olarak belirlenmiştir. Tek geçerli hız değerinin bulunamadığı durumlarda her bir engel için farklı hız değerlerinin bulunabilmesi amacıyla O1P2 veya O2P1 güvenli durumlarının sağlandığı güvenli hız değerleri hesaplanmıştır. Bu hesap yapılırken v_{ref} ve v_{max} arasındaki hız değerleri ile öncelikle O1P2 güvenli durumunun sağlanıp sağlanmadığı her bir hareketli engel için ayrı ayrı kontrol edilmiştir. Bu kontrol için hareketli engelin O1 pozisyonuna ulaşması için gereken süre t_{O1} hesaplanmıştır. Mobil

robotun ise P2 pozisyonuna ulaşması için gereken süre t_{P2} her bir olası hız değeri için hesaplanmıştır. Bu süre değerlerine göre hareketli engel O1 pozisyonuna varmadan mobil robot P2 noktasına varmışsa O1P2 güvenli bir durum anlamına gelmektedir ($t_{O1} > t_{P2}$). Bu durumun sağlandığı hız değerleri kaydedilip en yüksek değerdeki hız, geçerli hız olarak belirlenerek mobil robotun bu hızla hareket ederek ilgili hareketli engelden kaçınması sağlanmıştır.

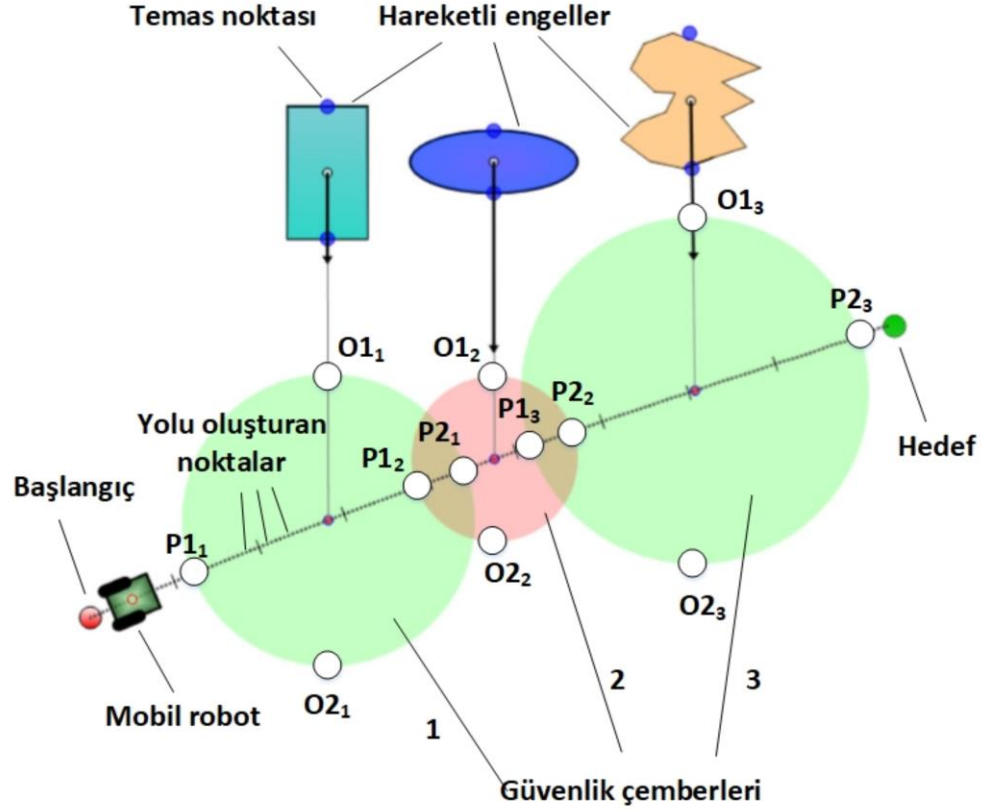
O1P2 durumunun hiçbir hız değeri için güvenli olmaması durumunda ise algoritma benzer kontrolü O2P1 durumu için yapmaktadır. Hareketli engelin O2 pozisyonuna mobil robotun da P1 pozisyonuna ulaşabilmesi için gereken süre değerleri t_{O2} ve t_{P1} hesaplanmaktadır. Bu durumda ise hareketli engel O2 pozisyonuna ulaştığında mobil robotun P1 pozisyonuna ulaşmış olması gerekmektedir ($t_{O2} < t_{P1}$). Bu durum sağlandığında hareketli engel ile mobil robot arasında güvenli bir mesafenin bulunduğu anlamına gelmektedir.

4.1.3 Belli Hareketli Engellerin Dikkate Alınmaması

İki farklı hareketli engelin güvenlik çemberleri iç içe geçtiğinde merkezleri de birbirine yaklaşmaktadır ve bu durum mobil robot ile hareketli engellerin çarpışmasına sebep olabilmektedir. Bu problem üç adet güvenlik çemberinin iç içe geçtiği özel durum için ele alınmıştır (Şekil 4.3).

Algoritmanın ilk engel için O1P2 durumunun, ikinci engel için O2P1 durumunun güvenli olduğunu tespit ettiği varsayıldığında, bu durumda mobil robot önce $P2_1$ pozisyonuna daha sonra da iç içe geçme durumundan dolayı $P1_2$ pozisyonuna geriye doğru hareket etmesi gerekmektedir. Robotun geriye doğru hareket etmesini engellemek için ikinci engelin güvenli durumu O1P2 olarak algoritma tarafından değiştirilmektedir. Daha sonra değiştirilen bu güvenli duruma göre üçüncü engelin güvenli durumu kontrol edilmektedir. Eğer üçüncü engelin güvenli durumu O2P1 ise mobil robot $P2_2$ pozisyonundan $P1_3$ pozisyonuna doğru geriye doğru hareket etmesi gerekmektedir. Bu durumda algoritmanın bu adımında ikinci engelin tehlike oluşturup oluşturmadığı kontrol edilmektedir. İkinci engel $O1_2$ pozisyonuna ulaştığında, üçüncü engelde $O2_3$ noktasını geçmiş ise ikinci engel tehlike oluşturmuyor anlamına gelmektedir. Tehlike oluşturmuyor ise bu engel algoritma tarafından dikkate

alınmamaktadır. Dikkate alınmayan engelin güvenlik çemberi farklı bir renk ile Şekil 4.3'teki gibi gösterilmektedir. Üçüncü engelin güvenli durumu O1P2 olduğu durum için robotun geriye doğru hareket etmesine gerek kalmamaktadır.

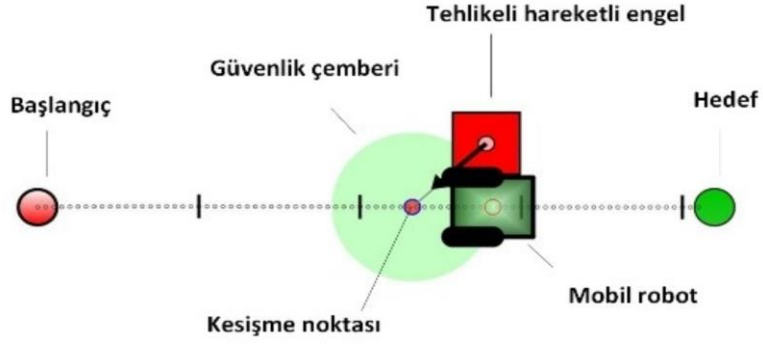


Şekil 4.3: İç içe geçmiş güvenlik çemberleri.

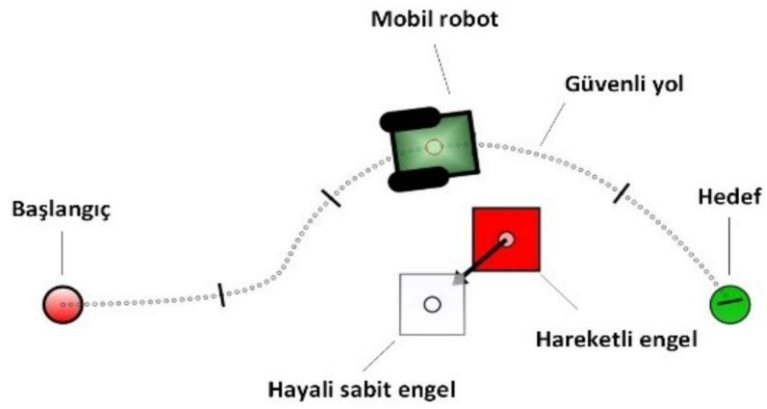
4.2 Hayali Engel (IO) Aşaması

Bölüm 4.1.1'de belirtildiği üzere algoritmanın SC aşamasında hareketli engeller güvenli ve tehlikeli olarak belirlenmektedir. Şekil 4.4a'da O1P2 güvenli durumu için tehlike arz eden bir durum verilmiştir. Algoritmanın IO aşamasında ise bu tehlikeli engel güvenli hale getirilmiştir.

Tehlikeli hareketli engelin güvenlik çemberinin merkezine hayali bir sabit engel eklenerek yol planlaması tekrar yapılmaktadır (Şekil 4.4b).



(a)

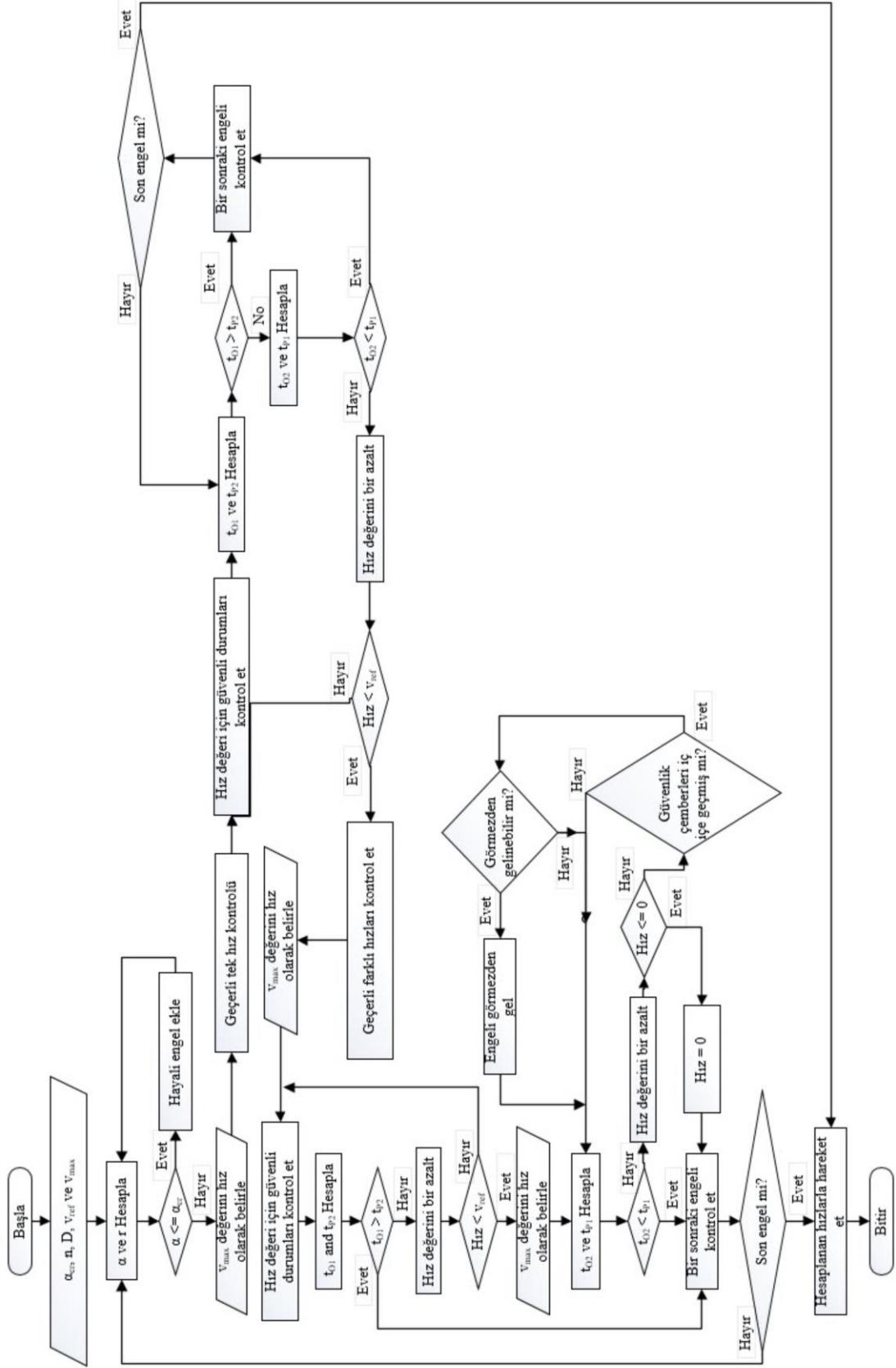


(b)

Şekil 4.4: Hareketli engel durumları a) Tehlikeli durum, b) Hayali engel ile güvenli durum.

Belirlenen yol değiştiği için tehlikeli hareketli engelin yolu kestiği nokta (Şekil 4.4a'daki pembe renkli nokta), güvenlik çemberinin merkez noktası ve yaklaşma açısı değişmektedir. Tehlikeli engelin yaklaşma açısı, mobil robot ile hareketli engel arasında güvenli bir mesafenin sağlandığı durum oluşana kadar hayali engel ekleme prosedürü devam ettirilmektedir. Bu durum hesaplama yükünü arttırmaktadır. Ancak bu aşama sadece tehlikeli hareketli engeller için aktif hale getirilmektedir.

SC-IO algoritmasının akış diyagramı Şekil 4.5'te verilmiştir.



Şekil 4.5: SC-IO algoritmasının akış diyagramı.

5. GOAVOID YAZILIMI

Bu tez çalışmasının birinci kısmında geliştirilen ve Bölüm 4'te detaylıca anlatılmış olan SC-IO hareket planlama algoritmasının performansının değerlendirilebilmesi için robotun hareket edeceği ortamı simüle eden bir yazılım çok önem arz etmektedir. Bu sebeple GoAvoid ismi verilen yazılım Microsoft® Visual Studio.NET platformunda C# programlama dili kullanılarak geliştirilmiştir.

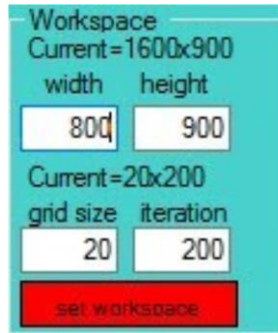
5.1 SC-IO Algoritmasının Eklenmesi

GoAvoid yazılımına SC-IO algoritması dört aşamada eklenmiştir. Bu aşamalar aşağıda listelenmiştir.

- Çalışma alanının oluşturulması
- Hareketli engellerin eklenmesi
- Güvenlik çemberinin eklenmesi
- Hayali engelin eklenmesi

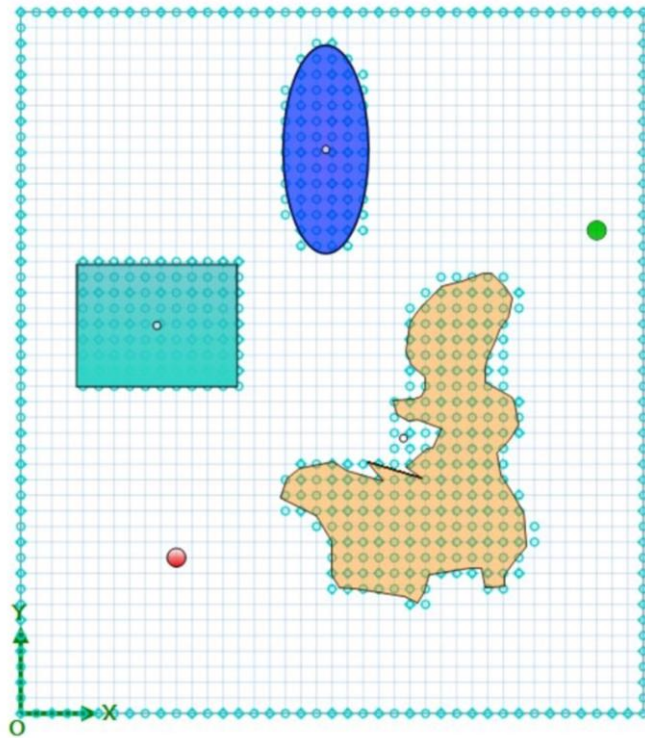
5.1.1 Çalışma Alanının Oluşturulması

Kullanıcının, çalışma alanını oluşturabilmesi için yazılıma ayar paneli eklenmiştir. Bu ayar paneli Şekil 5.1'de gösterilmiştir.



Şekil 5.1: Çalıştırma alanı oluşturma ayar paneli.

Oluşturulmak istenen çalışma alanının genişliği, yüksekliği, ızgara çizgilerinin sıklığı ve potansiyel alan metodu hesaplamalarında kullanılacak olan iterasyon sayısının ayarlanabildiği bu ayar paneli ile çalışma alanının oluşturulması sağlanmıştır. Birer adet dikdörtgensel, eliptik ve çokgensel engelin bulunduğu bir çalışma alanı örnek olarak Şekil 5.2’de gösterilmiştir. Şekilde, kırmızı renkli nokta hareketin başlangıç noktasını, yeşil renkli nokta ise hareketin hedef noktasını göstermektedir. Ayrıca, yatay ve dikey doğrular ızgara çizgilerini ve turkuaz renkli noktalar ise çalışma alanı sınırlarının ve engellerin işgal ettiği ızgara noktalarını ifade etmektedir.



Şekil 5.2: Oluşturulmuş çalışma alanı.

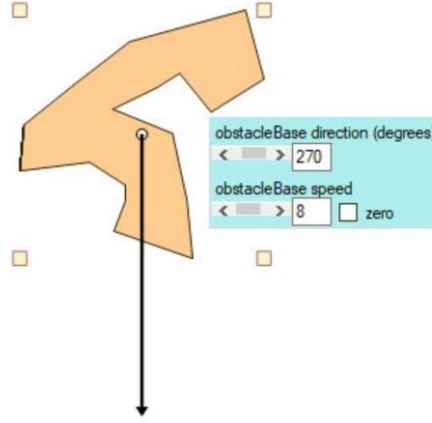
5.1.2 Hareketli Engellerin Eklenmesi

Yazılıma hareketli engellerin eklenmesi iki adımda gerçekleştirilmiştir ve bu adımlar aşağıda listelenmiştir.

- Hareketli engellerin hız ve doğrultusunun ayarlanması
- Engel ile yol kesişiminin belirlenmesi

5.1.2.1 Hareketli Engel Parametrelerinin Ayarlanması

Bu aşamada çalışma alanına eklenen engellerin hızları ve doğrultuları belirlenmektedir. Bu tez çalışmasında hareketli engellerin belirlenen doğrultu boyunca hareket ettiği durumlar için mobil robot hareket planlaması problemini çözmek amacıyla SC-IO hareket algoritması geliştirilmiştir. Başka bir deyişle, hareketli engellerin devamlı yön değiştirdiği durumlar bu tez çalışmasında incelenmemiştir. Her bir hareketli engelin farklı hareket doğrultusu ve hız değeri ile hareket etmesini sağlayabilmek amacıyla yazılıma Şekil 5.3'te görüldüğü gibi bir ayar ekranı eklenmiştir.



Şekil 5.3: Engelin hız ve doğrultu ayarlama ekranı.

5.1.2.2 Engel İle Yol Kesişiminin Belirlenmesi

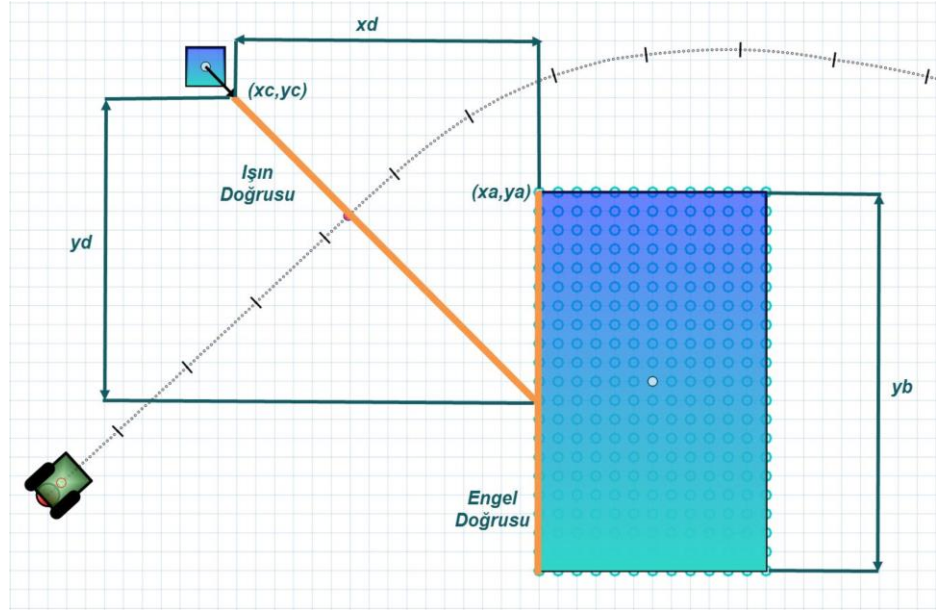
Hareketli engelin merkezinden, potansiyel alan metodu ile belirlenen yola doğru bir ışın doğrusu çizilerek engelin yolu kestiği nokta belirlenmektedir. Fakat hareketli engelin sabit engele çarptığı, dolayısıyla mobil robota bir tehlike teşkil etmediği durum için de yol ile kesişim noktası hesaplanmaktadır ve robotun hızı değiştirilmektedir. Bu yüzden hareketli engelin sabit engelle çarpışıp çarpışmadığını kontrol eden bir algoritma geliştirilmiştir. Bu algorithmada doğrunun parametrik denkleminde yararlanılmıştır.

Doğrunun parametrik denklemi doğru üzerindeki herhangi bir noktanın koordinatını bulmak için kullanılmaktadır ve (5.1) eşitliği ile verilmiştir.

$$X = x_1 + t(x_2 - x_1), Y = y_1 + t(y_2 - y_1) \quad (5.1)$$

(x_1, y_1) doğrunun başlangıç, (x_2, y_2) bitiş koordinatları olmak üzere doğrunun parametrik denklemindeki t parametresi yardımıyla doğru üzerindeki herhangi bir koordinat değeri (X, Y) hesaplanabilmektedir. Örneğin doğrunun başlangıç ve bitiş koordinatlarının bilindiği bir durumda $t=0,5$ değeri doğrunun tam orta noktasının koordinat değerini vermektedir. Tam tersi durumda ise herhangi bir koordinatın doğru üzerinde olup olmadığı da t parametresi yardımıyla kontrol edilebilmektedir. Koordinat değerinin doğrunun üzerinde olması için, t parametresinin 0 ile 1 arasında bir değere sahip olması gerekmektedir.

Yazılımda hareketli engelin merkezinden çizilen ışın doğrusu ile dikdörtgen şeklindeki sabit engeli oluşturan dört adet doğrunun ayrı ayrı kesişip kesişmediği kontrol edilmiştir. Işın doğrusu ve sabit engel doğruları için parametrik denklem yazılarak t_1 ve t_2 parametrelerinin değerleri kontrol edilmiştir (Şekil 5.4).



Şekil 5.4: Işın doğrusu ile sabit engel doğrusunun kesişmesi.

Engel doğrusunun parametrik denklemi (5.2)'deki, ışın doğrusunun parametrik denklemi ise (5.3)'teki eşitlik ile ifade edilmiştir.

$$X = x_a + t_1(x_b), Y = y_a + t_1(y_b) \quad (5.2)$$

$$X = x_c + t_2(x_d), Y = y_c + t_2(y_d) \quad (5.3)$$

İki ifade birbirine eşitlendiğinde elde edilen denklem sistemi (5.4)'te verilmiştir.

$$x_a + t_1(x_b) = x_c + t_2(x_d), y_a + t_1(y_b) = y_c + t_2(y_d) \quad (5.4)$$

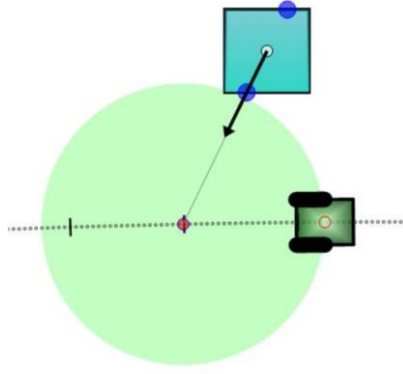
Denklem sistemi çözüldüğünde elde edilen t_1 ve t_2 parametreleri (5.5) ve (5.6) eşitlikleri ile verilmiştir.

$$t_1 = \frac{(-y_a)(x_d) + (y_c)(x_d) + (y_d)(x_a) - (y_d)(x_c)}{(x_b)(y_d) - (y_b)(x_d)} \quad (5.5)$$

$$t_2 = \frac{(-x_a)(y_b) + (x_b)(y_a) - (x_b)(y_c) - (x_c)(y_b)}{(x_b)(y_d) - (y_b)(x_d)} \quad (5.6)$$

5.1.2.3 Güvenlik Çemberinin Eklenmesi

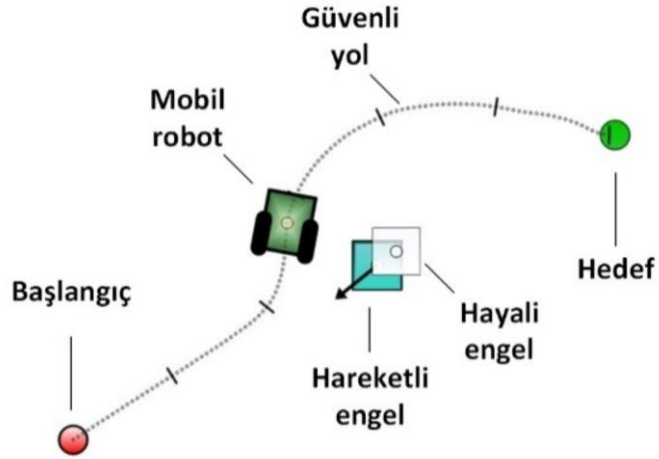
SC-IO hareket planlama algoritması geliştirilirken mobil robotun, yarıçapı ayarlanabilir bir güvenlik çemberi içerisinde hareketli engelin konumunun kontrol edilerek engellerden kaçınması hedeflenmiştir. Hareketli engelin güvenli duruma göre belirlenen temas noktası, güvenlik çemberinin bir ucuna ulaştığında, robotun da güvenlik çemberinin diğer ucuna ulaşması için gerekli olan robot hızları hesaplanmıştır. Hareketli engellerin boyutları da dikkate alındığı için robot ile engelin güvenlik çemberinin içerisinde iken birbirlerine çarpmamaları için her hareketli engelin kritik yaklaşma açısı (α_{cr}) hesaplanmıştır. Bu değer altında olan yaklaşma açılarında hareketli engel ve mobil robotun birbirine çarpma riski oluşmaktadır. Bu nedenle Şekil 5.5'te gösterildiği gibi robot ile engel arasında güvenli bir mesafenin olması gerekmektedir.



Şekil 5.5: Güvenli mesafe.

5.1.3 Hayali Engelin Eklenmesi

Bazı durumlarda güvenlik çemberinin yarıçapının değiştirilmesinin güvenli bir hareket planlaması için yeterli olmadığı görülmüştür. Örnek olarak robota tam karşıdan yaklaşan hareketli engel için yolun değiştirilmesi gerektiği tespit edilmiştir. Bu kapsamda hareketli engelin yol ile kesiştiği noktaya sabit bir hayali engel eklenerek yolun yeniden hesaplanabilmesi için kodlamalar yapılmıştır (Şekil 5.6).



Şekil 5.6: Hayali engelin eklenmesi.

6. SC-IO ALGORİTMASININ SİMÜLE EDİLMESİ

SC-IO algoritması farklı sayıda sabit ve hareketli engellere sahip olan boyutlarına göre yüksek boyutlu, orta boyutlu ve rastgele parametrelili ortamlar olarak adlandırılan üç farklı dinamik ortamda simüle edilmiştir. Çalışma ortamı ızgara hücrelerinden oluşmaktadır ve her bir ızgara hücresi 20 cm olarak kabul edilmiştir. Algoritmanın simüle edilmesi için GoAvoid yazılımı kullanılmıştır. Simülasyonlar Microsoft Windows® 10 işletim sistemine, Intel® Core™ i5-4460 3.2 GHz işlemciye ve 8 GB RAM kapasitesine sahip bir bilgisayarda gerçekleştirilmiştir. Daha önce de belirtildiği gibi yol planlama algoritmalarının içinden potansiyel alan metodu tercih edilerek kullanılmıştır. Geliştirilen algoritmanın genelleştirilmiş bir algoritma olduğunun gösterilebilmesi amacıyla üçüncü örnekte tüm parametrelerin değerleri rastgele olarak yazılım tarafından belirlenmiştir. Ayrıca ORCA metodunun kütüphanesi karşılaştırma yapılabilmesi amacıyla yazılıma eklenmiştir (Van Den Berg ve diğ. 2010).

Mobil robot ve engeller yazılımın her bir simülasyon adımında hareket etmektedirler ve hareket ederken de yol noktalarını kullanmaktadırlar. Bu sebeple hız değerleri ifade edilirken nokta/adım birimi kullanılmaktadır. Yol kordinatları ve mesafe değerleri bilindiğinden dolayı cm/adım olarak da ifade edilebilmektedirler. Adım zamanı hem robotun hem de engelin hareket etme periyot zamanını ifade etmektedir. Bu zaman, her ikisi içinde her zaman aynı değerdedir. Simülasyonlarda bu adım süresi her zaman sabit olmamaktadır ve işlemci zamanına göre devamlı değişmektedir. Bu süre yaklaşık 10 milisaniye (ms) olarak ölçülmüştür. Bu sebeple sonuçların en doğru şekilde ifade edilebilmesi amacıyla değer birimleri ifade edilirken adım kavramı kullanılmıştır. Simülasyonlarda kullanılan parametreler aşağıda açıklanmıştır ve ilk iki örnek için kullanılan değerler Tablo 6.1’de verilmiştir.

- v_{ref} : Mobil robotun referans hız değeri (nokta/adım)
- v_{max} : Mobil robotun ulaşabileceği maksimum hız değeri (nokta/adım)
- n_1 : Yüksek boyutlu hareket ortamı için potansiyel alan metodunda yapılan iterasyon sayısı
- n_2 : Orta boyutlu hareket ortamı için potansiyel alan metodunda yapılan iterasyon sayısı

- D_1 : Yüksek boyutlu hareket ortamında robot merkez noktası ile engel temas noktası arasındaki mesafe (cm)
- D_2 : Orta boyutlu hareket ortamında robot merkez noktası ile engel temas noktası arasındaki mesafe (cm)

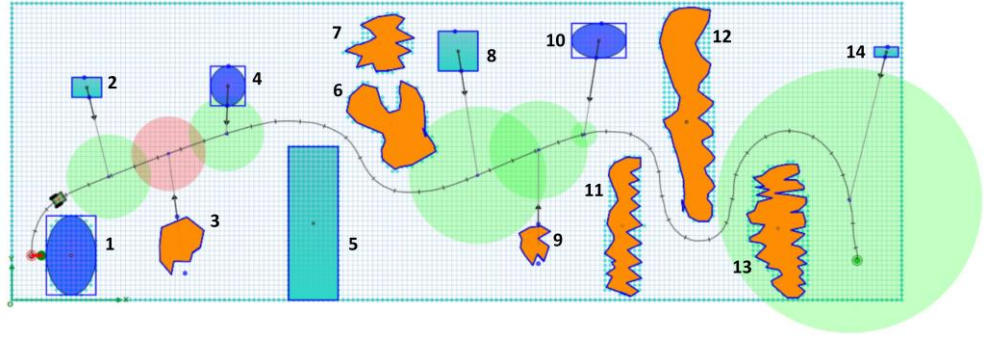
Tablo 6.1: İlk iki hareket ortamı için parametre değerleri.

Parametre	Değer
v_{ref}	50 nokta/adım
v_{max}	70 nokta/adım
n_1	500 kere
n_2	200 kere
D_1	50 cm
D_2	34 cm

6.1 Yüksek Boyutlu Hareket Ortamı

Bu örnekte algoritmanın Güvenlik Çemberi (SC) aşamasının nasıl performans gösterdiğinin gözlenebilmesi için Şekil 6.1’de görüldüğü gibi yedisi sabit ve yedisi hareketli engel olmak üzere toplam ondört engelin olduğu bir hareket ortamı incelenmiştir. Daha önce de bahsedildiği gibi engeller dikdörtgensel, eliptik ve çokgensel şeklinde olabilmektedir. Tüm engellerin merkez koordinatları ve hız değerleri Tablo 6.2’de verilmiştir. Tablonun kısa ve kolay anlaşılabilir olması için eliptik ve çokgensel engellerin tüm koordinatlarının verilmesi yerine bu engelleri çevreleyen dikdörtgensel alanların boyutları ilgili tabloda verilmiştir.

Yüksek boyutlu hareket ortamının ebatları $3600 \times 1200 \text{ cm}^2$ ve hareketin başlangıç ve hedef noktalarının koordinatları ise sırasıyla (80,180) ve (3420,160) olarak belirlenmiştir. Şekil 6.1’de görüldüğü üzere sadece sabit engellerin konumları dikkate alınarak potansiyel alan metodu ile 46529 noktadan oluşan güvenli bir yol hesaplanmıştır.



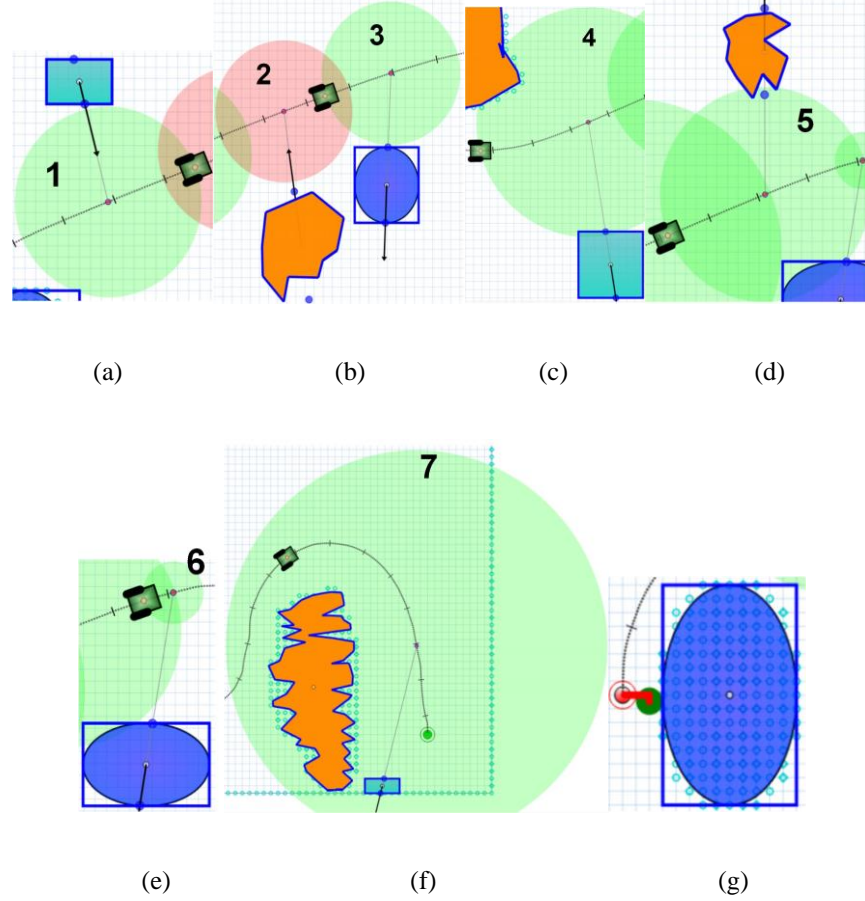
Şekil 6.1: Yüksek boyutlu hareket ortamı.

Tablo 6.2: Yüksek boyutlu hareket ortamındaki engellerin bilgileri.

Engel	Merkez koordinat	Hız (cm/adım)	Alan (cm ²)
1	(240, 180)	0	200x320
2	(278, 959)	1,9	120x80
3	(688, 196)	0,5	180x229
4	(879, 1001)	2,5	140x160
5	(1220, 310)	0	200x620
6	(1535, 708)	0	360x357
7	(1488, 1040)	0	280x320
8	(1797, 1060)	1	160x160
9	(2130, 200)	0,5	140x160
10	(2379, 1076)	0,5	220x140
11	(2470, 300)	0	140x560
12	(2730, 720)	0	220x880
13	(3100, 290)	0	240x540
14	(3546, 1035)	0,6	97x40

Mobil robotun yol üzerindeki hareketi Şekil 6.2’de verilmiştir. Şekil 6.2a’da görüleceği üzere hareketli engel 1 nolu güvenlik çemberinin sınırları içerisine girmeden önce mobil robot güvenlik çemberini referans hız değeri olarak belirlenen 50 nokta/adım hız ile terk ederek engelden kaçınmaktadır. Daha sonra 20 nokta/adım hız ile 2 numaralı kırmızı renkli güvenlik çemberinin ait olduğu çokgensel engeli görmezden gelerek geçmektedir ve 3 numaralı güvenlik çemberinin sınırlarına ulaştıktan sonra eliptik engelden de kaçınmaktadır (Şekil 6.2b). Robot, Şekil 6.2c’de görülen güvenli durumu sağlayabilmek amacıyla yavaşlayarak hızını 12 nokta/adım

değerine düşürmek suretiyle dikdörtgensel engelden de başarı ile kaçınmaktadır. Daha sonra 5 nolu güvenlik çemberine doğru yol olarak çokgensel engelden 8 nokta/adım hız değeri ile hareket ederek kaçınmaktadır (Şekil 6.2d).



Şekil 6.2: Yüksek boyutlu hareket ortamında hareket planlaması.

Şekil 6.2e’de görüldüğü gibi eliptik engel, 6 numaralı güvenlik çemberinin sınırlarını terk etmiş vaziyettedir. Bu sebepten dolayı ilgili engel robota herhangi bir tehlike teşkil etmemektedir ve robot referans hız değeri olan 50 nokta/adım hızıyla hareket ederek bu engeli de güvenli bir şekilde geçmektedir. Daha sonra robot yavaşlayarak 23 nokta/adım hız ile hareket ederek dikdörtgensel hareketli engelden de başarıyla kaçınmaktadır (Şekil 6.2f). Son olarak hareketli engel, 7 numaralı güvenlik çemberinin sınırları içerisindeki hareket alanı sınırına ulaştığında robot referans hız değeri olarak belirlenen 50 nokta/adım ile hareket ederek hedefine güvenli bir şekilde ulaşmaktadır. İlk hareketli engel hariç tüm hareketli engeller O2P1 güvenli durumu sağlanarak geçilmiştir. Tüm hareketli engellerin hareket doğrultuları, yaklaşma açıları (α) ve hesaplanan güvenlik çemberlerinin yarıçap değerleri (r) Tablo 6.3’te verilmiştir. Ayrıca bu tez çalışmasında geliştirilen SC-IO algoritmasının performansını

karşılaştırmak amacıyla yazılıma eklenen ORCA metodunun Van Den Berg ve diğ. (2010) bu örnekte daha ilk sabit engeli geçerken başarısız olduğu gözlemlenmiştir (Şekil 6.2g).

Tablo 6.3: Yüksek boyutlu hareket ortamında bulunan hareketli engellerin doğrultu ve güvenlik çemberi yarıçap değerleri.

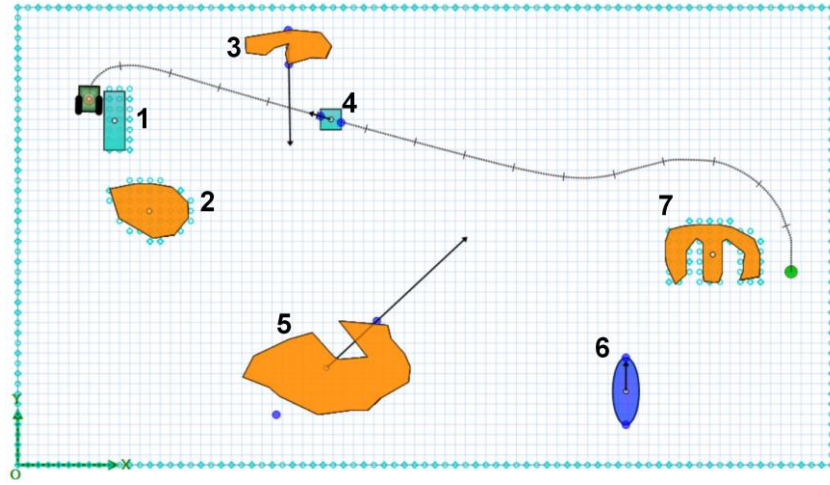
Engel	Doğrultu	α	r
2	284°	81,42°	170,7 cm
3	98°	77,02°	150,6 cm
4	268°	70,91°	151,3 cm
8	279°	76,74°	282,5 cm
9	90°	68,2°	200,4 cm
10	261°	67,1°	52,90 cm
14	256°	25°	538,8 cm

6.2 Orta Boyutlu Hareket Ortamı

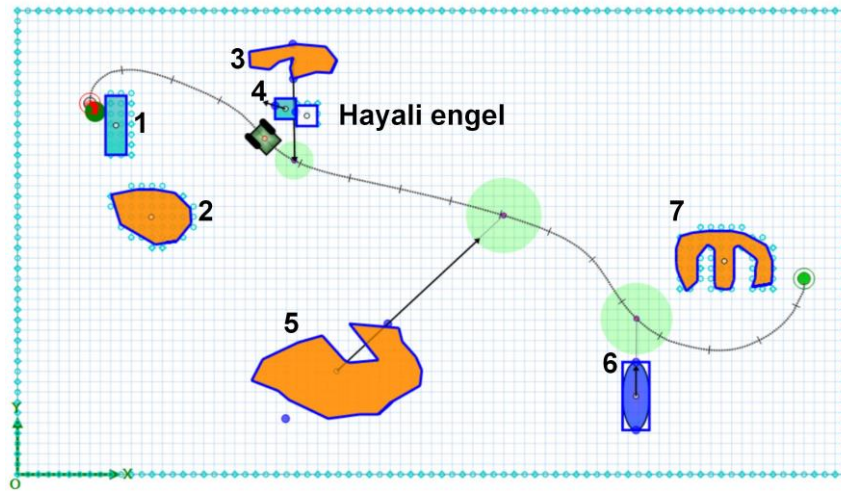
İkinci örnek olarak, birinci örneğe göre hesaplama yükü daha az olan bir orta boyutlu hareket ortamı incelenmiştir. Bu ortamda Şekil 6.3'te görüldüğü gibi toplam yedi adet engel bulunmaktadır. Bu engellerin üç adedi sabit ve kalan dört adedi ise hareketli engellerden oluşmaktadır. Sabit engellerin biri dikdörtgensel diğerleri ise çokgenseldir. Hareketli engeller ise bir adet dikdörtgensel, bir adet eliptik ve iki adet çokgensel engellerden oluşmaktadır. Tüm engellerin merkez koordinatları, hız değerleri ve bu engelleri çevreleyen dikdörtgensel alanların boyutları Tablo 6.4'te verilmiştir.

Tablo 6.4: Orta boyutlu hareket ortamındaki engellerin bilgileri.

Engel	Merkez koordinat	Hız (cm/adım)	Alan (cm ²)
1	(190, 677)	0	40x114
2	(258, 500)	0	155x107
3	(532, 822)	0,3	169x67
4	(615, 680)	1,5	40x40
5	(607,191)	0,2	329x184
6	(1195, 145)	0,1	52x132
7	(1366, 414)	0	186x117



(a)

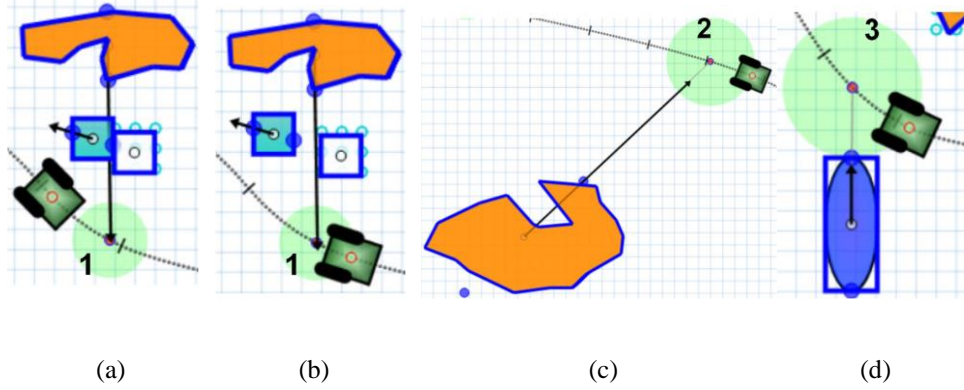


(b)

Şekil 6.3: Orta boyutlu hareket ortamında yol planlaması a) Hayali engel (IO) aşaması aktif değilken, b) Hayali engel (IO) aşaması aktif iken.

Bu hareket ortamının boyutları $1600 \times 900 \text{ cm}^2$ ve hareketin başlangıç ve hedef noktalarının koordinatları ise sırasıyla $(140,720)$ ve $(1520,380)$ olarak belirlenmiştir. Şekil 6.3a'da görüldüğü gibi 4 numaralı engel mobil robotun üzerine doğru hareket etmektedir ve bu engelden sadece sabit engellerin konumlarının dikkate alınması ile hesaplanan yol takip edilerek kaçınılması imkânsızdır. Bu sebeple, bu tez çalışmasında geliştirilen SC-IO algoritmasının Hayali engel (IO) aşaması bu örnekte aktif olmaktadır. Bölüm 4.2'de detaylıca anlatıldığı üzere bu aşamada engel ile yol kesişim

noktasına sabit hayali bir dikdörtgensel engel eklenerek, yolun tekrar hesaplanması temel olarak alınmaktadır. Şekil 6.3b’de görüldüğü üzere eklenen hayali engelin konumu da dikkate alınarak 16558 noktadan oluşan güvenli yeni bir yol hesaplanmıştır.



Şekil 6.4: Orta boyutlu hareket ortamında hareket planlaması.

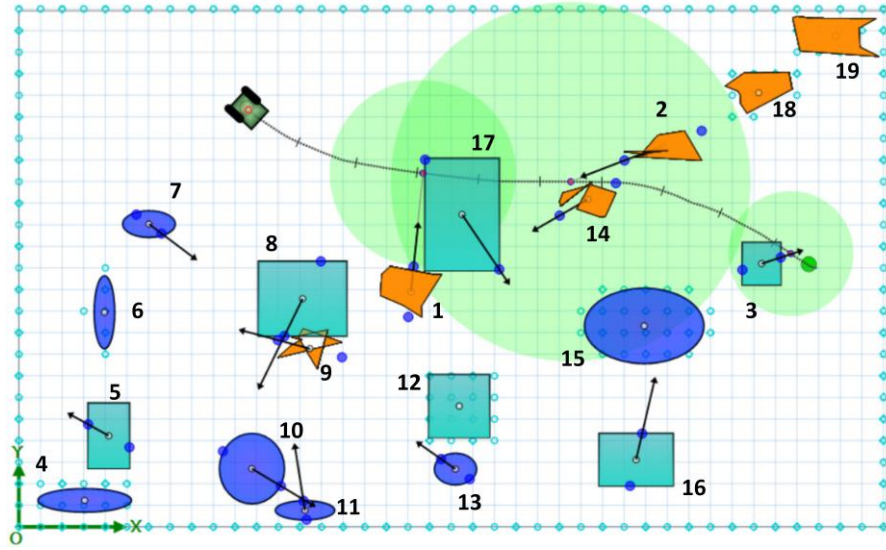
Böylece robot bu tehlike arz eden engelin etrafından dolaşarak güvenli bir şekilde bu engelden kaçınmıştır (Şekil 6.4a). Daha sonra robot ilk çokgensel engeli Şekil 6.4b’deki gibi 1 numaralı güvenlik çemberini kullanarak, ikinci çokgensel engeli ise 2 numaralı güvenlik çemberini kullanarak geçmiştir (Şekil 6.4c). Son olarak, robot eliptik engelden Şekil 6.4d’de görüldüğü gibi 3 numaralı güvenlik çemberini kullanarak kaçınmıştır. Bu örnekte robot 62 nokta/adım değerinde tek bir hızla hedefine güvenle ulaşmıştır. Şekil 6.4’ten de görüleceği üzere tüm güvenli durumlar OIP2 olarak SC-IO algoritması tarafından belirlenmiştir. Hareketli engellerin hareket doğrultuları ve yaklaşma açıları (α) ve güvenlik çemberlerinin yarıçap değerleri (r) Tablo 6.5’te verilmiştir.

Tablo 6.5: Orta boyutlu hareket ortamındaki hareketli engellerin doğrultu ve güvenlik çemberi yarıçap değerleri.

Engel	Doğrultu	α	r
3	84°	271°	37,6 cm
4	201°	163°	IO aşaması ile
5	43°	43°	73,3 cm
6	90°	90°	69,7 cm

6.3 Rastgele Parametrel Hareket Ortamı

Bu tez çalışmasında detaylı olarak açıklanan SC-IO algoritması genelleştirilmiş bir algoritma olarak geliştirilmiştir. Bu sebeple rastgele belirlenen parametrelerle de işlem yapabilme kabiliyetine sahiptir. Bu örnekteki ortam ve parametreler bu algoritma için geliştirilen GoAvoid yazılımı tarafından rastgele belirlenmiştir. Şekil 6.5'te görüldüğü gibi hareket ortamında altı adedi sabit ve onüç adedi hareketli engel olmak üzere toplam ondokuz adet engel bulunmaktadır. Sabit engeller bir adet dikdörtgen, üç adet eliptik ve iki adet çokgensel engellerden oluşmaktadır. Hareketli engeller ise beş adet dikdörtgen ve dörder adet eliptik ve çokgensel engellerden oluşmaktadır. Engeller rastgele parametrelerle yazılım tarafından oluşturulduğu için engellerin birbiri ile çarpışmaları bu örnek için göz ardı edilmiştir. Tüm engellerin merkez koordinatları, hız değerleri ve bu engelleri çevreleyen dikdörtgen alanların boyutları Tablo 6.6'da verilmiştir.



Şekil 6.5: Rastgele parametrel hareket ortamı.

Hareket ortamı rastgele parametrelerle oluşturulduktan sonra, ortamın boyutu $1406 \times 836 \text{ cm}^2$, ızgara boyutu 35 cm, referans hız değeri (v_{ref}) 38 nokta/adım, maksimum hız değeri (v_{max}) 70 nokta/adım, iterasyon sayısı (n) 848 ve güvenli mesafe değerinin de (D) 93 cm olarak belirlendiği gözlemlenmiştir. Hareketin başlangıç ve hedef noktalarının koordinatları ise rastgele olarak sırasıyla (369,679) ve (1279,426) olarak yazılım tarafından belirlenmiştir. Şekil 6.5'te görüldüğü üzere potansiyel alan metodu kullanılarak sadece sabit engellerin konumları dikkate alınarak 9841 noktadan

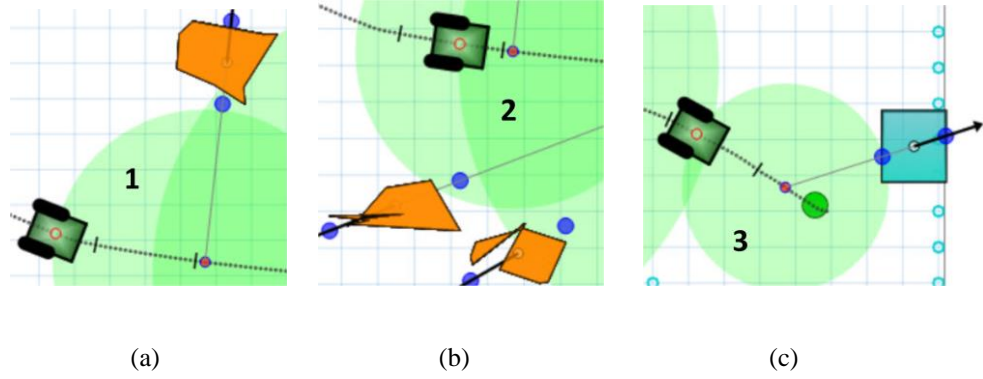
oluşan güvenli bir yol hesaplanmıştır. Mobil robotun hareketi 516 adımda tamamlanmıştır ve hareketli engellerden kaçındığı adımlar Şekil 6.6’da verilmiştir.

Tablo 6.6: Rastgele parametrelili hareket ortamındaki engellerin bilgileri.

Engel	Merkez koordinat	Hız (cm/adım)	Alan (cm ²)
1	(634, 368)	4,5	99x82
2	(1048, 620)	1,6	125x48
3	(1189, 422)	4,8	62x69
4	(107, 43)	0	151x38
5	(151, 145)	1,9	67x106
6	(139, 348)	0	33x118
7	(208, 494)	1,3	85x43
8	(466, 383)	4,6	143x121
9	(477, 288)	2	104x65
10	(367, 101)	4,2	105x114
11	(464, 23)	1,5	96x31
12	(713, 196)	0	98x102
13	(713, 90)	2,5	68x51
14	(924, 533)	0,9	91x57
15	(1013, 326)	0	193x122
16	(997, 99)	3,6	120x85
17	(712, 515)	3,4	120x182
18	(1198, 704)	0	108x66
19	(1323, 796)	0	139x65

Şekil 6.6a’da görüldüğü gibi 1 numaralı hareketli engel, ilgili güvenlik çemberi sınırlarını terk ettiğinde, robot 18 nokta/adım hız ile aynı güvenlik çemberi sınırlarına girmektedir. Daha sonra robot yavaşlayarak hızını 4 nokta/adım değerine düşürmek suretiyle 2 numaralı çokgensel engelden başarıyla kaçınmaktadır ve aynı zamanda hiç bir tehlike teşkil etmeyen 14 numaralı hareketli engeli de güvenli bir şekilde geçmektedir (Şekil 6.6b). Son olarak 3 numaralı dikdörtgensel engelden kaçınabilmek amacıyla robot, referans hız değeri olan 38 nokta/adım hız ile hareket ederek bu

engelden de başarıyla kaçınarak Şekil 6.6c’de görüldüğü gibi hedefine güvenli bir şekilde ulaşmaktadır. Bu örnekte algoritma tarafından tüm güvenli durumlar O2P1 olarak belirlenmiştir. Tüm hareketli engellerin hareket doğrultuları, yaklaşma açıları (α) ve kullandıkları güvenlik çemberlerinin yarıçap değerleri (r) Tablo 6.7’de verilmiştir.



Şekil 6.6: Rastgele parametrelili hareket ortamında hareket planlaması.

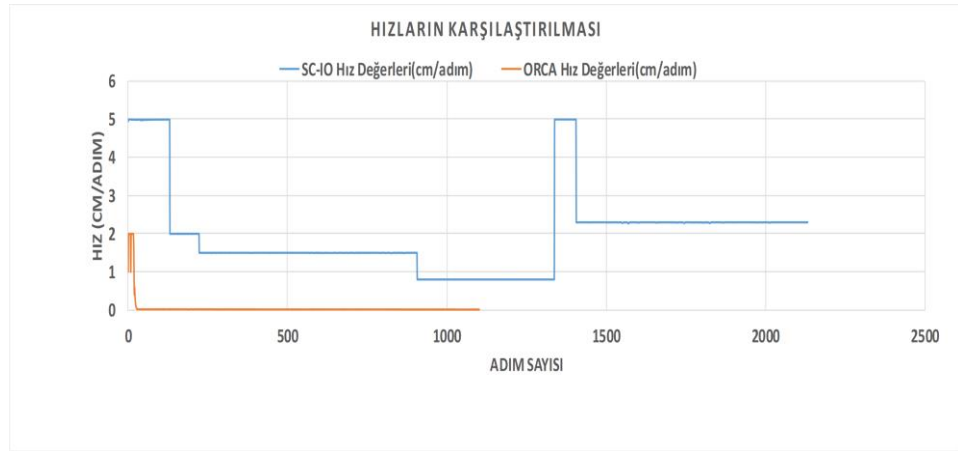
Tablo 6.7: Rastgele parametrelili hareket ortamındaki hareketli engellerin parametrelerinin değerleri.

Engel	Doğrultu	α	r
1	84°	88,17°	151 cm
2	201°	21,88°	291,4 cm
3	18°	54,61°	101,4 cm

6.4 ORCA Metodu İle Karşılaştırılması

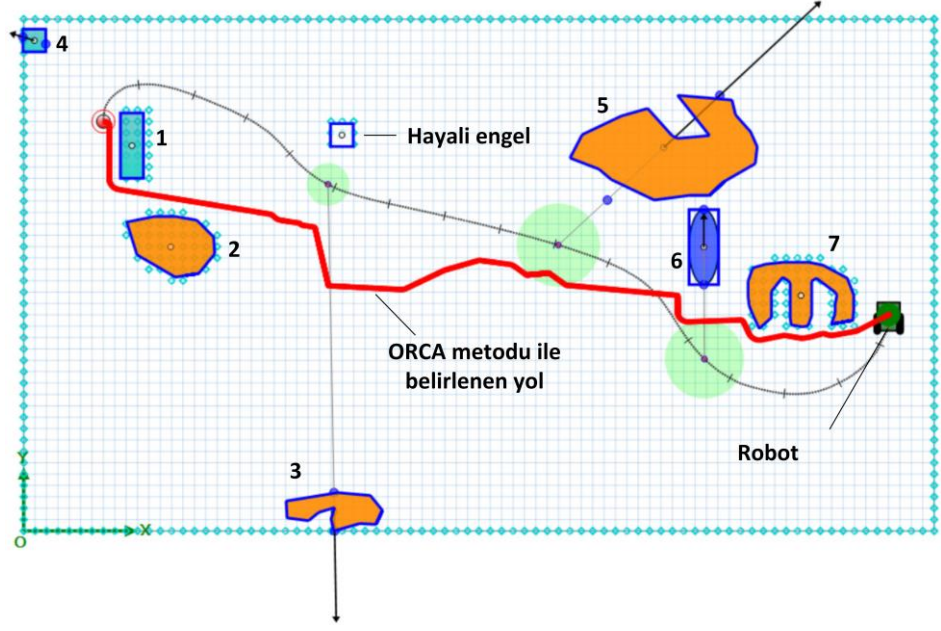
Daha önceki bölümlerde de bahsedildiği gibi, bu tez çalışmasında geliştirilen SC-IO algoritmasının uygulanabilirliğinin gösterilebilmesi amacıyla GoAvoid yazılımına ORCA kütüphanesi Van Den Berg ve diğ. (2010) eklenmiştir ve bu iki metot birbirleri ile karşılaştırılmıştır. SC-IO algoritması yay parçalarından oluşan eliptik engeller de dâhil olmak üzere farklı geometrik şekillerdeki engellerle çalışabilecek şekilde geliştirilmiştir. Ancak ORCA metodu sadece doğru parçalarından oluşan engeller ile çalışabilmektedir. Bu sebeple ORCA metodu ile karşılaştırma yapılırken eliptik engeller dikdörtgensel engellere dönüştürülerek bu metotta hesaplar yapılmıştır. SC-IO metodunda ise eliptik engel olarak dikkate alınmıştır. Yeşil daire ORCA metodundaki mobil robotu temsil etmekte iken kırmızı renkli kalın çizgi ise

ORCA kütüphanesi kullanılarak elde edilen sonuçlara göre oluşturulan yolu göstermektedir (bkz. Şekil 6.2g). İlk örnekte ORCA metodu kullanıldığında eliptik engelin kenarına gelince mobil robotun durduğu ve daha ilk sabit engeli geçerken hareket planlamasında başarısız olduğu yine ilgili şekilde gösterilmektedir. Daha önce de bahsedildiği üzere ORCA metodu her bir hareket adımı için tekrar hesaplama yapmaktadır. Yedi tanesi sabit ve yedi tanesi hareketli engel olmak üzere toplam ondört adet engelden oluşan yüksek boyutlu hareketli ortamı hesaplama yükünün fazla olmasından dolayı ORCA metodunun başarısız olmasına sebep olmaktadır. Diğer taraftan bu tez çalışmasında geliştirilen SC-IO algoritması ise robot hareketine başlamadan önce Güvenlik çemberi (SC) aşamasında sadece bir kere hesap yapmaktadır. Bu durum hesaplama yükünün düşük olmasına ve toplam hesaplama süresinin kısa olmasına sebep olmaktadır. İlgili örnek için SC-IO algoritması 2553 milisaniyede (ms) tüm hesaplamalarını gerçekleştirmiştir ve 2316 hareket adımında hedefine güvenli bir şekilde ulaşmıştır. Her iki metodun hız profilleri yüksek boyutlu hareket ortamı için Şekil 6.7’de verilmiştir.



Şekil 6.7: Yüksek boyutlu hareket ortamı için iki metodun hızlarının karşılaştırılması.

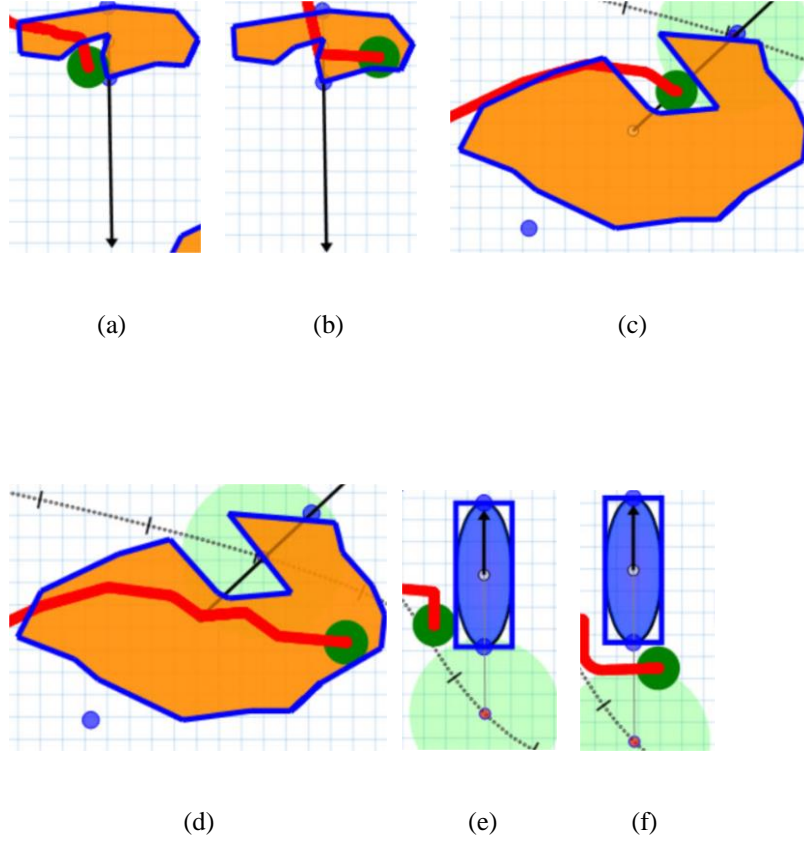
Orta boyutlu hareket ortamında ORCA metodu ile yapılan hareket planlamasında robot 1, 2 ve 7 numaralı sabit engellerden Şekil 6.8’de görüldüğü gibi başarıyla kaçınmaktadır.



Şekil 6.8: Orta boyutlu hareket ortamı için ORCA metodu ile yapılan yol planlaması

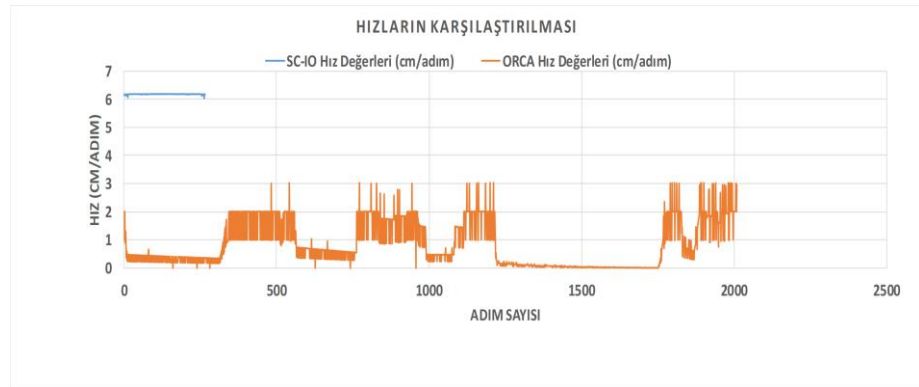
Robot hareketli engellere ulaştığında ise 3 numaralı engelin kenar çizgilerini takip etmektedir (Şekil 6.9a). Daha sonra Şekil 6.9b’de görüldüğü gibi robot bu engelin içinden geçmektedir. Robot hareketine devam ederken 5 numaralı engele ulaşarak bu engelin de kenar çizgilerini takip etmektedir (Şekil 6.9c). Yine aynı şekilde bu engelin de içinden geçerek Şekil 6.9d’de görüldüğü gibi engele çarpmaktadır. Son olarak robot 6 numaralı eliptik hareketli engele ulaşarak, ORCA metodundaki eliptik engelin dikkate alınamaması eksikliği nedeniyle bu engelin dikdörtgensel kenar çizgilerini takip ederek engelin yoldan çekilmesini beklemektedir (Şekil 6.9f).

Tüm bu hareketi ORCA metodu 2008 adımda tamamlamaktadır. Hareketi boyunca robotun, çokgensel engellerin içinden geçerek onlarla çarpışması sebebiyle bu hareket ortamında ORCA metodunun güvenli hareket planlamasını başaramadığı gözlemlenmiştir. Bu tez çalışmasında geliştirilen SC-IO algoritması ise aynı hareket ortamında hareketini 265 adımda tamamlayarak hareket planlamasını başarılı bir şekilde gerçekleştirmiştir (bkz. Şekil 6.3 ve Şekil 6.4). Her iki metodun her bir adımdaki hız değerlerinin karşılaştırılması Şekil 6.10’da verilmiştir.

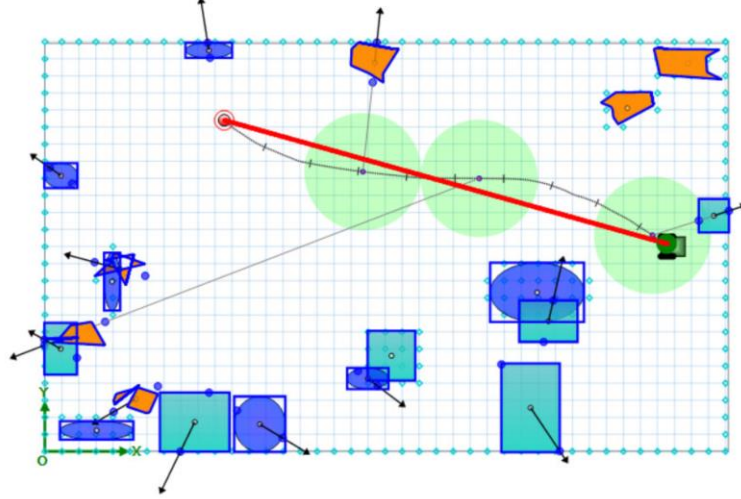


Şekil 6.9: ORCA metodu ile yapılan hareket planlaması.

Rastgele parametrelili hareket ortamında ise engeller çok hızlı hareket ettiği için, ORCA metodu ile yol planlaması yapılırken robotun hız ve doğrultusunu değiştirmesine gerek kalmamaktadır. Bu sebeple ORCA metodu boş bir hareket alanında hareket ederek 614 adımda yol planlamasını başarılı bir şekilde gerçekleştirebilmektedir (Şekil 6.11). Bu hareket ortamı için de hız karşılaştırılması Şekil 6.12’de verilmiştir.

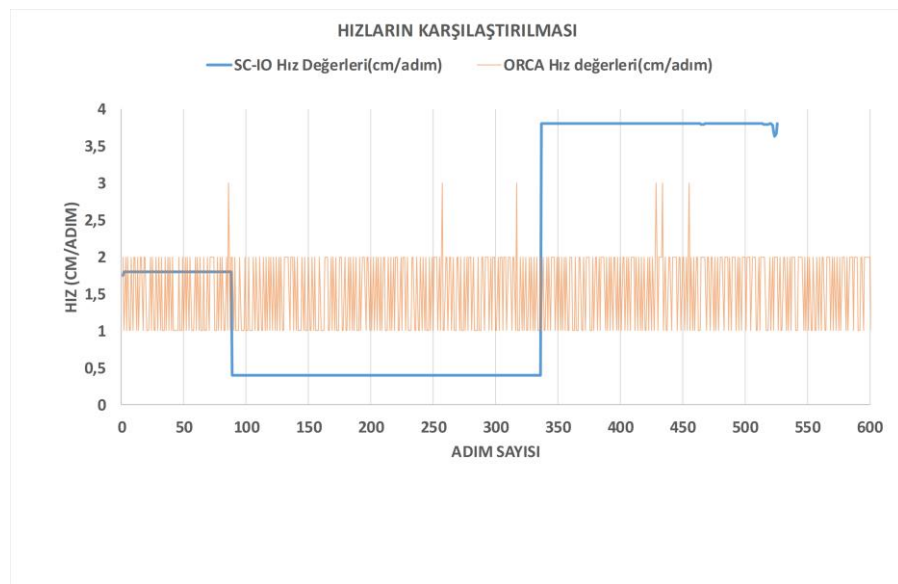


Şekil 6.10: Orta boyutlu hareket ortamı için iki metodun hızlarının karşılaştırılması.



Şekil 6.11: Rastgele parametrelı hareket ortamında ORCA metodu yardımıyla yapılan yol planlaması.

Sonuç olarak ORCA metodunun engelleri katı bir cisim olarak dikkate almak yerine onların sadece köşe çizgilerini dikkate alarak hesaplamalarını gerçekleştirdiği gözlemlenmiştir. Bu yaklaşım, metodun çokgensel engeller gibi düzensiz köşe çizgilerine sahip engellerden kaçınmaya çalıştığında başarısız olmasına sebep olmaktadır. Ancak bu tez çalışmasında sunulan SC-IO algoritmasının tüm engellerin temas noktalarını belirleyerek onları katı bir cisim olarak dikkate alması sebebiyle her türlü geometrik şekle sahip engellerden kaçınabilmektedir. Ayrıca gerektiğinde algoritmanın Hayali engel (IO) aşaması devreye alınarak tam karşıdan gelen hareketli engellerden, mevcut yolunu değiştirerek kaçınabilme kabiliyetine de sahiptir.



Şekil 6.12: Rastgele parametrelı hareket ortamı için iki metodun hızlarının karşılaştırılması.

7. HİPER-GEREĞİNDEN ÇOK SERBESTLİK DERECELİ SERİ MANİPÜLATÖRÜN BİR MOBİL ROBOTUN ÜZERİNE YERLEŞTİRİLMESİ

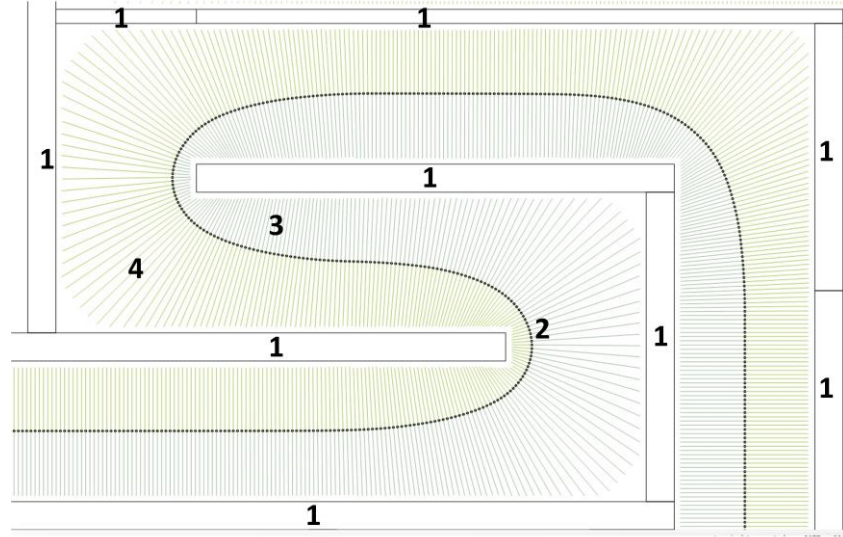
Bu tez çalışmasının ikinci kısmında, mobil robot üzerinde hiper-gereğinden çok serbestlik dereceli bir seri manipülatörün taşınması ve bu seri manipülatörün de dar alanlarda manevra yaparak belirli bir hedef noktasına ulaşması konusu üzerine çalışılmıştır. Bu hususta, daha önceki bölümlerde de anlatılan SC-IO algoritması kullanılarak manipülatör güvenli bir şekilde mobil robot üzerinde taşınmıştır. Daha sonra manipülatör için önce iki boyutlu hareket ortamlarında daha sonra da üç boyutlu hareket ortamlarında Bölüm 3'te detaylıca anlatılan potansiyel alan metodu ve yeni geliştirilen ışın konsepti kullanılarak yol ve hareket planlama algoritmaları geliştirilmiştir.

7.1 İki Boyutlu Hareket Ortamı İçin Yol Planlama Algoritması

7.1.1 Global Yol Noktaları ve Işın Doğrularının Elde Edilmesi

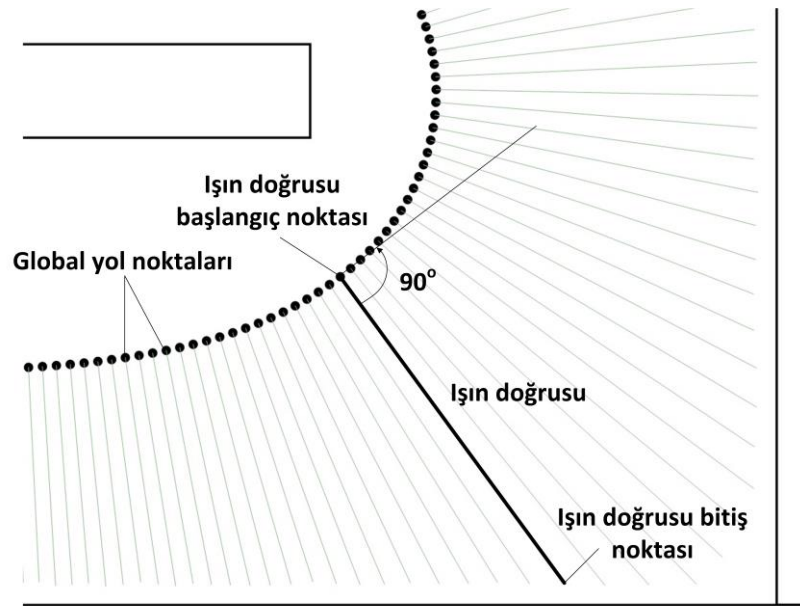
Potansiyel alan metodu kullanılarak, serbest yolun ortasındaki noktalardan oluşan global bir yol elde edilmiştir. Bu yolun yumuşatılarak olabildiğince düzgün olması çok önem arz etmektedir. Şekil 7.1'de 2 olarak numaralandırılan yumuşatılmış yolun bir kısmı gösterilmiştir ve yol noktaları engel olmayan serbest bölgelerde konumlanmış vaziyettedir. Aynı şekil üzerinde, engeller 1, sağ tarafta bulunan ışın doğruları 3 ve sol tarafta bulunan ışın doğruları ise 4 olarak numaralandırılmıştır. Bu yol manipülatörün yol planlaması için gerekli olan tüm bilgileri içermektedir.

Yol elde edildikten sonra, ışın doğruları oluşturulmuştur. Her ışın doğrusu, ilgili yol noktasından başlayıp engel sınırında son bulmaktadır. Işın doğrusu, yol ile arasındaki açı 90° olacak şekilde çizilmektedir.



Şekil 7.1: Global yol noktaları ve temel tanımlamalar.

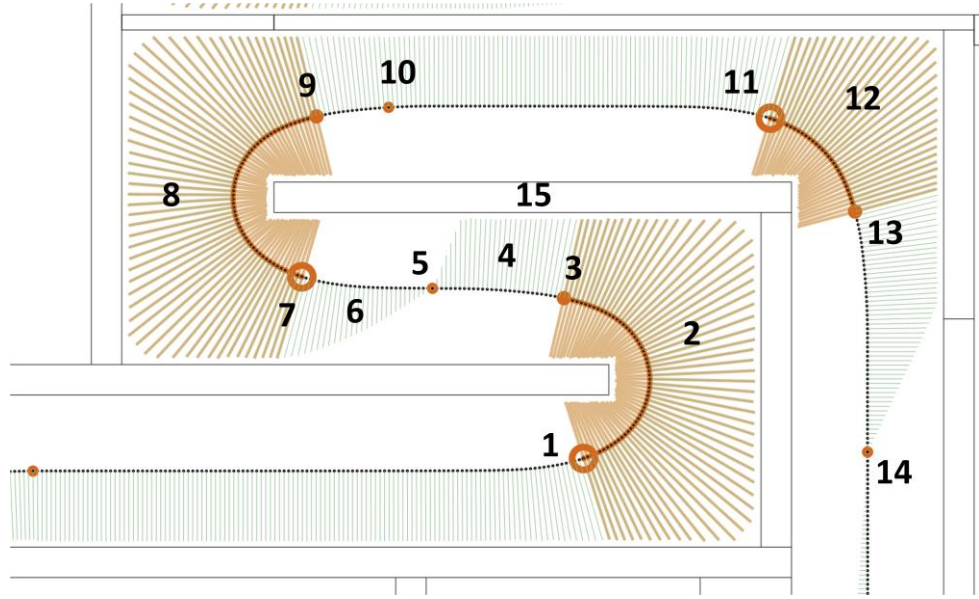
Şekil 7.2’de bir adet sağ ışın doğrusu gösterilmiştir. Aynı prosedür, sol ışın doğruları için de uygulanmaktadır. Eğer ışın doğrusu engel sınırına ulaşmadan önce ulaşabileceği maksimum uzunluğa ulaşırsa, o uzunlukta sona ermektedir. Özellikle dar alanlardaki kritik bölgelerde bu durum ortaya çıkmaktadır. Şekil 7.1’de görüldüğü gibi ışın doğrularının kapladığı alan manipülatörün sınırlandırılması gereken boş alandır. Ancak, dar alanda manevra yapılması gereken durumlarda, bu boş alanın sadece bazı bölümlerinin kullanılması gerekmektedir.



Şekil 7.2: Işın doğrusu tanımlamaları.

7.1.2 Kritik Bölgelerin Belirlenmesi

Sağ ve sol ışın doğrularının uzunlukları ve eğrilikleri her bir yol noktası için elde edilmiştir. Bu değerlerin daha önceden belirlenen bir büyüklüğün altında olduğu yol noktaları tespit edilerek kritik bölgeleri oluşturan yol noktaları belirlenmiştir. Örnek olarak, Şekil 7.3'te üç adet kritik bölge gösterilmiştir. İlk kritik bölge 2 olarak numaralandırılmıştır ve 1 numaralı noktadan başlayıp, 3 numaralı noktada sona ermektedir. İkinci kritik bölge ise 8 olarak numaralandırılmıştır ve 7 numaralı noktadan başlayıp, 9 numaralı noktada sona ermektedir. Son kritik bölge ise 12 olarak numaralandırılmıştır ve 11 numaralı noktadan başlayıp, 13 numaralı noktada sona ermektedir.



Şekil 7.3: Yol belirleme tanımlamaları.

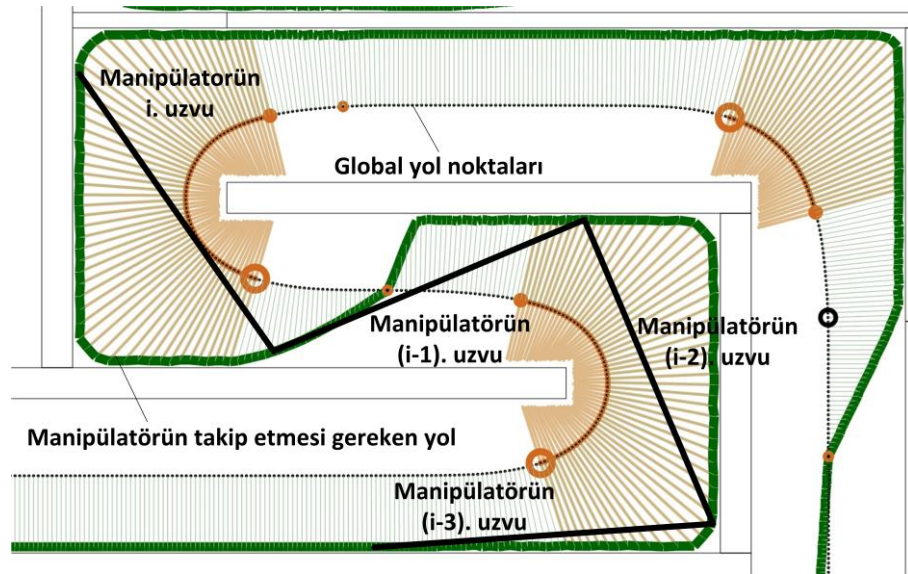
2 numaralı kritik bölge sağ tarafta olmakta iken 8 numaralı diğer kritik bölge sol tarafta olmaktadır (Şekil 7.3). Başka bir deyişle, bu iki kritik bölge birbirine ters yönde olmaktadır. Bu sebeple yol planlaması yapılırken burada yön değişimi yapılması gerekmektedir. Şekil 7.4'te görüldüğü üzere, yön değişimi 4 ile 6 numaralı bölgelerin tam orta noktasında bulunan 5 olarak numaralandırılmış olan yol noktasında gerçekleşmektedir. Çünkü 3 ve 7 numaralı noktaların arasındaki mesafe

manipülatorün bir uzvunun uzunluğuna göre çok kısa olmaktadır. Bu sebeple yön değişiminin 5 nolu noktada gerçekleşmesi gerekmektedir. Bir başka önemli husus ise programlama yapılırken yön değişimi için dizi değerleri arasında değişimler yapılması gerekliliğidir ve bu durum programlamayı zorlaştırmaktadır. Bu zorluk sağ dizi değerlerinin sol dizi değerlerine kopyalanarak, programlamada tek bir dizi kullanılarak aşılmıştır.

Şekil 7.3'te gösterilen 8 numaralı ikinci kritik bölge ve 12 numaralı üçüncü kritik bölge ele alındığında, iki kritik bölgenin de yönünün sol tarafta olduğu ve 10 numaralı noktada herhangi bir yön değişiminin gerekmediği görülmektedir. En son yol değişiminin 14 nolu noktada olması gerekmektedir. Eğer kritik bölgeler arasındaki mesafe manipülator uzuv uzunluğundan büyükse, yön değişiminin tam orta nokta yerine daha önceki bir noktada olması gerekmektedir.

7.1.3 Manipülator İçin Güvenli Yolun Belirlenmesi

Kritik bölgelerin ve yol değişim noktalarının belirlenmesinden sonra, manipülatorün takip etmesi gereken yol Şekil 7.4'te kalın yeşil çizgilerle gösterildiği gibi elde edilmiştir ve dört adet manipülator uzvunun uç noktalarının güvenli yol noktalarını takip etmesi simüle edilmiştir.



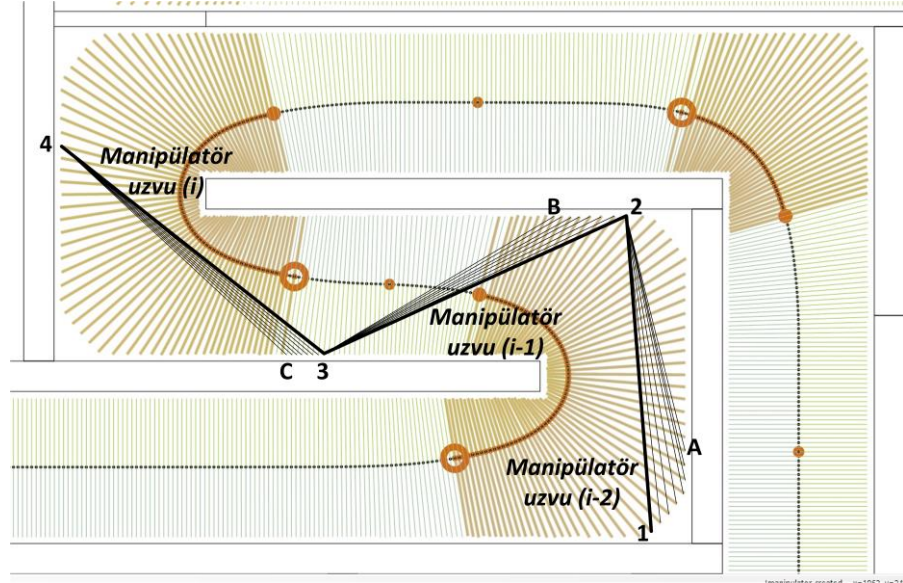
Şekil 7.4: Manipülator için belirlenen yol.

7.1.4 Kontrol Stratejisi

Bu tez çalışmasında geliştirilen yol planlama algoritması basit bir kontrol stratejisine sahip olduğu için herhangi bir akış şemasına ihtiyaç duymamaktadır. Yol planlaması için yapılması gereken tek şey aşağıdaki prensibin yerine getirilmesidir.

“Manipülâtörün son uzvunun uç noktası hedef noktaya ulaşana kadar, manipülâtörün tüm uzuvlarının uç noktalarının sol ve sağ ışın doğrularının uç noktalarından oluşan güvenli yol noktalarının takip edilmesi sağlanmalıdır.”

Manipülâtör uzuvlarının uç noktalarının belirlenen güvenli yol üzerinde hareket etmesi gerçek zamanlı olarak basit nümerik bir prosedür yardımıyla gerçekleştirilmiştir. Üç adet arka arkaya sıralanmış birbirine bağlı manipülâtör uzvu sırasıyla, i , $(i-1)$, $(i-2)$ olarak numaralandırılmıştır ve bu uzuvların güvenli yol noktalarına yerleşmesi Şekil 7.5’te gösterilmiştir.



Şekil 7.5: Manipülâtörün yol üzerine yerleştirilmesi.

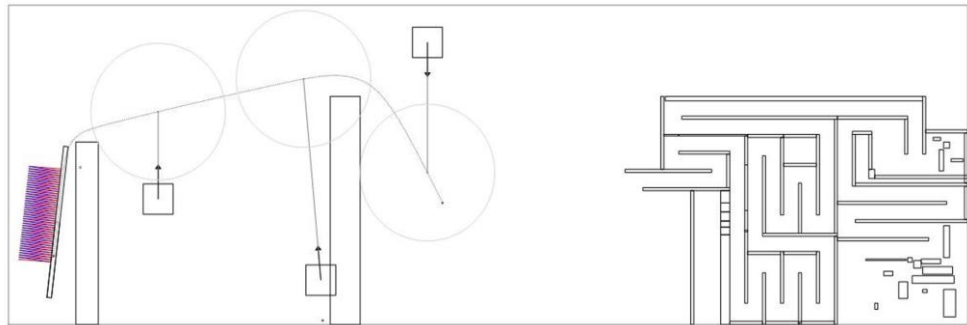
i olarak numaralandırılan uzvun yakın uç noktasının 4 numaralı noktaya hareket ettiği durum göz önüne alındığında, bu uzvu C noktasından 3 noktasına kadar her bir nokta için kontrol edilmektedir. Bu noktaların 4 numaralı nokta ile arasındaki mesafenin ilgili uzvu uzunluğuna yakın ya da tam eşit olduğu durumlarda ilgili uzvun uç noktası belirlenen noktaya yerleştirilmek suretiyle tüm manipülâtör uzuvlarının hareket planlaması gerçekleştirilmektedir. Bu örnek için robot uzvunun yerleştirilmesi

4 ve 3 numaralı noktalar arasında gerçekleşmektedir (Şekil 7.5). Benzer kontrol prosedürü (*i-1*) ve (*i-2*) olarak numaralandırılan uzuvlar için de yine aynı şekilde uygulanmaktadır. Bu uzuvlar için araştırma B noktasından 2 noktasına ve A noktasından 1 noktasına kadar yapılmaktadır. 2 ve 1 noktaları bu araştırma sonucunda belirlenmiştir. Sonuç olarak üç adet manipülâtör uzvunun üzerine yerleştirilmesi gereken noktalar 3, 2 ve 1 olarak belirlenmiştir.

7.2 İki Boyutlu Hareket Ortamı İçin Simülasyonlar

Simülasyonların gerçekleştirilebilmesi amacıyla Microsoft® Visual C#.NET ortamında bir bilgisayar yazılımı geliştirilmiştir ve bu yazılım Intel® Core™ i7-3520M 2.90 GHz işlemcisine ve 8 GB RAM değerine sahip bir dizüstü bilgisayarda çalıştırılmıştır. Grafikler için Graphics Device Interface (GDI+) kütüphanesi kullanılmıştır.

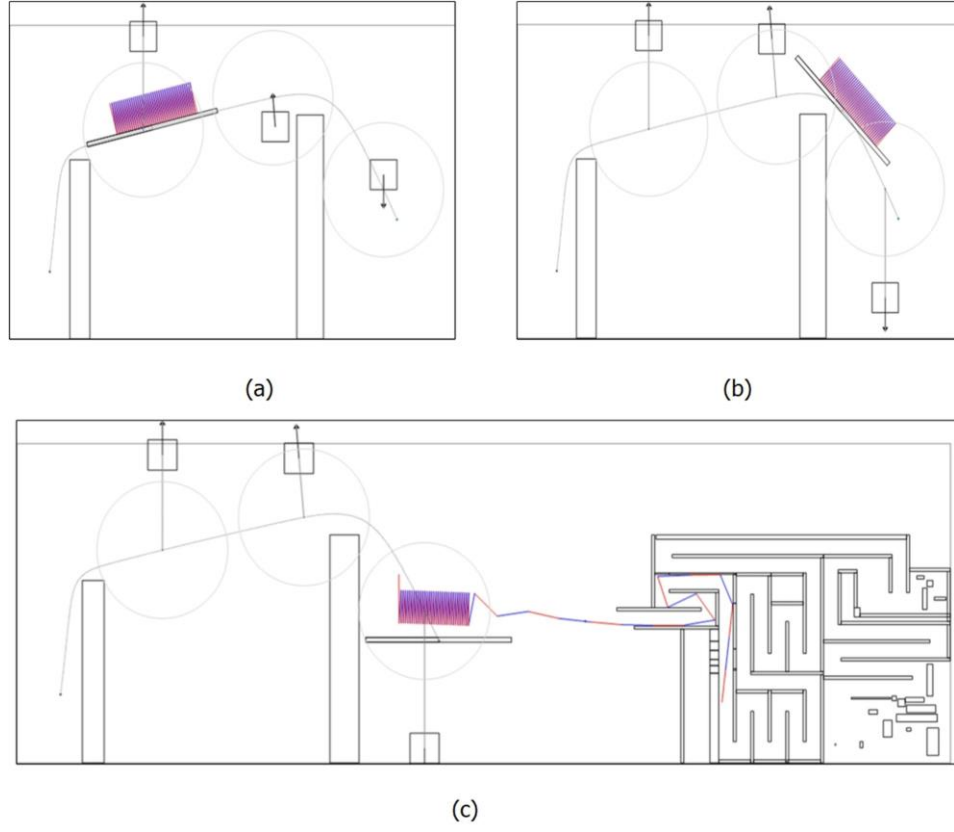
Bu bölümde, iki boyutlu hareket ortamı için sadece bir adet örnek verilmiştir. Bu örneğin hareket ortamı Şekil 7.6’da gösterilmiştir. Şekilden de görüldüğü üzere hareket ortamının sol tarafında mobil robotun üzerine yerleştirilmiş 83 adet uzva sahip bir manipülâtör bulunmaktadır. Ayrıca hareket ortamında sabit ve hareketli engeller mevcuttur. Hareket ortamının sağ tarafında ise en zor manevralardan basit manevralara kadar çeşitli manevra bölgeleri vardır.



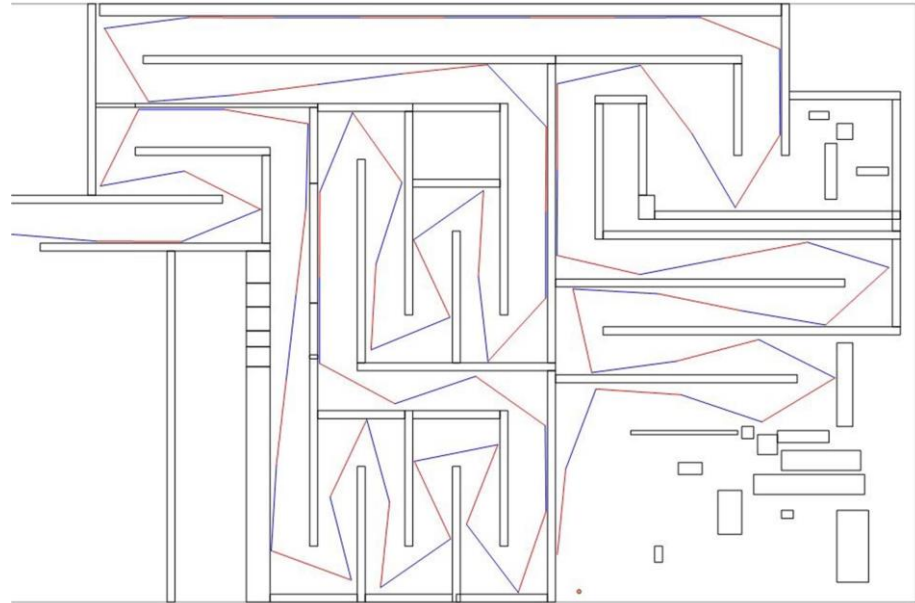
Şekil 7.6: İki boyutlu hareket ortamı.

Öncelikle manipülâtör, sabit ve hareketli engeller arasından güvenli bir şekilde mobil robot üzerinde taşınarak manevra bölgelerine sahip hareket alanına taşınmaktadır (Şekil 7.7). Daha sonra manipülâtör, geliştirilen algoritma yardımıyla

dar alan bölgelerinde gerekli manevraları yaparak hedefine Şekil 7.8’de görüldüğü gibi güvenli bir şekilde ulaşmaktadır.

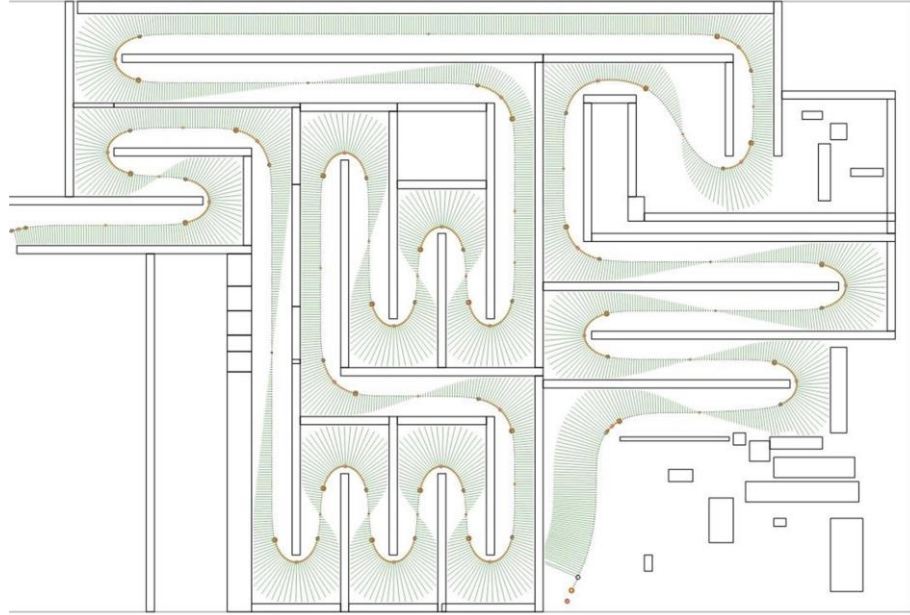


Şekil 7.7: Mobil robotun ve üzerindeki manipülörün hareketi a) İlk sabit ve hareketli engellerden kaçınması, b) İkinci sabit ve hareketli engellerden kaçınması, c) Manipülörün yol planlaması.

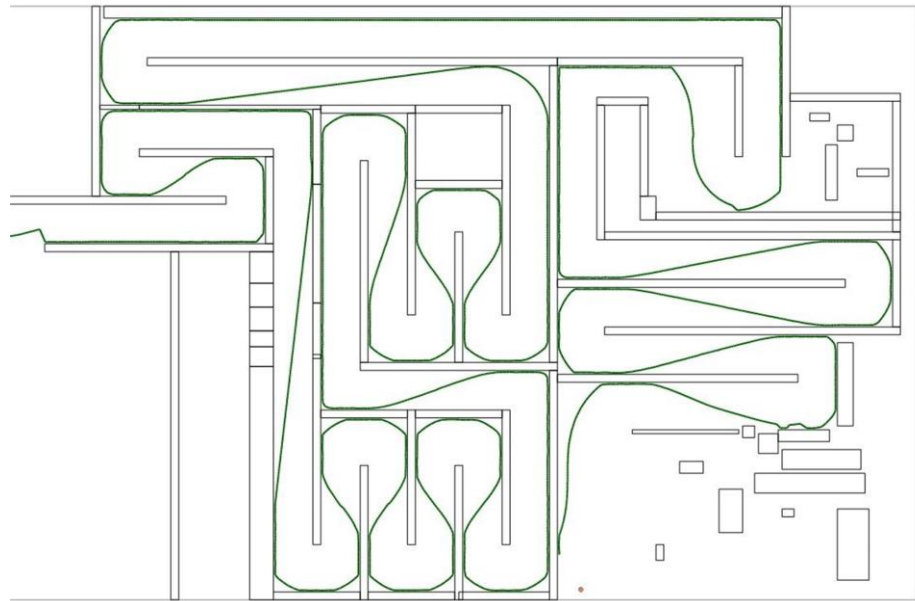


Şekil 7.8: Dar bölgelerde manipülörün hedefine güvenle ulaşması.

Metodun nasıl çalıştığının görselleştirilebilmesi amacıyla, Şekil 7.9’da kritik bölgeler ve ışın doğruları, Şekil 7.10’da ise dar alanda manevra yapılırken manipülatörün uç noktasının takip etmesi gereken yol kalın yeşil çizgilerle gösterilmiştir. Metodun verimliliğindeki başarısı bu belirlenen yoldan kaynaklanmaktadır. Manipülatörün hareketi boyunca uç noktasının takip ettiği noktalardan oluşan alan bir başka deyişle süpürdüğü alan ise Şekil 7.11’de verilmiştir.



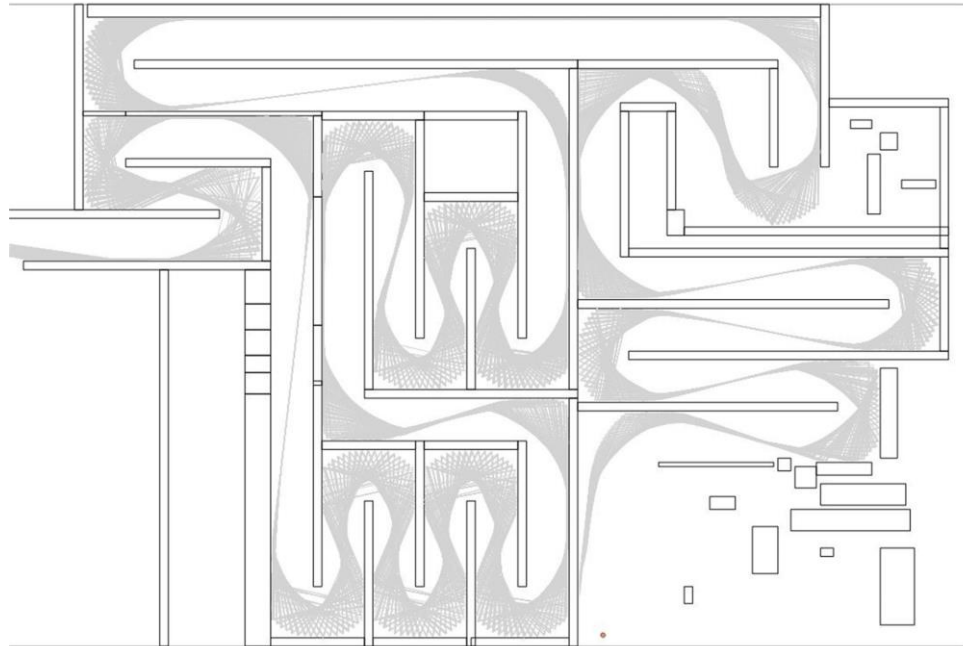
Şekil 7.9: Kritik bölgeler ve ışın doğruları.



Şekil 7.10: Manipülatörün uç noktasının takip etmesi gereken yol.

Bu alan manipülatörün hareketi için belirlenen tüm konfigürasyonların bir araya getirilmesi ile elde edilmiştir ve manipülatörün uzuv uzunluğu ile karşılaştırıldığında çok dar bölgelerden oluşmaktadır.

Bu örnekte iki nokta arasındaki mesafe 12 piksel olarak alınmıştır. Toplam 2384 adet ışın doğrusu oluşturulmuştur. Potansiyel alan 2400 ms'de 7000 iterasyon sonucunda hesaplanmıştır. Bu kadar yüksek iterasyona ihtiyaç duyulmasının sebebi çok sayıda engel içeren bir hareket ortamı olmasından dolayıdır. Ayrıca potansiyel alan hesaplamalarının bütün hareket alanına yayılması gerekmektedir. Global yol belirlendikten sonra, manipülatörün uç noktasının takip etmesi gereken yol, geliştirilen algoritma ile 15 ms gibi çok kısa bir sürede belirlenmiştir. Manipülatörün hedef noktasına ulaşması için her bir uzvun yol noktalarına yerleştirilmesi ise 752 ms'de tamamlanmıştır. Hareket simüle edildiğinde, manipülatörün hedefine 32 saniyede (s) güvenli bir şekilde ulaştığı gözlemlenmiştir. Bu süre çok uzun olarak görülse de çok dar alana sahip bir hareket alanı ve yavaş çalışan GDI+ kütüphanesi dikkate alındığında bu süre makul hale gelmektedir.

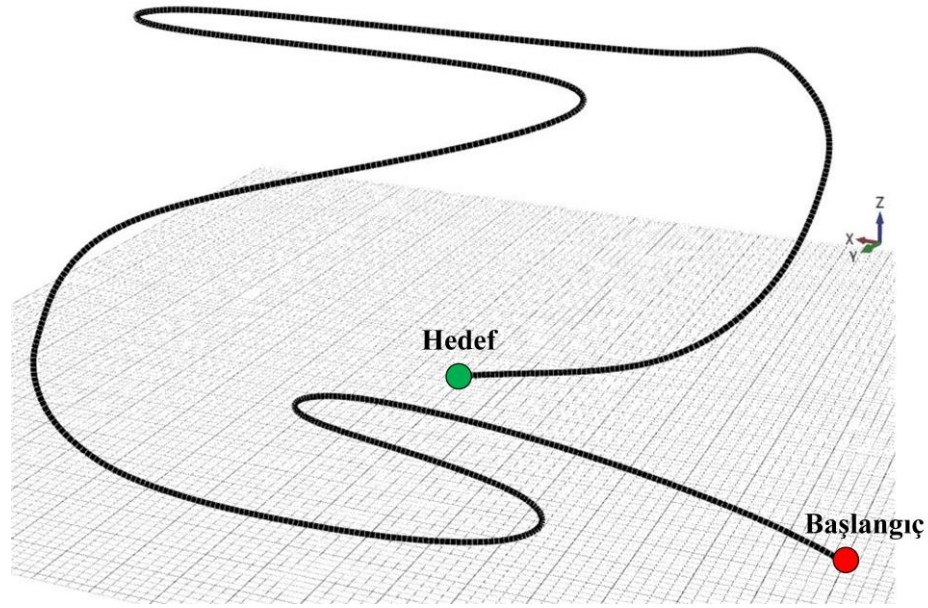


Şekil 7.11: Hareketi boyunca manipülatörün süpürmesi gereken alanlar.

7.3 Üç Boyutlu Hareket Ortamı İçin Yol Planlama Algoritması

7.3.1 Işın Çiftlerinin Elde Edilmesi

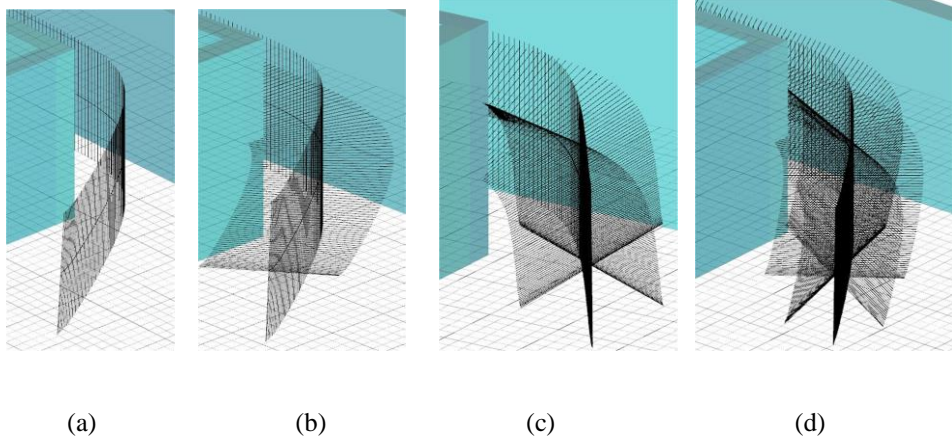
Üç boyutlu hareket ortamlarında potansiyel alan metodu ile global yol hesabından daha önce Bölüm 3.1.3'te ve 3.1.4'te bahsedilmiştir. Şekil 7.12'de örnek olarak potansiyel metodu ile elde edilen global yol gösterilmiştir.



Şekil 7.12: Üç boyutlu hareket ortamında potansiyel alan metodu uygulaması (Engeller gösterilmemiştir).

İki boyutlu hareket ortamları ile kıyaslandığında, üç boyutlu hareket ortamlarında ışın doğrularını elde etmek daha zor olmaktadır. Sol ve sağ ışın doğruları iki boyutlu hareket ortamlarında tüm serbest bölgeleri kapsamakta iken üç boyutlu hareket ortamlarında bu durum geçerli olmamaktadır. Şekil 7.13a'da görüldüğü üzere ışın çiftlerinin oluşturulması gerekmektedir ve bu ışın çiftleri tam olarak serbest bölgeleri kapsamamaktadır. Bu sebeple, üç boyutlu hareket ortamları için farklı bir prosedürün uygulanması gerekmektedir. Serbest bölgelerin kapsanabilmesi amacıyla, ardışık iki yol noktasının oluşturduğu doğruya dik olacak şekilde ışın doğruları oluşturulmuştur. Işın çifti sayısı arttıkça, daha fazla serbest alan kapsamaktadır. Her ne kadar ışın çifti sayısının artması çözünürlüğü arttırsa da, hem hesaplama yükünün

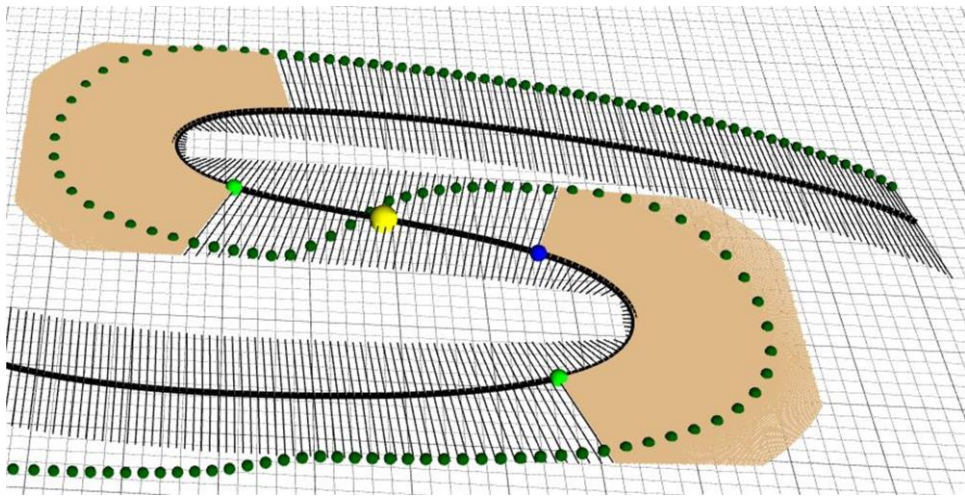
artmasına sebep olmaktadır hem de manipülatörün manevra kabiliyetine herhangi bir katkıda bulunmamaktadır. 1, 2, 3 ve 4 adet ışın çiftleri Şekil 7.13'te gösterilmiştir.



Şekil 7.13: Işın çiftleri a) Bir adet ışın çifti, b) İki adet ışın çifti, c) Üç adet ışın çifti, d) Dört adet ışın çifti.

7.3.2 Kritik Bölgelerin Belirlenmesi ve Yön Değişimi

Bölüm 7.1.2'de verilmiş olan iki boyutlu hareket ortamında kritik bölgelerin belirlenmesinde kullanılan kriterler üç boyutlu hareket ortamları için de kullanılmıştır. Sadece üç boyutlu hareket ortamı için farklı olarak her bir ışın çifti için ayrı ayrı kritik bölgeler belirlenmiştir. Örnek olarak, kritik bölgeler ve kritik bölgeler arasında belirlenen üç adet nokta Şekil 7.14'te gösterilmiştir.



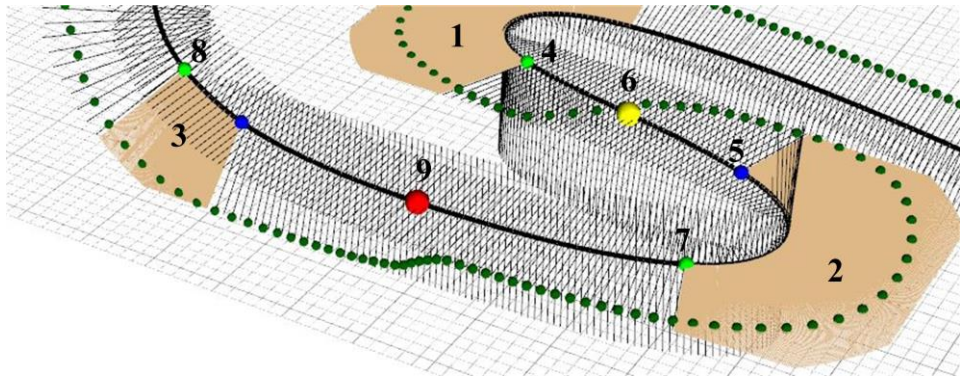
Şekil 7.14: Tek ışın çifti üzerindeki kritik bölgeler ve noktalar.

Şekil 7.14'te görüldüğü üzere, ilk kritik bölgenin son ışın doğrusu yeşil renkli küre ile gösterilirken, ikinci kritik bölgenin ilk ışın doğrusu mavi renkli küre ile gösterilmiştir. Bu iki noktanın orta noktası ise sarı renkli küre ile gösterilirken, belirlenen yol ise yeşil renkli ard arda noktalarla gösterilmiştir. Bu noktalar yol değişim noktasının belirlenmesinde kullanılmaktadır ve bu değişim sarı küre ile gösterilen noktada gerçekleşmektedir.

7.3.3 Işın Çifti Değişirme Noktalarının Belirlenmesi

Kritik bölgelerde, bir ışın çifti kritik bölgeye sahip iken diğerinin kritik bölgeye sahip olmadığı bir durum gerçekleşebilmektedir. Böyle bir durumda, yol planlaması yapılırken manipülatörün manevra yapabilmesini sağlamak amacıyla, kritik bölgeye sahip olmayan ışın çiftinden kritik bölgeye sahip olan ışın çiftine geçiş yapılması gerekmektedir. Bu geçiş Bölüm 7.3.2'de detaylıca açıklanmış olan iki boyutlu hareket ortamında tek bir ışın çifti üzerinde gerçekleştirilen yön değişiminden farklı olmaktadır. Üç boyutlu hareket ortamında ışın çiftleri arasında değişim yapılmak suretiyle manipülatörün gerekli manevrayı yapabilmesi sağlanmıştır.

Şekil 7.15'te görüldüğü üzere, 1 ve 2 olarak numaralandırılmış kritik bölgeler ilk ışın çiftine ait iken, 3 olarak numaralandırılan kritik bölge ise ikinci ışın çiftine aittir. Yol noktaları, yeşil renkli ard arda noktalarla gösterilmiştir ve ışın değişimi iki ışın çiftinin bitiş noktaları olan 7 ve 8 numaralı noktaların tam orta noktası olan 9 numaralı noktada gerçekleşmektedir.



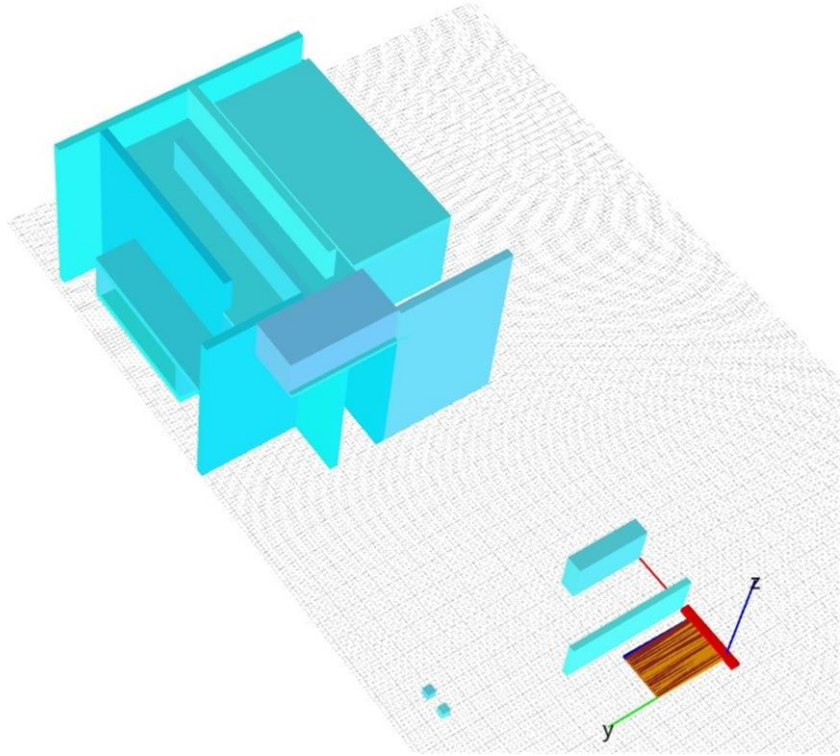
Şekil 7.15: İki ışın çifti üzerindeki kritik bölgeler ve noktalar.

Daha önce de bahsedildiği gibi engellerle dolu dar alanlardan oluşan üç boyutlu hareket ortamlarında manipülatör için güvenli yol belirlendikten sonra, manipülatörün uç noktasının bu yol noktalarını takip etmesi suretiyle, manipülatörün engellerden kaçınarak hedef noktasına güvenle ulaşması sağlanmaktadır.

7.4 Üç Boyutlu Hareket Ortamı İçin Simülasyonlar

Simülasyonların gerçekleştirilebilmesi amacıyla Microsoft® Visual C#.NET ortamında bir bilgisayar yazılımı geliştirilmiştir ve bu yazılım Intel® Core™ i7-6700HQ 2.60 GHz işlemcisine ve 16 GB RAM değerine sahip bir dizüstü bilgisayarda çalıştırılmıştır. Grafikler için WPF (Windows Presentation Foundation) ve ücretsiz olarak sunulmuş olan kütüphane kullanılmıştır (Helix Toolkit 2022).

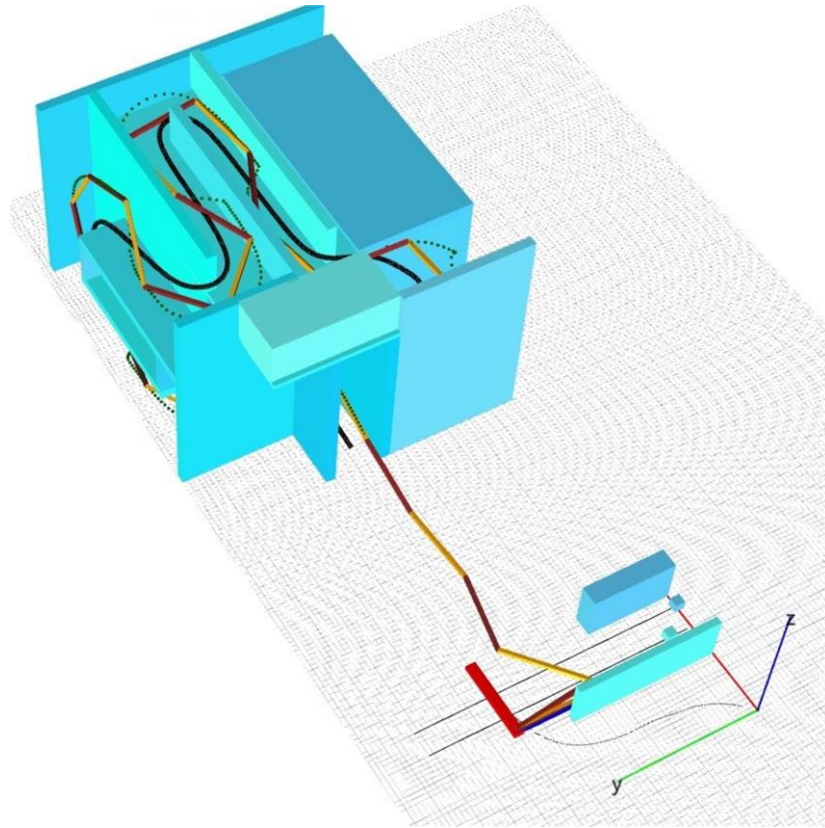
Bu bölümde iki adet hareket ortamı için manipülatörün yol planlaması yapılmıştır. İlk hareket ortamı Şekil 7.16'da görüldüğü üzere birbirini takip eden birçok dar alan bölgelerinden oluşmaktadır.



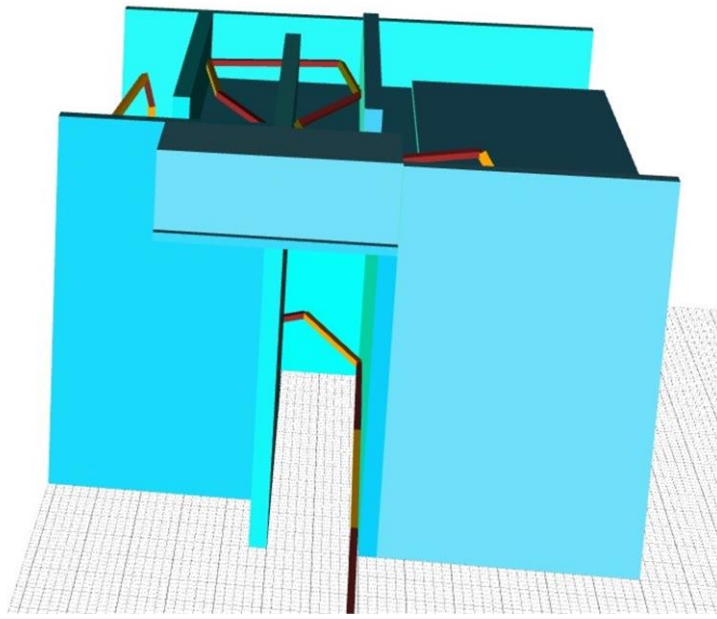
Şekil 7.16: İlk örneğin üç boyutlu çalışma ortamı.

Hareket ortamının sadece birçok dikdörtgensel engellerden oluşmasına rağmen, bu geometrik yapıdaki engeller manipülatör yol planlaması yapılırken en zor manevra stratejisinin geliştirilmesini gerektirmektedir. Diğer bir deyişle, manipülatör bu geometrik yapıdaki engellerden manevra yaparak kaçınabildiği takdirde, birçok küçük ya da büyük engellerin olduğu hareket ortamlarında daha kolay bir şekilde manevra yaparak engellerden kaçınabilme yeterliliğine sahip olmaktadır. Çünkü potansiyel alan metodu ile global yol hesaplanırken bu durum herhangi bir önem arz etmemektedir. İlk hareket ortamında orijin noktasına yakın bir yerde mobil robotun üzerinde konumlanmış olan 35 adet uzva sahip bir manipülatör bulunmaktadır (Şekil 7.16). Ayrıca sabit ve hareketli engeller manipülatörün manevra yapacağı ana engel yapısının dışında bulunmaktadır.

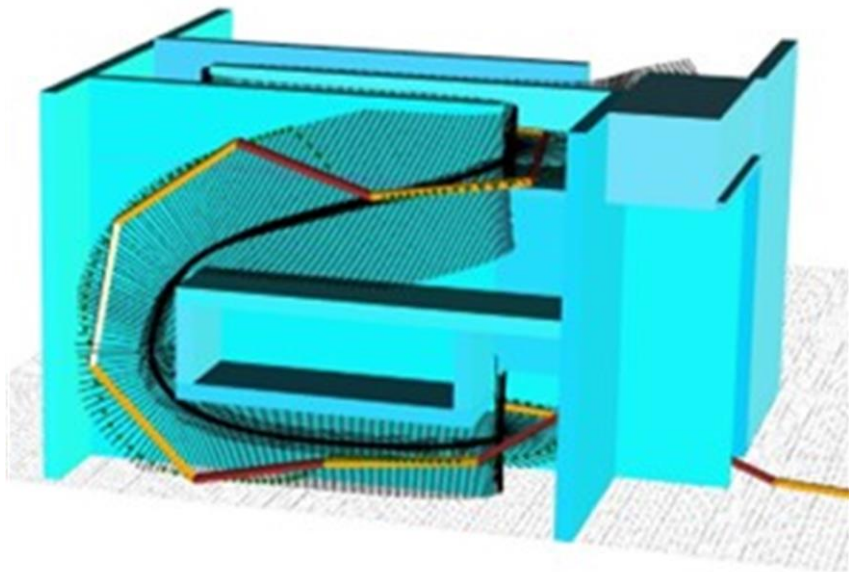
Şekil 7.17’de görüldüğü üzere manipülatör, mobil robot üzerinde taşınarak ana engel yapısının girişine sabit ve hareketli engeller arasından engellere çarpmadan ulaştırılmaktadır. Daha sonra, manipülatör ana engel yapısının içerisinde manevra yaparak hedef noktaya güvenle varmaktadır.



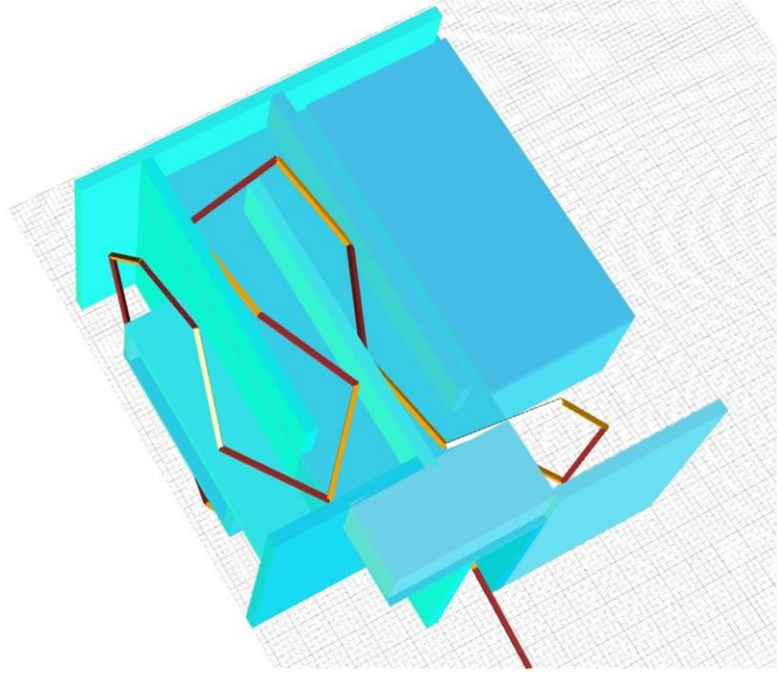
Şekil 7.17: Mobil robot üzerinde manipülatörün hareketi.



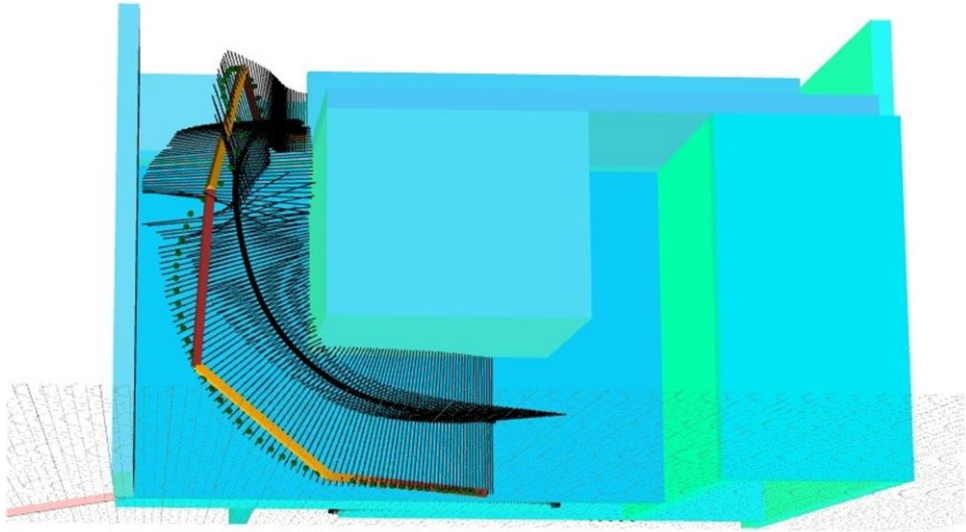
(a)



(b)



(c)

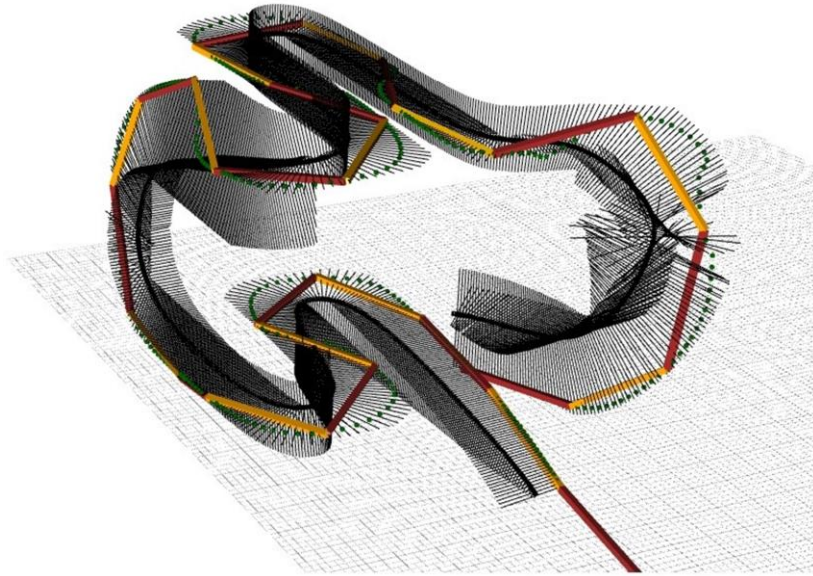


(d)

Şekil 7.18 (devam): Manipülator hareketinin farklı açılardan görüntüleri a) Hareketin başlangıcı, b) Işın çiftinin değiştirilmesi, c) Yardımcı elemanlar olmadan dar alan manevralarının yapılması, d) Hareketin bitişi.

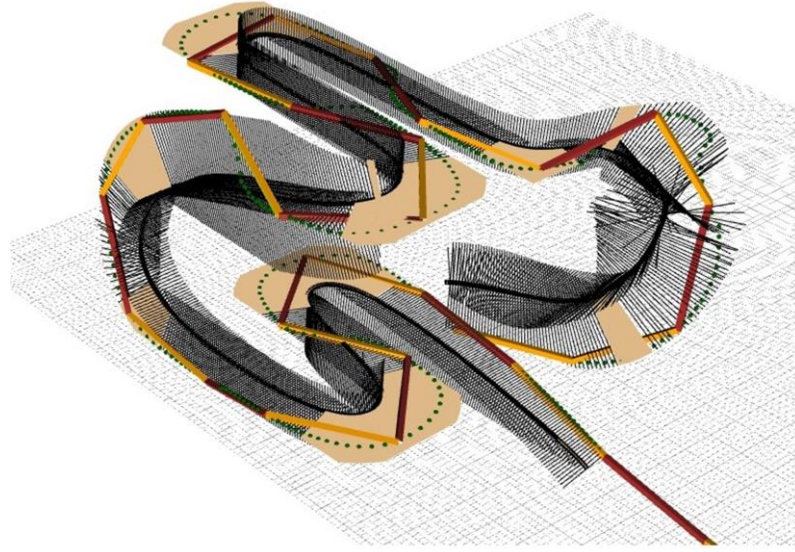
Hareket ortamının üç boyutlu olduğu dikkate alındığında, manipülatörün hareketini göstermek çok kolay olmamaktadır. Bu sebeple, Şekil 7.18’de manipülatörün hareketinin farklı açılardan resimleri, ışın doğruları, belirlenen yol noktaları vb. yardımcı elemanları da içerecek şekilde verilmiştir. Şekilden de görüldüğü üzere, manipülatör oldukça dar bölgelerden manevra kabiliyetini kullanarak hedef noktasına güvenli bir şekilde ulaşmaktadır.

Metodun çalışma prensibinin anlaşılabilmesi amacıyla, ek resimler eklenmiştir. Şekil 7.19’da engeller gizlenerek ışın çiftleri ve metot ile belirlenen yol verilmiştir. Şekil 7.20’de kritik bölgeler de gösterilmiştir.



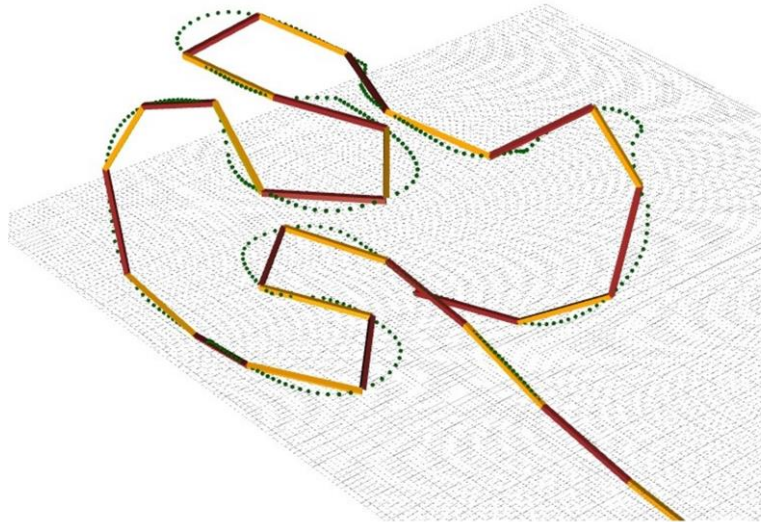
Şekil 7.19: Işın çiftleri ve manipülatör yolunun gösterilmesi.

Şekil 7.21’de ise sadece yeşil renk ile manipülatörün uç noktasının takip etmesi gereken yol verilerek, manipülatör uzuvlarının yola nasıl yerleştiği gösterilmiştir. Daha önce de bahsedildiği üzere geliştirilen metodun verimliliği elde edilen yoldan kaynaklanmaktadır. Bu örnek için ard arda iki nokta arasındaki mesafe (d_p) 16 piksel olarak alınmıştır. Toplamda 6564 adet ışın doğrusu geliştirilen metot tarafından çizilmiştir. Potansiyel alan metodu ile 13022 ms’de 600 iterasyon yapılarak global yol hesaplanmıştır. Yüksek iterasyon sayısının gerekliliğinin sebebi daha önce de bahsedildiği üzere hareket ortamının büyüklüğü ve sayıca çok engellerden oluşmasıdır.



Şekil 7.20: Kritik bölgeler.

Potansiyel alan metodunun hedef noktasına doğru güvenli bir yol hesaplayabilmesi için potansiyel alan değerlerinin yeterince yayılması gerekmektedir. Bu durum da gerekli olan iterasyon sayısını arttırmaktadır.

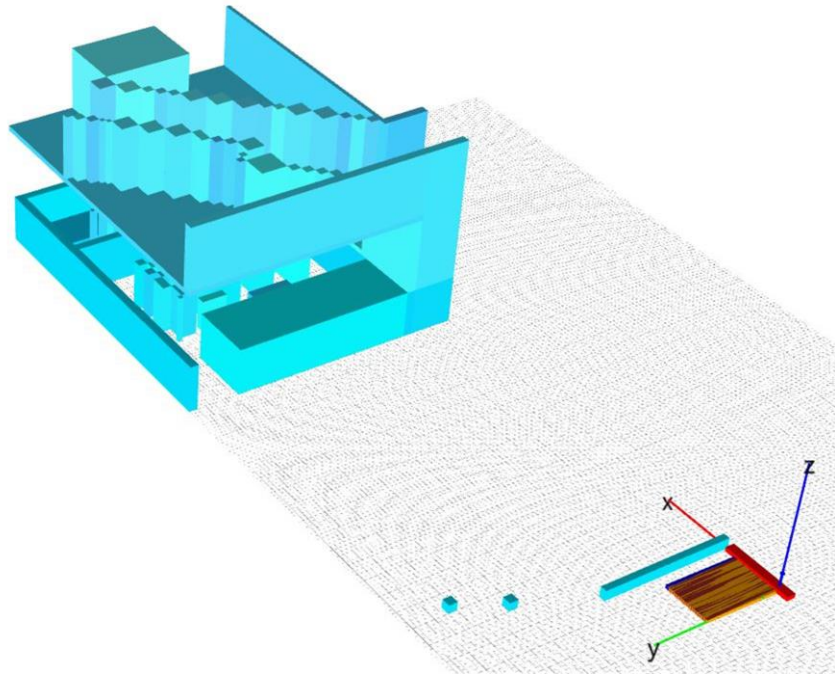


Şekil 7.21: Üç boyutlu hareket ortamında dar alan manevralarının yapılması.

Global yol noktaları belirlendikten sonra, bu noktalardan ışın doğrularının çizilmesi 15738 ms'de tamamlanmıştır. Manipülator için gerekli olan yol, üç boyutlu bir hareket ortamı için şaşırtıcı bir şekilde çok kısa bir sürede 117 ms'de elde edilmiştir. Manipülator hedef noktasına 14623 ms'de ulaşmıştır.

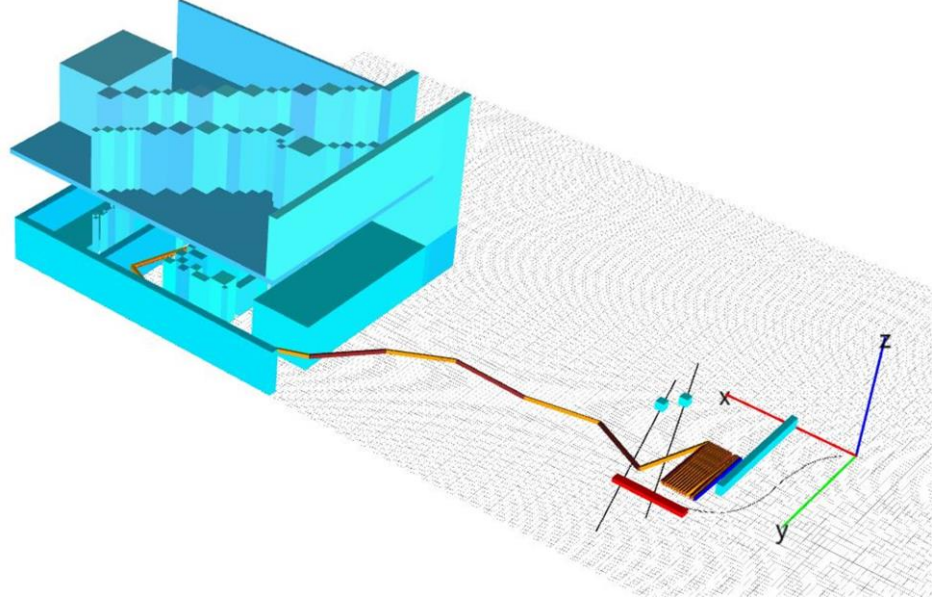
Işın doğrusu sayısı hesaplama sürelerini etkilemektedir. Örneğin d_p değeri 8 piksel olarak alındığında, ışın doğrusu sayısı tam iki katına çıkarak 13128 değerine yükselmektedir. Bu d_p değeri ile potansiyel alan metodu 600 iterasyon ile 13280 ms'de hesaplanmıştır. Global yol noktaları belirlendikten sonra, bu noktalardan ışın doğrularının çizilmesi ise 32500 ms'de tamamlanmıştır. Manipülator için gerekli olan yol, üç boyutlu bir hareket ortamı için yine şaşırtıcı bir şekilde çok kısa bir süre olan 281 ms'de elde edilmiştir. Manipülator ise hedef noktasına 30251 ms'de ulaşmıştır.

İkinci örneğin hareket ortamında da ilk örneğe benzer şekilde orijin noktasına yakın bir yerde mobil robotun üzerinde konumlanmış olan 35 adet uzva sahip bir manipülator ve bu manipülatorün manevra yapacağı ana engel yapısının dışında sabit ve hareketli engeller bulunmaktadır (Şekil 7.22).



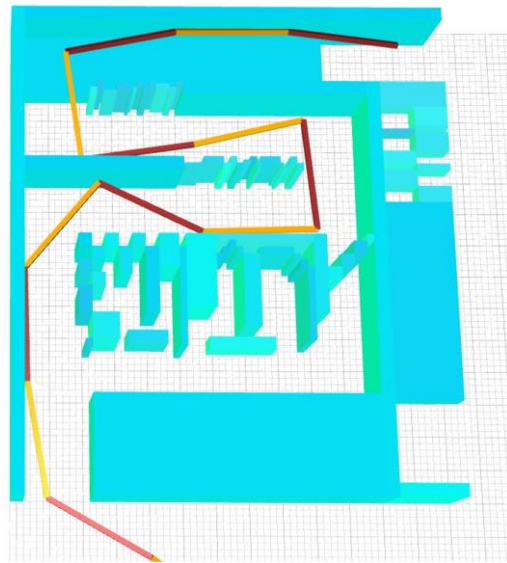
Şekil 7.22: İkinci örneğin üç boyutlu hareket ortamı.

Şekil 7.23'te görüldüğü üzere manipülatör, mobil robot üzerinde taşınarak ana engel yapısının girişine sabit ve hareketli engeller arasından engellere çarpmadan ulaştırılmaktadır. Daha sonra, manipülatör ana engel yapısının içerisinde manevra yaparak hedef noktaya güvenli bir şekilde ulaşmaktadır.



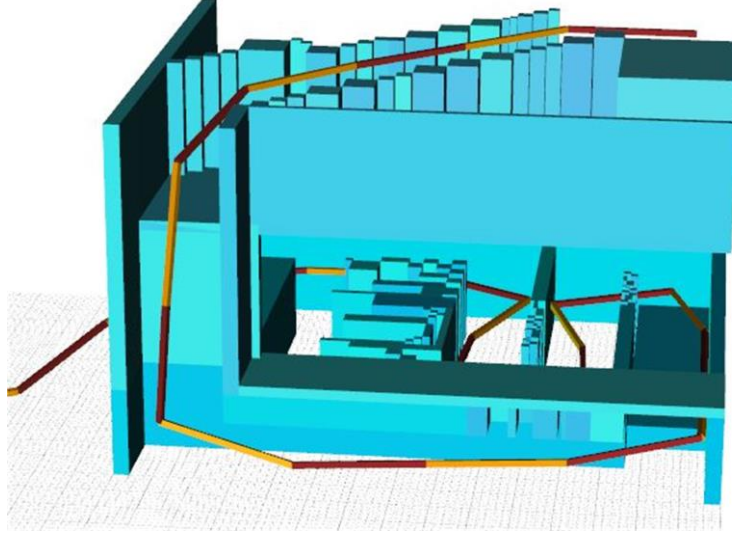
Şekil 7.23: Manipülatörün mobil robot üzerinde taşınması.

Manipülatör, ana engel yapısına ulaştıktan sonra Şekil 7.24'te gösterilen dar bölgelerden manevra yaparak kaçınmaktadır.



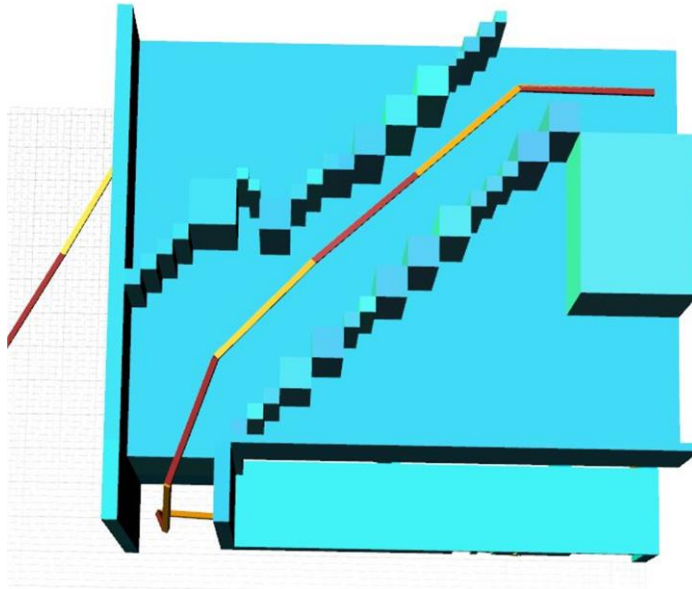
Şekil 7.24: Manipülatörün dar bölgelerden manevra yaparak kaçınması.

Daha sonra ana engel yapısının Şekil 7.25'te verilen bölümüne ulaşarak dar bölgeden geçmektedir.



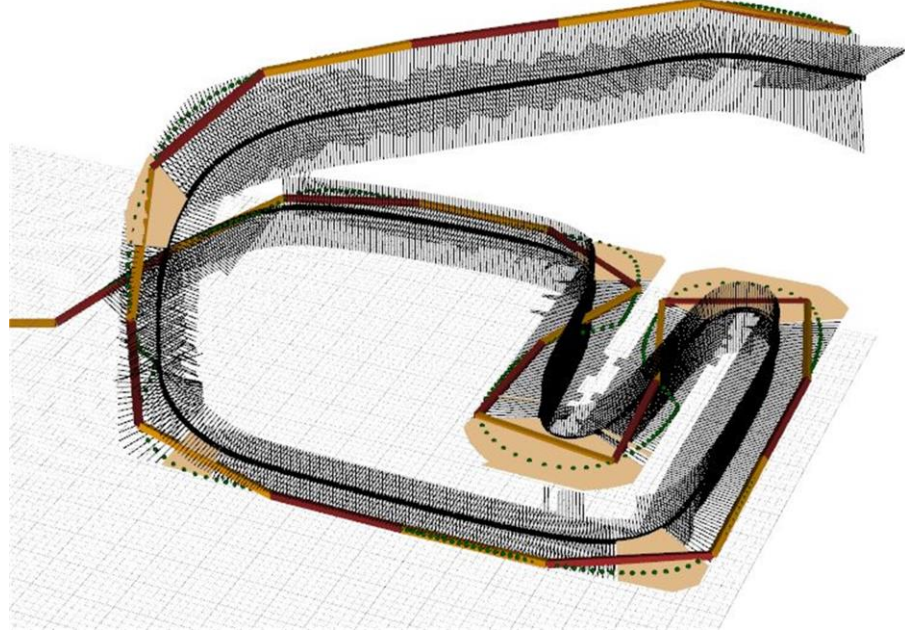
Şekil 7.25: Manipülâtörün dar alandan güvenle geçmesi.

Son olarak, manipülâtör yukarıya doğru hareket ederek, birçok küçük ya da büyük engellerin oluşturduğu bölgeden, engellerden kaçınarak hedefine başarılı bir şekilde ulaşmaktadır (Şekil 7.26).



Şekil 7.26: Manipülâtörün hedef noktasına ulaşması.

Engellerin gizlendiği hareket ortamında, manipülatör için belirlenen yol yardımcı elemanlar ile birlikte Şekil 7.27’de verilmiştir.



Şekil 7.27: Manipülatörün ikinci örnekteki alanda hareketi (Engeller gösterilmemiştir).

Bu örnekte potansiyel alan hesabı yapılırken, ilk örnekten farklı olarak 650 iterasyon kullanılmıştır. Bu örnek için de ard arda iki nokta arasındaki mesafe (d_p) 16 piksel olarak alınmıştır. Geliştirilen metot tarafından 7296 adet ışın doğrusu çizilmiştir. Potansiyel alan metodu ile global yolun hesaplanması 13822 ms sürmüştür. İlk örneğe göre daha yüksek iterasyon sayısının gerekliliğinin sebebi hareket ortamının büyüklüğü ve bu örneğin birçok küçük ve büyük engellerden oluşmasıdır. Global yol noktalarının elde edilmesinden sonra, bu noktalardan ışın doğrularının çizilmesi 15124 ms’de tamamlanmıştır. Manipülatör için gerekli olan yol üç boyutlu bir hareket ortamı için şaşırtıcı bir şekilde çok kısa bir sürede, 56 ms’de tamamlanmıştır. Son olarak, manipülatör hedef noktasına 29578 ms’de ulaşmıştır.

Daha önce de bahsedildiği gibi, ışın doğrusu sayısı hesaplama sürelerini etkilemektedir. Örneğin bu örnek için d_p değeri 8 piksel olarak alındığında, ışın doğrusu sayısı tam iki katına çıkarak 14592 adet olmaktadır. 650 iterasyon sonucunda potansiyel alan metodu yardımıyla global yolun hesaplanması 14722 ms sürmüştür. Daha sonra bu noktalardan ışın doğrularının çizilmesi 32763 ms’de tamamlanmıştır.

Manipülâtör için gerekli olan yol, üç boyutlu bir hareket ortamı için yine şaşırtıcı bir şekilde çok kısa bir süre olan 123 ms’de elde edilmiştir. Manipülâtör ise hedef noktasına 39264 ms’de ulaşmıştır.

Sonuç olarak, üç boyutlu hareket ortamlarında engellerin büyüklüğü, geometrik şekli ve sayısı hesaplama yüklerini ciddi bir şekilde arttırmamaktadır. Çünkü potansiyel alan metodunda sadece engellerin işgal ettiği ızgara noktalarının değerleri değişmektedir. Hareket ortamının büyüklüğü arttıkça sadece hesaplamadaki ızgara nokta sayısı artmaktadır ve değerlerin yakınsanabilmesi için yeterli bir iterasyon sayısına ihtiyaç duyulmaktadır. Işın doğrusu oluşturma sürelerinin ise ışın sayısı ile doğru orantılı olarak değiştiği gözlenmiştir. Fakat geliştirilen algoritma ile manipülâtörün uç noktasının takip etmesi gereken yolun belirlenmesi süreci milisaniyeler mertebesinde tamamlandığı için ışın doğrusu sayısı arttıkça hesaplama süresinde kayda değer bir artışın olmadığı sonucuna varılmıştır.

7.5 Manevra Kabiliyetinin Matematiksel Olarak Belirlenmesi

Manevra kabiliyetinin matematiksel olarak ifade edilmesinde Conkur ve diğ. (2005) çalışmalarında bahsettiği hareket ortamı karşılaştırmalı değerlendirme için kullanılmıştır (Şekil 7.28). Eşitlik (7.1)’de verilen l değeri manevra kabiliyeti hakkında bir ölçü olarak kabul edilmiştir. l_{uzuv} manipülâtörün uzvunun uzunluğunu ifade ederken, l_{max} ise geometrik olarak mümkün olan maksimum manipülâtör uzuv uzunluğunu ifade etmektedir. l değerinin alabileceği maksimum değer ise 1 olmaktadır.

$$l = \frac{l_{uzuv}}{l_{max}} \quad (7.1)$$

Şekil 7.28’de gösterilen l_1 ve l_{max} değerleri hem iki boyutlu hem de üç boyutlu simülasyonlarındaki hareket ortamlarında sırasıyla 200 ve 440 piksel olarak alınmıştır. Fakat l_{uzuv} değerleri iki boyutlu ve üç boyutlu hareket ortamlarında değişiklik göstermektedir. İki boyutlu hareket ortamı için l_{uzuv} değeri 430 piksel olarak alınmakta iken, üç boyutlu hareket ortamı için bu değer 350 piksel olarak alınmıştır. Bu durum

8. SONUÇ VE ÖNERİLER

Bu tez çalışmasında hem mobil robotun, hem de gereğinden/hiper gereğinden çok serbestlik dereceli manipülatörün, iki ve üç boyutlu hareket ortamlarındaki belirli bir başlangıç noktasından belirli bir hedef noktasına doğru engellerden kaçınarak ulaşabilmesini sağlamak için yol ve hareket planlama problemleri ele alınmıştır. Mobil robot için sabit ve hareketli engellerin bulunduğu hareket ortamları için hareket planlaması yapılırken, gereğinden/hiper gereğinden çok serbestlik dereceli manipülatörün hareket planlaması sadece sabit engellerden ve dar alanlardan oluşan hareket ortamları için yapılmıştır. Hareket planlama problemlerinin çözümü amacıyla her iki robot için ayrı iki adet hareket algoritması geliştirilmiştir.

Mobil robotun hareket planlamasını, sabit ve hareketli engellerle dolu alanlarda gerçekleştiren “Hayali Engellerle Güçlendirilmiş Güvenlik Çemberleri” (SC-IO) isimli bir algoritma geliştirilerek sunulmuştur. Bu algorithmada, herhangi bir yol planlama algoritması ile sadece sabit engellerin konumları dikkate alınarak tek bir hesaplama sonucunda elde edilen güvenli yol temel alınarak hareket planlaması yapılmıştır. Algoritma, Güvenlik Çemberi (SC) ve Hayali Engel (IO) olmak üzere iki aşamadan oluşmaktadır. SC aşamasında hareketli engellerin yol ile kesiştiği noktalar tespit edilerek bu noktalar merkezleri olacak şekilde güvenlik çemberleri eklenmiştir. Mobil robot ile hareketli engelin aynı anda güvenlik çemberinin içerisinde bulunmaması mobil robot ile hareketli engelin çarpışmasını engellemek için yeterli olmaktadır. Güvenlik çemberinde mobil robot ile hareketli engellerin arasında güvenli bir mesafenin olabilmesi için gerekli olan hareketli engellerin yaklaşma açısı ve kritik yaklaşma açısı hesaplanmıştır. Mevcut hareketli engellerin yaklaşma açıları ile değerlendirilerek hareketli engeller güvenli ve tehlikeli olarak belirlenmiştir. Güvenli olan hareketli engeller için mobil robotun tüm olası hız değerleri için, robot ve hareketli engelin konumları dikkate alınarak robotun her bir hareketi için güvenli hız değerleri hesaplanmıştır. Tehlikeli hareketli engeller için algoritmanın IO aşaması etkin hale getirilerek güvenlik çemberinin merkezine kare şeklinde sabit hayali bir engel eklenerek hem güvenli yol hem de tehlikeli hareketli engellerin yaklaşma açılarının değişmesi sağlanmıştır. Tehlikeli hareketli engeller güvenli olana kadar hayali engel ekleme işlemi devam ettirilmiştir. Engeller güvenli olduğunda tekrar

algoritmanın SC aşaması çalıştırılarak güvenlik çemberlerinin eklenmesi sağlanarak mobil robotun güvenli hız hesabı yapılmıştır.

Algoritma üç farklı hareket ortamında simüle edilerek sonuçları verilmiştir. Bu simülasyonlarda yol planlaması için potansiyel alan metodu kullanılmıştır. Sonuç olarak metodun mobil robotun hareket planlamasını güven içinde gerçekleştirdiği gözlemlenmiştir. Fakat tehlikeli engel sayısı arttıkça eklenmesi gereken hayali engelin sayısının artmasının hesaplama süresini arttırdığı tespit edilmiştir. Ancak hesaplamanın mobil robotun hareketine başlamadan önce tek bir seferde yapılmasından dolayı bu durum tolere edilebilecek bir durum olarak görülmüştür. Bu algorithmada hareketli engellerin boyutu büyüdükçe güvenlik çemberinin boyutunun büyütülmesi gerekmektedir. Bu durum mobil robotun çok düşük hızlarla hareket etmesine sebep olmaktadır. Daha sonra yapılacak olan çalışmalarda bu durumun incelenmesi planlanmaktadır.

Gereğinden/Hiper gereğinden çok serbestlik dereceli manipülatörün yol ve hareket planlaması için potansiyel alan metodu ile belirlenmiş olan yol noktalarından ışın doğruları çizilerek, dar alanların oluşturduğu kritik bölgeler ve bu bölgelerden kaçınabilmek için gerekli olan manevraların yönleri tespit edilmiştir. Bu bilgiler kullanılarak yol ve hareket planlamaları gerçekleştirilmiştir. İki boyutlu hareket ortamlarında sola ve sağa doğru çizilmiş olan bir adet ışın çifti manipülatör tarafından gerekli manevraların yapılabilmesini sağlamak için yeterli olmasına rağmen, üç boyutlu hareket ortamlarında daha fazla ışın çiftine ihtiyaç olduğu tespit edilmiştir. Bu sebeple iki adet ışın çifti kullanılarak yol ve hareket planlaması gerçekleştirilmiştir. Belirlenen yol üzerinde aynı ışın çifti üzerinde yön değişimi ve farklı ışın çiftleri arasında ışın çifti değişimi yapılması suretiyle güvenli yol noktaları belirlenmiştir. Geliştirilen bu yol ve hareket planlama algoritmaları birleştirilerek, manipülatörün mobil robot üzerinde taşındığı, iki ve üç boyutlu hareket ortamlarında simüle edilmiştir. Simülasyonlar sonucunda iki boyutlu hareket ortamlarında mobil robotun sabit ve hareketli engellere çarpmadan üzerinde taşıdığı manipülatörü belirlenen hedef noktasına güvenle ulaştırdığı ve manipülatörün de dar alanlardan gerekli manevrayı yaparak sabit engellerden kaçınarak hedefine ulaştığı deneysel olarak kanıtlanmıştır. Simülasyonlar sonucunda manipülatörün iki boyutlu hareket ortamlarında manevra yapılabilecek alanın %98'ini, üç boyutlu hareket ortamları için ise %80'ini kullandığı

belirlenmiştir. Bu değerler manipölatörün fiziksel olarak manevra yapabileceği müddetçe neredeyse hiç bir manevra alanını boşa harcamadığını göstermektedir. Bu tezde sunulan örnek hareket simülasyonları, geliştirilmiş olan her iki algoritmanın basit ve sağlam bir yapıya sahip olduklarını göstermektedir. Özellikle hiper gereğinden çok serbestlik dereceli manipölatör için üç boyutlu hareket ortamlarında geliştirilen metot literatürde bulunan metotlar ile kıyaslandığında, metodun manevra kabiliyetinin benzer metotlara göre oldukça üstün olduğu bariz bir şekilde ortaya çıkmaktadır.

9. KAYNAKLAR

Aine, S., Swaminathan S., Narayanan V., Hwang V. and Likhachev M., “Multi-Heuristic A*”, *Int. J. Rob. Res.*, 35, 224-243, (2015).

Alonso-Mora, J., Beardsley, P. and Siegwart, R., “Cooperative collision avoidance for nonholonomic robots”, *IEEE Trans. Rob.*, 34 (2), 404-420, (2018).

Ananthanarayanan, H. and Ordonez, R., “A fast converging optimal technique applied to path planning of hyper-redundant manipulators”, *Mech. Mach. Theory*, 118, 231-246, (2017).

Ayten, K. K., Sahinkaya, M. N. and Dumlu, A., “Real time optimum trajectory generation for redundant/hyper-redundant serial industrial manipulators”, *Int. J. Robot. Syst.*, 14 (6), 1-14, (2017).

Azariadis, P. N. and Aspragathos, N. A., “Obstacle representation by Bumpsurfaces for optimal motion-planning”, *Rob. Auton. Syst.*, 51, 129–150, (2005).

Bulut, Y. and Conkur, E. S., “A real-time path-planning algorithm with extremely tight maneuvering capabilities for hyper-redundant manipulators”, *Eng. Sci. Technol. An Int. J.*, 24 (1), 247-258, (2021).

Burhanuddin, L. A., Islam, M. N. and Yusof, S. M., “Evaluation of collision avoidance path planning algorithm”, *Int. Conf. Res. Innov. Inf. Syst.*, 360-365, (2013).

Canali, C., Pistone, A., Ludovico, D., Guardiani, P., Gagliardi, R., Verme, L., et al., “Design of a novel long-reach cable-driven hyper-redundant snake-like manipulator for inspection and maintenance”, *Appl. Sci.*, 12 (7) , 3348, (2022).

Castillo, O., Trujillo, L. and Melin, P., “Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots”, *Soft Comput.*, 11, 269-279, (2007).

Chen, J., Tan, C., Mo, R., Zhang, H., Cai, G. and Li, H., “Research on path planning of three-neighbor search A* algorithm combined with artificial potential field”, *Int. J. Adv. Rob. Syst.*, 18 (3), 1-13, (2021).

- Chiacchio, P., Chiaverini, S., Sciavicco, L. and Siciliano, B., “Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy”, *Int. J. Rob. Res.*, 10 (4), 410-425, (1991).
- Chirikjian, G. S. and Burdick, J. W., “An obstacle avoidance algorithm for hyper-redundant manipulators”, *IEEE Int. Conf. Robot. Autom.*, 625–631, (1990).
- Choset, H. and Henning, W., “A follow-the-leader approach to serpentine robot motion planning”, *J. Aerosp. Eng.*, 12 (2), 65–73, (1999).
- Chu, X., Hu, Q. and Zhang, J., “Path planning and collision avoidance for a multi-arm space maneuverable robot”, *IEEE Trans. Aerosp. Elect. Syst.*, 54 (1), 217-232, (2018).
- Collins, T. and Shen, W.-M., “PASO: An integrated, scalable PSO-based optimization framework for hyper-redundant manipulator path planning and inverse kinematics”, *Information Sciences Institute Technical Report*, (2016).
- Conkur, E. S., “Path following algorithm for highly redundant manipulators”, *Rob. Auton. Syst.*, 45, 1–22, (2003).
- Conkur, E. S., “Path planning using potential fields for highly redundant manipulators”, *Rob. Auton. Syst.*, 52, 209–228, (2005).
- Conkur, E. S. and Buckingham R., “Clarifying the definition of redundancy as used in robotics”, *Robotica*, 15, 583–586, (1997^a).
- Conkur, E. S. and Buckingham, R., “Manoeuvring highly redundant manipulators”, *Robotica*, 15, 435–447, (1997^b).
- Conkur, E. S., Buckingham, R. and Harrison, A., “The beam analysis algorithm for path planning for redundant manipulators”, *Mechatronics*, 15, 67–94, (2005).
- Dave, F. and Anthony, S., “Using interpolation to improve path planning: The Field D* algorithm”, *J. F. Rob.*, 23 (2), 79-101, (2006).
- De Maeyer, J., Versteyhe, M. and Demeester, E., “Sampling-based tube following for redundant, planar robotic manipulators”, *IEEE Int. Conf. Emerg. Technol. Fact. Autom.*, 752–758, (2018).

Du, Z.-C., Ouyang, G.-Y., Xue, J. and Yao, Y.-B., “A review on kinematic, workspace, trajectory planning and path planning of hyper-redundant manipulators”, *10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 444-449, (2020).

Duan, J., Wang, B., Cui, K. and Dai, Z., “Path planning based on NURBS for hyper-redundant manipulator used in narrow space”, *Appl. Sci.*, 12 (3), 1314, (2022).

Fahimi, F., Ashrafiuon H. and Nataraj, C., “Obstacle avoidance for spatial hyper-redundant manipulators”, *The First Asian Conference on Multibody Dynamics (ACMD)*, 247-254, (2002).

Fiorini, P. and Shiller, Z., “Robot motion planning in dynamic environments using velocity obstacles”, *Int. J. Rob. Res.*, 17 (7), 760-772, (1998).

Fraichard, T. and Asama, H., “Inevitable collision states - a step towards safer robots”, *Adv. Rob.*, 18 (10), 1001-1024, (2004).

Ge, S. S. and Cui, Y. J., “Dynamic motion planning for mobile robots using potential field method”, *Auton. Rob.*, 13, 207-222, (2002).

Graham, A. and Buckingham, R., “Real-time collision avoidance of manipulators with multiple redundancy”, *Mechatronics*, 3 (1), 89–106, (1993).

Gu, M. and Huang, Y., “Dynamic obstacle avoidance of mobile robot based on adaptive velocity obstacle”, *36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 776-781, (2021).

Gyenes, Z. and Szadeczky-Kardoss, E. G., “Motion planning for mobile robots using the safety velocity obstacles method”, *Proceedings of the International Carpathian Control Conference*, 389-394, (2018).

Gyenes, Z. and Szadeczky-Kardoss, E. G., “Motion planning for mobile robots using the collidable velocity obstacles method”, *30th Mediterranean Conference on Control and Automation (MED)*, 737-742, (2022).

Hart, P. E., Nilsson, N. and Raphael, B., “A formal basis for the heuristic determination of minimum cost paths”, *IEEE Trans. Syst. Sci. and Cybern.*, 4, 100-107, (1968).

Helix Toolkit, (25 September 2022), <https://github.com/helix-toolkit>.

Hou W., Xiong, Z., Wang, C. and Chen, H., “Enhanced ant colony algorithm with communication mechanism for mobile robot path planning”, *Rob. Auton. Syst.*, 148, 103949, (2022).

Islam, M. N., Tamura, S., Murata, T. and Yanase, T., “Evaluation of a new backtrack free path planning algorithm for manipulators”, *IEEJ Trans. Electron. Inf. Syst.*, 128, 1293–1302, (2008).

Jamali, A., Khan, M. R., Osman, M. S., Rahman, M. M., Ashari, M. F., Jamaludin, M. S., et al., “Collision free control of variable length hyper redundant robot manipulator”, *Appl. Mech. Mater.*, 541–542, 1107–1114, (2014).

Jia, L., Huang, Y., Chen, T., Guo, Y., Yin, Y. and Chen, J., “MDA+RRT: A general approach for resolving the problem of angle constraint for hyper redundant manipulator”, *Exp. Syst. Appl.*, 193, 116379, (2022).

Kang, N. K., Son, H. J. and Lee, S. H., “Modified A-star algorithm for modular plant land transportation”, *J. Mech. Sci. Technol.*, 32, 5563-5571, (2018).

Khatib, O., “Real-time obstacle avoidance for manipulators and mobile robots”, *Int. J. Rob. Res.*, 5 (1), 90–98, (1986).

Ko, N. Y. and Lee, B. H., “Avoidability measure in moving obstacle avoidance problem and its use for robot motion planning”, *Proceedings of the International Conference on Intelligent Robots and Systems*, 3, IEEE, 1296-1303, (1996).

Koenig, S. and Likhachev, M., “D* Lite”, *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 476-483, (2002).

Kufoalor, D. K. M., Brekke, E. F. and Johansen, T. A., “Proactive collision avoidance for ASVs using a Dynamic Reciprocal Velocity Obstacles Method”, *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE, 2402-2409, (2018).

Lamini, C., Benhlima, S. and Elbekri, A., “Genetic algorithm based approach for autonomous mobile robot path planning”, *Procedia Comput. Sci.*, 127, 180-189, (2018).

Le, A. V., Prabakaran V., Sivanantham, V. and Mohan, R. E., “Modified a star algorithm for efficient coverage path planning in tetris inspired self reconfigurable robot with integrated laser sensor”, *Sensors*, 18 (8), 2585, (2018).

- Leena, N. and Saju, K. K., “A survey on path planning techniques for autonomous mobile robots”, *IOSR J. Mech. and Civ. Eng. (IOSR-JMCE)*, 8, 76-79, (2014).
- Li, W., Yang, C., Jiang, Y., Liu, X and Su, C. Y., “Motion planning for omnidirectional wheeled mobile robot by potential field method”, *J. Adv. Transp.*, 2017, 1-11, (2017).
- Li, J., Xu, B., Yang, Y. and Wu, H., “Quantum ant colony optimization algorithm for AGVs path planning based on Bloch coordinates of pheromones”, *Nat. Comput.*, 19, 673-682, (2020).
- Liang, T.-C. and Liu, J.-S., “An improved trajectory planner for redundant manipulators in constrained workspace”, *J. Robot. Syst.*, 16 (6), 339–351, (1999).
- Likhachev, M., Gordon, G. J. and Thrun, S., “ARA*: Anytime A* with provable bounds on sub-optimality”, *Adv. in Neural Inf. Process. Syst.*, 16, 767–774, (2004).
- Likhachev, M., Ferguson, D., Gordon, G., Anthony, S. and Sebastian, T., “Dynamic A*: An anytime, replanning algorithm”, *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS)*, 262-271, (2005).
- Liljebäck, P., Pettersen, K. Y., Stavdahl, Ø and Gravdahl, J. T., “A review on modelling, implementation, and control of snake robots”, *Rob. Auton. Syst.*, 60 (1), 29–40, (2012).
- Lin, C.-C. and Chuang, J.-H., “A potential-based path planning algorithm for hyper-redundant manipulators”, *J. Chin. Inst. Eng.*, 33 (3), 415-427, (2010).
- Lin, Y., Wang, J., Xiao, X., Qu, J. and Qin, F., “A snake-inspired path planning algorithm based on reinforcement learning and self-motion for hyper-redundant manipulators”, *Int. J. Robot. Syst.*, 19 (4), 1-13, (2022).
- Liu, J., Yang, J., Liu, H., Tian, X. and Gao, M., “An improved ant colony algorithm for robot path planning”, *Soft Comput.*, 21, 5829-5839, (2017).
- Liu, Z., Jiang, Z., Xu, T., Cheng, H., Xie, Z. and Lin, L., “Avoidance of high-speed obstacles based on velocity obstacles”, *Proceedings of the International Conference on Robotics and Automation, IEEE*, 7624-7630, (2018).

- Ma, S., Hirose, S. and Yoshinada, H., “Development of a hyper-redundant multijoint manipulator for maintenance of nuclear reactors”, *Adv. Rob.*, 9 (3), 281–300, (1994).
- Ma, S. and Konno, M., “Obstacle avoidance scheme for hyper-redundant manipulators - global motion planning in posture space”, *IEEE Int. Conf. Robot. Autom.*, 161–166, (1997).
- Mac, T. T., Copot, C., Tran, D. T. and De Keyser, R., “Heuristic approaches in robot path planning: A survey”, *Rob. Auton. Syst.*, 86, 13-28, (2016).
- Mac, T. T., Copot, C., Tran, D. T. and De Keyser, R., “A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization”, *Appl. Soft Comput.*, 59, 68-76, (2017).
- Machmudah, A., Parman, S., Abbasi, A., Solihin, M. I., Manan, T. S. A., Beddu, S., et al., “Cyclic path planning of hyper-redundant manipulator using whale optimization algorithm”, *Int. J. Adv. Comput. Sci. Appl.*, 12 (8), 677-686, (2021).
- McLean, A. and Cameron, S., “Snake-based path planning for redundant manipulators”, *IEEE Int. Conf. Robot. Autom.*, 275–282, (1993).
- McLean, A. and Cameron, S., “The virtual springs method: Path planning and collision avoidance for redundant manipulators”, *Int. J. Rob. Res.*, 15, 300-319, (1997).
- Mester, G., “Intelligent mobile robot motion control in unstructured environments”, *Acta Polytech. Hung.*, 7 (4), 153-165, (2010).
- Miao, Y., Gao, F. and Zhang, Y., “Gait fitting for snake robots with binary actuators”, *Sci. China Technol. Sci.*, 57 (1), 181–191, (2014).
- Minnetoğlu, O., “Engelli bir alan içinde hedefine otomatik olarak ulaşabilen bir mobil robotun kabiliyetinin araştırılması”, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Makina Mühendisliği Anabilim Dalı*, Denizli, (2013).
- Mohanan, M. G. and Salgoankar, A., “A survey of robotic motion planning in dynamic environments”, *Rob. Auton. Syst.*, 100, 171-185, (2018).
- Montiel, O., Orozco-Rosas, U. and Sepúlveda, R., “Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles”, *Expert Syst. Appl.*, 42, 5177-5191, (2015).

Motahari, A., Zohoor, H. and Korayem, M. H., “A new obstacle avoidance method for discretely actuated hyper-redundant manipulators”, *Sci. Iranica B*, 19 (4), 1081-1091, (2012).

Nakamura, Y., *Advanced robotics: redundancy and optimization*, Addison Wesley Publishing Company, (1991).

Ngwenya, T., Ayomoh, M. and Yadavalli S., “Virtual obstacles for sensors incapacitation in robot navigation: A systematic review of 2D path planning”, *Sensors*, 22 (18), 6943, (2022).

Oral, T. and Polat, F., “MOD* Lite: An incremental path planning algorithm taking care of multiple objectives”, *IEEE Trans. Cybern.*, 46, 245-257, (2016).

Orozco-Rosas, U., Montiel, O. and Sepúlveda, R., “Mobile robot path planning using membrane evolutionary artificial potential field”, *Appl. Soft Comput.*, 77, 236-251, (2019).

Owen, E. and Montano, L., “Motion planning in dynamic environments using the velocity space”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2833-2838, (2005).

Poty, A., Melchior, P. and Oustaloup, A., “Dynamic path planning by fractional potential”, *Second IEEE International Conference on Computational Cybernetics*, IEEE, 365-371, (2004).

Przybylski, M. and Putz, B., “D* Extra Lite: A Dynamic A* with search-tree cutting and frontier-gap repairing”, *Int. J. Appl. Math. Comput. Sci.*, 27 (2), 273–290, (2017).

Przybylski, M., “AD*-Cut: A search-tree cutting anytime dynamic A* algorithm”, *Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling*, AAAI Press, 494-499, (2018).

Qian, S., Zi, B., Shang, W.-W. and Xu, Q.-S., “A review on cable-driven parallel robots”, *Chin. J. Mech. Eng.*, 31, 66, (2018).

Qin, G., Wu, H., Cheng, Y., Pan, H., Zhao, W., Shi, S., et al., “Adaptive trajectory control of an under-actuated snake robot”, *App. Math. Mod.*, 106, 756-769, (2022).

Qixin, C., Yanwen, H. and Jingliang, Z., “An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot”, *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE, 3331-3336, (2006).

Raja, R. and Dutta, A., “Path planning in dynamic environment for a Rover using A* and potential field method”, *Proceedings of the International Conference on Advanced Robotics*, IEEE, 578-582, (2017).

Reznik, D. and Lumelsky, V., “Sensor-based motion planning in three dimensions for a highly redundant snake robot”, *Adv. Robot.*, 9 (3), 255–280, (1994).

Sanchez-Ibanez, J. R., Perez-del-Pulgar, C. J. and Garcia-Cerezo, A., “Path planning for autonomous mobile robots: A review”, *Sensors*, 21 (23), 7898, (2021).

Schilling, R. J., Read, R., Lovass-nagy, V. and Walker, G., “Path tracking with the links of a planar hyper-redundant robotic manipulator”, *J. Robot. Syst.*, 12 (3), 189–197, (1995).

Seder, M. and Petrovic, I., “Dynamic window based approach to mobile robot motion control in the presence of moving obstacles”, *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, 1986-1991, (2007).

Seereeram, S. and Wen, J. T., “A global approach to path planning for redundant manipulators”, *IEEE Trans. Robot. Autom.*, 11, 152–160, (1995).

Sharma, D. and Dubey, S. K., “Anytime A* algorithm - An extension to A* algorithm”, *Int. J. Sci. & Eng. Res.*, 4, 1-4, (2013).

Snape, J., Van Den Berg, J., Guy, S. J. and Manocha, D., “The hybrid reciprocal velocity obstacle”, *IEEE Trans. Rob.*, 27, 696-706, (2011).

Song, R., Liu, Y. and Bucknall, R., “Smoothed A* algorithm for practical unmanned surface vehicle path planning”, *Appl. Ocean Res.*, 83, 9-20, (2019^a).

Song, B., Wang, Z., Zou, L., Xu, L. and Alsaadi, F. E., “A new approach to smooth global path planning of mobile robots with kinematic constraints”, *Int. J. Mach. Learning Cybern.*, 10 (1), 107-119, (2019^b).

Stentz, A., “Optimal and efficient path planning for partially known environments”, *Int. J. Rob. Autom.*, 10, 89-100, (1993).

Stentz, A., “The Focussed D* algorithm for real-time replanning”, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, British Library Conference Proceedings, 1652-1659, (1995).

Stern, R., Felner, A., Van Den Berg, J., Puzis, R., Shah, R. and Goldberg, K., “Potential-based bounded-cost search and anytime non-parametric A*”, *Artif. Intell.*, 214, 1-25, (2014).

Sun X.,William, Y. and Koenig, S., “Moving Target D* Lite”, *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, 1, 67-74, (2010).

Takahashi, O. and Schilling, R. J., “Motion planning in a plane using generalized Voronoi diagrams”, *IEEE Trans. Robot. Autom.*, 5 (2), 143–150, (1989).

Tang, L., Wang, J., Zheng, Y., Gu, G., Zhu, L. and Zhu, X., “Design of a cable-driven hyper-redundant robot with experimental validation”, *Int. J. Adv. Robot. Syst.*, 14, 1–12, (2017).

Tang, L., Zhu, L. M., Zhu, X. and Gu, G., “A serpentine curve based motion planning method for cable-driven snake robots”, *25th Int. Conf. Mechatronics Mach. Vis. Pract.*, 5–10, (2018).

Tang, L., Huang, J., Zhu, L.-M., Zhu, X. and Gu, G., “Path tracking of a cable-driven snake robot with two-level motion planning method”, *IEEE/ASME Trans. Mechatronics*, 24 (3), 935-946, (2019).

Tappe, S., Pohlmann, J., Kotlarski, J. and Ortmaier, T., “Towards a follow the-leader control for a binary actuated hyper-redundant manipulator”, *IEEE Int. Conf. Intell. Robot. Syst.*, 3195–3201, (2015).

Tho, T. P. and Think, N. T., “An overview of cable-driven parallel robots: Workspace, tension distribution, and cable sagging”, *Math. Prob. in Eng.*, 2022, 2199748, (2022).

Tuncer, A. and Yildirim, M., “Dynamic path planning of mobile robots with improved genetic algorithm”, *Comput. & Electr. Eng.*, 38 (6), 1564-1572, (2012).

Van Den Berg, J., Lin, M. and Manocha D., “Reciprocal velocity obstacles for real-time multi-agent navigation”, *Proceedings of the International Conference on Robotics and Automation*, IEEE, 1928-1935, (2008).

Van Den Berg, J., Guy, S. J., Snape, J., Lin, M. and Manocha, D., “RVO2 Library: Reciprocal collision avoidance for real-time multi-agent simulation [online]”, (25 September 2022), <https://gamma.cs.unc.edu/RVO2/downloads/>, (2010).

Van Den Berg, J., Guy, S. J., Lin, M. and Manocha, D., “Reciprocal n-body collision avoidance”, *Springer Tracts in Adv. Rob.*, 70, 3-19, (2011).

Wang, W., Zhu, M., Wang, X., He, S., He, J. and Xu, Z., “An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators”, *Int. J. Robot. Syst.*, 15 (5), 1-13, (2018).

Wang, Y., Xiaoxiao, L., Zhang, J., Li, S., Xu, Z. and Zhou, X., “Review of wheeled mobile robot collision avoidance under unknown environment”, *Sci. Prog.*, 104 (3), 1-26, (2021).

Wang, H., Lou, S., Jing, J., Wang, Y., Liu, W. and Liu, T., “The EBS-A* algorithm: An improved A* algorithm for path planning”, *Plos One*, 17 (2), e0263841, (2022).

Wei, H., Zheng, Y. and Gu, G., “RRT-based path planning for follow-the-leader motion of hyper-redundant manipulators”, *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 3198-3204, (2021).

Willms, A. R. and Yang, S. X., “An efficient dynamic system for realtime robot-path planning”, *IEEE Trans. Syst., Man, Cybern. Part B*, 36 (4), 755-766, (2006).

Willms, A. R. and Yang, S. X., “Real-time robot path planning via a distance propagating dynamic system with obstacle clearance”, *IEEE Trans. Syst., Man, Cybern. Part B*, 38 (3), 884-893, (2008).

Xidias, E. K. and Aspragathos, N. A., “Time sub-optimal path planning for hyper-redundant manipulators amidst narrow passages in 3D workspaces”, (eds: M. Ceccarelli and V. A. Glazunov), *Advances on Theory and Practice of Robots and Manipulators*, Cham, Springer, 445-452, (2014).

Xidias, E. K., “Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces”, *Robot. Comput.-Integr. Manuf.*, 50, 286-298, (2018).

- Xie, H., Wang, C., Li, S., Hu, L. and Yang, H., “A geometric approach for follow-the-leader motion of serpentine manipulator”, *Int. J. Adv. Robot. Syst.*, 16, 1–18, (2019).
- Xu, W., Mu, Z., Liu, T. and Liang, B., “A modified modal method for solving the mission-oriented inverse kinematics of hyper-redundant space manipulators for on-orbit servicing”, *Acta Astronaut.*, 139, 54-66, (2017).
- Xu, T., Zhang, S., Jiang, Z., Liu, Z. and Cheng, H., “Collision avoidance of high-speed obstacles for mobile robots via maximum-speed aware velocity obstacle method”, *IEEE Access*, 8, 138493-138507, (2020).
- Xue, F. and Fan, Z., “Kinematics and control of a cable-driven snake-like manipulator for underwater application”, *Mech. Sci.*, 13, 495-504, (2022).
- Yang, S. X. and Meng, M., “An efficient neural network approach to dynamic robot motion planning”, *Neural Netw.*, 13 (2), 143-148, (2000).
- Yang, H., Qi, J., Miao, Y., Sun, H. and Li, J., “A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization”, *IEEE Trans. on Ind. Electr.*, 66 (11), 8557-8566, (2019).
- Yuan, J., Wang, H., Lin, C., Liu, D. and Yu, D., “A novel GRU-RNN network model for dynamic path planning of mobile robot”, *IEEE Access*, 7, 15140-15151, (2019).
- Zafar, K., Baig, A. R., Bukhari, N. and Khan, A., “Optimization of route planning using simulated ant agent system”, *Int. J. Comput. Appl.*, 4 (8), 1-4, (2010).
- Zeng, M. R., Xi, L. and Xiao, A. M., “The free step length ant colony algorithm in mobile robot path planning”, *Adv. Rob.*, 30 (23), 1-6, (2016).
- Zhang, D., Chen, Y., Huang, C. and Gao, M., “Improved path planning algorithm on the rugged road”, *Automatika*, 57, 477-483, (2016).
- Zhang, C., “Path planning for robot based on chaotic artificial potential field method”, *IOP Conference Series: Materials Science and Engineering*, 317, 2-6, (2018^a).
- Zhang, H., Lin, W. and Chen, A., “Path planning for the mobile robot: A review”, *Symmetry*, 10 (10), 1-17, (2018^b).

Zhang, B., Li, G., Zheng, Q., Bai, X., Ding, Y. and Khan, A., “Path planning for wheeled mobile robot in partially known uneven terrain”, *Sensors*, 22 (14), 5217, (2022).

Zheng, Y., Wu, B., Chen, Y., Zeng, L., Gu, G. and Zhu, X., “Design and validation of cable-driven hyper-redundant manipulator with a closed-loop puller-follower controller”, *Mechatronics*, 78, 102605, (2021).

Zhu, Q., Han, Y., Liu, P., Xiao, Y., Lu, P. and Cai, C., “Motion planning of autonomous mobile robot using Recurrent Fuzzy Neural Network trained by extended Kalman filter”, *Comput. Intell. Neurosci.*, 2019, 1-16, (2019).

Zhu, Y., Ma, R. and Yihong, Q., “Improvement of ant colony method track planning based on artificial potential field method”, *16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1394-1398, (2020).