

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

ÇOKLU MOBİL ROBOTLARIN KOORDİNASYONU

ONUR AKBATI

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

**DANIŞMAN
PROF. DR. GALİP CANSEVER**

İSTANBUL, 2013

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOKLU MOBİL ROBOTLARIN KOORDİNASYONU

Onur AKBATI tarafından hazırlanan tez çalışması 19/12/2013 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Galip CANSEVER
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. Galip CANSEVER
Yıldız Teknik Üniversitesi

Prof. Dr. Haluk Görgün
Yıldız Teknik Üniversitesi

Doç. Dr. Mehmet Akar
Boğaziçi Üniversitesi

Yrd. Doç. Dr. Akın Delibaşı
Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. Ahu Ece Hartavi Karcı
Yeni Yüzyıl Üniversitesi

Bu alıřma, Yıldız Teknik Üniversitesi Bilimsel Arařtırma Projeleri Koordinatörlüğü' nün 2012-04-04-KAP02 numaralı projesi ile desteklenmiřtir.

ÖNSÖZ

Bu tezin hazırlanması sırasında bana her konuda yardımcı olan, yardım ve katkıları ile beni yönlendiren, değerli hocam ve danışmanım Prof. Dr. Galip CANSEVER'e ve çalışmanın uygulama aşaması gerekli malzemelerin temini için projemizi destekleyen Yıldız Teknik Üniversitesi Bilimsel Araştırmalar Koordinatörlüğü'ne teşekkürlerimi sunarım. Ayrıca "Bilim İnsanı Destekleme Programı" kapsamında, doktora öğrenimim süresince bana maddi destek sağlayan TÜBİTAK'a teşekkür ederim.

Bu tez çalışması süresince benden desteğini esirgemeyen aileme ve Arş. Gör. Şirin Akkaya'ya teşekkürü borç bilirim.

Aralık, 2013

Onur AKBATI

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	vii
KISALTMA LİSTESİ.....	viii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ	xi
ÖZET	xii
ABSTRACT.....	xiii
BÖLÜM 1	
GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	4
1.3 Hipotez	4
BÖLÜM 2	
HIZ ENGELLERİ	6
2.1 Basit Hız Engelleri.....	6
2.2 Karşılıklı Hız Engelleri	9
2.3 Karşılıklı Çok Sayıda Aracın Çarpışmasının Engellenmesi	11
2.4 İvme Hız Engelleri.....	15
2.5 Hibrit Hız Engelleri	16
BÖLÜM 3	
ARAÇ MODELİ VE PARK PROBLEMİ	18
3.1 Araç Modeli.....	18
3.2 Park Problemi.....	20

BÖLÜM 4	
GELİŞTİRİLEN YÖNTEM	23
BÖLÜM 5	
UYGULAMA	33
5.1 Giriş	33
5.2 Görüntü Geribeslemesine Dayalı Araç Konum Tespiti	34
5.3 Park Problemi Uygulaması	35
5.4 Çarpışmadan Kaçınma Uygulaması.....	37
BÖLÜM 6	
SONUÇLAR VE ÖNERİLER.....	42
KAYNAKLAR	44
EK-A	
A-1 PROGRAM KODLARI	47
ÖZGEÇMİŞ	48

SİMGE LİSTESİ

$v_{A,B}$	A aracının B aracına göre bağıl hızı
$\lambda_{A,B}$	Bağıl hız vektörü doğrultusunda çizilen ışın
VO	Hız engeli kümesi
$CC_{A,B}$	A ve B robotları arasındaki çarpışma koniği
$S_{(.)}$	Seçilebilir hız kümeleri
$\alpha_{(.)}^{(.)}$	Risk faktörü
$v_{(.)}$	Hız vektörü
$u_{(.)}$	Kontrol girişleri
$x_{(.)}$	x eksenindeki koordinat bilgisi
$y_{(.)}$	y eksenindeki koordinat bilgisi
θ	Aracın yönelim açısı
e_d	Konum hatası
e_θ	Açı hatası
r_{\min}	Sınırlama değeri
v_{pref}	Tercih edilen hız vektörü
r	Robotların yarıçapı
ϵ_H	Yörünge takip hatası
$\hat{p}^k(.)$	Bir sonraki adımda beklenen robot pozisyonu
δ	Zaman boyutunda tanımlanmış kontrol parametresi
v_{\max}	Maksimum robot hızı
α	Konik açısı
β	VO, RVO ve HRVO için orta çizgi açısı
p	İki robotun merkezi arasındaki uzaklık

KISALTMA LİSTESİ

AVO	Acceleration Velocity Obstacle
CL	Center Line
CVO	Common Velocity Obstacle
HRVO	Hybrid Reciprocal Velocity Obstacle
iGPS	Intelligent Global Positioning System
ORCA	Optimal Reciprocal Collision Avoidance
RVO	Reciprocal Velocity Obstacle
TS	Tarama Sayısı
VO	Velocity Obstacle

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1 A aracı ve B nesnesi	6
Şekil 2.2 Çarpışma koniği.....	7
Şekil 2.3 Hız engeli.....	8
Şekil 2.4 Erişilebilir Hız Kümesi.....	8
Şekil 2.5 Kaçınma Kümeleri	9
Şekil 2.6 Karşılıklı hız engeli.....	10
Şekil 2.7 Farklı parametreler için hız engeli	11
Şekil 2.8 Kısıtlı zaman aralığında hız engeli.....	12
Şekil 2.9 ORCA yarı düzlemleri	13
Şekil 2.10 Birden çok araç için ORCA yarı düzlemleri	13
Şekil 2.11 Hareket kabiliyeti kısıtlı araç yapısı	14
Şekil 2.12 ORCA doğruları ve P_{AHV} poligonu.....	15
Şekil 2.13 İvme-hız engeli ve seçilebilir hız kümeleri.....	16
Şekil 2.14 Karşılıklı hız engellerinde ters tarafı seçme durumu	16
Şekil 2.15 Hibrit hız engellerinin geometrik yapısı.....	17
Şekil 3.1 Araç Modeli.....	18
Şekil 3.2 Araç koordinatlarının tanımlanması	19
Şekil 3.3 Robot park problemi.....	20
Şekil 4.1 Hız vektörünün tek hız engeli içinde kalma durumu	23
Şekil 4.2 Hız vektörünün çok sayıda hız engeli içinde kalma durumu.....	24
Şekil 4.3 Çok sayıda hız engelini olduğu durum.....	24
Şekil 4.4 Izgaralama yönteminin geometrik gösterimi.....	26
Şekil 4.5 Hız engelini eşitsizliklerle ifade edilmesi	26
Şekil 4.6 HRVO kümesinin hesaplanması	27
Şekil 4.7 Konik açısının gösterimi	28
Şekil 4.8 Benzetim blok diyagramı	30
Şekil 4.9 Robot blok diyagramı.....	31
Şekil 4.10 HRVO hesaplayıcı blok diyagramı	31
Şekil 4.11 Navigasyon haritası.....	32
Şekil 5.1 Qbot mobil robotu	33
Şekil 5.2 Uygulama sisteminin yapısı	36
Şekil 5.3 Açık hatasının zamana göre değişimi	37
Şekil 5.4 Konum hatasının zamana göre değişimi.....	37
Şekil 5.5 Park için oluşturulan çemberler.....	38

Şekil 5.6	Algoritma akış diyagramı	39
Şekil 5.7	Tek robota ait blok diyagramı.....	40
Şekil 5.8	Uygulamada kullanılan araçlar	41
Şekil 5.9	Uygulama süresince araçların izlediği yol haritaları	41

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 4.1 Benzetime ait parametreler	32
Çizelge 5.1 Hedef takip uygulaması parametreleri	36
Çizelge 5.2 Araçlara verilen hedef nokta bilgisi	40

ÇOKLU MOBİL ROBOTLARIN KOORDİNASYONU

Onur AKBATI

Elektronik Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Prof. Dr. Galip CANSEVER

Bu tez çalışmasında çoklu mobil robot (araç) sistemlerinde araçların birbirleri arasında haberleşmeksizin çarpışmadan kaçınarak verilen görevleri yerine getirmesi için uygulamaya yönelik bir yöntem geliştirilmesi hedeflenmiştir. Buradaki asıl problem araçların çarpışmalardan kaçınmasıdır. Literatürde bu probleme bir çözüm olarak sunulan hız engelleri ile navigasyon kavramı kullanılarak başarılı sonuçlar elde edilmiştir. Buradaki amaç, bir aracın hareketli bir nesneye göre olan bağıl hızının kullanılması sonucunda aracın hareketli nesne ile çarpışmasına sebep olacak hız vektörleri kümesinin belirlenmesi ve aracın yeni hız vektörünü bu küme dışından seçerek çarpışmadan kaçınmasıdır. Yakın tarihlerde yapılan çalışmalarda “Hız Engeli” yöntemleri geliştirilmiş ve araçların aynı algoritmaya göre davranış sergileyeceği düşünülerek yöntem ve uygulamalar gerçekleştirilmiştir; “Karşılıklı Hız Engeli”, “Hız-İvme Engeli”, “Hibrit Hız Engeli” bunların bir kısmıdır. Günümüze kadar yapılan çalışmalar ele alındığında çoğu çalışmada araç dinamiklerinin, park probleminin ve işlem yükünün tam olarak ele alınmadığı görülmektedir. Bunun bir sebebi de algoritmanın yapısının basit tutularak işlem yükünün azaltılmasıdır. Bu tez çalışmasında işlem yükünü azaltan yeni bir yöntem oluşturularak, robotlar üzerinde gerçekleştirilmiştir.

Anahtar Kelimeler: Mobil robotlar, hız engeli, çarpışma önleme, QBot

COORDINATION OF MULTIPLE MOBILE ROBOTS

Onur AKBATI

Department of Electrical Engineering

PhD. Thesis

Adviser: Prof. Dr. Galip CANSEVER

In this thesis, development of a method regarding the multiple mobile robots (vehicles) systems performing predefined tasks without communication between each other while avoiding collisions is aimed. Here the main problem is collision avoidance. In literature, the concept of navigation with velocity obstacles given as a solution has achieved successful results. The aim of this concept is to determine the colliding velocity sets with using relative velocity of a vehicle according to a moving object and to prevent collision by choosing a new velocity vector outside of this set. In the recent studies “velocity obstacle” based methods has been improved and solutions for vehicles acting in the same manner are put into practice. “Reciprocal Velocity Obstacles”, “Acceleration Velocity Obstacle” and “Hybrid Velocity Obstacle” are some of them. Researches until today show that vehicle dynamics, parking problem and computational load are not considered at all. Reducing the computational load by keeping the algorithm in a basic form is one reason here. In the thesis, using this information, a new method has been developed and applied on mobile robots.

Keywords: Mobile robots, velocity obstacle, collision avoidance, QBot

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

1.1 Literatür Özeti

İki ana başlık altında incelenen literatür çalışmasının ilk bölümünde hız engelleri ile ilgili yapılan çalışmalar incelenmiş, ikinci bölümünde de görsel geri besleme ile yörünge takibi ile ilgili çalışmalar incelenmiştir.

Hız engeli kavramı ilk olarak 1998 yılında Fiorini ve Shiller tarafından ortaya konmuştur. Bu çalışmada amaç hareketli ve statik nesnelerin bulunduğu ortamlarda tek bir aracın çarpışmadan kaçınması için hız engeli olarak tanımlanan küme dışında bir vektörün seçimidir [1]. Prassler, Bank ve Kluge çalışmalarında bir insanın hareketi ile uyumlu bir şekilde çalışan mobil robotun hareket planlamasını klasik hız engelleri kullanarak gerçekleştirmişlerdir. Bu çalışmayı tekerlekli sandalye ve benzeri rehabilitasyon araçları üzerinde denemişlerdir [2]. Abe ve Matsou, çoklu otonom mobil araçların çarpışmasını önlemek amacı ile yaptıkları çalışmalarında kooperatif çarpışmadan kaçınma yöntemini kullanılmıştır. Önerilen yöntem bu tür uygulamalarında çok sık kullanan hareketli mobil robotlar arasında çarpışmayı algılamayı sağlayan hız engeli yönteminin bir uzantısıdır. Uygulama göz önüne alındığında, kullanılan yöntem gerçek zamanlı her örnekleme zamanında yerel ölçümlere ve yerel planlamaya dayalıdır. Hız engeli yöntemi sadece çarpışma olabilecek hız vektörleri kümesini sağlar ve hız seçimi ile ilgili bir sınırlama yapmaz. Önerilen yöntemde robotlar açık iletişim olmadan çarpışma bilgilerini paylaşır. Çarpışma bilgisini paylaşmak genel hız engellerini kullanarak robotların birbirleri ile işbirliği yapmasını sağlar. İyi görev paylaşımı ve anlık olarak çarpışmayı önlemek için hızları değerlendiren hız indeksleri tanımlanmıştır. Hız indeksleri görev hızlarının

ağırlıkları toplamından hesaplanır ve CVO hız indeksleri de genel hız indekslerinin en iyi sonuç verenleri arasından seçilir gelecek çarpışmaları önlemek için hızları yönlendirir [3]. Sonraki bir çalışmada Large, Sekhavat, Shiller ve Laugier, dinamik çevrede gerçek zamanlı global hareket planlama üzerine bir çalışma yapmışlardır. Bu alanda yapılan birçok yaklaşımın ağır hesaplama yükü ve tatmin edilemeyen gerçek zamanlı kısıtlamalardan şikâyetçi olması nedeni ile bu çalışmada doğrusal olmayan hız engeli yaklaşımı öne sürülmüştür. Bu yaklaşımda robot tarafından kabul edilen engellerin pozisyonları ve bilinen ya da tahmin edilen yörüngeleri kinematik ve dinamik kısıtlamalar dikkate alınarak hız uzayında haritalanmıştır. Sonuç olarak çarpışmadan bağımsız hız haritaları oluşturulmuş ve yerel hedef odaklı engelden kaçınmayı gerçekleştirmek için risk kavramı geliştirilmiştir. Grafik genişleme teknikleri ile bu yöntem birleştirilerek hareketli ortamda değişen global hareket planlayıcısı tanımlanmıştır [4]. Aynı ekip bu konuda benzer bir çalışma daha yapmış ve hareketli ortamda değişen global hareket planlayıcısı ile birlikte yerel engelden kaçınma sisteminden oluşan, robot için kabul edilebilir hızları kullanarak tamamen otonom bir navigasyon müdürlü oluşturmuşlardır [5]. Myers ve diğerleri çalışmalarında dinamik engelden kaçınma problemine çözüm olarak gradyan hız engeli algoritmasını kullanarak yeni bir yaklaşım geliştirmiştir. Çalışmada özel tasarlanmış engel tespiti için kullanılan LADAR (lazer destekli radar) sensörü bulunduran hareket kabiliyeti kısıtlı araba benzeri araç ile gerçek zamanlı uygulama yapılmıştır [6]. Bu tür çalışmalarda engelin pozisyonu ve hızı (yön ve genliği) ile ilgili mümkün olduğu kadar çok bilgiye ihtiyaç vardır. Fakat bazı durumlarda engel aracın gömülü sensörüne çok yakın bulunur ve burada ölü noktalar oluşturur. Hatta bu ölü noktaların arkasına diğer engellerde saklanabilir. Torres, Breno ve diğerleri bu probleme çözüm getirmek için sadece iki boyutlu lazer tarama verilerine dayalı bir engel izleme modülü oluşturulmuştur. Engel izleme modülü, engel tanıma, engel sınıflandırma ve engel izleme olarak üç ana bölümden oluşmaktadır ve buna ek olarak, belirli zaman aralığında yasak olan tekerlek açıları kümesine karar vermek için kullanılan hız engeli yaklaşımı geliştirilmiştir [7]. Claes, Hennes ve diğerleri çalışmalarında robotların güvenli hareketleri kararlaştırmak için çevredeki robotların o anki pozisyon ve hızlarını geometrik olarak göstermiş ve her robotun etrafındaki hız engellerini oluşturmak için yerleşik konum belirleme ve yerel

iletişim kullanmışlardır. Çalışmada kullanılan bölge yoğunluğunun kestirimi belirsizlikler altında çarpışmadan bağımsız izlenebilecek konumları verir. Birçok algoritmada robotlar sınırlı yarıçaplarıyla değerlendirilirken, bu çalışmada dışbükey yaklaşımı kullanılmıştır [8]. Berg ve diğerleri 2008 yılında karşılıklı hız engellerini tanıtmış ve ilk olarak iki aracın bulunduğu durumlarda bu metodun kullanımına ait sonuçları sunmuş [9] ve daha sonra bunu çok sayıda aracın olduğu durumlara uyarlamıştır [10]. Alonso-Mora ve diğerleri ise farklı araç tiplerine uyarlanması üzerine çalışmalar gerçekleştirmiştir [11], [12]. Jiawei, Xunyu ve Xiafu çalışmalarında önceki çoklu-mobil robot hareket planlama yöntemlerinden farklı olarak, öne sürdükleri yöntemde temel olarak dinamik kısıtlamalar ve kontrol verimliliğine odaklanmışlardır. Robotun kontrol hızını planlamak için hız değişim uzayı kavramı tanımlanmış ve değerlendirme fonksiyonu hız değişim uzayını temel alarak en iyi kontrol hızına karar vermek için tasarlanmıştır [13]. Berg, Snape ve diğerleri hız engellerine araçların ivmelenmesini de ekleyerek daha gerçekçi bir yaklaşımda bulunmuş ve ivme-hız engellerini tanıtmıştır [14]. Dubey, Krishna ve diğerleri çalışmalarında alan programlanabilir kapı dizileri (FPGA) kullanarak ivmelenme sınırlamalarıyla yönelimsiz robotlar için çarpışmadan kaçınma tabanlı ivme hız engellerini gerçekleştirmişlerdir. Uygulamada özellikle çarpışma önleyici paralel mimari kullanılarak FPGA'in paralel işlem yapabilme özelliğinden faydalanmak istenmiştir [15]. Snape, Berg ve diğerleri sonrasında gerçekleştirdikleri çalışmalarında hibrit hız engelleri kavramını ortaya koymuştur. Burada karşılıklı hız engellerinin kullanıldığı durumlarda oluşan bazı problemlerin çözümüne yönelik daha kapsayıcı bir yöntem geliştirilmiştir [16].

Görsel geri besleme ile hedef takibi konusunda Y. Kanayama ve diğerleri doğrusallaştırma kullanarak temel bir çalışma gerçekleştirmiştir [17]. Bu konuda farklı yöntemler kullanılmıştır, mesela R. Fierro, F. L. Lewis geri adımlamalı kontrol metodunu kullanırken [18], K. H. Park, S. B. Cho ve Y. W. Lee eniyi denetim üzerine çalışmıştır [19]. C. Y. Tsai ve K. T. Song kutup atama metodundan faydalandığı bir çalışma gerçekleştirmiştir [20]. G. Scagliave diğerleri doğrusal enterpolasyon metodunu kullanmıştır [21]. C.Y. Tsai, K. T. Song ve J. M. Jung, J. H. Kim dayanıklı denetleyici tasarımı üzerine çalışmalar gerçekleştirmiştir [22], [23]. G. Yuan ve diğerleri ise yapay sinir ağlarından faydalanmıştır [24]. Konuyla ilgili bir diğer sınıflandırma uygulamaya

yönelik olarak da yapılabilir. Bazı durumlarda görsel veriler yerine diğer algılayıcılar ile elde edilen geri besleme bilgisinden yararlanılmıştır. Örneğin A. Treptow ve diğerleri termal kameraların kullanımına yönelik bir çalışma gerçekleştirmiştir [25]. Y. Hada ve ekibi ise iGPS ve odometrinin kullanımına yönelmiştir [26]. H Zhang ve J. P. Ostrowski işlem yükünün azaltılması yönünde çalışmış, K. H. Park ve diğerleri ve L. Rui ve ekibi takip işlemi sırasında engellerle karşılaşılması durumunu ele almıştır [27], [19], [28]. J. B. Coulaud sınırlı özelliklere sahip bir hattın takibine yönelik çalışmıştır [29]. L. Freda ve G. Oriolo ise görsel verinin görüş alanında tutulması problemine yönelik çözümler üretmiştir [30]. Bunun dışında bazı çalışmalarda hız kestirimciler kullanılmıştır [22], [31], [32]. Çoğu çalışmada aracın hız modeli kullanılmışken [18] ve [23] tork modelinden yararlanılmıştır böylece araç dinamikleri daha kapsamlı olarak ele alınmıştır. Ek olarak [15] ve [16] çalışmalarında genişletilmiş kalman filtresinden faydalanmıştır.

1.2 Tezin Amacı

İnsansız kara araçlarının gelişimi ile beraber otonom araçlarda engelden kaçma ve park problemleri ortaya çıkmış ve bununla ilgili çalışmalar yapılmış ve yapılmaya devam edilmektedir. Engelden kaçma probleminde hareketli engeller büyük sorun teşkil etmektedir. Bu sorunların en büyüğü aynı yapıya sahip birden çok aracın birbirleri ile etkileşim halinde olduğu hareketli engel problemidir. Literatür çalışmasında bahsedildiği gibi hız engelleri kullanılarak oluşturulan metodlar başarılı çözümler getirmektedir. Bu çalışmada hız engelleri ile oluşturulan Hibrit hız engelleri yönteminin uygulanmasında alternatif olarak kullanılabilir ızgaralama yöntemi ile çözüm aranmıştır. Oluşturulan bu yöntem sayesinde algoritmanın gömülü olarak çalıştırıldığı platform için işlem yükü azaltılmış ve uygulanabilirlik elde edilmiştir.

Araçların hedef noktaya ulaştıklarında ortaya çıkan park problemi için görüntü geri beslemesinden faydalanarak geri beslemeli doğrusallaştırma yöntemi ile çözüm aranmıştır ve bu farklı bir uygulama ile tek araç üzerinde test edilmiştir.

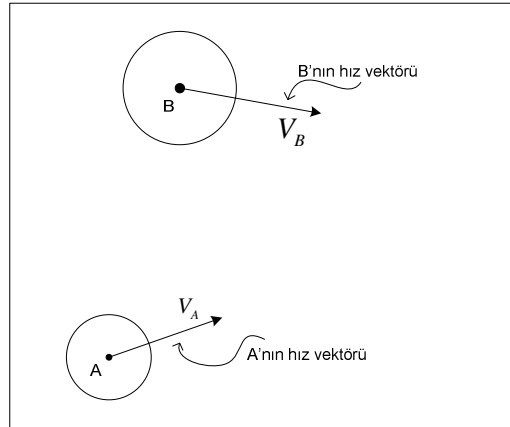
1.3 Hipotez

Hibrit hız engellerinin kullanımı beraberinde araçlar için yeni bir hız vektörünün belirlenmesi problemini doğurmaktadır. Şöyle ki diğer araçlar için oluşturulan Hibrit hız

engellerinin bileşiminden doğan geometrik şekil üzerinde en uygun sonucu bulmak için kullanılan yöntemin işlem yükü yüksektir. Buna karşın yaklaşık olarak bulunabilecek olan sonuçlarda çarpışmaları önlemek açısından yeterli olmaktadır. Bu amaçla ızgaralama metodu ile daha az işlem yüküne sahip algoritmalar oluşturulabilmektedir.

2.1 Basit Hız Engelleri

Fiorini ve Schiller [1] 1998 yılında gerçekleştirdikleri çalışmada hız engeli kavramını ortaya koymuşlardır. Bu kavramın amacı A ile temsil edilen robot ve B ile temsil edilen hareketli bir nesnenin çarpışmasını önlemek için hız uzayında hız engeli adı altında bir kümenin elde edilmesidir. A aracı yeni hız vektörünü bu kümenin dışında seçtiği sürece çarpışmanın gerçekleşmeyeceği ispat edilmiştir.



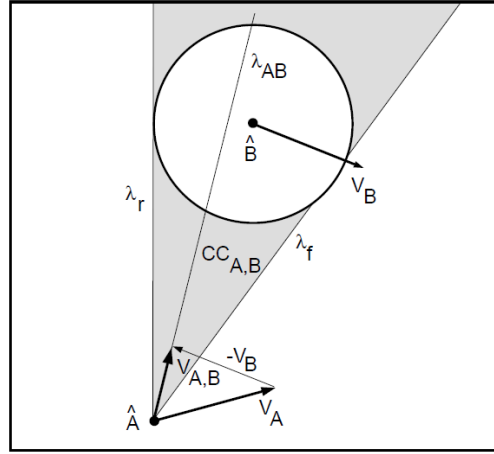
Şekil 2.1 A aracı ve B nesnesi

Bu kümenin oluşturulmasında izlenmesi gereken geometrik metot ise şu şekilde sunulmuştur; öncelikle yeni bir biçim uzayına geçiş yapılır, burada A aracı noktasal olarak \hat{A} ile temsil edilir, B hareketli nesnesi ise A 'nın yarıçapı kadar genişletilerek

\hat{B} ile temsil edilir. Bu tanımlamalardan sonra çarpışma koniği (Collision cone) aşağıdaki şekilde ifade edilir.

$$CC_{A,B} = \{v_{A,B} \mid \lambda_{A,B} \cap \hat{B} \neq \emptyset\} \quad (2.1)$$

$v_{A,B}$ \hat{A} nın \hat{B} ye göre bağıl hızını ifade etmektedir. $\lambda_{A,B}$ ise bu bağıl hız vektörü doğrultusunda çizilen ışını ifade etmektedir. Yukarıdaki kümeden de anlaşılacağı gibi çarpışmanın oluşması için $\lambda_{A,B}$ bu koniğin içinde olmalıdır, aksi durumda çarpışma riski bulunmadığı için koniğin hesaplanmasına da gerek yoktur. Şekil 2.2' de çarpışma koniği görülmektedir.



Şekil 2.2 Çarpışma koniği

Bu adımdan sonra hız engeli (VO) Denklem 2.2'deki şekilde ifade edilir.

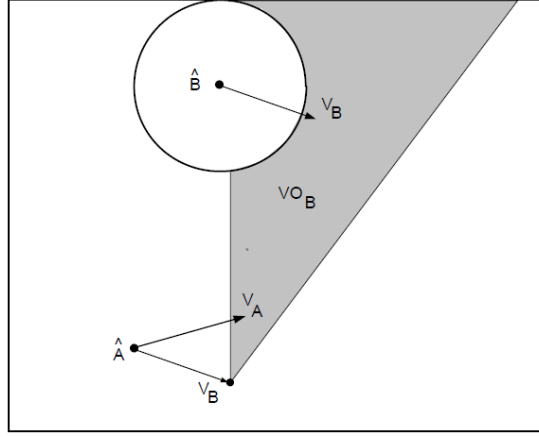
$$VO = CC_{A,B} \oplus v_B \quad (2.2)$$

Burada \oplus operatörü Minkowski toplamını ifade etmektedir.

Tanım 2.1 Minkowski toplamı A ve B gibi iki konum vektörü kümesi ele alındığında A kümesindeki her elemanın B kümesindeki her elemana eklenmesi ile elde edilir. Denklem 2.3'te bu toplam gösterilmiştir.

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \quad (2.3)$$

Tanım 2.1'e göre hız engeli çarpışma koniğinin v_B vektörü kadar ötelenmesi ile elde edilir. Şekil 2.3'de bu durum geometrik olarak gösterilmiştir.



Şekil 2.3 Hız engeli

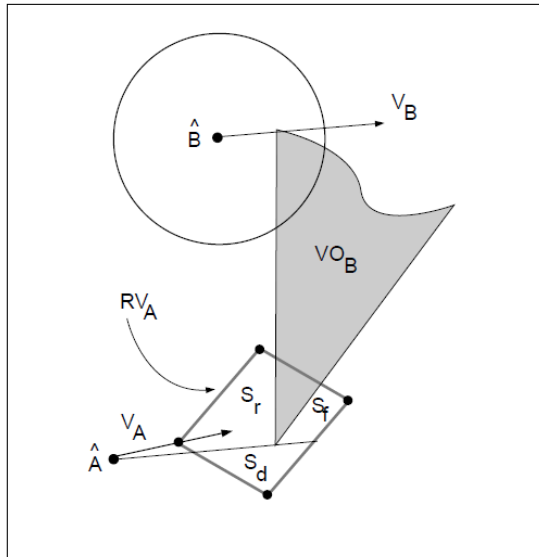
Hız engeli uzayı v_A 'ya göre, kaçınan (avoiding) ve çarpışan (colliding) hızlar olarak ikiye ayırır. Çarpışmanın önlenmesi için yeni v_A hızı bu kümenin dışında seçilmelidir.

$$A(t) \cap B(t) = 0 \text{ eğer } v_A(t) \notin VO(t) \quad (2.4)$$

Birden fazla B hareketli nesnesi olması durumunda, her bir nesne için bulunan hız engeli kümelerinin birleşimi kullanılır.

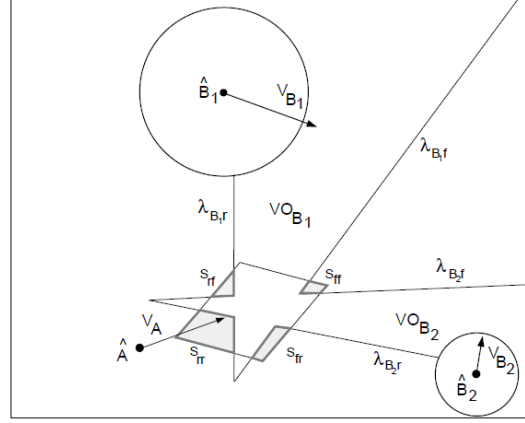
$$VO = \bigcup_{i=1}^m VO_{Bi} \quad (2.5)$$

Bundan sonra A aracının gerçekleştirilebilir ivmelenmesi (feasible accelerations) göz önüne alınır. Bu da erişilebilir hızların (reachable velocities) kümesini verecektir.



Şekil 2.4 Erişilebilir Hız Kümesi

Burada hız engeline göre yapılacak seçimlere göre (S_f, S_r, S_d) A aracı B nesnesinin arkasından (S_r) ya da önünden (S_f) geçebileceği gibi bu nesneden uzaklaşarak da (S_d) çarpışmadan kaçınabilir. Birden çok hareketli B nesnesinin olması durumunda seçenekler azalabilmektedir. Şekil 2.5' de bu durum görülmektedir.



Şekil 2.5 Kaçınma Kümeleri

Çalışmalarının bundan sonraki kısmında Fiorini ve Schiller [1] en uygun hız vektörünün seçimi için değişik arama yöntemleri önermiş ve bunların sonuçlarını vermişlerdir.

Günümüze yakın zamanda yapılan çalışmalarda ise hız engelleri popülerliğini korumuş ve kısmi değişiklik yapılarak belli ölçütlere göre başarımın artırılması yoluna gidilmiştir. Fakat farklı olarak bu kavramın sürü robotların olduğu sistemlere uyarlanması üzerine çalışılmıştır.

2.2 Karşılıklı Hız Engelleri

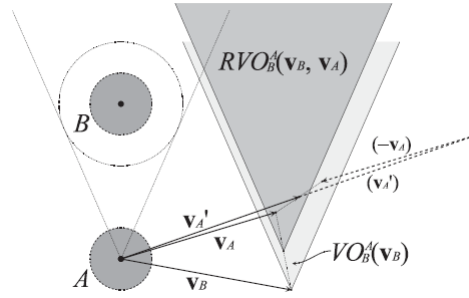
Yakın zamanda yapılan çalışmalardan biri Van den Berg, Lin ve Manocha [9] tarafından 2008 yılında gerçekleştirilmiştir. Burada Karşılıklı Hız Engelleri (Reciprocal Velocity Obstacles) adı altında yeni bir yöntem oluşturulmuştur. Eski metoda göre farklılık ise burada sabit hızda hareket eden nesnelere yerine bütün nesnelere aynı şekilde karar veren robotlar olduğu düşünülmüştür.

Fakat en yalın şekliyle hız engellerinin kullanması beraberinde kararsızlık sorununu getirmiştir. Bunun sebebi bağıl hızlara göre elde edilen hız engellerinin karşılıklı iki araç için simetrik şekilde oluşmasıdır. Bunun sonucunda da araçların birbirinden habersiz olarak yeni hızlarını seçtikten sonra çarpışma tehlikesi atlatılır. Fakat bu durumda

yeniden uygun görünen eski hızlarına tekrar döndüklerinde ise aynı sonucun elde edilmesiyle araçların kararsız davranarak çarpışmaya sürüklenmesidir. Karşılıklı hız engeli (RVO) şu şekilde tanımlanmıştır;

$$RVO_B^A(v_B, v_A) = \{v'_A \mid 2v'_A - v_A \in VO_B^A(v_B)\} \quad (2.6)$$

Buradaki amaç A ve B araçlarının eşit çabayla kaçıışı için aynı tarafı kullanmasını sağlamaktır.



Şekil 2.6 Karşılıklı hız engeli

Eğer araçların harcayacakları çaba eşit değil ise bu durum için daha genelleştirilmiş denklem formu Denklem 2.7'de görülmektedir.

$$RVO_B^A(v_B, v_A, \alpha_B^A) = \{v'_A \mid \frac{1}{\alpha_B^A} v'_A + (1 - \frac{1}{\alpha_B^A}) v_A \in VO_B^A(v_B)\} \quad (2.7)$$

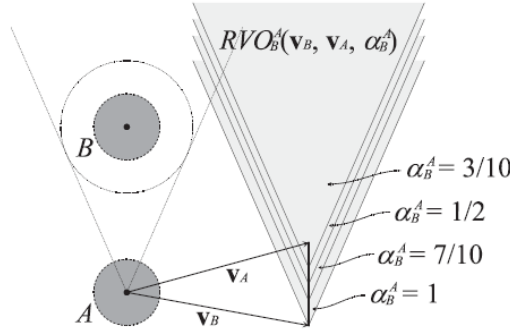
Bu durumda araçların çaba miktarlarını belirlemek üzere α_A^B ve α_B^A arasındaki ilişki Denklem 2.8'deki gibi seçilmelidir.

$$1 - \alpha_A^B = \alpha_B^A \quad (2.8)$$

Normal durumda bu ilişki Denklem 2.9'deki gibi olmalıdır.

$$\alpha_A^B = \alpha_B^A = \frac{1}{2} \quad (2.9)$$

Genelleştirilmiş form kullanıldığında elde edilen RVO'lar Şekil 2.7'de gösterilmiştir.



Şekil 2.7 Farklı parametreler için hız engeli

Son olarak sabit hıza sahip nesnelerin (O) ve diğer aynı davranışı sergileyen araçların olduğu ortamlar için toplam RVO şu şekilde verilmiştir.

$$RVO^i = \bigcup_{j \neq i} RVO_j^i(v_j, v_i, \alpha_j^i) \cup_{o \in o} VO_o^i(v_o) \quad (2.10)$$

Buradan görüldüğü gibi sabit hızlı nesnelerin karşılıklı bir hareketi olmayacağı için bu tip nesneler için sadece hız engellerinin hesaplanması yeterlidir.

2.3 Karşılıklı Çok Sayıda Aracın Çarpışmasının Engellenmesi

Bundan sonra 2009' da Van den Berg ve diğerleri [10] tarafından "Reciprocal n-body collision avoidance" karşılıklı çok sayıda aracın çarpışması ile ilgili yeni bir çalışma gerçekleştirilmiştir. Bu çalışmadaki amaç ise hesaplama yükünü azaltmak ve her aracın hedef noktalarına ulaşmak için kullanacağı v_{opt} hızına en yakın hız vektörünü elde etmektir.

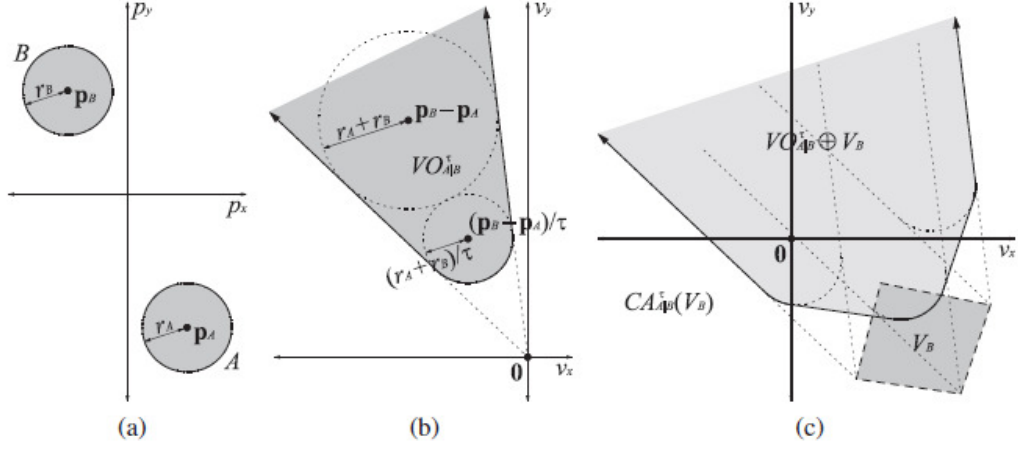
Bu yöntem için şu aşamalar izlenmektedir. İlk olarak belli bir τ zaman dilimi için hız engeli tanımlanır. Şöyle ki $D(p, r)$, p noktasında merkezlenmiş r yarıçaplı bir açık disk olsun,

$$D(p, r) = \{q \mid \|q - p\| < r\} \quad (2.11)$$

Buna göre belli bir zaman diliminde oluşabilecek çarpışmayı tanımlayan hız engeli Denklem 2.12'deki gibi ifade edilir.

$$VO_{AB}^\tau = \{v \mid \exists t \in [0, \tau] : tv \in D(p_B - p_A, r_A + r_B)\} \quad (2.12)$$

Denklem 2.11 ve 2.12 geometrik olarak Şekil 2.8'deki gibi gösterilebilir.



Şekil 2.8 Kısıtlı zaman arlığında hız engeli

Bundan sonra A ve B araçları için V_A ve V_B kümelerinin olduğu düşünülür ve çarpışma önleyecek hızlar kümesi şu şekilde tanımlanır.

$$CA_{A/B}^{\tau}(V_B) = \{v \mid v \notin VO_{A/B}^{\tau} \oplus V_B\} \quad (2.13)$$

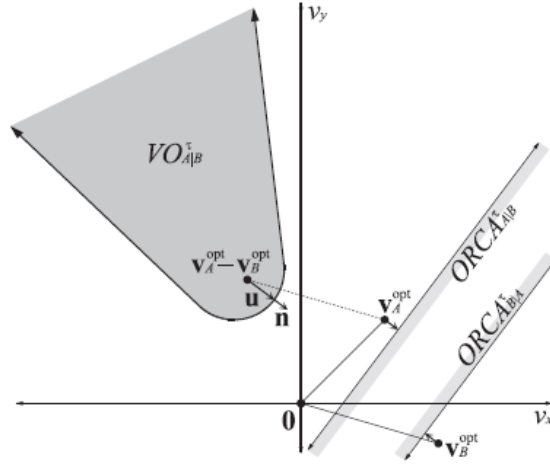
Bu durum Şekil 2.8 (c)' de görüldüğü gibidir. Eğer aşağıdaki şartlar sağlanıyor ise,

$$V_A \subseteq CA_{A/B}^{\tau}(V_B) \text{ ve } V_B \subseteq CA_{B/A}^{\tau}(V_A) \quad (2.14)$$

Bu iki kümeye karşılıklı çarpışma önleyici (reciprocally collision avoiding) adı verilir. Ayrıca, ifade Denklem 2.15'deki gibi olursa bu durumda karşılıklı azami (reciprocally maximal) kümeler adını alır.

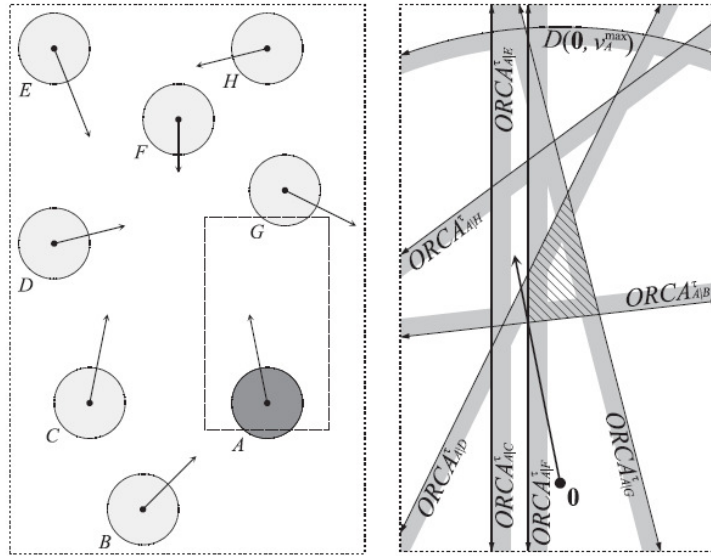
$$V_A = CA_{A/B}^{\tau}(V_B), V_B = CA_{B/A}^{\tau}(V_A) \quad (2.15)$$

Bütün bu tanımları sağlayacak olan kümelere de ORCA (Optimal Reciprocal Collision Avoidance) kümeleri denir. Bu kümelerin en önemli özelliği araçların hedef noktalarına giderken korumak istedikleri v_{pref} hız vektöründen en az değişikliği yaparak bu kümelerin içine girebilmelidir. Bir diğer önemli nokta bu kümeler birer yarı düzlem verir ve Şekil 2.9' da bu durum görülmektedir.



Şekil 2.9 ORCA yarı düzlemleri

Birden çok aracın bulunduğu durumlarda bu kümelerin oluşturacağı yarı düzlemlerin kesişiminden oluşan bölge çarpışma önleyecek olan kümeyi verir. Şekil 2.10' da bu durum görülmektedir.

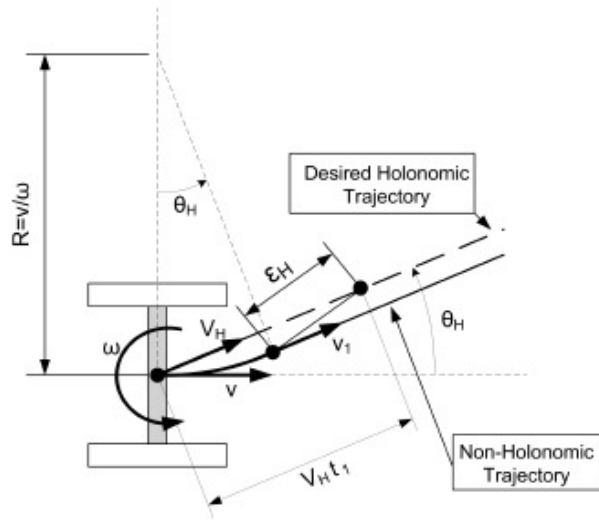


Şekil 2.10 Birden çok araç için ORCA yarı düzlemleri

Burada önemli olan nokta araçların hedeflerine giderken seçtiği en uygun hızın (v_{opt}) genişliğidir (v_{opt} yön olarak zaten hedefe doğru gitmeyi amaçlar). Bu hız yüksek seçildiğinde araçların çarpışma önleyici bir küme bulması mümkün olmadığı, sıfır olarak seçildiğinde de çok aracın bulunduğu ortamlarda kilitlemeler söz konusu olduğu belirtilmiştir [10]. Bu çalışmalarda önemli olan bir diğer nokta araçların her yöne

hareket edebilir (omni-directional) olduğunun düşünülmesidir. Oysaki uygulama da kullanılan araçlar hareket kabiliyeti kısıtlı yapıdadır.

Bu yöntemin hareket kabiliyeti kısıtlı araçlarda uygulanması üzerine bir çalışma 2010 senesinde Alonso-Mora ve diğerleri [11] tarafından gerçekleştirilmiştir. Bu çalışmada öncelikle hareket kabiliyeti kısıtlı bir aracın bu yapısına uygun olmayan bir yörüngeyi belli bir ε_H hatası ile izlenebileceği üzerinde durulmuştur. Şekil 2.11’de bu durum görülmektedir.



Şekil 2.11 Hareket kabiliyeti kısıtlı araç yapısı

Şekil 2.11’de tanımlanan araç ve parametreler için yörünge izlenirken oluşacak takip hatası Denklem 2.16’da ifade edilmiştir.

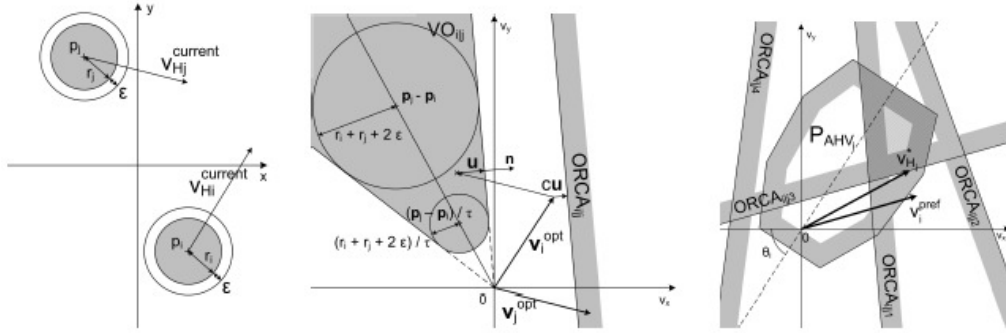
$$\begin{aligned} \varepsilon_H^2(v, w, V_H, \theta_H) &= (V_H t_1 - R \sin(\theta_H))^2 + (R(1 - \cos(\theta_H)))^2 \\ &= V_H^2 t_1^2 - \frac{2V_H t_1 \sin(\theta_H)}{\omega} v + \frac{(R(1 - \cos(\theta_H)))^2}{\omega^2} v^2 \end{aligned} \quad (2.16)$$

Bundan sonra takip hatasını belli bir ε değerinden küçük tutabilecek izin verilen holonomik hızlar kümesi (allowed holonomic velocities) belirlenir.

$$S_{AHV} = \{v_H \in \mathfrak{R}^2 \mid \exists (v(\tau), \omega(\tau)) \in S_{NHC}, \left\| p + \tau \cdot v_H - \hat{p}^k(\tau) \right\| \leq \varepsilon, \forall \tau \geq 0\} \quad (2.17)$$

Burada $\hat{p}^k(\tau)$ bir sonraki adımda beklenen robot konumunu ifade eder. Sonraki adımda bu kümeyi kapsayan bir poligon elde edilir ve ORCA yarı düzlem doğrularının

oluşturacağı çarpışma önleyici hızlar kümesi ile kesişimi göz önüne alınarak yeni hız vektörü seçilir. Şekil 2.12’de bu durum görülmektedir.



Şekil 2.12 ORCA doğruları ve P_{AHV} poligonu

Bir diğer önemli nokta, araçların yarıçaplarının takip hatası sebebiyle ε kadar genişletilmesidir.

Benzer bir çalışma daha sonra Alonso-Mora ve diğerleri [14] tarafından hareket kabiliyeti kısıtlı araçlar için gerçekleştirilmiştir. Bu çalışmalarda hesaba katılmayan bir diğer nokta aracın kinematik olarak anlık hız değişimlerini gerçekleştiremeyecek olmasıdır.

2.4 İvme Hız Engelleri

Van den Berg ve diğerleri [6] bu durumu ele alan “İvme-Hız Engelleri” isimli yeni bir metod geliştirmiştir. Buradaki amaç kendi kapalı çevrim hız denetleyicisine sahip olan araçların üstel olarak yeni bir hıza geçiş yapacağını hesaba katarak ivme-Hız engellerini hesaplamaktır. Başlangıç olarak çok yönlü hareket eden ve ivmelenmesi kısıtlanmış bir araç ele alınırsa,

$$a_A(t) = v_A'(t) = \frac{v_A' - v_A(t)}{\delta}, \|a_A\| < a_A^{\max} \quad (2.18)$$

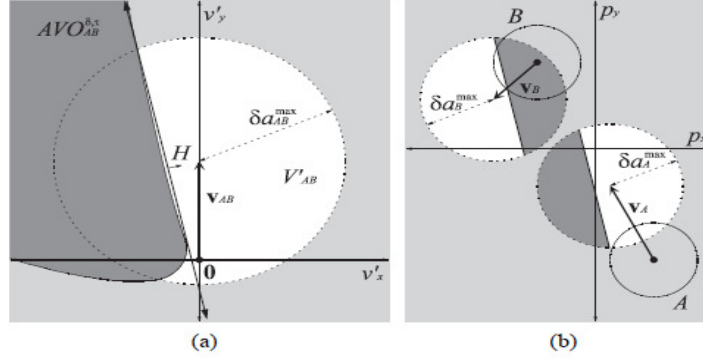
Burada δ zaman boyutunda tanımlanmış bir kontrol parametresidir. Bu denklemden hareketle hızın zamana göre olan fonksiyonu elde edilirse,

$$v_A(t) = v_A' - e^{-t/\delta} (v_A' - v_A) \quad (2.19)$$

Şeklinde elde edilecektir. Bu bilgiler ışığında belli bir zaman aralığı için ivme-hız engeli (AVO) Denklem 2.20’de tanımlanmıştır.

$$AVO_{AB}^{\delta,\tau} = D\left(\frac{\delta(e^{\frac{t}{\delta}} - 1)v_{AB} - p_{AB}}{t + \delta(e^{\frac{t}{\delta}} - 1)}, \frac{r_{AB}}{t + \delta(e^{\frac{t}{\delta}} - 1)}\right) \quad (2.20)$$

Buradan hareketle ivme-hız engeli ve aracın ivmesinin ORCA yarı düzlemleri ile kesişimi sonucu oluşan seçilebilir hız kümeleri Şekil 2.13' de verilmiştir.

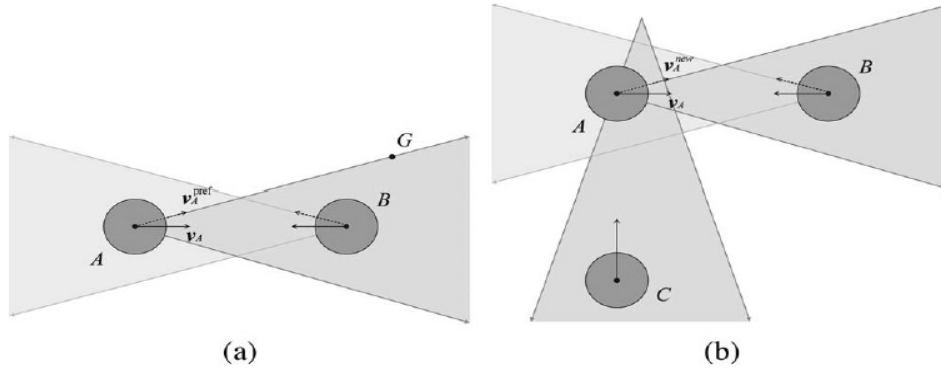


Şekil 2.13 İvme-hız engeli ve seçilebilir hız kümeleri

Bu yöntem uygulanırken ortaya çıkan önemli bir problem oluşan ivme hız kümelerinin dış-bükey olmamasıdır. Bu sebeple çıkan şeklin önce dış-bükey zarfı bulunur. Tanım gereği ORCA yarı düzlem doğrularının bulunabilmesi için bu gereklidir.

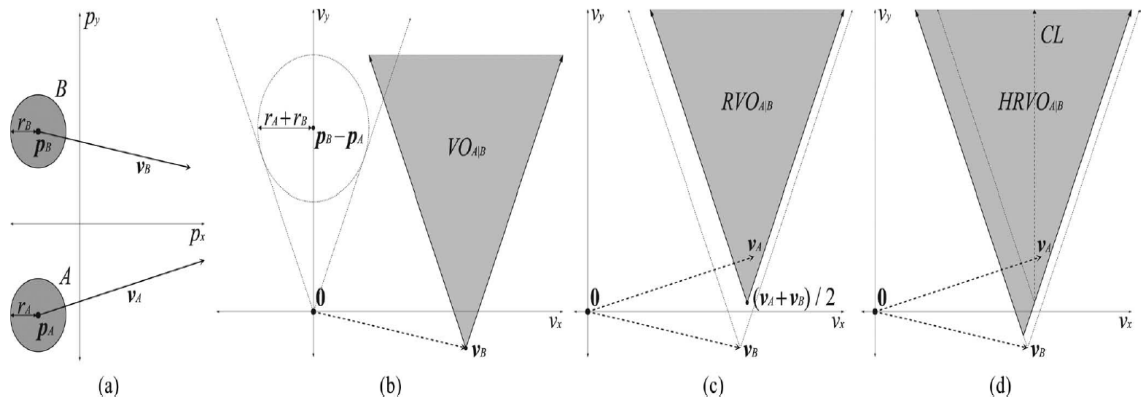
2.5 Hibrit Hız Engelleri

Hibrit hız engelleri karşılıklı hız engellerine alternatif olarak oluşturulmuştur. Karşılıklı hız engellerinin kullanımı her ne kadar başarılı sonuçlar vermiş olsa da yolları kesişen araçların birbirini aynı taraftan geçmesi zorunluluğu kimi zaman istenmeyen sonuçlar doğurmuştur. Bazı durumlarda aracın ters tarafı seçmesi en uygun veya zorunlu çözüm halini alabilir. Şekil 2.14'de bu durumun ortaya çıktığı iki farklı senaryo görülmektedir.



Şekil 2.14 Karşılıklı hız engellerinde ters tarafı seçme durumu

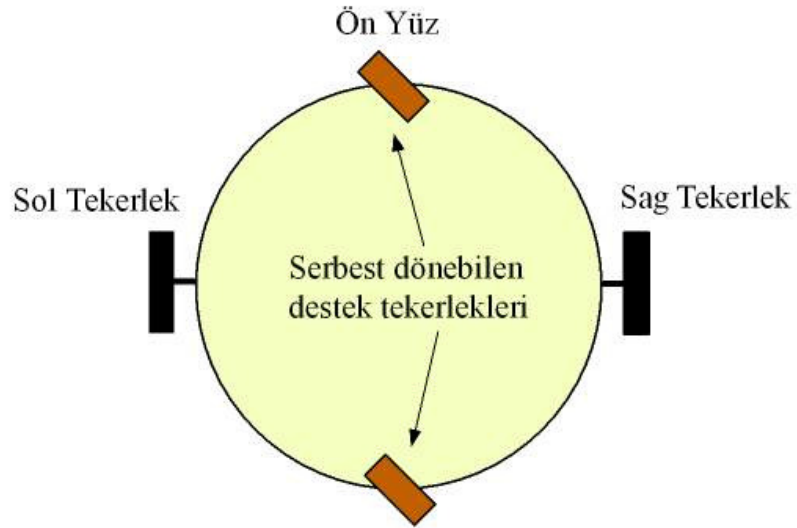
Hibrit hız engelleri ters tarafın seçilebilmesini sağlamak amacıyla, karşılıklı hız engelinin ters taraf olarak nitelendirilen kolunu klasik hız engelinin koluna öterler. Böylece araçlardan biri ters tarafı seçtiğinde tüm sorumluluğu üstlenmiş olur. Bu tanım gereği hibrit hız engeli oluşturulurken öncelikle hız vektörünün hız engelinin orta çizgisine göre hangi tarafında kaldığına bakılmaktadır, buna göre diğer taraf klasik hız engeline kadar genişletilecektir. Şekil 2.15’de bu küme geometrik olarak gösterilmiştir.



Şekil 2.15 Hibrit hız engellerinin geometrik yapısı

ARAÇ MODELİ VE PARK PROBLEMİ**3.1 Araç Modeli**

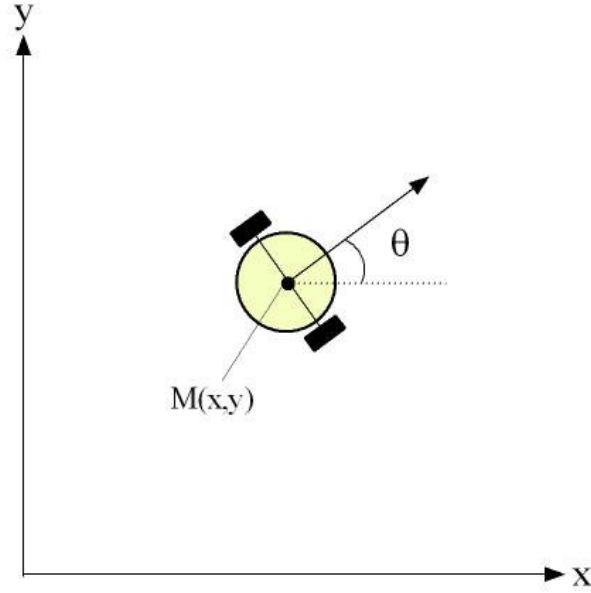
Araç geometrik yapısının oluşturulmasında öncelikle literatürdeki kullanılan modeller gözlemlenmiş ve araç modeli olarak diferansiyel sürüşe sahip, dairesel gövdeli araç modeli benimsenmiştir. Uygulamada kullanılan QBot aracı da aynı modele sahiptir. Şekil 3.1’de kullanılan araç modeli gösterilmiştir.



Şekil 3.1 Araç Modeli

Kullanılacak olan aracın üzerinde iki adet kendi ekseninde serbest dönebilen destek tekerlekleri bulunmaktadır. Bu tekerleklerin oluşturacağı sürtünme ve diğer etkiler araç modelinde ihmal edilmiştir. Aracın hareket denklemleri yazılırken araç

merkezi Şekil 3.2 de görülen aracın tam orta noktası seçilmiştir. Buradaki amaç araç yapısının simetrik olmasını sağlayıp denklemleri basitleştirmektir.



Şekil 3.2 Araç koordinatlarının tanımlanması

Öncelikle aracın merkezinin (M) konumsal olarak değişimini ele alacak olursak, aracın tekerleklerinin sahip oldukları hız vektörleri sağ tekerlek için v_1 ve sol tekerlek için v_2 olarak tanımlanır. Bu durumda araç merkezinin (M) sahip olduğu hızın vektörel değeri Denklem 3.1'deki gibi gösterilir.

$$v_m = \frac{v_1 + v_2}{2} \quad (3.1)$$

Bu sonuca bağlı olarak, aracın x ve y koordinatlarına göre sahip olduğu hız vektörleri Denklem 3.2'deki gibi ayrıştırılabilir.

$$v_x = v_m \cos \theta = \frac{v_1 + v_2}{2} \cos \theta \quad (3.2)$$

$$v_y = v_m \sin \theta = \frac{v_1 + v_2}{2} \sin \theta$$

Burada θ aracın oryantasyon açısını ifade etmektedir. Açısal hız Denklem 3.3'deki gibi ifade edilir.

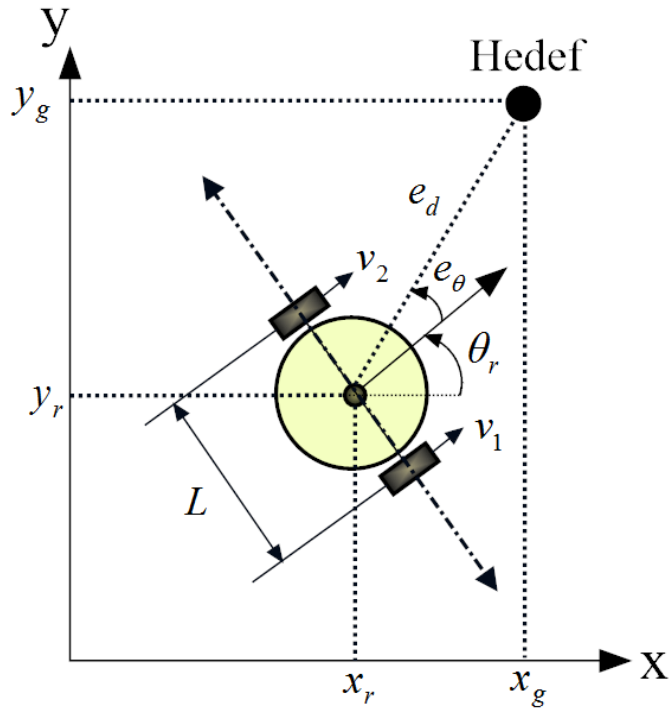
$$\dot{\theta} = \frac{v_1 - v_2}{L} \quad (3.3)$$

Bu denklemin elde edilmesi ile v_1 ve v_2 hızlarına bağlı olarak kartezyen koordinatlarda aracın yapacağı konum değişikliği hesaplanabilmektedir. Sonuç olarak mobil robotun matematiksel modelini ifade eden denklemlerin matris formu Denklem 3.4'deki gibi ifade edilir.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cos \theta & \frac{1}{2} \cos \theta & 0 \\ \frac{1}{2} \sin \theta & \frac{1}{2} \sin \theta & 0 \\ \frac{1}{L} & -\frac{1}{L} & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \theta \end{bmatrix} \quad (3.4)$$

3.2 Park Problemi

Aracın hedeflenen bir noktaya ulaşma problemi Şekil 3.3'deki gibi ifade edilir.



Şekil 3.3 Robot park problemi

Burada v_1 ve v_2 aracın tekerlek hızları, L tekerlekler arasındaki mesafe, e_d hedef noktaya olan uzaklık hatası, e_θ 'da açı hatası olarak tanımlanır. x_r ve y_r , aracın konumunu ve x_g ve y_g 'de aracın hedef noktasının konumunu ifade eder. Bu bilgiler kullanılarak aracın tekerlek hızlarına bağlı matematiksel modeli aşağıdaki gibi yeniden ifade edilir.

$$\begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cos \theta_r & \frac{1}{2} \cos \theta_r \\ \frac{1}{2} \sin \theta_r & \frac{1}{2} \sin \theta_r \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (3.5)$$

Park problemine çözüm olarak sunulan geribesleme ile doğrusallaştırma yöntemi için tekerlek hızlarını kullanmak yerine aşağıdaki dönüşümler kullanılmıştır.

$$v_1 = u_1 + u_2 \quad (3.6)$$

$$v_2 = u_1 - u_2$$

Hata terimlerinin matematiksel ifadesi Denklem 3.7 ve 3.8'deki gibi ifade edilir. r_{\min} pozitif skaler terimi aracın park edeceği noktayı aşmaması için belirlenmiş bir sınır değerdir.

$$e_d = \sqrt{(x_p - x_r)^2 + (y_p - y_r)^2} - r_{\min} \quad (3.7)$$

$$e_\theta = \arctan\left(\frac{y_p - y_r}{x_p - x_r}\right) - \theta_r \quad (3.8)$$

Hata terimleri kullanılarak robotun dinamik denklemleri aşağıdaki gibi elde edilir.

$$e_d = \frac{x_r(x_p - x_r) + y_r(y_p - y_r)}{\sqrt{(x_p - x_r)^2 + (y_p - y_r)^2}} \quad (3.9)$$

$$e_\theta = \frac{x_r(y_p - y_r) - y_r(x_p - x_r)}{(e_d + r_{\min})^2} - \theta_r \quad (3.10)$$

Denklemler yeniden düzenlenirse;

$$e_d = -u_1 \frac{(x_p - x_r) \cos \theta_r + (y_p - y_r) \sin \theta_r}{e_d + r_{\min}} \quad (3.11)$$

$$e_\theta = u_1 \frac{(y_p - y_r) \cos \theta_r - (x_p - x_r) \sin \theta_r}{(e_d + r_{\min})^2} - \frac{2u_2}{L} \quad (3.12)$$

Denklem 3.11 ve 3.12'de kullanılan ifadelerde trigonometrik ifadeler yerine yazılırsa dinamik denklemler Denklem 3.11 ve 3.12'deki halini alır ve sonuç olarak yerel

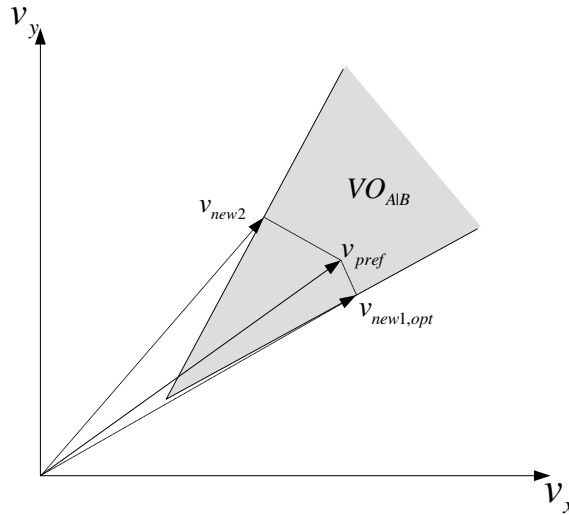
koordinatlar dinamik ifadelerden çıkarılmış olur. Dinamik denklemler, trigonometrik fonksiyonlar, hata ifadeleri ve kontrol sinyallerinden oluşur.

$$e_d = -u_1 \cos e_\theta \quad (3.11)$$

$$e_\theta = \frac{u_1 \sin e_\theta}{e_d + r_{\min}} - \frac{2u_2}{L} \quad (3.12)$$

GELİŞTİRİLEN YÖNTEM

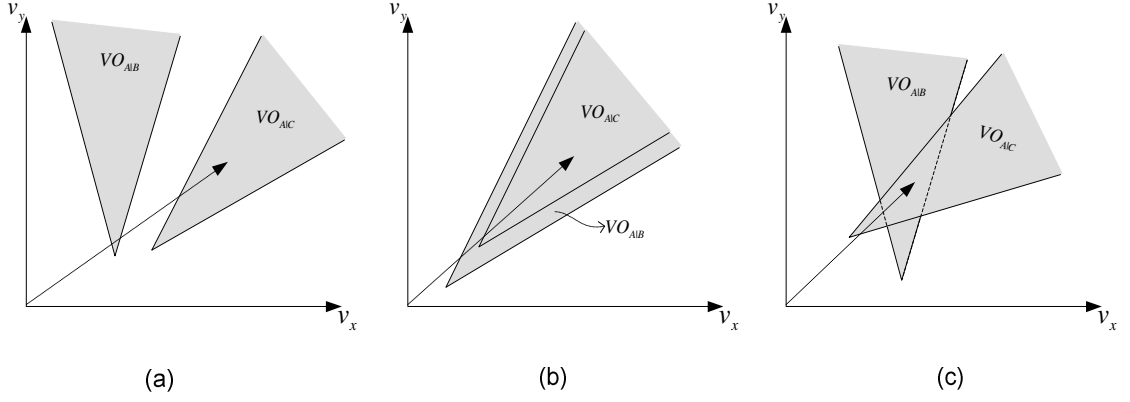
İki araç arasında hız engelleri kullanılarak yapılan bir navigasyon çalışması çok sayıda aracın bulunduğu navigasyon çalışmasına göre işlem yükü açısından daha kolaydır. Bunun sebebi bir araca ait tercih edilen hız vektörünün tek bir hız engeli içinde kaldığı durumda eniyi çözümün hız engelini oluşturan hız koniğinin kollarına dik olarak çizilen vektörün tercih edilen hız vektörüne eklenmesi ile elde ediliyor olmasıdır. Burada çift sonuç elde edilmesine karşın en uygun çözüm en az değişimi sağlayan fark vektörüdür ve bu tek bir sonuç elde edilmesini sağlar. Bu durum Şekil 4.1’de görülmektedir.



Şekil 4.1 Hız vektörünün tek hız engeli içinde kalma durumu

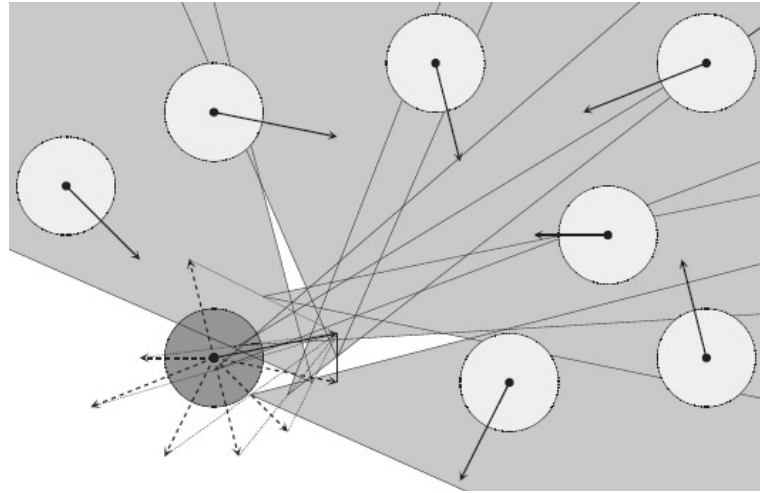
Birden çok aracın bulunduğu durumda ise hız engellerinin bileşiminden ortaya çıkan (genellikle dış bükey) kümeye göre eniyi hız vektörünün aranması söz konusudur. Bu durumda farklı senaryolar ortaya çıkmaktadır. Bunlar hız engellerinin ayrık olması, iç

içerisinde olması veya kesişerek dış bükey hız engelleri bileşkesi oluşturması olarak sıralanabilir. Şekil 4.2’de çift hız engelinin bulunduğu farklı senaryolar ele alınmıştır.



Şekil 4.2 Hız vektörünün çok sayıda hız engeli içinde kalma durumu

Bu şartlar altında eniyi hız vektörünü bulmak zorlaşmaktadır. Eniyi hız vektörü tek hız engeline olduğu gibi hız engelini oluşturan konik kollara dik olarak çizilen vektörün eklenmesi ile elde edilebileceği gibi bazı durumlarda bu çözüm başka bir kesişen hız engelinin içinde kalabilmektedir. Bu durumda hız engellerinin kesişim noktalarının çözüm için kontrol edilmesi gerekmektedir ve birden çok noktanın testine dayanan bir algoritma ile en uygun olan vektör karşılaştırmaların sonucunda elde edilebilmektedir. Çok sayıda hız engelinin olduğu durumlarda bu ağır bir işlem yükü oluşturmaktadır. Hız engellerinin birleşimi sonucunda ortaya çıkan bu durum Şekil 4.3’te görülmektedir.



Şekil 4.3 Çok sayıda hız engelinin olduğu durum

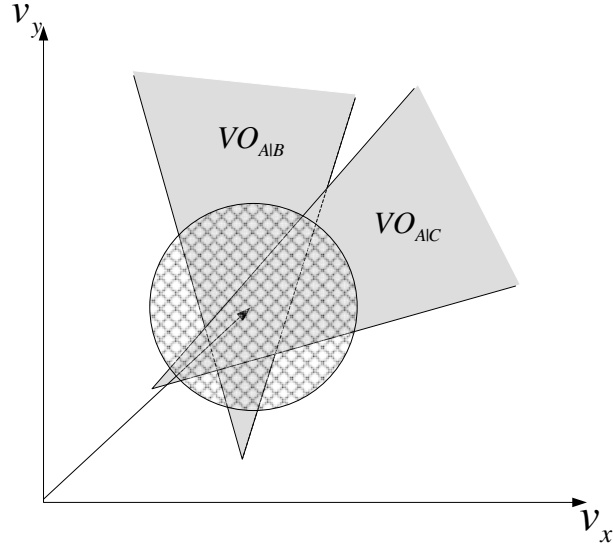
Alternatif bir çözüm olarak tercih edilen hız vektörü v_{pref} etrafında eniyi çözümün aranmasına yönelik bir yöntem uygulanabilir. Sonuç olarak v_{pref} vektörüne en yakın çözüm de çarpışmalardan kaçınma için yeterlidir. Bu amaçla bu vektörün kendisinden başlanarak yakınındaki sonuçların taranması ile tarama aralığına bağlı olarak en uyguna yakın bir çözüm elde etmek mümkündür. Bu amaçla ızgaralama metodundan faydalanılabilir.

Izgaralama metodu ile oluşan geometrik şeklin eşitsizlikler ile elde edilen ifadesinin farklı değerler için taranması mümkündür. Taranacak olan alanın belirlenmesinde ise robotun kinematik olarak gerçekleştirebileceği hız kümelerinden faydalanılır. Bir diğer kistas olarak robotun sürekli olarak hedef noktasına yaklaşması seçilebilir. Bu şartları sağlamak amacı ile $v_{pref} = v_{max} / 2$ kabulünün yapılması taranacak bölgenin yapısını basitleştirmektedir. Bu kabulün yapılması ile robotun seçebileceği hız kümesi v_{pref} merkez olmak üzere v_{pref} 'in skaler değeri yarıçaplı daire şeklini almaktadır. Bu şekilde robotun alabileceği en büyük hız vektörü $v = v_{max}$ genliğine, en düşük genlikli vektör ise $v = 0$ genliğine sahip olacaktır.

Izgaralama metodu ile taranacak olan alanın homojene yakın bir şekilde taranması için kartezyen ızgaralama yerine farklı bir kutupsal ızgaralama metodunun kullanılmasını gerektirmektedir. Bu amaç ile dairesel alan eşit aralıklı halkalara bölünür ve her halka üzerinde eşit yay miktarında ilerlenerek tarama gerçekleştirilir. Örneğin merkezden sınıra kadar olan alan m adet halkaya ayrıldığı takdirde merkeze en yakın olan ilk halka üzerinde n adet nokta taranır, bir sonraki halka $2n$ nokta için taranır ve son halka (dairenin dış sınırı) $m \times n$ adet nokta olarak taranacaktır. Tarama sayısını (TS) belirleyecek olan formül Denklem (4.1)'te verilmiştir.

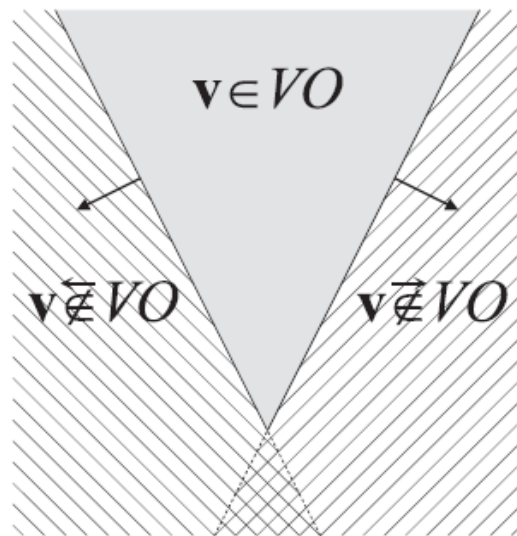
$$TS = \frac{m(m+1)}{2}n \quad (4.1)$$

Izgaralama metodu kullanılarak gerçekleştirilen taramaya dayalı temsili geometrik yapı Şekil 4.4'te gösterilmiştir.



Şekil 4.4 Izgaralama yönteminin geometrik gösterimi

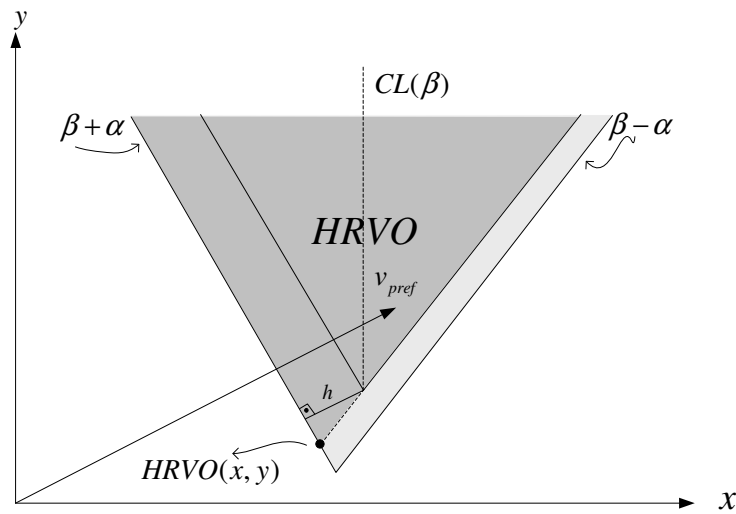
Izgaralama metodunun kullanılabilmesi için öncelikle hız engellerinin belirlenmesi gerekmektedir. Bu amaçla öncelikle hız engellerinin robot konumlarına göre açıları ve başlangıç noktasına ait koordinatlar belirlenir. Daha sonra bu değerlere bağlı olarak oluşan konik 2-boyutlu uzayın içinde oluşturulan iki yarı uzayın birleşiminin dışında kalan küme olarak ifade edilir. Bu şekilde her hız engeli iki adet eşitsizliğin sağlanması ile elde edilen küme haline getirilmiş olur. Şekil 4.5'te birden çok hız engelinin bulunduğu durumlarda ise diğer hız engelleri de benzer şekilde oluşturularak izgaralama yöntemi ile tarama gerçekleştirilir.



Şekil 4.5 Hız engelini eşitsizliklerle ifade edilmesi

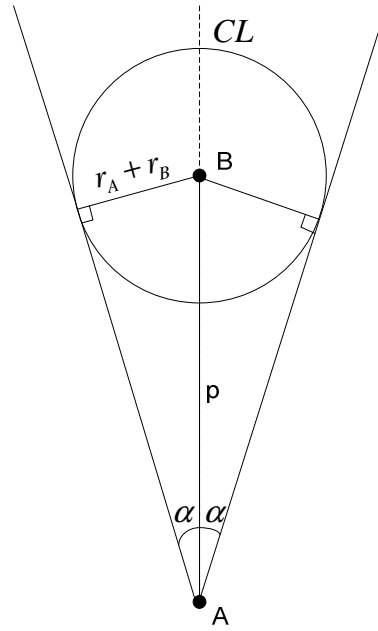
Çarpışmadan kaçınma yönteminin uygulaması amacıyla öncelikle kullanılacak olan hız engellerinin seçimi gerekmektedir. Başarımları sebebi ile uygulama için hibrit hız engelleri seçilmiştir. Hibrit hız engelleri çok sayıda aracın bulunduğu ortamlarda hedef noktaya ulaşım açısından en uygun çözümlerin elde edilmesini sağlamaktadır. Buna karşın hibrit hız engellerini hesaplanması, klasik hız engelleri ve karşılıklı hız engellerinin hesaplanması kadar basit değildir. Hibrit hız engelleri klasik hız engelleri ve karşılıklı hız engellerinin kombinasyonu olarak elde edilmektedir.

Hibrit hız engellerinin hesaplanmasında algoritma açısından şu yol izlenmiştir: Öncelikle tercih edilen hız vektörünün bu hız engeli içinde olup olmadığı kontrol edilir ve eğer değilse klasik hız engelini kullanılması yeterlidir. Bunun sebebi, tanım gereği karşılıklı hız engelini tercih edilen hız vektörünün klasik hız engeli kümesi içinde olmadığı durumda tanımsız olmasıdır. Aynı şekilde tercih edilen hız vektörü klasik hız engeli içerisinde ise mutlaka bir karşılıklı hız engeli kümesi tanımlanabilir. Tercih edilen hız vektörü klasik hız engeli içinde ise klasik ve karşılıklı hız engelini başlangıç noktası hesaplanır. İki hız engeli için de açılar aynı olduğu gibi hibrit hız engelleri için de bu durum geçerlidir. Hibrit hız engelini başlangıç noktası iki yerde olabilmektedir. Doğru noktanın seçimi ise tercih edilen hız vektörünün karşılıklı hız engelini ortasından geçen çizginin hangi tarafında kaldığına bağlıdır. Şekil 4.6'da temsili bir hız engeli görülmektedir.



Şekil 4.6 HRVO kümesinin hesaplanması

Şekil 4.6'da görüldüğü gibi tercih edilen hız vektörünün klasik hız engeli (VO) içinde kalması durumunda mutlaka bir karşılıklı hız engeli (RVO) oluşmaktadır ve bu klasik hız engeli içinde kalmaktadır. Hibrit hız engeli (HRVO) ise hız vektörünün RVO'nun orta çizgisinin (CL) kaldığı tarafına bağlı olarak iki şekilde belirlenmektedir. Şekil 4.6'da hız vektörü CL çizgisinin sağ tarafında kaldığı için HRVO, RVO'nun sol tarafa doğru VO koniğinin sol koluna kadar genişletilmesi ile elde edilmiştir. HRVO'nun kullanılabilmesi için öncelikle HRVO koordinatlarının hesaplanması gerekmektedir ($HRVO(x, y)$). Burada β açısı hız engellerinin paralel olması sebebiyle VO, RVO ve HRVO için orta çizginin açısını ifade etmektedir. α açısı ise hız engellerinin oluşmasında rol oynayan robotların birbirine olan uzaklığına ve çaplarına bağlı olarak elde edilen konik açıdır, aynı şekilde paralellik sebebiyle VO, RVO ve HRVO için bu açı eşittir. Bu durumda konik açı toplamda 2α olarak ifade edilebilir. Şekil 4.7'de bu açının nasıl oluştuğu görülmektedir.



Şekil 4.7 Konik açısının gösterimi

Şekil 4.7'de görüldüğü gibi A ve B robotlarından A noktaya indirgenmiş buna karşın B robotunun iki robotun çapları toplamına eşit olacak kadar büyütülmüştür. Bu durumda iki robotun arasındaki merkezler uzaklığı p ile ifade edilmiştir. Buna dayanarak α açısı trigonometrik olarak Denklem 4.2'de ifade edilmiştir.

$$\alpha = a \sin \frac{r_a + r_b}{p} \quad (4.2)$$

Bunun dışında β açısı ise benzer şekilde oluşan hız engelinin hız koordinat sisteminde ifade edilirken orta çizgisini (CL) tanımlamaktadır. Hız engelinin başlangıç noktasının belirlenmesi için öncelikle karşılıklı hız engelinin ilgili kolunun doğru denklemi ile ifade edilmesi gerekmektedir. Hız vektörünün Şekil 4.6'da olduğu gibi orta çizginin sağ tarafında kaldığı düşünülerek Denklem 4.3'te VO 'nun ilgili koluna ait doğru denklemi, Denklem 4.4'te ise RVO ' nun ilgili koluna ait doğru denklemi verilmiştir.

$$\tan(\beta + \alpha) = \frac{v_y - VO_y}{v_x - VO_x} \quad (4.3)$$

$$\tan(\beta + \alpha) = \frac{v_y - RVO_y}{v_x - RVO_x} \quad (4.4)$$

Burada VO_x ve VO_y , VO koniğinin başlangıç noktasını ifade etmektedir. VO ' nun tanımı gereği bu nokta hız koordinat sisteminde v_B vektörünü ifade eder. RVO_x ve RVO_y ise RVO koniğinin başlangıç noktasını ifade etmektedir. Bu nokta ise v_A ve v_B vektörlerinin ortalamasını ifade eder. Buradan hareketle Denklem 4.5'te ve Denklem 4.6'daki şekilde ifade edilebilirler.

$$VO_x = v_{Bx}, \quad VO_y = v_{By} \quad (4.5)$$

$$RVO_x = \frac{v_{Ax} + v_{Bx}}{2}, \quad RVO_y = \frac{v_{Ay} + v_{By}}{2} \quad (4.6)$$

Bu denklemler ile iki kol arasındaki dik uzaklık hesaplanır. Bu uzaklık tanım gereği pozitif bir değerdir ve Denklem 4.7'de de en son hali verilmiştir.

$$h_1 = \left| \left(\frac{v_{Ay} - v_{By}}{2} \right) \cos(\beta + \alpha) - \left(\frac{v_{Ax} - v_{Bx}}{2} \right) \sin(\beta + \alpha) \right| \quad (4.7)$$

Daha sonra bu uzaklık karşılıklı hız engelinin diğer koluna ait olan açı değerleri yardımıyla trigonometrik denklemler kullanılarak karşılıklı hız engelinin başlangıç noktası ile hibrit hız engelinin başlangıç noktası arasındaki farkın kartezyen değerler cinsinden hesaplanmasını sağlar. Denklem 4.7'de ve Denklem 4.8'de $HRVO$ koniğinin

başlangıç noktasına ait koordinatlar RVO koniğinin başlangıç noktası koordinatları cinsinden ifade edilmiştir.

$$HRVO_x = RVO_x - \frac{h_1}{\sin(2\alpha)} \cos(\beta - \alpha) \quad (4.8)$$

$$HRVO_y = RVO_y - \frac{h_1}{\sin(2\alpha)} \sin(\beta - \alpha) \quad (4.9)$$

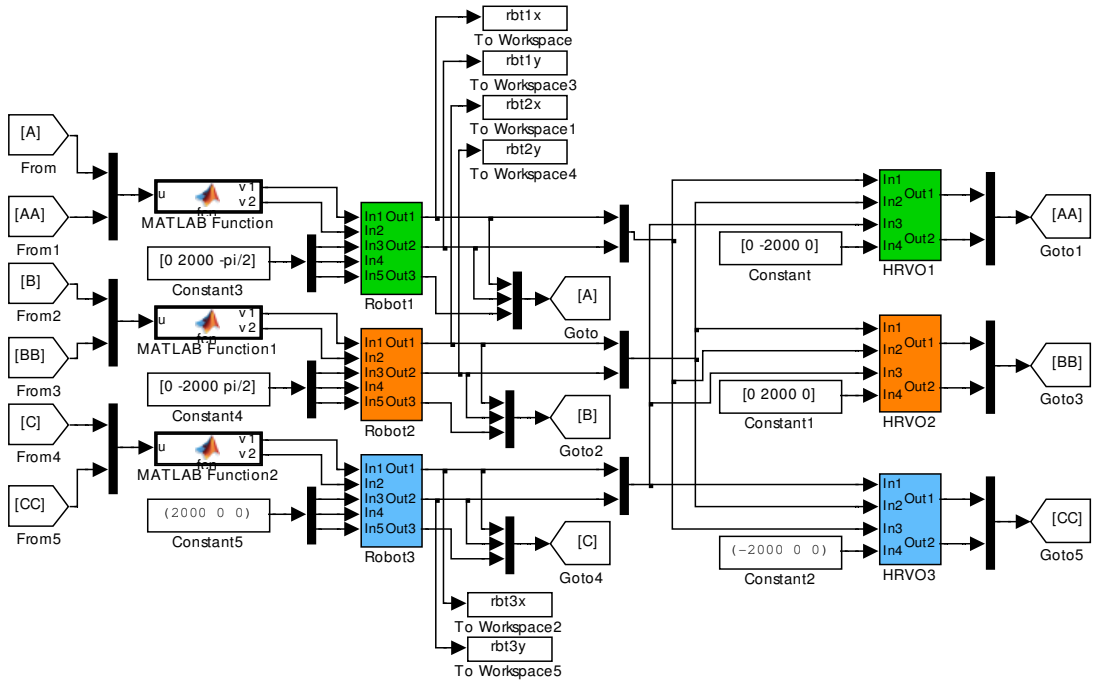
Tercih edilen hız vektörünün RVO orta çizgisinin sol tarafında kalması durumunda ise konik kolları değişeceğinden aynı yol izlenerek Denklem 4.10, 4.11 ve 4.12'deki gibi elde edilir.

$$h_2 = \left| \left(\frac{v_{Ay} - v_{By}}{2} \right) \cos(\beta - \alpha) - \left(\frac{v_{Av} - v_{Bx}}{2} \right) \sin(\beta - \alpha) \right| \quad (4.10)$$

$$HRVO_x = RVO_x - \frac{h_2}{\sin(2\alpha)} \cos(\beta + \alpha) \quad (4.11)$$

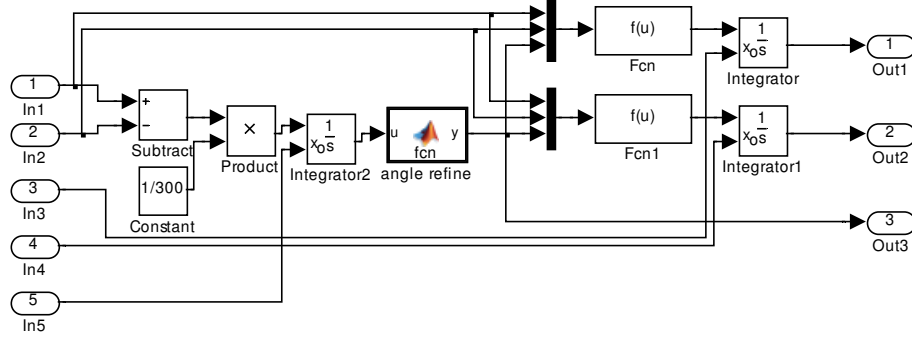
$$HRVO_y = RVO_y - \frac{h_2}{\sin(2\alpha)} \sin(\beta + \alpha) \quad (4.12)$$

Bu sonuçlar ışığında ızgaralama metodunun işlevselliği MATLAB Simulink platformunda benzetimi gerçekleştirilmiştir. Sisteme blok diyagramı Şekil 4.8'de görülmektedir.



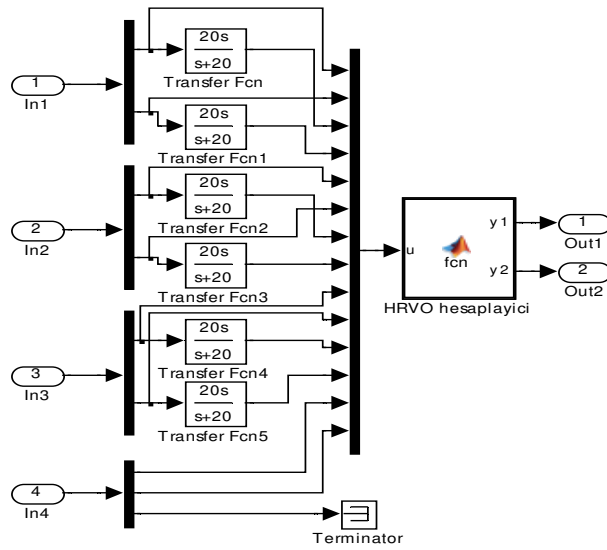
Şekil 4.8 Benzetim blok diyagramı

Burada üç aracın bulunduğu bir çarpışmadan kaçınma problemi oluşturulmuştur, araç hedef noktaları ve başlangıç değerleri sabitler şeklinde ifade edilmiştir. Ayrıca karmaşıklığı önlemek amacı ile robot modelleri alt sistem olarak ifade edilmiştir. Robot modelini oluşturan alt yapı Şekil 4.9'da görülmektedir.



Şekil 4.9 Robot blok diyagramı

Bunun dışında robot bloklarının üreteceği güncel konum bilgileri hibrit hız engellerinin ızgaralama metodu ile hesaplandığı başka bir alt bloğa girilmiştir. Ek olarak araç hızları konum bilgilerinin filtrelenmiş bir türev işleminden geçirilmesiyle elde edilir. Bütün konum, hız bilgileri ve hedef noktaları oluşturulan matlab fonksiyonuna girilerek, çarpışmalardan kaçınılması için izlenmesi gereken yeni hız vektörü elde edilmiştir. Şekil 4.10'da hız engellerinin uygulandığı alt sistem yapısı görülmektedir.



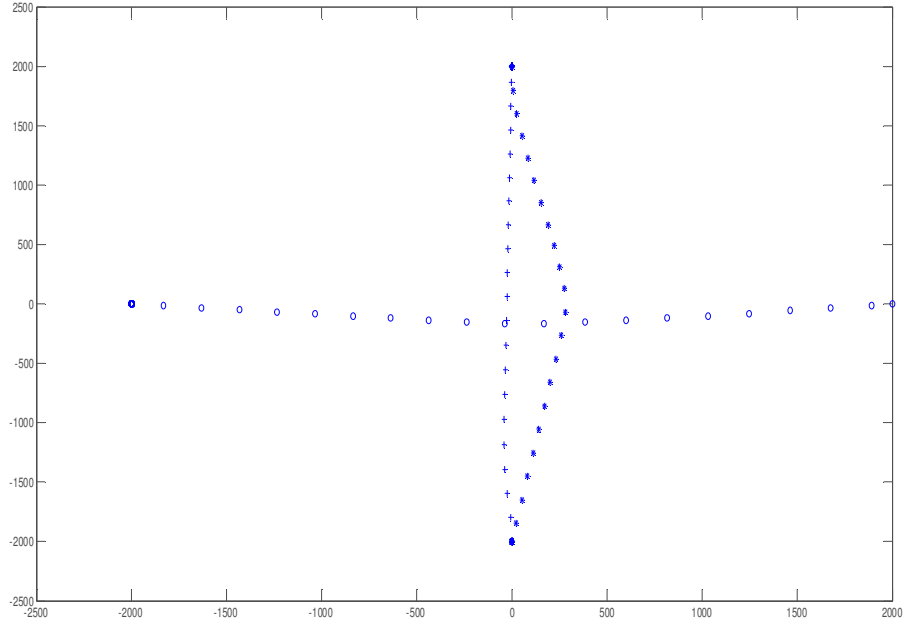
Şekil 4.10 HRVO hesaplayıcı blok diyagramı

Benzetime ait parametreler Çizelge 4.1’de verilmiştir.

Çizelge 4.1 Benzetime ait parametreler

	Başlangıç noktası	Hedef noktası
Robot 1	$x= 0, y=2000$	$x= 0, y=-2000$
Robot 2	$x=0, y=-2000$	$x= 0, y=2000$
Robot 3	$x=2000, y=0$	$x=0, y=-2000$

Benzetim sonucunda elde edilen araçlara ait navigasyon haritası Şekil 4.11’deki gibidir.



Şekil 4.11 Navigasyon haritası

Elde edilen sonuçlar araçların çarpışmadan kaçınarak hedef noktalarına gittiklerini göstermektedir. Benzetim çalışmasında park problemi ele alınmamıştır, bu nedenle araçlar hedef noktalarına vardıklarında kararsız bir tutum sergilemektedir. Araçların yönelim açıları irdelenmediği için bu durum görülememektedir.

5.1 Giriş

Uygulamanın gerçekleştirilebilmesi için Quanser firması tarafından üretilen QBot robotlarından faydalanılmıştır. QBot benzer akademik uygulamalarda sıklıkla kullanılan iRobot firması tarafından üretilen iRobot Create platformunun üzerine inşa edilmiştir. iRobot Create platformu diferansiyel sürüş tekniğini kullanmaktadır. Araç iki adet hareketi sağlayan DC motor tahrikli tekerlek ve araç dengesini sağlayan iki adet serbest dönebilen tekerlek ile donatılmıştır. Şekil 5.1’de uygulamada kullanılan Qbot mobil robotun resmi verilmiştir.



Şekil 5.1 Qbot mobil robotu

iRobot Create kendi içinde kapalı çevrim olarak tekerlek hızlarını kodlayıcı yardımı ile denetlemektedir. QBot ise bu robotların üzerine gumstix gömülü bilgisayar modüllerinin eklenmesi ile oluşturulmuştur. Modüller üzerinde Linux tabanlı işletim sistemi çalışmaktadır. Ayrıca modüller kablosuz iletişim olanağı sağlayarak robot ile veri alışverişinde kolaylık sağlamaktadır.

Quanser firmasının Quarc yazılımının kullanımı ile MATLAB Simulink yazılımı üzerinde oluşturulmuş olan kontrol algoritmaları/blokları derlenerek gerçek zamanlı olarak QBot robotları üzerinde koşturulabilmektedir. Uygulamanın gerçeğe yakınlığı açısından bu durum büyük önem arz etmektedir. Bu sayede robotlar kendi hesaplamalarına dayanarak çarpışmalardan kaçınmaya yönelik manevralar yapabilmektedir.

5.2 Görüntü Geribeslemesine Dayalı Araç Konum Tespiti

Uygulamalarda araç konumunun belirlenen koordinat sistemine göre konum ve açı bilgilerinin belirlenmesi gerekmektedir. Bu amaç ile araçların veya hedef noktaların örüntüler yardımıyla işaretlenmesi gerekmektedir. Araç park problemi ile ilgili uygulamalarda hedef noktanın bir örüntüden ibaret olması benimsenmiş ve araç üzerine monte edilen bir kameradan alınan görüntü bilgisine dayalı hata miktarlarını tespit eden bir sistem oluşturulmuştur. Çok sayıda aracın bulunduğu çarpışmadan kaçınma uygulaması için ise örüntüler araç üzerine yerleştirilmiştir. Örüntülerin koordinat verileri Arma [33] isimli, OpenCV tabanlı ve açık kaynak kodu sunulan bir kütüphane yardımı ile elde edilmektedir. Arma bilgisayar üzerinde çalışarak ilgili kamera görüntüsü içinde bulunan örüntüleri algılar ve koordinat bilgilerini elde eder, asıl amacı arttırılmış gerçeklik uygulamalarıdır. Kaynak kodlarına yapılan eklemeler ile oluşturulan programın koordinat bilgilerini robotlara iletebilmesi için TCP/IP haberleşmesinden faydalanılmıştır. Program içine yerleştirilen iş parçaları (threads) ile her robot için TCP/IP soketleri oluşturulmuştur. Böylece robotlar kablosuz iletişim ağı üzerinden görüntü işleme yazılımına bağlanarak koordinat bilgilerini elde eder. Bunun dışında veri aktarım miktarı ve hız oranı uygulamanın gereksinimlerine göre ayarlanmalıdır.

5.3 Park Problemi Uygulaması

Park probleminin çözümlenmesi amacı ile öncelikle örüntü takibine dayalı bir kontrol sistemi oluşturularak robotlar üzerinde uygulanmıştır. Daha önceki bölümde elde edilmiş olan dinamik denklemler (5.1) ve (5.2)'de verilmiştir.

$$e_d = -u_1 \cos e_\theta \quad (5.1)$$

$$e_\theta = \frac{u_1 \sin e_\theta}{e_d + r_{\min}} - \frac{2u_2}{L} \quad (5.2)$$

Bu denklemlerin doğrusal olmayan yapısı sebebi ile geri besleme doğrusallaştırması yapılarak dinamik denklemlerin üstel kararlı hale getirilmesi hedeflenmiştir. Bu amaçla u_1 ve u_2 için (5.3), (5.4) verilen denklemler seçilmiştir.

$$u_1 = \frac{\lambda_1 e_d}{\cos e_d} \quad (5.3)$$

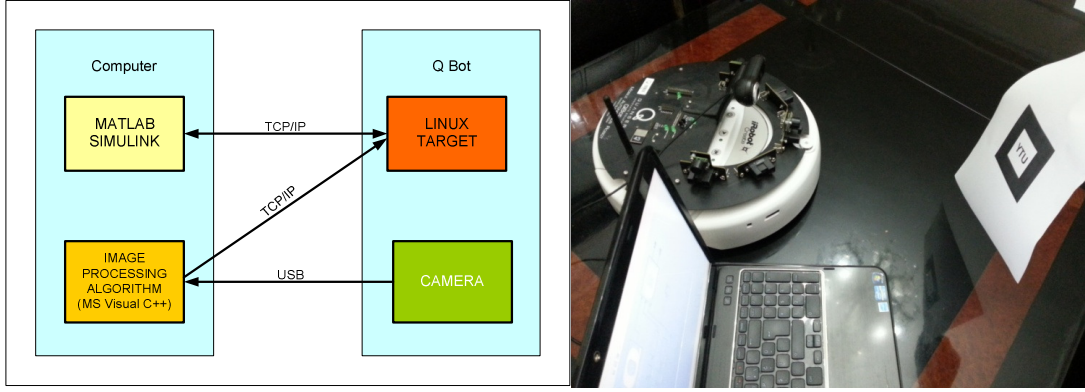
$$u_2 = \frac{\lambda_2 L e_\theta}{2} + \frac{\lambda_1 L e_d \tan(e_\theta)}{2(e_\theta + r_{\min})} \quad (5.4)$$

Bu seçimin yapılması ile dinamik denklemler aşağıda gösterilen basitleştirilmiş (5.5) ve (5.6) denklemleri ile gösterilir.

$$e_d = -\lambda_1 e_d \quad (5.5)$$

$$e_\theta = -\lambda_2 e_\theta \quad (5.6)$$

Robot konumunun elde edilmesi için daha önceki bölümlerde bahsedilen görüntü işleme programından faydalanılmıştır. Robot üzerine yerleştirilen bir kamera sayesinde robot için problem örüntünün kamera yatay ekseninde ortalanması (açı hatası) ve hizalanması (uzaklık hatası) problemi haline gelmiştir. Uygulama sisteminin yapısını gösteren temsili diyagram Şekil 5.2'de görülmektedir.



Şekil 5.2 Uygulama sisteminin yapısı

Sonraki adımda MATLAB Simulink kullanılarak gerçek zamanlı uygulama robot üzerinde uygulanmıştır. Çizelge 5.1’de uygulamaya ait parametreler görülmektedir.

Çizelge 5.1 Hedef takip uygulaması parametreleri

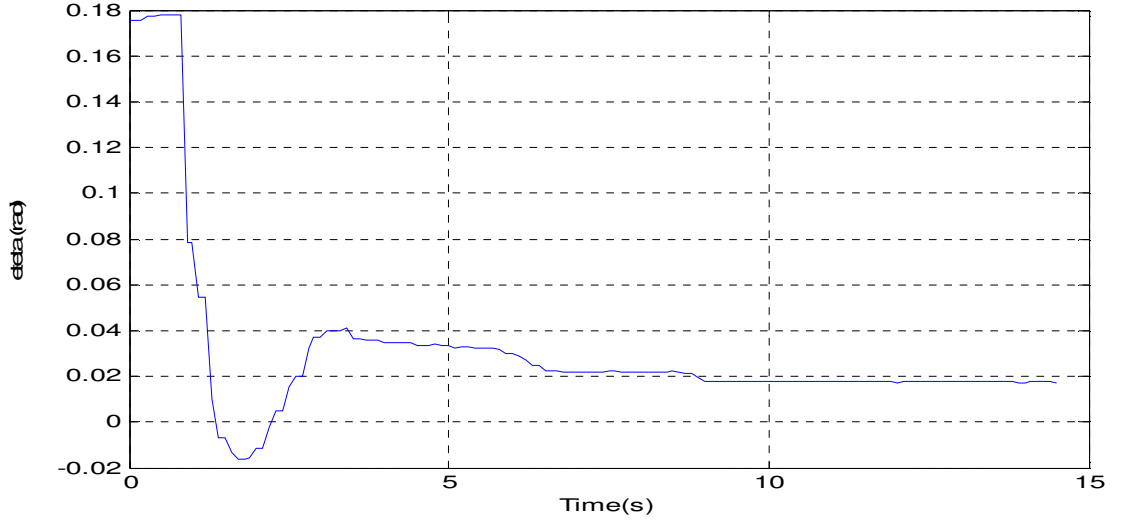
λ_1	λ_2	L	r_{min}	$e_d(0)$	$e_\theta(0)$
1	1.2	250mm	400mm	550mm	0.18

Uygulama için parametrelerin atanması ile elde edilen kontrol sinyallerinin son hali Denklem (5.7) ve (5.8)’de görülmektedir.

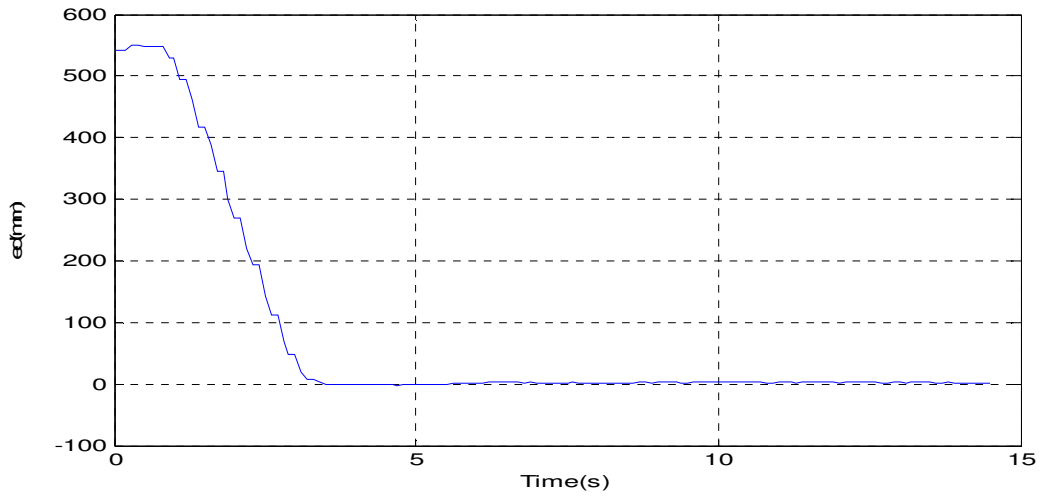
$$u_1 = \frac{e_d}{\cos e_\theta} \quad (5.7)$$

$$u_2 = 150e_\theta + \frac{125e_d \tan e_\theta}{e_d + 400} \quad (5.8)$$

Bu parametrelerin kullanımı ile elde edilen sonuçlar gerçek zamanlı olarak Şekil 5.3 ve Şekil 5.4’te verilmiştir.



Şekil 5.3 Açık hatasının zamana göre değişimi



Şekil 5.4 Konum hatasının zamana göre değişimi

5.4 Çarpışmadan Kaçınma Uygulaması

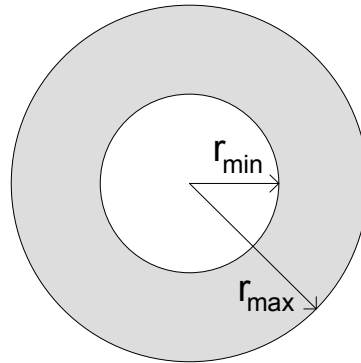
Çarpışmadan kaçınma uygulamasında daha önceki hedef takip probleminde olduğu gibi görüntü geri beslemesinden faydalanılmıştır. Bu amaçla robotları kuş bakışı gören bir kamera kullanılmıştır. Daha önce önerilen yöntem ek olarak araç park ve çarpışmadan kaçınma adımları QBot'ların simetrik yapısı sebebi ile robotlar çift yönlü hareket ettirilmiştir. Bunun anlamı robotların açılmalari $+\pi/2$ 'den fazla ve $-\pi/2$ 'den az olması durumunda robotun tekerleklerinin ters yöne hareket ederek robotun geriye doğru gitmesidir. Bu amaçla açılmalari π kadar ötelenir. Tekerlek hızları yer ve

işaret değiştirir. Böylece araçların karşılaşabileceği maksimum açı hatası $\pi/2$ radyan olmaktadır.

Burada oluşabilecek bir sorun araç park problemi için oluşturulan denklemlerin tanım bölgesidir. Park uygulamasında kullanılan Denklem 5.7 ve 5.8 ele alındığında bu denklemlerin $\pm\pi/2$ için tanımsız olması söz konusudur. Park uygulamasında böyle bir problem ile karşılaşılmasının sebebi kamera açısının bu noktaları göremeyecek kadar dar olmasıdır. Bu problemi çözmek amacı ile bahsedilen tanımsız değerlere yakın noktalarda tanımlı olacak şekilde değer ataması yapılmıştır (fonksiyon sınırlandırılmıştır).

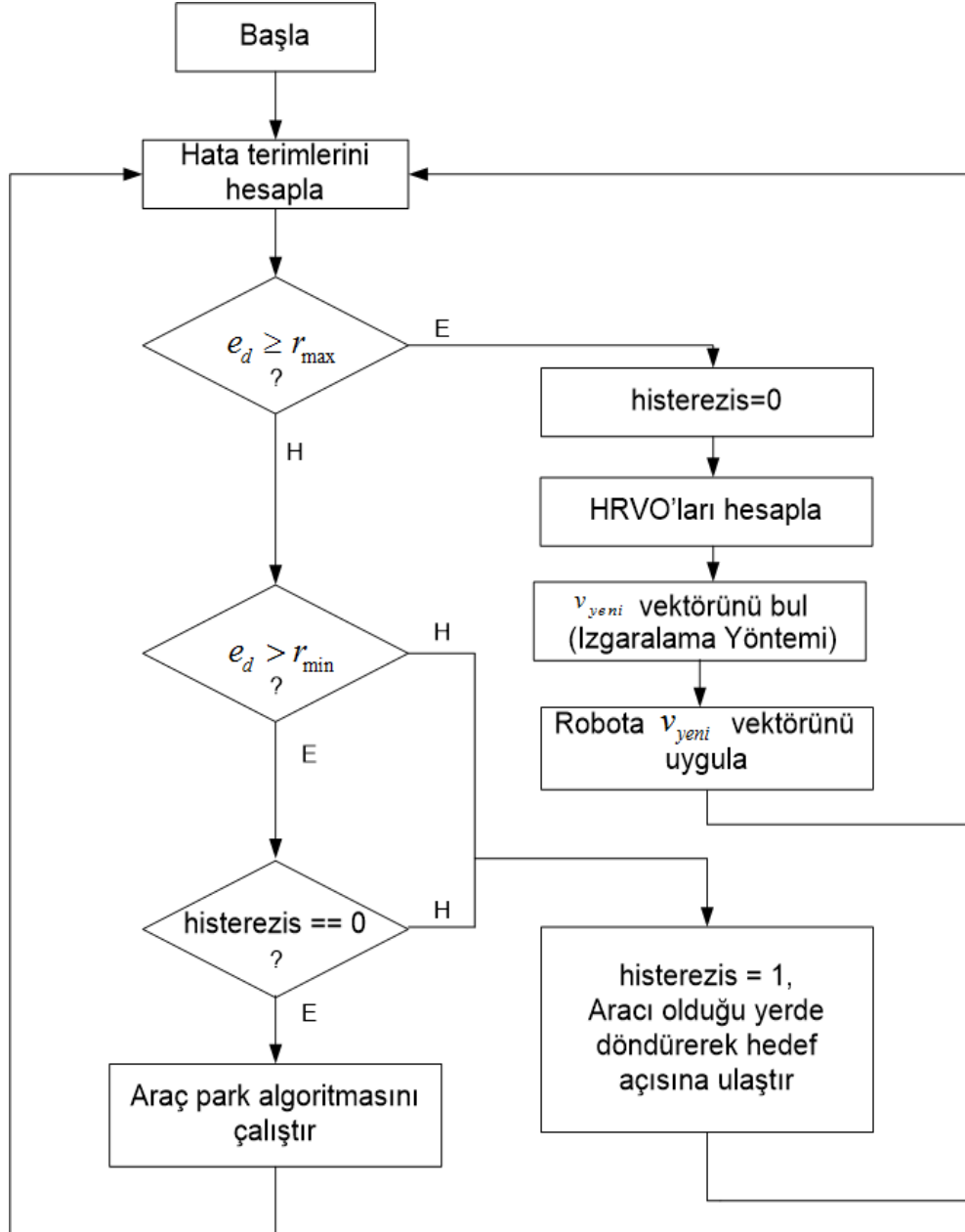
Uygulamada ortaya çıkan bir diğer problem de araçların tekerlek hızlarının sınırlı olmasıdır. Bu durumda elde edilen kontrol sinyallerinin öngörüldüğü gibi uygulanamaması sorunu ortaya çıkar ve bu sistemi kararsızlığa sürükleyebilmektedir. Bu sorunu aşmak için hata değerlerinde limitleme yapılmaktadır. Örneğin çok uzaktaki bir hedef noktaya gidilmesi büyük bir e_d hatası doğurmakta bu da sonuç olarak u_1 sinyalini uygulanamaz değerlere ulaştırmaktadır.

Uygulamada araçların görevi tamamlanana kadar üç farklı kontrol algoritması kullanılır. Bunlardan ilki aracın hedef noktaya çarpışmalardan kaçarak gitmesini sağlayan hibrit hız engellerine ve ızgaralama yöntemine dayanan yeni vektör belirleme sistemidir. Eğer araç hedef noktaya çok yakınsa birinci algoritma yerine ikinci algoritma olarak park etme algoritması devreye girer. Araç park noktasının çevresi iki adet çember ile sınırlandırılmıştır. Şekil 5.5'te görüldüğü gibi bunlardan biri r_{min} diğeri ise r_{max} tır.



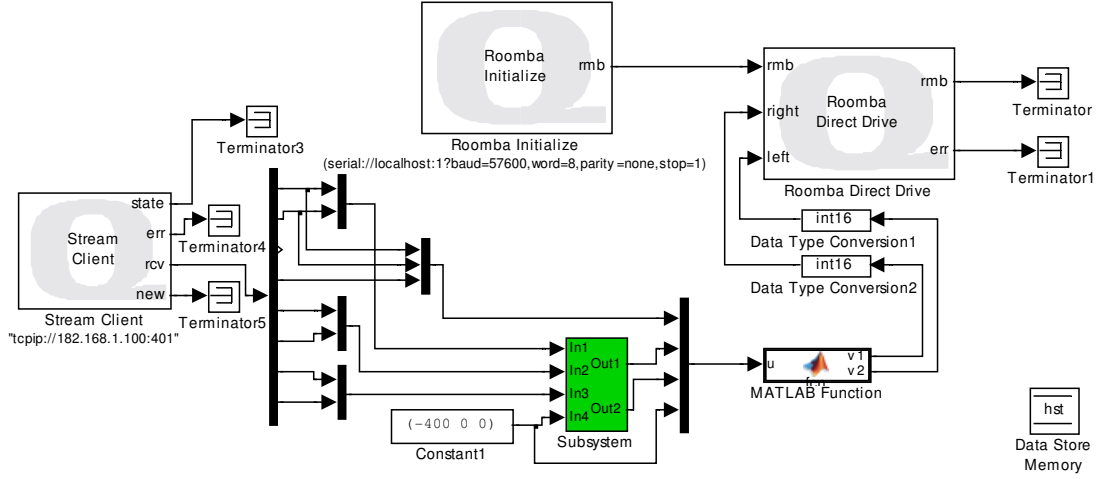
Şekil 5.5 Park için oluşturulan çemberler

Araç dışarıdan r_{max} çemberi içine girdiğinde hız engellerinin hesaplamasını bırakarak park algoritmasını devreye sokar ve r_{min} çemberini hedef olarak alır. r_{min} çemberine ulaştığında araç aç hedef aç değerine ulaşana kadar kendi çevresinde döner ve durur. Burada oluşabilecek kararsız durumların önüne geçmek için iki çember arası geçişlerde histerezis uygulanmıştır. Tüm sisteme ait algoritmaya ait akış diyagramı Şekil 5.6'da verilmiştir.



Şekil 5.6 Algoritma akış diyagramı

Bu yazılım için oluşturulan blok diyagramlar Şekil 5.7'de verilmiştir.



Şekil 5.7 Tek robota ait blok diyagramı

Uygulamada araçların görevi herhangi farklı noktalardan yola çıkarak x koordinat ekseninin üzerinde yan yana park edilmesidir. Araçlara verilen hedef bilgileri Çizelge 5.2’ de görülmektedir.

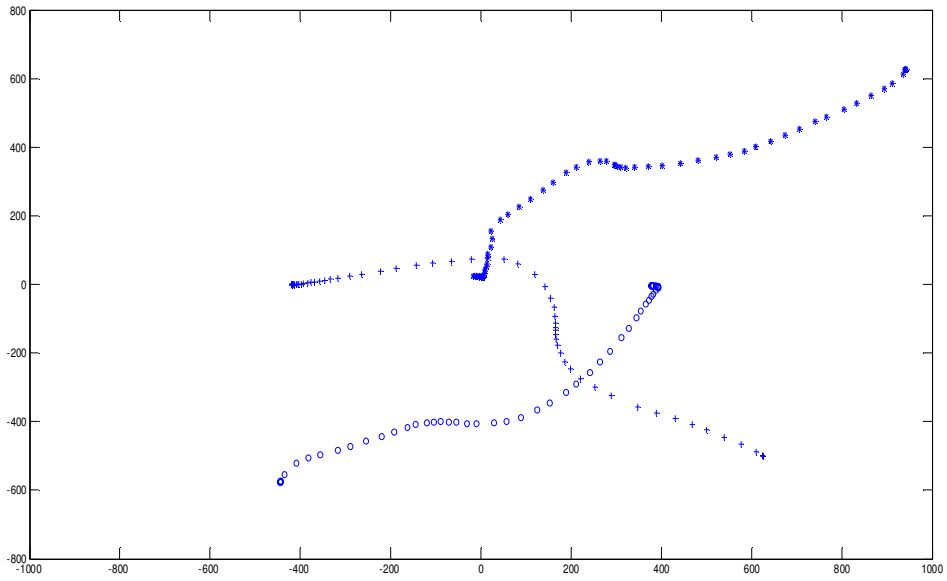
Çizelge 5.2 Araçlara verilen hedef nokta bilgisi

	Hedef noktası
Robot 1	x= -400, y=0
Robot 2	x= 0, y=0
Robot 3	x=400, y=0

Uygulamada kullanılan araçların resmi Şekil 5.8’ da görülmektedir. Uygulama süresince araçların izlediği yol haritası ise Şekil 5.9’da verilmiştir.



Şekil 5.8 Uygulamada kullanılan araçlar



Şekil 5.9 Uygulama süresince araçların izlediği yol haritaları

SONUÇLAR VE ÖNERİLER

Yapılan çalışmada işlem yükü azaltılıp gömülü olarak robotların üzerinde algoritmanın koşturulması sağlanmıştır. Izgaralama yöntemi geliştirilerek bu işlem yükünün daha da azaltılması mümkün görülmektedir. Benzer şekilde araçların park problemine de yeni bir yaklaşım getirilmiştir.

Robotların üzerindeki işlemcinin kapasitesi sebebi ile araçların azami hızları sınırlandırılmıştır. Tarama zamanının azaltılması için algoritmanın hafifletilmesi veya taranan nokta sayısının azaltılması gerekmektedir.

Uygulamada zaman gecikmeleri de sınırlayıcı bir faktördür. Görüntü işleme ve koordinatları belirlemek için oluşturulan algoritma bilgisayar üzerinde çalıştırılmaktadır. Buna bağlı olarak örüntülerin tespiti ve TCP/IP haberleşmesi ihmal edilebilir gecikmeler oluşturmaktadır.

QBot robotları uygulamanın gerçek zamanlı olarak yapılmasına olanak tanımıştır. Literatürdeki çalışmalarda genelde tekerlek hızları hesaplanarak gönderilmektedir. Bu durumda gerçek anlamda bir uygulama yapılamamaktadır.

QBot robotlarının kullanımında gözlemlenen bir kusur; aracın çok düşük hız değerlerini tekerleklere uygulamamasıdır. Bu durum ihmal edilebilir seviyede olsa da sürekli hal hatalarının oluşmasına sebep olmaktadır. Ek olarak robotlara sadece tekerlek hızlarının giriş olarak uygulanıyor olması robot üzerinde tork kontrolüne dayalı algoritmaların koşturulmasını olanaksız kılmaktadır.

Robotlar üzerinde ayrıca dıř ortamla ilgili bilgi almasını sađlayan algılayıcılar mevcuttur, fakat uygulama esnasında ihtiyaç duyulan seviyede bilgi sađlamadıkları için bu algılayıcılardan faydalanılmamıştır.

KAYNAKLAR

- [1] Fiorini, P., ve Shiller, Z., (1998). "Motion Planning in Dynamic Environments Using Velocity Obstacles", *The International Journal of Robotics Research*, 17(7): 760-772.
- [2] Prassler, E., Bank, D., ve Kluge, B., (2001). "Motion Coordination Between a Human and a Robotic Wheelchair", In *Robot and Human Interactive Communication*, 10th IEEE International Workshop, 412-417.
- [3] Abe, Y., ve Yoshiki, M., (2001). "Collision Avoidance Method for Multiple Autonomous Mobile Agents by Implicit Cooperation", In *Intelligent Robots and Systems*, IEEE/RSJ International Conference, 3: (1207-1212).
- [4] Large, F., Sekhavat, S., Shiller, Z., ve Laugier, C., (2002). "Towards Real-Time Global Motion Planning in a Dynamic Environment Using the NLVO Concept", In *Intelligent Robots and Systems*, IEEE/RSJ International Conference, 1: (607-612).
- [5] Large, F., Sckhavat, S., Shiller, Z., ve Laugier, C., (2002). "Using Non-linear Velocity Obstacles to Plan Motions in a Dynamic Environment", In *Control, Automation, Robotics and Vision*, ICARCV 2002. 7th International Conference, 2: (734-739).
- [6] Myers, T. J., Noel, T., Parent, M., ve Vlacic, L., (2005). "Autonomous Motion of a Driverless Vehicle Operating Among Dynamic Obstacles". In *Decision and Control and 2005 European Control Conference*, 44th IEEE Conference, 5071-5076.
- [7] Megda, P. T., Esteves, B. A., ve Becker, M., (2011). "Determining Forbidden Steering Directions for a Passenger Car in Urban Environments Based on the Velocity Obstacle Approach and Use of Trackers", In *Robotics Symposium*, IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC), 1-6.
- [8] Claes, D., Hennes, D., Tuyls, K., ve Meeussen, W., (2012). "Collision Avoidance Under Bounded Localization Uncertainty", In *Intelligent Robots and Systems (IROS)*, IEEE/RSJ International Conference, 1192-1198.

- [9] Van den Berg, J., Lin, M., ve Manocha, D., (2008). "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation", In Robotics and Automation, ICRA 2008. IEEE International Conference, 1928-1935.
- [10] Van Den Berg, J., Guy, S. J., Lin, M., ve Manocha, D., (2011). "Reciprocal n-body Collision Avoidance", In Robotics Research, 3-19.
- [11] Alonso-Mora, J., Breitenmoser, A., Beardsley, P., ve Siegwart, R., (2012). "Reciprocal Collision Avoidance for Multiple Car-Like Robots", In Robotics and Automation (ICRA), IEEE International Conference, 360-366.
- [12] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., ve Siegwart, R., (2013). "Optimal Reciprocal Collision Avoidance for Multiple Non-holonomic Robots", In Distributed Autonomous Robotic Systems, 203-216.
- [13] Jiawei, H., Xunyu, Z., ve Xiafu, P., (2012). "Velocity Variation Space Based Multiple Agents Motion Planning for Mobile Robot", In Control Conference (CCC), 4820-4824.
- [14] Dubey, R., Pradhan, N., Krishna, K. M., ve Chowdhury, S. R., (2012). "Field Programmable Gate Array (FPGA) Based Collision Avoidance Using Acceleration Velocity Obstacles". In Robotics and Biomimetics (ROBIO), IEEE International Conference, 2333-2338.
- [15] Van den Berg, J., Snape, J., Guy, S. J., ve Manocha, D., (2011). "Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles", In Robotics and Automation (ICRA), IEEE International Conference, 3475-3482.
- [16] Snape, J., van den Berg, J., Guy, S. J., ve Manocha, D., (2011). "The Hybrid Reciprocal Velocity Obstacle. Robotics, IEEE Transactions on Robotics, 27:696-706.
- [17] Kanayama, Y., Kimura, Y., Miyazaki, F., ve Noguchi, T., (1990) "A Stable Tracking Control Method for an Autonomous Mobile Robot", IEEE International Conference on Robotics and Automation, 384-389.
- [18] Fierro, R. ve Lewis, F. L., (1995). "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In Decision and Control", 34th IEEE Conference, 4: 3805-3810.
- [19] Park, K. H., Cho, S. B. ve Lee, Y. W., (2001). "Optimal Tracking Control of a Nonholonomic Mobile Robot", ISIE 2001, IEEE International Symposium, 3: 2073-2076.
- [20] Tsai, C. Y. ve Song, K. T., (2005). "Visual Tracking Control of a Mobile Robot Using a New Method in Image Plane", 12th IEEE Conference Advanced Robotics, 540-545.
- [21] Scaglia, G., Rosales, A., Quintero, L., Mut, V., ve Agarwal, R., (2010). "A Linear-Interpolation-Based Controller Design for Trajectory Tracking of Mobile Robots", Control Engineering Practice, 18(3): 318-329.

- [22] Tsai, C. Y., ve Song, K. T., (2009). "Dynamic Visual Tracking Control of a Mobile Robot with Image Noise and Occlusion Robustness", *Image and Vision Computing*, 27(8):1007-1022.
- [23] Yang, J. M., ve Kim, J. H., (1999). "Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots", *Robotics and Automation, IEEE Transactions*, 15(3): 578-587.
- [24] Yuan, G., Yang, S. X., ve Mittal, G. S., (2001). "Tracking Control of a Mobile Robot Using a Neural Dynamics Based Approach", In *Robotics and Automation, ICRA. IEEE International Conference*, 1: 163-168.
- [25] Treptow, A., Cielniak, G., ve Duckett, T., (2006). "Real-time People Tracking for Mobile Robots Using Thermal Vision". *Robotics and Autonomous Systems*, 54(9): 729-739.
- [26] Hada, Y., Hemeldan, E., Takase, K., ve Gakuhari, H., (2003). "Trajectory Tracking Control of a Nonholonomic Mobile Robot Using iGPS and Odometry", In *Multisensor Fusion and Integration for Intelligent Systems, MFI2003, IEEE International Conference*, 51-57.
- [27] Zhang, H., ve Ostrowski, J. P., (2002). "Visual Motion Planning for Mobile Robots". *Robotics and Automation, IEEE Transactions*, 18(2):199-208.
- [28] Rui, L., Zhijiang, D., ve Lining, S., (2009). "Moving Object Tracking Based on Mobile Robot Vision". In *Mechatronics and Automation, ICMA, International Conference*, 3625-3630.
- [29] Coulaud, J. B., Campion, G., Bastin, G., ve De Wan, M., (2006). "Stability Analysis of a Vision-Based C design for an Autonomous Mobile Robot", *Robotics, IEEE Transactions*, 22(5): 1062-1069.
- [30] Freda, L., ve Oriolo, G., (2007). "Vision-Based Interception of a Moving Target with a Nonholonomic Mobile Robot", *Robotics and Autonomous Systems*, 55(6):419-432.
- [31] Chen, J., Dixon, W. E., Dawson, M., ve McIntyre, M., (2006). "Homography-Based Visual Servo Tracking Control of a Wheeled Mobile Robot". *Robotics, IEEE Transactions*, 22(2):406-415.
- [32] Ma, Y., Kosecka, J., ve Sastry, S. S., (1999). "Vision Guided Navigation for a Nonholonomic Mobile Robot", *Robotics and Automation, IEEE Transactions*, 15(3): 521-536.
- [33] xanthippi.ceid.upatras.gr, ARma library: Pattern tracking for Augmented Reality, <http://xanthippi.ceid.upatras.gr/people/evangelidis/arma/>, 09/2013.

A-1 PROGRAM KODLARI

Uygulamada ve benzetim alıřmalarında kullanılan kodlar tez ieriđine eklenmemiřtir. İhtiya duyulduđu taktirde oakbati@gmail.com mail adresinden istenebilir.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Onur AKBATI
Doğum Tarihi ve Yeri : 24/05/1983
Yabancı Dili : İngilizce
E-posta : oakbatı@yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Elektrik Müh.	Yıldız Teknik Üniversitesi	2008
Lisans	Elektrik Müh.	Yıldız Teknik Üniversitesi	2005
Lise	Fen Bilimleri	Burak Bora Anadolu Lisesi	2001

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2005	Yıldız Teknik Üniversitesi	Araştırma Görevlisi
2005	Teksan Jenaratör	Elektrik Müh.

YAYINLARI

Makale

1. Uçun, L., **Akbatı, O.**, Cansever O., Ethernet Based Automation Network Systems Mobile Laboratory: A Case Study in Nonlinear Control of Water Tank System, Turkish Journal of Electrical Engineering & Computer Sciences, (ISSN: 1300-0632) (**SCI-Expanded**), DOI: 10.3906/elk-1309-127.

Bildiri

1. **Akbatı, O.**, Cansever G., Control of Pattern Tracking Nonholonomic Mobile Robot with Feedback Linearization, Eleco 2013, Bursa, Kasım 28-30, 512-515.
2. Uçun L., **Akbatı O.**, Cansever G., (2012). "CoNeT Mobil Laboratory for Ethernet Based I/O Systems in Industrial Applications", ICCAS 2012 International Conference of Control, 18-22, Jeju Island, Kore, Ekim 17-21, 18-22.
3. **Akbatı, O.**, Cansever, G., (2008). "Çizgi Takip Robotunun Modellenmesi ve Denetleyici Tasarımı", Otomatik Kontrol Ulusal Toplantısı, İstanbul, Kasım 13-15, 870-875.
4. **Akbatı, O.**, Cansever, G., (2007). "Depo Robot Sistemi", 4. Otomasyon Sempozyumu, Samsun, Mayıs 23-25, 31-33.

Projeler

- 1- Öğrenci Robot Projesi, YTÜ Araştırma Fonu, Proje No: 25-04-02-04, Araştırmacı, Eylül 2005 - Eylül 2006.
Proje Uygulama Yeri: Yıldız Teknik Üniversitesi
- 2- CoNeT: Co-operative Network Training, Avrupa Birliği LLP Projesi, Proje No: 502106-LLP-1-2009-1-BE-ERASMUS-ECUE, Araştırmacı, Eylül 2009 - Eylül 2012.
Proje Uygulama Yeri: Karel de Grote University, Belçika
- 3- Yetersiz Eyleyici Mekanik Sistemler için Lineer Olmayan Kontrolcü Tasarımı ve Uygulaması, YTÜ Araştırma Fonu, Proje No: 2010-04-02-KAP01, Araştırmacı, Mayıs 2010 - Mayıs 2011.
Proje Uygulama Yeri: Yıldız Teknik Üniversitesi
- 4- İnsansız Kara Araçlarının Koordinasyonu, YTÜ Araştırma Fonu, Proje No: 2012-04-04-KAP02, Araştırmacı, Nisan 2012 - Devam Ediyor.
Proje Uygulama Yeri: Yıldız Teknik Üniversitesi
- 5- Hız kestirimi için gözleyici tasarımı ve çıkış geri beslemeli denetleyici tasarımı ve uygulamaları, YTÜ Araştırma Fonu, Proje No: 2012-04-04-KAP01, Araştırmacı, Nisan 2012 - Devam Ediyor.
Proje Uygulama Yeri: Yıldız Teknik Üniversitesi

Ödüller:

1. Üniversiteler Arası “Abu Robocon 2004” Robot yarışmasında Türkiye Üçüncülük Ödülü.
2. “XPLORE New Automation Award 2005” yarışmasında “Depo Robotu Otomasyon Sistemi” projesi ile Fabrika Kategorisinde Uluslararası Üçüncülük Ödülü.
3. “XPLORE New Automation Award 2008” yarışmasında “Gökdelen Camlarını Temizleyen Robot” projesi ile Bina Kategorisinde Uluslararası İkincilik Ödülü.