

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL ROBOT NAVİGASYON SİSTEMİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Aykut ÖZDEMİR

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

EYLÜL 2016

MOBİL ROBOT NAVİGASYON SİSTEMİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

**Aykut ÖZDEMİR
(518131025)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Volkan SEZER

EYLÜL 2016

İTÜ, Fen Bilimleri Enstitüsü'nün 518131025 numaralı Yüksek Lisans Öğrencisi Aykut ÖZDEMİR, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “MOBİL ROBOT NAVİGASYON SİSTEMİ GELİŞTİRİLMESİ” başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Yrd. Doç. Dr. Volkan SEZER**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Metin GÖKAŞAN**
İstanbul Teknik Üniversitesi

Yrd. Doç. Dr. Janset DAŞDEMİR
Yıldız Teknik Üniversitesi

Teslim Tarihi : **28 Nisan 2016**
Savunma Tarihi : **26 Ağustos 2016**





Eşime ve aileme,



ÖNSÖZ

Tezimi gerçekleştirmemde büyük emeği olan danışmanım Yrd. Doç. Dr. Volkan SEZER'e ve çalışmalarım boyunca beni motive eden çalışma arkadaşlarıma teşekkür ediyorum.

Ayrıca her zaman yanımda olan eşime ve aileme sonsuz şükranlarımı sunuyorum.

Eylül 2016

Aykut ÖZDEMİR
Araştırma Görevlisi





İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ	xiii
ŞEKİL LİSTESİ	xv
ÖZET	xvii
SUMMARY	xix
1. GİRİŞ	1
1.1 Çalışmanın Ana Hatları	2
2. KULLANILAN ARAÇLAR	5
2.1 Turtlebot	5
2.2 Robot İşletim Sistemi (ROS).....	5
2.2.1 Bağlantı altyapısı	6
2.2.2 Robota özgü özellikler.....	7
2.2.3 ROS araçları	7
3. MOBİL ROBOTLARDA KONUMLANDIRMA	9
3.1 Konumlandırma Problemlerinin Çeşitleri	9
3.2 Literatürde Bulunan Yöntemler.....	9
3.3 Uygulama	11
4. YOL PLANLAMA YÖNTEMLERİ VE GÜVENLİ RRT* YAKLAŞIMI ÖNERİSİ	15
4.1 Literatür Araştırması	15
4.2 RRT* Yöntemi.....	18
4.3 Safe-RRT* Yöntemi	20
4.3.1 Uzaklık terslerinin toplamı yöntemi	23
4.3.2 MR-RRT* yöntemi maliyet yaklaşımı	24
4.4 Safe-RRT* ve RRT* Yöntemlerinin Karşılaştırılması	24
5. YOL TAKİBİ, ENGELDEN KAÇMA VE ÖNERİLEN FGM-DW YAKLAŞIMI	29
5.1 Robot Hareket Modeli	30
5.2 Pure Pursuit	32
5.3 Engelden Kaçınma.....	34
5.3.1 Literatürde bulunan yöntemler	34
5.3.2 FGM yöntemi	35
5.3.3 DWA yöntemi	37
5.3.4 FGM-DW yöntemi	38
5.3.4.1 FGM ile yönlendirici açının hesaplanması	39

5.3.4.2 Ulaşılabilir ve çarpışmasız hız çiftlerinin hesaplanması	40
5.3.4.3 Amaç fonksiyonuna göre değerlendirme.....	40
6. DEĞİŞMEZ VE DEĞİŞKEN ORTAMLARDA NAVİGASYON DENEYLERİ.....	45
6.1 Değişmez Çevrelerde Mobil Robot Navigasyonu	46
6.2 Değişken Çevrelerde Mobil Robot Navigasyonu	47
7. SONUÇ VE ÖNERİLER	53
KAYNAKLAR.....	55
ÖZGEÇMİŞ	59



KISALTMALAR

AMCL	: ADAPTIVE MONTE CARLO LOCALIZATION
APF	: ARTIFICIAL POTENTIAL FIELD
CC-RRT*	: CHANCE CONSTRAINED RRT*
DWA	: DYNAMIC WINDOW APPROACH
FGM	: FOLLOW the GAP METHOD
IMU	: INERTIAL MEASUREMENT UNIT
MCL	: MONTE CARLO LOCALIZATION
MR-RRT*	: MINIMUM RISK RRT*
ND	: NEARNESS DIAGRAM
ORM	: OBSTACLE RESTRICTION METHOD
PRM	: PROBABILISTIC ROADMAP METHOD
ROS	: ROBOT OPERATING SYSTEM
RRT	: RAPIDLY EXPLORING RANDOM TREE
SLAM	: SIMULTANEOUS LOCALIZATION and MAPPING
VFH	: VECTOR FIELD HISTOGRAM



ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1: RRT* ve Safe-RRT* yöntemleri ile bulunan yol uzunlukları ve engellere mesafe değerleri.	27
--	----





ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 : Kobuki tabanı ve özellikleri.....	6
Şekil 2.2 : Turtlebot mobil robot platformu yapısı.	6
Şekil 3.1 : Konumlandırma aşamaları.....	10
Şekil 3.2 : Kullanılan çevre haritası.	12
Şekil 3.3 : AMCL konumlandırma aşamaları.	13
Şekil 4.1 : RRT* algoritma yapısı.....	20
Şekil 4.2 : ChooseParent fonksiyonu.	20
Şekil 4.3 : ReWire fonksiyonu.....	21
Şekil 4.4 : Yeni ChooseParent fonksiyonu.....	23
Şekil 4.5 : Yeni ReWire fonksiyonu.....	23
Şekil 4.6 : Safe-RRT* yöntemi sonucunda bulunan yollar.	26
Şekil 4.7 : Safe-RRT* (k=0.05) ve RRT* yöntemi ile 3000 düğüm sayısında oluşan ağaç yapıları ve bulunan yollar.	28
Şekil 5.1 : Robotun koordinat eksenindeki hareketi.	30
Şekil 5.2 : Robotun dairesel hareketi ve dönme yarıçapı.....	31
Şekil 5.3 : Hareket modeli ile bulunan yol ve kontrol sinyalleri-1.	32
Şekil 5.4 : Hareket modeli ile bulunan yol ve kontrol sinyalleri-2.	32
Şekil 5.5 : Takip mesafesinin takip kalitesine etkisi.	33
Şekil 5.6 : Engellerin robot yarıçapında genişlemesi ve robotun tarama yaptığı bölge.....	36
Şekil 5.7 : Kutupsal dönüştürülmüş engel uzaklıklarının grafiği.....	36
Şekil 5.8 : Hedef açısı (yeşil ok), boşluk açısı (kırmızı ok) ve FGM sonucu hesaplanan hedef açısı (pembe ok) gösterimi.....	37
Şekil 5.9 : Hedef açısı (kırmızı ok), uygun hız çiftleri (yeşil), uygun olmayan hız çiftleri (kırmızı) ve DWA sonucu hesaplanan en iyi hız çifti (mavi) ile gösterilmiştir.....	38
Şekil 5.10 : FGM yönteminin koridor ortamında P kontrolör ile oluşan üç farklı durum.	39
Şekil 5.11 : FGM-DW yöntemi sonucu robotun takip ettiği yol.	42
Şekil 5.12 : Robot çizgisel hızının zamana göre değişimi.	42
Şekil 5.13 : FGM-DW ile farklı güvenlik parametreleri (alfa=0.5,1,2) için elde edilen simülasyon sonuçları.	43
Şekil 5.14 : Gerçek deneylerin yapıldığı test ortamı.....	43
Şekil 5.15 : FGM-DW ile gerçek ortamda farklı güvenlik parametreleri (alfa=0.5,1,2) için elde edilen sonuçlar.	44
Şekil 6.1 : Harita üzerinde konumlandırma yönteminin gerçekleştirilmesi.	45
Şekil 6.2 : Ortam haritasının robot yarıçapı oranında genişletilmesi.....	46
Şekil 6.3 : Değişmez ortamda mobil robot navigasyon sistemi.....	47

Şekil 6.4	: İki farklı durum için robotun başlangıç (siyah), hedef (kırmızı) noktaları için hesaplanan Safe-RRT* yolu (mavi), Pure Pursuit kullanılarak robotun gittiği gerçek yörünge (yeşil) ile gösterilmiştir.	48
Şekil 6.5	: Değişken ortamda mobil robot navigasyon sistemi.....	48
Şekil 6.6	: Üç farklı durum için robotun başlangıç (siyah), hedef (kırmızı) noktaları için hesaplanan Safe-RRT* yolu (mavi), robotun FGM-DW kullanılarak gittiği gerçek yörünge (yeşil) ile gösterilmiştir.	50
Şekil 6.7	: İki farklı durum için robotun başlangıç (siyah), hedef (kırmızı) noktaları için hesaplanan Safe-RRT* yolu (mavi), robotun FGM-DW kullanılarak gittiği gerçek yörünge (yeşil) ile gösterilmiştir.	51



MOBİL ROBOT NAVİGASYON SİSTEMİ GELİŞTİRİLMESİ

ÖZET

Mobil robotlarda hareket planlama problemi birçok birbiri ile bağlantılı problemi kapsamaktadır. Mobil robotun çevreyi algılaması, bu algılamalardan çalıştığı ortamı keşfetmesi, bu keşfettiği ortamda algılayıcılar vasıtasıyla kendi konumunu kestirmesi, başka bir konuma gitmesi için takip etmesi gereken yörüngeyi hesaplanması ve bu yörüngeyi güvenli bir şekilde takip etmesi bu problemlere örnek olarak gösterilebilir. Bu sebeple farklı algılayıcı, robot ve ortam çeşitliliğine sahip robotik sistemlere özel navigasyon yöntemleri ortaya atılmıştır.

Bu çalışmada ilk olarak, diferansiyel sürüş sistemine sahip mobil robot platformu ile haritanın bilindiği kapalı bir ortamda iki boyutlu derinlikli tarama bilgisi yardımıyla robotun harita üzerindeki konumunun belirlenmesi ve harita üzerindeki başka bir noktaya ulaşacak yörüngeyi planlaması gerçekleştirilmiştir. Daha sonra bulunan yörüngelerin Pure Pursuit yöntemi ile takip edilmesi gerçekleştirilmiştir. Ayrıca daha sonra robotun haritada bulunmayan engellere çarpmasını önlemek amacıyla, iki boyutlu tarama bilgisi ile çalışan engelden kaçınma yöntemi geliştirilmiştir. Engelden kaçınma yönteminin yol planlama yöntemi ile birlikte uyum içinde çalışması gerçekleştirilmiştir. Robotun konumunun kestirilmesi robotun kinematik hesaplarından kaynaklanan hataları azaltmak amacıyla kullanılmıştır. Mobil robotun kendi algılayıcıları (enkoder,imu) yardımıyla elde ettiğimiz veriler, tekerleklerdeki kayma ve eylemsizlik gibi etkiler ilk olarak küçük konum belirsizlikleri oluştursa da zamanla bu belirsizlikler toplanarak artmaktadır. Konum belirsizliği, mobil robot üzerindeki Microsoft firmasına ait Kinect algılayıcısı kullanılarak konumlandırma yöntemi yardımıyla azaltılmıştır.

Şimdiye kadar harita üzerinde konumu bilinen bir robotun yol planlaması için çeşitli metotlar geliştirilmiştir. Karmaşık ve çok boyutlu haritalar için örnekleme tabanlı planlayıcılar önerilmiştir. Robotun kinematik kısıtlarını hesaba katıp, aynı zamanda kademeli olarak bulunan yolu kısaltmaya çalışan RRT* yöntemi popüler örnekleme tabanlı planlayıcılardan birisidir. Bu yöntem sayesinde harita üzerinde robotun takip edebileceği yumuşak geçişli yollar ile bu yolları takip etmesi için gerekli kontrol girişleri bulunabilmektedir. Fakat bu kontrol girişleri yalın haliyle mobil robota uygulanırsa robotun beklenen yoldan saptığı ve engellere çarptığı görülecektir.

Ayrıca RRT* yönteminin kademeli olarak bulunan yolu kısaltmaya çalışmasından dolayı harita üzerindeki engellere teğet yollar bulması muhtemeldir. Bu yolların takip edilmesi ise ölçüm ve hareket belirsizlikleri sebebiyle robotun engellere sürtme veya çarpmasına sebep olabilir. Bu çalışmada daha güvenli yolların planlanması için daha önce ortaya atılmış olan MR-RRT* yöntemi ile benzerlik gösteren Safe-RRT* yöntemini önerilmektedir. Bu yöntem kademeli olarak yolumuzu kısaltmaya çalışırken aynı zamanda engellere yakınlığına göre yörüngeleri cezalandırarak en kısa ve güvenli yolun seçilmesini sağlamaktadır. Alınan sonuçlar karşılaştırıldığında

Safe-RRT* yöntemi ile, RRT* yöntemi kadar kısa olmasa da çok daha güvenli yollar bulunabilmektedir. MR-RRT* yönteminde sadece engellere belirli bir mesafe yakındaki yörüngelerin cezalandırılması yapılırken, Safe-RRT* yönteminde ise bütün yörüngelere engellere uzaklığı ile orantılı bir cezalandırma yapılmaktadır.

Bu bahsedilen yöntemler sonucu bulunan yolun çevredeki engellerin hareket etmediği ve konumlarının kesin olarak bilindiği varsayılırsa takip edildiğinde hedef noktasına varması beklenir. Yol takip yöntemleri yardımıyla bu yolların takip edilmesi sağlanabilmektedir fakat çevredeki engellerin hareketli ve konumlarının önceden bilinemediği durumlarda engelden kaçınma yöntemlerine gereksinim duyulmaktadır. Bu yöntemler robotun sensörleri yardımıyla çevreyi algılayıp engellerden kaçınarak hedefine ulaşması için tasarlanmıştır.

FGM ve DWA, farklı altyapıları kullanan ve kendilerini kanıtlamış engelden kaçınma yöntemleridir. FGM, ilk olarak alınan derinlikli tarama bilgisi yardımıyla robot çevresindeki boşlukların hesaplanmasını gerçekleştirir. Daha sonra robotu hedef noktasına ulaştırabilecek güvenli yönelimin hesaplanması gerçekleştirilir. DWA yönteminde ise ilk olarak robot dinamikleri kullanılarak engellere çarpmadan durabilen, ulaşılabilir hız kümesinin hesaplanması gerçekleştirilir. Daha sonra elde edilen bu hız kümesinden, amaç fonksiyonu sonucu en yüksek değere sahip hız çifti bulunur. FGM yönteminin güvenli yönelimi, DWA yönteminin çarpışma kontrolü ve amaç fonksiyonu içerisinde gerekli görülen özellikler birleştirilerek FGM-DW yöntemi oluşturulur. Bu yeni yaklaşım, yol planlama yöntemi ile uyumlu çalışacak şekilde tasarlanmış, böylece engellerin değişken olduğu ortamlarda güvenli şekilde hedefe ulaşabilen bir navigasyon sistemi geliştirilmiştir.

MOBILE ROBOT NAVIGATION SYSTEM

SUMMARY

Mobile robot navigation problem is a widely researched area by scientists. This problem includes perception, mapping, localization, path planning and path tracking problems. Several sensors, different kinematic structures and different environmental complexities require specific solutions.

Motion planning techniques are designed to find geometrically admissible trajectory pairs connecting vehicle initial position and goal location without collisions. These methods can be divided into two major parts, namely global and reactive motion planning.

Global motion planning methods use priori information of obstacles and generate trajectories between initial position and goal position inside configuration space. This type of methods presume that obstacles are static and map is not updated by means of sensory information. Probabilistic roadmaps (PRMs), rapidly-exploring random trees (RRTs), potential field methods, and cell decomposition based methods belong this category. These methods are problematic in case information of obstacles are inaccurate or not available. Moreover their execution time increases exponentially as a consequence of model and world complexities.

There are many proposed methods which aim to solve path planning problem. Sampling based planners are proposed for complex and multidimensional environments. These methods solve path planning problem fast by means of its randomness. RRT* algorithm takes into account the robot's kinematic constraints also gradually optimizes the path. This method yields obstacle free paths which are smoothly connected each other. It also stores the control signals which create the trajectories between points. Although this algorithm yields necessary control sequences for reaching from initial position to goal position, collisions would occur due to the wheel slippages and inertial forces. In order to avoid from these collisions, position correction with localization and path tracking algorithms should be used.

In addition, it is common that RRT* algorithm finds paths which are close to obstacles due to its optimality in terms of shortest path. If these paths are directly tracked, it is possible to meet up with dangerous maneuvers. For this reason, we propose a new approach which is called Safe-RRT* in this thesis. This approximation not only finds the near optimal paths but also safer paths. We compare RRT* and Safe-RRT* approximations according to shortest path optimality and safety criterion. From the results of the experiments, it is observed that even RRT*'s path optimality is better, our proposed method finds safer paths than RRT* method.

In other words if environment model and robot motion are uncertain; consequently global motion plan becomes unrealistic from the standpoint of obstacle avoidance. Reactive motion planning methods focus on changing robot maneuvers from sensory information and goal position. These methods recursively detect obstacles and avoid

from them while driving the robot toward the goal point. Their key advantage over global planners is low computational complexity since these methods use only small portion of information about the environment. Reactive motion planning methods can further be divided into two types in accordance with their final output.

First type of reactive motion planning methods called directional approaches, which compute the appropriate vehicle heading angle for obstacle free navigation. Artificial Potential Field (APF), Vector Field Histogram (VFH), VFH+, Nearness Diagram (ND), Obstacle Restriction Method (ORM) and Follow the Gap Method (FGM) are popular methods in this category. Potential field and vector field methods use repulsive and attractive forces which are generated by obstacles and goal point respectively. ORM uses repulsive angle sets and sub-goals, ND and FGM methods use gap arrays located between obstacles to determine the appropriate heading angle. Although these approaches efficiently generate the direction outputs, they are not sufficient for taking into account the robot dynamics.

Unlike directional approaches, velocity space approaches use robot's dynamic properties in order to perform obstacle free navigation. These methods presume that vehicle travels along arcs of circles. For instance Curvature Velocity Method (CVM) calculates possible curvatures with eliminating ones which collide with obstacles. After pruning redundant ones, most appropriate velocity pair is calculated by CVM's objective function. Velocity Obstacles method (VO) uses dynamic objects velocities in its workspace. This velocities restrict robot's current velocity space. Then final velocity vector is selected among admissible velocities due to convergence measure to goal point. Dynamic window approach (DWA) considers reachable velocities within a short time interval, named as admissible velocities. This method eliminates velocity pairs which are not able to stop the robot without collision. Remaining velocity pairs evaluated by its objective function and velocity pair which has maximum value is executed by this method.

In static environments, our preconditions is map, initial position and goal position are known by robot. Navigation starts with localization on the map. Secondly, path planning finds path from robot position to goal point. Lastly, path following method tracks the path founded by path planning algorithm.

Robot localization is crucial for path tracking because perception data except laser scan is not enough to solve the localization problem. Encoders and inertial measurement unit (IMU) are not enough to predict position accurately because of wheel slippages and inertial force effects. In this thesis, this uncertainty is overcome by using adaptive monte carlo localization algorithm. Adaptive monte carlo localization example works well with Microsoft's Kinect sensor's scan feature. We also use ROS (Robot Operating System) features which facilitate communication between devices. ROS also has various support (e.g., tutorials, demos, drivers) for robotic applications. "Turtlebot" robotic platform is used for experimental tests of this study. Turtlebot has ROS support and has technical and software auxiliary resources.

In dynamic environments, path tracking methods faces collisions due to dynamic obstacles. Reactive motion planners solve this situation by means of sensor measurements.

In this thesis, we propose another method approach named as FGM-DW which uses the strongest parts of FGM and DWA together, in order to navigate the robot in

configuration space. This method retrieves final heading angle from FGM method. Then it uses admissible velocity set calculation stage used as in original DWA method. Finally, it calculates the most appropriate control signal from its new objective function. In dynamic environments, the newly proposed Safe-RRT* algorithm FGM-DW approach are combined to navigate the robot.





1. GİRİŞ

Mobil robotlarda hareket planlama problemi, üzerinde sıkça durulan konulardan birisidir. Algılayıcı ve eyleyici teknolojisinin gelişmesi ve ihtiyaçları daha çok karşılayan yöntemlerin ortaya atılması ile bu alanda büyük ilerleme kaydedilmiştir.

Mobil robotlar kendilerine verilen görevleri gerçekleştirebilmeleri için ilk olarak çevre hakkında bilgiye sahip olmaya ve bu çevrede kendini konumlandırmaya ihtiyaç duyarlar. Ayrıca bu cihazların kendilerine verilen görevleri gerçekleştirebilmeleri için eyleyicilerini ne zaman ve ne şekilde kullanmaları gerektiğini bilmesi gerekir. Tek bir problem olarak görünmesine rağmen bu problem, içerisinde birçok farklı problemi barındırmaktadır.

Mobil robotların hareket planlaması problemi birçok farklı yöntemin birlikte kullanılması ile çözülebilmektedir. Hareket planlama problemi birbirine bağımlı farklı problemleri içerisinde barındırmaktadır. Örneğin eş zamanlı haritalama ve konumlandırma (SLAM) uygulamalarında konumlandırma işlemi için haritaya, haritanın ölçümlere göre güncellenmesi için ise konumlandırma yöntemi ile konum tahmini yapılmasına ihtiyaç duyulur. Bunun gibi birbirine bağımlı problemlere hareket planlama içerisinde sıkça karşılaşılmaktadır. Örneğin, mobil robotun doğru şekilde konumlandırılmadığı takdirde harita üzerinde istenilen şekilde ilerlemesi mümkün olmamaktadır. Bulunan yollar engelsiz olsa dahi yol takip yöntemi aşırı salınımlı hareketler meydana getirdiğinde engellere çarpma gerçekleşebilmektedir. Bu nedenle doğru bir navigasyon gerçekleştirebilmek için bütün araçların birbiri ile uyumlu ve hatayı en aza düşürecek şekilde çalışması gerekir.

Bir mobil robotun başlangıç noktasından hedef noktasına gitmesi istendiğinde şu sorularla karşılaşılmaktadır:

- Başlangıç ve bitiş noktaları nasıl temsil edilebilir?
- Robot hareket ettikten sonra hangi konumda olduğunu nasıl öğrenebilirim?

- Robotun başlangıç noktasından hedef noktasına gitmek için takip etmesi gereken en uygun yolu nasıl hesaplayabilirim?
- Bulunan bu yolu en uygun şekilde nasıl takip edebilirim?

Bu soruların bütün cevaplarını karşılayan bir sistemin kendisine verilen hareket planlama görevini gerçekleştirebileceğini söyleyebiliriz. Mobil robotumuzun enkoder, eylemsizlik hesaplama ünitesi (imu) vasıtasıyla çıkarımda bulunduğu konumu ile gerçek konumu arasında uyumsuzluğun bulunması başka bir algılayıcı yardımı ile konumunun doğrulanmasını gerekli kılmaktadır. Kendini konumlandırma probleminin aşılması ileride takip etmemiz gereken yolu bulduğumuzda daha doğru bir şekilde bulunan yolu takip etmemizi sağlayacaktır.

Bu problem çözüldükten sonra yol planlama aşaması devreye girecektir. Kullanılacak yol planlama yöntemi robotun kıstaslarını göz önüne alarak takip edebileceği kısa ve güvenli yolları bulmamızı sağlamalıdır. Robotun kıstaslarını göz önüne alıp, yolun uzunluğunu kademeli olarak azaltan yol planlama algoritması güvenlik kaygılarını da karşılayacak şekilde oluşturulmuştur. Bulunan sonuçlar karşılaştırılmış ve kayda değer ölçüde güvenlik kaygılarımızın karşılandığı görülmüştür.

Haritanın değişmez ve doğru, ayrıca konum kestiriminin yeterli olduğu kabulleri doğrultusunda mobil robotun yol planlama yöntemi sonucu bulunan yolu takip etmesi ile hedefine ulaşabileceğini söyleyebiliriz. Harita içerisindeki engellerin konumlarının değişken veya yanlış olması durumunda ise bulunan bu yolların takip edilmesi robotun engeller ile çarpışmasına sebep olabilmektedir. Bu problemin çözümü için ise engelden kaçınma yöntemleri ortaya atılmıştır. Tezin son kısmında engellerden kaçınma yöntemleri incelenmiş ve robot dinamiklerini ve güvenliği ön plana alan yeni bir engelden kaçınma yöntemi geliştirilmiştir. Bu yöntemin yol planlama algoritması ile birlikte çalıştırılması gerçekleştirilmiştir.

1.1 Çalışmanın Ana Hatları

Bu tezde yapılan çalışmalar şu sırayla anlatılmaktadır. Tezin ikinci bölümünde kullanılan yazılımsal ve donanımsal araçlardan bahsedilecektir. Üçüncü bölümde ise robot konumlandırma üzerine yapılan çalışmalar ve kullanılan yöntem anlatılacaktır. Dördüncü bölümde ise kullanılan yörünge planlama yöntemleri hakkında bilgi

verilecek ve uygulama inceliklerinden bahsedilecektir. Beşinci bölümde dördüncü bölüm tarafından elde edilen yörüngelerin takip edilmesi için kullanılan yol takip yöntemi ve kullanılan engelden kaçınma yöntemi hakkında bilgi verilecektir. Altıncı bölümde ise önerilen navigasyon yöntemleri incelenecek ve değerlendirilmesi yapılacaktır. Son bölümde ise bu tezde elde ettiğimiz sonuçlar ile bu çalışmanın üzerine ileride yapılabilecek geliştirmeler üzerinde durulmuştur.



2. KULLANILAN ARAÇLAR

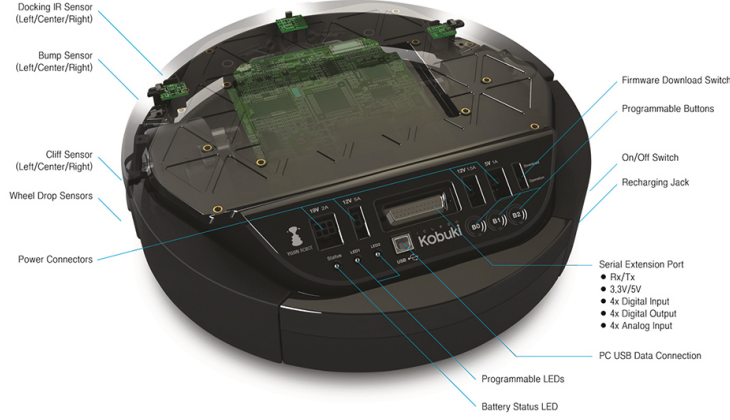
Tez çalışmalarının gerçekleştirilmesinde çeşitli donanımsal ve yazılımsal araçlar kullanılmıştır. Turtlebot [2], içerisinde birçok algılayıcı bulunan, Robot Operating System (ROS) [4] destekli mobil robot geliştirme platformudur. ROS robotik uygulamaların çeşitli bilgilere ulaşması ve manipüle etmesi amacıyla kullanılmasının dışında çeşitli görselleştirme ve simülasyon ortamlarına sahaip olmasından dolayı robotik çalışmalarda sıklıkla kullanılmaktadır.

2.1 Turtlebot

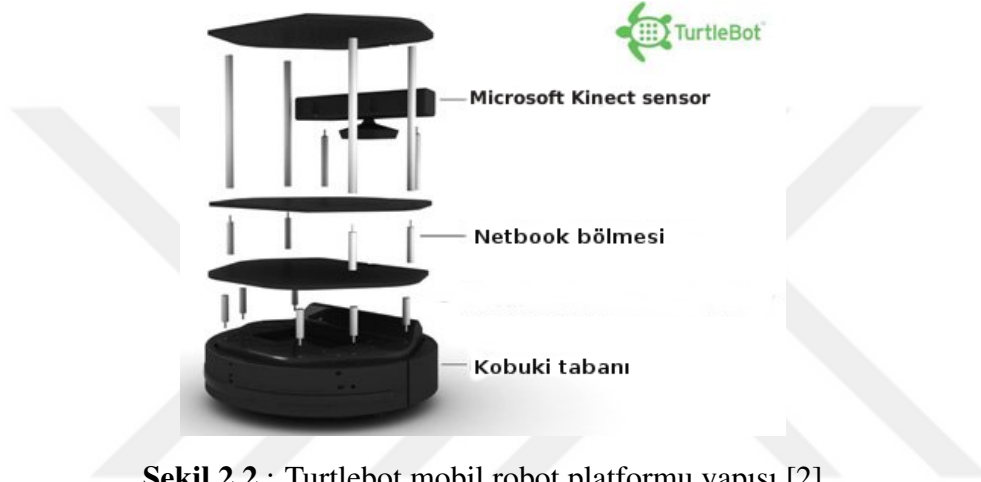
Turtlebot mobil robot platformu yapısal anlamda üç parçadan oluşur. Bu parçalar taşıyıcı iskelet, derinlikli tarama yapabilen algılayıcı ve diferansiyel sürüş sistemine sahip Kobuki [1] tabanıdır. Şekil 2.1 'de görüldüğü gibi üzerinde çeşitli algılayıcılar eklenmiş diferansiyel sürüş sistemine sahip mobil cihaz Kobuki tabanı olarak adlandırılır. Bu cihazın üzerine taşıyıcı bir iskelet yerleştirilmiştir. Bu iskeler üzerine takılacak çeşitli çevre birimlerini (Dizüstü,algılayıcılar vb.) taşıması amacıyla eklenmiştir. Ayrıca üzerinde derinlik taraması yapabilen algılayıcıların (Kinect, Asus Xtion vb.) sabitlenmesi için bölümü bulunmaktadır. Turtlebot mobil robot platformu Şekil 2.2 'de gösterilmiştir. Kinect algılayıcısı ise yüksek çözünürlüklü kamera, derinlik tarama özelliği ve mikrofon dizisi içermesinden dolayı ses tanıma, görüntü işleme, hareketlerin algılanması, haritalama ve konumlandırma gibi birçok projede kullanılabilecek çok yönlü bir cihazdır.

2.2 Robot İşletim Sistemi (ROS)

ROS genel itibariyle robot uygulamaları için geliştirilmiş haberleşme, kontrol ve simülasyon ortamlarını içerisinde barındıran gelişmiş bir yazılım geliştirme alanıdır. Kütüphaneleri ve araçları birçok robotik platforma destek vermektedir ve karmaşık problemleri kolayca çözülebilmesine olanak sağlamaktadır. Alanında yaygın



Şekil 2.1 : Kobuki tabanı ve özellikleri [1].



Şekil 2.2 : Turtlebot mobil robot platformu yapısı [2].

olmasından dolayı farklı robotik alanlarda uzmanlaşmış araştırma guruplarının birlikte kapsamlı araştırmalar yapabilmesine olanak sağlamaktadır [3].

ROS ayrıca destek sağladığı robot ve algılayıcıların nasıl kullanılacağını anlattığı içerisinde açık kaynak kodlu birçok örnek bulunan destek sayfasına sahiptir [4]. Kullanıcılar bu destek sayfasına kendi yaptığı uygulamaları gönderebilmekte veya takıldığı kısımlarda kendi forumundaki başka kullanıcılardan yardım alabilmektedir. Başlıca özellikleri aşağıda detaylı olarak açıklanmıştır.

2.2.1 Bağlantı altyapısı

ROS mesaj gönderim için bir ara taşıyıcı görevi görebilmektedir. Mesajların yayınlanması ve mesajlara abone olma işlemlerinin belirli bir söz dizimi kuralına göre olması platform içerisinde oluşturulan kod parçalarının kendi aralarında kolayca haberleşmelerine imkan tanımaktadır. Ayrıca bu mesajların saklanabilmesi ve tekrar yayınlanabilmesi kullanışlı ROS özelliklerinden birkaçıdır. Eş zamanlı ve eşzamanlı

olmayan mesaj gönderim özellikleri sayesinde robot biliminin ihtiyaç duyduğu geliştirme ortamı görevlerini karşılayabilmektedir.

2.2.2 Robota özgü özellikler

Robotların konum, konum dönüşümleri ve vektörler gibi bilgilerini, ayrıca çeşitli algılayıcı bilgilerini (örneğin imu, lazer tarama, kamera gibi) bir standart içerisinde saklaması ROS ekosistemi içerisinde oluşturulan kodların farklı amaçlar için de kullanılabilmesine imkan tanımaktadır. Geometri kütüphanesi hareketli ve sabit robot uzuvlarının koordinat dönüşümlerinin kolayca hesaplanabilmesini sağlamaktadır. ROS, robot tanımlama dili kullanarak sıfırdan bir robotun modellenmesi ve bu model üzerinden simülasyon yapmaya imkan tanımaktadır. Robot içerisinde gerçekleşen olayların anlık olarak teşhisi ve gözlemlenmesi, çeşitli ROS araçları yardımıyla yapılabilmektedir. ROS ayrıca konum hesaplama, harita üzerinden robot konumlandırma, harita oluşturma ve mobil hareket planlama gibi işlemleri gerçekleştirebilecek çeşitli yeteneklerle donatılmıştır.

2.2.3 ROS araçları

ROS'un en önemli özelliklerinden birisi sahip olduğu araçlardır. ROS ile kullanıcı arayüzü gerektirmeksizin yalnızca komut satırı üzerinden mesajlar, servisler ve olaylar gözlenebilmektedir. Rviz ve rqt grafiksel araçları bilgi gözlem gereksinimlerini karşılamak için kullanılabilir.

Rviz aracı birçok algılayıcı mesajının, robot modellerinin ve haritaların farklı konum dönüşümleri ile gözlemlenebildiği üç boyutlu ROS görselleştirme aracıdır. Rqt ise farklı görevleri gerçekleştiren kod parçalarının birbiri ile ilişkilerini gözlemlemek, yayınlanan mesajların içeriği hakkında bilgi sahibi olmak ve bu içeriklerin kaydedilmesi ve tekrar yayınlanmasını sağlamak amacıyla kullanılan görsel arayüzlü ROS aracıdır.

Gazebo aracı, içerisinde birçok robot platformu ve birçok nesneyi (raf, masa, duvar vb.) barındıran üç boyutlu gerçek dünya modelleme ortamıdır. Bu araçta çevre koşulları yapay olarak oluşturularak uygulamaların gerçek ortamdaki önce simülasyon ortamında test edilmesi sağlanabilmektedir.



3. MOBİL ROBOTLARDA KONUMLANDIRMA

Robotların bilinen bir haritada nerede bulunduğunu kestirmek amacıyla çeşitli konum hesaplama yöntemleri ortaya atılmıştır. Konumlandırma yöntemleri bir robotun nerede olduğu hakkında hiçbir bilgisi olmadığı durumda (kaçırılmış robot problemi) veya bulunduğu konum hakkında şüphesi olduğunda konumunu kestirmek amacıyla kullanılmaktadır. Birçok eş zamanlı konumlandırma ve haritalama (SLAM) yöntemi daha doğru haritalar elde etmek için konumlandırma yöntemlerini kullanmaktadır.

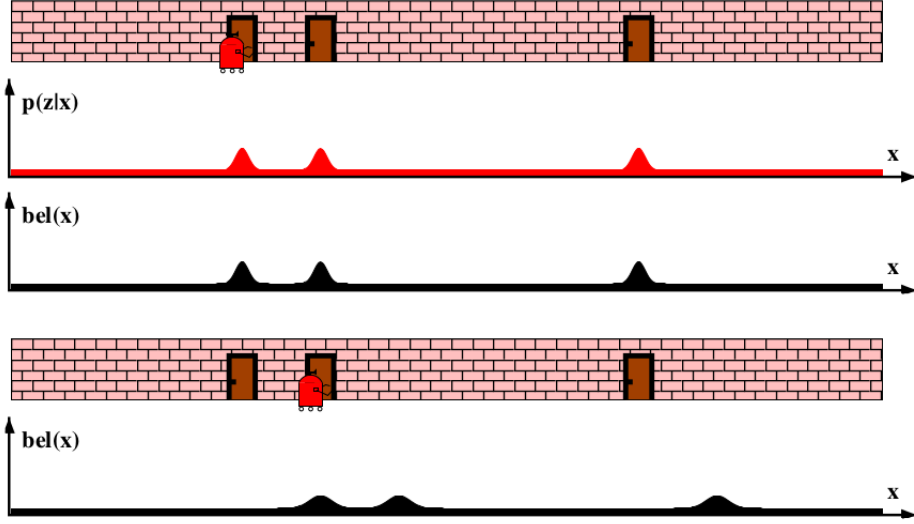
3.1 Konumlandırma Problemlerinin Çeşitleri

Robot konumlandırma problemi amaçlarına göre üç sınıfta incelenir. Bunlardan ilki olan pozisyon izleme yöntemi robotun başlangıçta konumunun bilindiğini varsayıp hareketi sırasında oluşan küçük belirsizlikleri azaltmaya yardım etmesi için kullanılır. Geniş çaplı konumlandırma yöntemi ise robotun çevrede bilinmeyen bir konumda bulunması ve nerede olduğunu bilmesi için çok az bilgiye sahip olduğu durumlarda kullanılır. Kaçırılmış robot yöntemi ise robotun çevredeki konumu hakkında bilgisi olduğu durumdan konumu hakkında hiçbir bilgisi olmadığı başka bir ortama götürülmesi durumunda kullanılmaktadır [5].

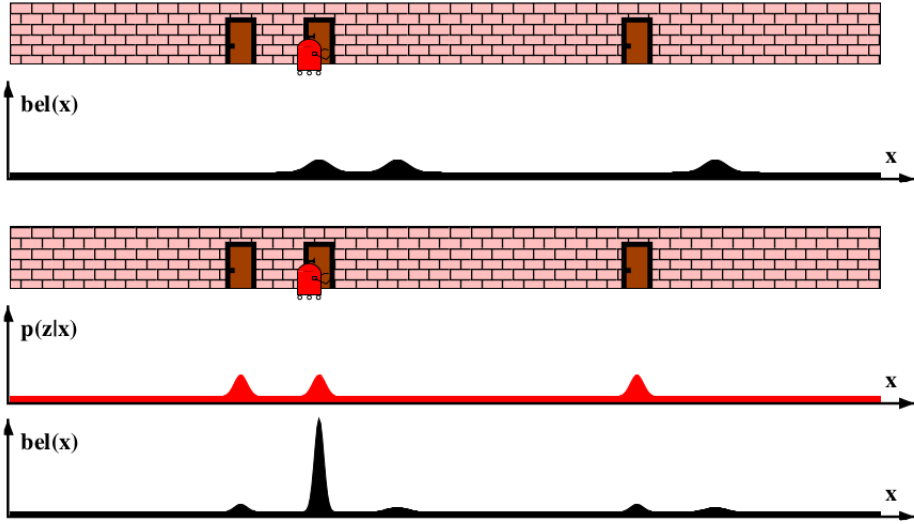
3.2 Literatürde Bulunan Yöntemler

Mobil robot konumlandırma problemi çözümünde iki aşamalı bir yaklaşım sergilenir. Birinci aşama tahmin aşaması olarak adlandırılır. Bu aşamada robotun bir sonraki konumunu tahmin etmek için robotun önceki konumu ve hareket modeli kullanılır. Daha sonra ise güncelleme aşamasına geçilir. Bu aşamada ise tahmin edilen konum algılayıcılar yardımıyla alınan bilgi ile birleştirilip daha doğru bir konum tahmininde bulunulur [6]. Şekil 3.1’de bu aşamalar gösterilmiştir.

Kalman filtresi [7] fikri robotun başlangıç noktasının, hareket ve algılayıcı modelinin gauss yoğunlukları ile ifade edebildiği durumlarda daha önce bahsettiğimiz iki aşamanın kapalı formda çözülebilmesini sağlamaktadır. Kalman filtresi tabanlı



(a) Tahmin aşaması.



(b) Güncelleme aşaması.

Şekil 3.1 : Konumlandırma aşamaları [5].

tekniklerin kesin ve düzgün bir şekilde robotun konum takibi problemini çözebildiği söylenebilir. Geniş çaplı konumlandırma problemlerinde pozisyon bilgisini tek bir gauss dağılımında tuttuğundan robotun birçok farklı bölgede bulunmasını hesaplaması ve çıkarımda bulunması düşünülemez. Ayrıca lineer olmayan sistemler ve takip hataları basit kalman filtresinin sorunları arasındadır.

Bu sorunların üstesinden gelinmesi için işaret içeren markov konumlandırma yöntemleri [8,9] ile ızgara tabanlı markov konumlandırma yöntemleri [10,11] ortaya atılmıştır. Çok başarılı sonuçlar vermesine rağmen bu yöntemler aşırı miktarda hesaplama yükü oluşturmaktadır.

Örnekleme tabanlı yöntemler ise yoğunlukları Gauss formunda temsil etmek yerine örnek dizileri şeklinde temsil etmeyi tercih eder. Bu sayede az hesaplama yükü ile konumlandırma problemlerinin üstesinden gelebilirler. Monte Carlo Lokalizasyonu (MCL) [6] en bilinen örnekleme tabanlı konumlandırma yöntemlerinden biridir.

Bu metodun aşamaları şu şekildedir. Önceki iterasyondan gelen örnek kümesindeki bütün örneklerle ve robota hareket modeli uygulanır. İkinci aşamada alınan gerçek ölçümler ile örneklerin pozisyonunda alınması gereken ölçümler karşılaştırılır. Gerçek ölçümlerle benzerliğine göre örneklerin ağırlıklandırılması gerçekleştirilir. Bu ağırlıklara göre ağırlığı yüksek örneklerin yanında yeni örnekler oluşturulur bu oluşturulan yeni örnekler eski örnek kümesi ile değiştirilir. Her adımda başa dönülerek bu adımlar takip edilir.

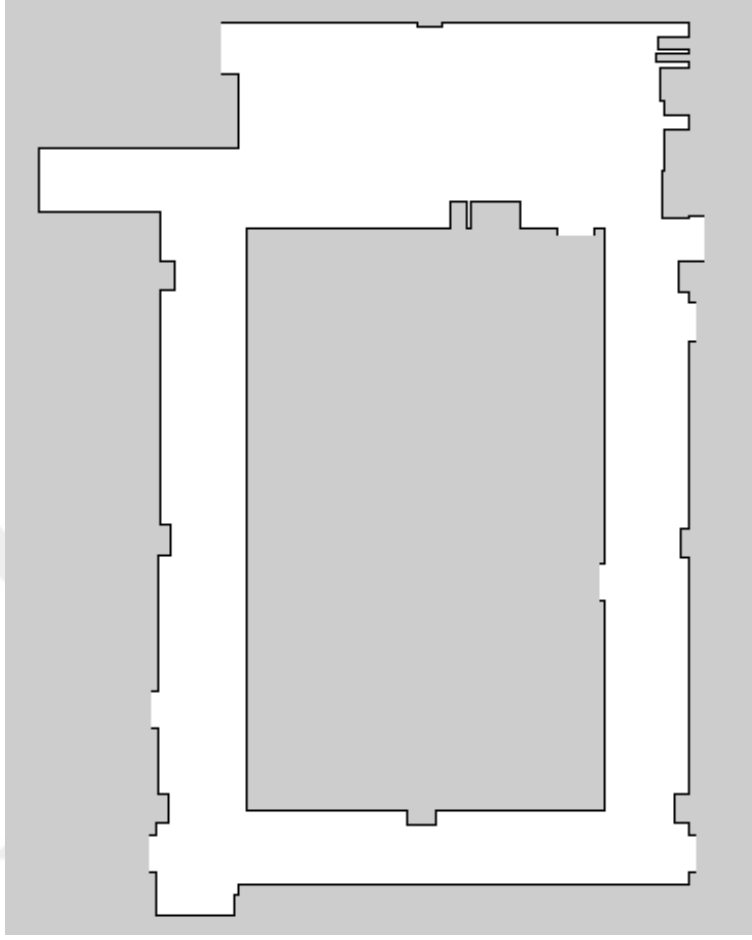
Alınan örnek sayısının konumdan şüphe edildiğinde artırılması, güvenildiğinde ise azaltılması MCL performansını artırmak için önerilmiş başka bir metottür. Bu metod Adaptive Monte Carlo Localization (AMCL) olarak adlandırılmıştır. Bu yöntemde eğer konum tahmini algılayıcı ölçümleri ile uyumlu ve örneklerin ağırlıkları yüksek ise örnek sayısının azaltılması, sensör ölçümlerinin tahminler ile uyumsuzluk yaşadığı durumlarda örnek sayısının artırılması önerilmiştir. Bu sayede konum belirsizlikleri belirli bir aralık bandı içerisinde tutulabilmektedir [12].

3.3 Uygulama

Bu uygulamada kullanılan ortam haritası Şekil 3.2 'de görüldüğü gibidir. Şekilde görülen 18.4m x 22.8m boyutlarında çalışma alanımızın 5cm çözünürlükte oluşturulmuş görüntüsüdür. Robot hareket modelinin lineer olmaması ve ortam haritasının geniş olmasından dolayı oluşabilecek işlem maliyetinin azaltılması amacıyla örnekleme tabanlı konum belirleme yöntemi tercih edilmiştir.

Turtlebot platformu içerisindeki enkoder,imu vasıtasıyla kendi konumunu tahmin edebilmekte fakat bu bilgi çevre ölçümleri ile doğrulanmadığından dolayı güvenilir bir kaynak olarak ele alınamamaktadır. Örnekleme tabanlı konumlandırma yöntemlerinden Adaptive Monte Carlo Localization (AMCL) yöntemi ROS aracılığıyla gerçekleştirilmiş ve ortam haritası üzerinde robot konumlandırma problemi çözülmüştür. Şekil 3.3'de görüldüğü gibi Kinect tarafından alınan ölçümlerle birlikte

Turtlebot harita üzerinde aşamalı bir şekilde konumlandırılmış, Rviz simülasyon ortamında görselleştirme araçlarının kullanılmasıyla gözlemlenmiştir.



Şekil 3.2 : Kullanılan çevre haritası.

Robot ilk olarak harita üzerinde başlangıç bölgesi olarak belirlenmiş bölgeye yakın bir konuma yerleştirilmiştir. Daha sonra ROS içerisinde bulunan AMCL kütüphanesi çalıştırılmış ve RViz simülasyon ortamı çalıştırılmıştır. Konumlandırma yöntemi ilk olarak başlangıç bölgesi üzerinden örnekler alır. Daha sonra robot hareket ettikçe bu örnekler de hareket ettirilir. Bu yöntemde alınan ölçümlerin ve haritanın karşılaştırılması ile uyumlu örneklerin tutulup uyumsuz olanların elenmesi sağlanmıştır.

Şekil 3.3'de görüldüğü üzere ilk olarak parçacıklar başlangıç noktasının çevresine dağıtılmıştır. Daha sonra robotun hareket etmesi ile parçacıklar da hareket ettirilmiş ve bu parçacıklardan ölçüm sonuçlarına benzer olmayanların elenmesi, benzer olanların ise çevresine yeni parçacıkların eklenmesi gerçekleştirilmiştir. Bu işlemin robotun her hareketinden sonra tekrarlanmasıyla parçacıklar gittikçe dar bir bölgede yoğunlaşmış

ve robotun gerçek konumu küçük bir hata payıyla belirlenmiştir. Konumlandırma yönteminin koridor tipi bölgelerde ayırt edici özellik tespit edememesi nedeniyle konum belirsizliğinin arttığı, köşe ve karmaşık engellere sahip bölgelerde ise konum belirsizliğinin azaldığı gözlemlenmiştir.



Şekil 3.3 : AMCL konumlandırma aşamaları, robotun olası konumları(kırmızı oklar), sensör ölçümleri (siyah kareler) ile gösterilmiştir.



4. YOL PLANLAMA YÖNTEMLERİ VE GÜVENLİ RRT* YAKLAŞIMI ÖNERİSİ

Yol planlama robotu başlangıç koordinatından hedef koordinatına götürebilecek yörünge dizisinin bulunması işlemidir. Bu yöntemler sonucu elde edilen yolların engeller ile kesişmemesi ve kullanılacak robotun hareket modeline uygun olması istenir. Bu yöntemler ortam haritasına, robot konumuna ve hedef noktasına ihtiyaç duymaktadır.

Dinamik modeli $x'(t) = f(x(t), u(t))$ şeklinde bir sistem düşünelim. Bu denklemden x durum uzayının u ise kontrol uzayının bir elemanı olsun. Engelli bölgeler X_{obs} ile ve serbest bölgeler X_{free} olarak adlandırılınsın. Engelli bölgeler, durum uzayından serbest bölgelerin çıkartılmasına ($X \setminus X_{free}$) eşit olacaktır. X_{hedef} robotun ulaşması istenilen hedef bölgeyi temsil etsin. Başlangıç konumundan x_{ilk} hedef bölgeye ulaşabilecek engelsiz yörüngelerin aranması işlemine yol planlama denir. Bu yöntem ile bulunan kontrol girişi ($u:[0,t]$) ise t süresi boyunca uygulandığında robotu başlangıç noktasından hedef bölgesine ulaştıracak kontrol sinyallerini temsil eder.

4.1 Literatür Araştırması

Yol planlama robotu başlangıç koordinatından hedef koordinatına götürebilecek sürüş kontrol sinyallerinin bulunması işlemidir. Bu sinyaller çevre koşullarının dayattığı sınırlar ve robotun kendi dinamikleri çerçevesinde hesaplanmalıdır. Bu işlemin gerçekleştirilmesi için ortaya atılan ilk yaklaşımlar hücre ayrıştırma temelli yöntemler [13], potansiyel alan yöntemleri [14] ve yol haritası yöntemleridir [15]. Ayrıca haritadaki belirsizlikleri de hesaba katabilecek algılayıcı tabanlı hareket planlama yöntemleri olarak böcek algoritmaları [16] da öne sürülen yöntemler arasındadır.

Hücre temelli ayrıştırma yaklaşımları harita üzerindeki engellerin haritayı boş alan parçalarına ayırdığı varsayımı ile bu parçalar üzerinde başlangıç noktasından bitiş noktasına ulaşmak için arama gerçekleştiren yöntemlerdir.

Potansiyel alan yaklaşımında ise engeller ve hedef noktaları baz alınarak oluşturulan yapay potansiyel alanın, nokta ile tanımlanan robotu yönlendirmesi hedeflenmiştir. Hedef noktasında bulunan çekici potansiyel alan robotu kendi üzerine çekmekte iken, engel noktalarının itici potansiyeli robotu engellerden uzaklaştırmaya çalışmaktadır. Haritanın herhangi bir noktasında bulunan robotun hedefin çekme ve yakınındaki engellerin itme kuvvetleri bileşkesi yönünde hareket etmesi ile hedef noktasına varması hedeflenmektedir.

Yol haritası yöntemleri ise engellerin geometrik şekilleri üzerinden birbirine bağlı çift yönlü engelsiz alanlarda bağlantılar oluştururlar. Bağlantı kümesine başlangıç ve bitiş noktasının da bağlanması ile başlangıçtan hedef noktasına en kısa yolun aranması sonucu takip edilmesi gereken yol bulunur. Başka başlangıç bitiş noktası verildiğinde yine aynı bağlantı kümesi kullanılarak yol planlama problemi çözülmeye çalışılır.

Böcek algoritmaları ise haritanın tamamının bilinmediği ve hareketli engellerin bulunduğu durumlarda mesafe veya yaklaşım algılayıcıları yardımıyla hedefe ulaşılmasını sağlarlar. Bu algoritmaların en belirgin özelliği, hedefe doğru hareket ederken bir engel ile karşılaştığında engelin köşelerini takip etmesidir. Bu yöntemde farklı çevreler için farklı ayrılma ve takip etme rutinlerine sahip Bug 1 ve Bug 2 yöntemleri ortaya atılmıştır [16].

Bunların dışında çalışma ortamının bölünerek bu bölümler üzerinden arama yapabilen kare arama algoritması A*[17] ortaya atılmıştır. Bu yöntem ile karesel bölünmüş bu alanlar içerisinde başlangıç ve bitiş kareleri arasında en kısa yolun bulunmasını sağlanmaktadır. Fakat haritanın karelere ayrılma çözünürlüğüne bağlı olmasından dolayı bu yöntem ile her koşulda sonuç almak mümkün olamamaktadır.

Çok boyutlu ve karmaşık engellerin bulunduğu ortamlarda yukarıda bahsedilen yöntemler hesaplama süresi bakımından maliyetlidir. Çalışma alanının bu yöntemler ile incelenmesi dinamik olarak takip edilmesi uygun olmayan yolların bulunmasına veya problemin çözümünün zorlaşmasına sebep olmaktadır. Bu tür ortamlarda yol planlama probleminin çözümü için çevrenin özelliklerini kullanan bahsedilen yöntemler yerine örnekleme tabanlı planlayıcılar ortaya atılmıştır.

Örnekleme tabanlı planlayıcıların en popüler olanları Olasılıksal Yol Haritası Metodu (PRM) [18] ve Hızlı Keşfeden Rastgele Ağaçlar (RRT) [19-20] yöntemleridir. Bu

iki yöntemin en belirgin farkı PRM'nin çift yönlü bağlantılar üzerinden arama gerçekleştirmesi, RRT algoritmasının ise tek yönlü bağlantılarla genişlemesidir. PRM yöntemindeki çift yönlü bağlantılarda hareket kısıtları dikkate alınamamakta fakat RRT algoritması her yeni başlangıç ve bitiş noktası için yeni bir ağaç genişletmesinden dolayı hareket kısıtları bu bağlantılarda kullanılabilir. PRM algoritmasında bağlantı bütünü bir kere oluşturulup bu bütün üzerinden arama gerçekleştirir. Bu yüzden PRM değişken engelli çevreler için uygun değildir fakat bağlantı bütünü tekrar hesaplama gerektirmediğinden oldukça hızlı sonuçlar üretir.

RRT algoritmasının en iyi çözümü bulması, keşfedilmemiş bölgelere genişleme eğiliminden dolayı mümkün olmamaktadır. Bu yüzden Karaman ve Frazzoli RRT algoritması ile oluşan ağaç yapısını geliştirmek için RRT* [21-22] algoritmasını ortaya atmıştır. Bu yöntem klasik RRT'nin en yakın düğümden yeni noktaya yönelme prosedürünü, yakın komşuları içinden, toplam maliyeti en aza düşürecek düğümden yeni noktaya yönelme olarak değiştirir. Ayrıca RRT* yöntemine yeni bağlanan düğümden komşu düğümlerinin maliyetini kısaltması için yeniden bağlama prosedürü eklenmiştir. Ayrıca bu yöntemin olasılıksal olarak kademeli bir şekilde en iyi sonuca yaklaşması gerektiği kanıtlanmıştır.

RRT* yönteminin çeşitli versiyonları zaman içerisinde geliştirilmiştir. Alınan örneklerin, üzerine etkiyen potansiyel alan ile veya üçgensel trigonometrik bağıntılardan yararlanılarak hedef ve başlangıç noktaları arasında bölgelere taşınması fikirleri ortaya atılmış [23,24] ayrıca harita üzerindeki Voronoi ve visibility graph yöntemleri ile elde edilen bölgelerden örnek alınması ile kısa yolların daha çabuk bulunması sağlanmıştır [25,26]. Ayrıca RRT*'ın çalışma esnasında elde ettiği yolların özellikleri kullanılarak yeni örnekler alınması fikirleri ortaya atılmıştır [27]. Bu yöntemlerin çoğu RRT*'ın en iyi sonuca yaklaşım hızını artırmak ve ilk çözümü bulmasını kolaylaştırmaya yönelik adımlardır.

Yakın zamanda yukarıda bahsedilen yöntemlerin haricinde çevre özelliklerini hesaba katarak güvenlik kaygılarını da gideren CC-RRT* [28] ve MR-RRT* [29] yöntemleri ortaya atılmıştır. CC-RRT* yöntemi durum uzayını kullanmak yerine olasılık uzayında belirsizlikleri de hesaba katılarak yol planlama problemini çözmektedir. Bu yöntem ile hesaplanan yörüngeler çarpışma riskleri ile değerlendirilirler. Bu yöntem çarpışma riski yüksek olan yörüngelerin maliyetlerini artırıp kısa yol olarak seçilmesini

önleyerek daha güvenli yolların tercih edilmesini sağlar. Ayrıca gerçek zamanlı olarak ağaç yapısını güncelleyebilmekte ve hareketi sırasında yolunun kapanması durumunda yenisini aramaktadır. MR-RRT* yöntemi ise durum uzayında engellere yakın geçen yörüngelerin maliyetlerinin artırılması temeline dayanır. Olasılıksal yöntemler yerine metrik maliyetler üzerinden hesaplamalarını gerçekleştirir. Bu yöntemlerin çevre özelliklerini kullanmalarından dolayı RRT* yöntemine göre ek hesaplama maliyeti getirdiği yadsınmaz bir gerçektir. Fakat bu yöntemler RRT* yönteminin engel köşelerine çok yakın bulunduğu yörüngeler yerine takip edilmesi daha güvenli yollar bulunmasını sağladıklarından avantajlıdır.

4.2 RRT* Yöntemi

RRT* yöntemi engelsiz ortamlarda düğüm (V) ve bağlantı (E) kümeleri içeren zamanla genişleyen bir ağaç yapısı ($T = (V, E)$) oluşturma temeline dayalı bir yol planlama yöntemidir. Bu yöntemde başlangıç noktası düğüm kümesinin ilk elamanıdır. Düğüm kümesi yeni bağlantılar yapıldıkça genişler. Bu genişleme harita üzerinden rastgele engelsiz bölgelerden örneklerin alınıp, bu örneklerle robot hareket kısıtlarına uygun bağlantılar kurulmasıyla gerçekleşir. Bağlantı kümesinde ise bu düğümlerin birbiri arasındaki ilişkiler saklanır.

Optimal yol planlama kademeli olarak maliyeti en aza indirme yönteminin adıdır. Yörünge maliyetleri $c(x)$ ile gösterilir ve her zaman değeri pozitifdir. RRT* yöntemi kademeli olarak yörünge maliyetlerini minimize ederek genişleyen bir ağaç yapısı oluşturur. RRT* yönteminin aşamaları Şekil 4.1, 4.2 ve 4.3'de görüldüğü gibidir. Aşamalarda kullanılan fonksiyonlar aşağıdaki gibidir.

Sample : Bu fonksiyon X_{free} içerisinde rastgele seçilmiş örneklerin alınması görevini görür.

Distance : Bu fonksiyon iki konum arasındaki optimal uzaklık ile dönüş yapar. Uzaklık ölçütü öklid uzaklığı alınmıştır.

Nearest : Bir konum ($z \in X$) ve ağaç yapısı ($T = (V, E)$) verilmiş olsun. Nearest fonksiyonu z konumuna düğüm kümesi içerisinde bulunan en yakın komşusunu *Distance* fonksiyonuna göre bulur.

Near : Bir konum ($z \in X$) ve ağaç yapısı ($T = (V, E)$) verilmiş olsun. Near fonksiyonu z 'nin çevresinde belirli bir yarıçap içerisinde kalan düğüm kümesi elemanlarının bulunmasını sağlar. Bu yarıçap sabit veya düğüm sayısı ile değişken olabilir.

ObstacleFree : Bu fonksiyon kontrol ettiği bağlantı engeller ile kesişiyorsa *True* kesişmiyorsa *False* çıkış değeri verir. Bu kontrolün yapılma yöntemi engellerin ve bağlantıların temsil şekli ile ilgilidir.

Steer : İki adet konum $z_1 \in X, z_2 \in X$ verilmiş olsun. Hareket modeli olarak da adlandırabileceğimiz bu fonksiyon sistemi z_1 noktasından z_2 noktasına sürebilen kontrol sinyalinin ve bu kontrol sinyali uygulandığında oluşan yörüngeyi hesaplanmasını sağlar.

InsertNode : Düğüm kümesinde bulunan bir düğüm ($z_{current} \in V$) ve bu düğümde ulaşılmış yeni bir konum ($z_{new} \notin V$) olduğu varsayalım. Bu metod içerisinde yeni konum düğüm kümesine eklenir ayrıca yeni düğüm bağlantı kümesine kök düğümü $z_{current}$ bilgisiyle beraber eklenir. Yeni noktanın maliyeti ise kök düğümü ile arasındaki yörünge maliyeti ve kök düğümünün maliyetinin toplamından Denklem 4.1'deki bağıntıdan hesaplanır.

$$cost(z_{new}) := cost(z_{current}) + c(x') \quad (4.1)$$

RRT* yöntemi RRT yöntemi gibi başlangıç noktasından başlayarak ağaç yapısını genişletmeye çalışır. RRT yöntemi yeni örneğe en yakın dal ucundan genişlemeye çalışırken, RRT* algoritması yeni örneğe en az maliyetle ulaşmak için örneğin çevresindeki komşularını maliyet fonksiyonuna göre değerlendirir. Engelsiz alan içerisinde bulunan maliyetin en aza indirilmesini sağlayacak düğümden bağlantı kurulur (Şekil 4.2).

ReWire fonksiyonunda ise ChooseParent fonksiyonu ile başarılı bağlantı kurabildiğimiz düğümden çevredeki komşu düğümlere bağlantı yapıp yapamayacağı kontrol edilir. Eğer bu noktadan çevre düğümlere yapılan bağlantı çevre düğümlerin maliyetini azaltıyorsa eski bağların kopartılıp yeni düğümden bu noktalara yeni bağlantılar kurulması gerçekleştirilir (Şekil 4.3).

```

Input:  $z_{init}$ 
Output:  $T = (V, E)$ 
 $T \leftarrow \text{InitializeTree}()$ 
 $T \leftarrow \text{InsertNode}(\emptyset, z_{init}, T)$ 
for  $i=1$  to  $i=N$  do
     $z_{rand} \leftarrow \text{Sample}(i)$ 
     $z_{nearest} \leftarrow \text{Nearest}(T, z_{rand})$ 
     $(x_{new}, u_{new}, T_{new}) \leftarrow \text{Steer}(z_{nearest}, z_{rand})$ 
    if  $\text{ObstacleFree}(x_{new})$  then
         $Z_{near} \leftarrow \text{Near}(T, z_{new}, x_{new})$ 
         $z_{min} \leftarrow \text{ChooseParent}(Z_{near}, z_{nearest}, z_{new}, x_{new})$ 
         $T \leftarrow \text{InsertNode}(z_{min}, z_{new}, T)$ 
         $z_{near} \leftarrow \text{ReWire}(T, Z_{near}, z_{min}, z_{new})$ 
    end
end
return  $T$ 

```

Şekil 4.1 : RRT* algoritma yapısı.

```

Input:  $Z_{near}, z_{nearest}, z_{new}, x_{new}$ 
Output:  $z_{min}$ 
 $z_{min} \leftarrow z_{nearest}$ 
 $c_{min} \leftarrow \text{Cost}(z_{nearest}) + c(x_{new})$ 
for  $z_{near} \in Z_{near}$  do
     $(x', u', T') \leftarrow \text{Steer}(z_{near}, z_{new})$ 
    if  $\text{ObstacleFree}(x')$  and  $x'(T') = z_{new}$  then
         $c' \leftarrow \text{Cost}(z_{near}) + c(x')$ 
        if  $c' < \text{Cost}(z_{new})$  and  $c' < c_{min}$  then
             $z_{min} \leftarrow z_{near}$ 
             $c_{min} \leftarrow c'$ 
        end
    end
end
return  $z_{min}$ 

```

Şekil 4.2 : ChooseParent fonksiyonu.

4.3 Safe-RRT* Yöntemi

Önerilen bu yöntemin amacı ise başlangıç noktasından hedef noktasına iteratif olarak en iyi sonuca yaklaşan RRT* yönteminin hedefe daha güvenli yollardan ulaşan yörüngelerin bulmasını sağlamaktır. RRT* yöntemi genel olarak en kısa yolları bulmaya çalıştığından engel noktalarının köşelerinden geçen yollar bulma eğilimindedir. Robot konum bilgisindeki küçük belirsizlikler veya çevresel ve robotun

```

Input:  $T, Z_{near}, z_{nearest}, z_{new}$ 
Output:  $T$ 
for  $z_{near} \in (Z_{near} \setminus z_{min})$  do
   $(x', u', T') \leftarrow Steer(z_{new}, z_{near})$ 
  if  $ObstacleFree(x')$  and  $x'(T') = z_{near}$  and  $Cost(z_{new}) + c(x') < Cost(z_{near})$ 
    then
       $T \leftarrow ReConnect(z_{new}, z_{near}, T')$ 
    end
  end
end
return  $T$ 

```

Şekil 4.3 : ReWire fonksiyonu.

hareket aksamı içerisindeki kusurlar yüzünden bulunan bu yollar takip edildiğinde engellere çarpma veya sürtünme gibi durumlar oluşabilmektedir.

RRT* yöntemi içerisinde iki adet maliyet değişkeni bulunmaktadır. $Cost(z)$ değişkeni z düğümünden ilk düğüme (x_{ilk}) ulaşılması için katedilmesi gereken yolun maliyetini, $c(x')$ değişkeni $Steer$ fonksiyonu ile bulunan iki düğümü birleştiren yörünge parçasının katetmesi gereken yolu gösterir. Bu maliyet değişkenleri yolların kısaltılması için $ChooseParent$ ve $ReWire$ bölümlerinde yolların optimizasyonu için kullanılmaktadır.

Safe-RRT* yöntemi ise RRT* yönteminin maliyet terminolojisinde buna bağlı olarak bağlantı seçiminde değişiklikler önermektedir. En kısa bağlantıyı seçmek yerine, güvenli ve kısa bağlantıların seçilmesi amaçlanmıştır. Bu amaçla RRT* algoritmasına SafetyCost adında yeni bir maliyet değişkeni eklenmiş ve bu değişken bağlantı seçiminde kullanılmıştır.

Safe-RRT* yönteminin modifiye edilmiş $ReWire$ ve $ChooseParent$ fonksiyonları Şekil 4.4 ve Şekil 4.5'te görüldüğü gibidir.

Modifiye edilmiş $ChooseParent$ fonksiyonunun işleme aşamaları aşağıdaki gibidir.

- Alınan örneğin en yakın komşusu ile bağ yaptıktan sonra düğüm maliyeti Denklem 4.2'de gösterilen bağıntıdan hesaplanır.
- Daha sonra örneğin çevresinde belirli bir yarıçap içerisinde kalan komşu düğüm kümesi bulunur.
- Düğüm kümesi içerisindeki elemanların yeni örnek ile engelsiz bağ yapıp yapamayacağına bakılır.

- Engelsiz bağ yapabildiği düğümlerden ulaşılma maliyeti Denklem 4.3’de gösterilen bağıntıdan hesaplanır.
- Bu maliyetler içerisinde en düşük değerin elde edildiği düğüm, alınan örneğin kök düğümü olarak belirlenir.

$$c_{new} \leftarrow Cost(z_{nearest}) + c(x_{new}) + SafetyCost(x_{new}) \quad (4.2)$$

$$c_{new} \leftarrow Cost(z_{near}) + c(x') + SafetyCost(x') \quad (4.3)$$

Denklemden bulunan x' ifadesi iki düğüm arasında *Steer* fonksiyonu sonucu bulunan yörünge parçasını temsil etmekte ve seçilen düğümlere göre değişiklik göstermektedir. Modifiye edilmiş ReWire fonksiyonun aşamaları aşağıdaki gibidir.

- Komşu düğüm kümesinden ChooseParent fonksiyonu ile bulunmuş örneğe en az maliyetle ulaştığımız düğüm çıkarılır.
- Alınan örnekten kalan düğümlere engelsiz bağ yapıp yapamayacağı kontrol edilir.
- Engelsiz bağ yapabildiği düğümler için maliyet Denklem 4.4’te gösterilen denklemden hesaplanır.
- Yeni kurulan bu bağlar ile komşu düğümlere daha az maliyetli bir yol bulunmuş ise komşu düğümlerin eski bağları kopartılıp örnekten bu düğümlere yeni bağlantılar kurulur.

$$c_{near} \leftarrow Cost(z_{new}) + c(x') + SafetyCost(x') \quad (4.4)$$

Bu yaklaşım bileşke maliyet olarak da adlandırabileceğimiz $Cost(z) + c(x') + SafetyCost(x')$ denkleminde elde edilen değerleri azaltmaya çalışmaktadır. SafetyCost maliyetinin bileşke maliyete engellere yakın yörünge parçaları için yüksek, engellerden uzak geçenler için düşük değerler eklemesi engellere yakın geçen yollardan ziyade engellere uzak geçen yolların ChooseParent ve ReWire aşamalarında seçilmesini sağlayacaktır. SafetyCost fonksiyonunun değerini hesaplamak için iki

Input: $Z_{near}, z_{nearest}, z_{new}, x_{new}$
Output: z_{min}

```

1  $z_{min} \leftarrow z_{nearest}$ 
2  $c_{min} \leftarrow \text{Cost}(z_{nearest}) + c(x_{new}) + \text{SafetyCost}(x_{new})$ 
3 for  $z_{near} \in Z_{near}$  do
4    $(x', u', T') \leftarrow \text{Steer}(z_{near}, z_{new})$ 
5   if  $\text{ObstacleFree}(x')$  and  $x'(T') = z_{new}$  then
6      $c' \leftarrow \text{Cost}(z_{near}) + c(x') + \text{SafetyCost}(x')$ 
7     if  $c' < \text{Cost}(z_{new})$  and  $c' < c_{min}$  then
8        $z_{min} \leftarrow z_{near}$ 
9        $c_{min} \leftarrow c'$ 
10    end
11  end
12 end
13 return  $z_{min}$ 

```

Şekil 4.4 : Yeni ChooseParent fonksiyonu.

Input: $T, Z_{near}, z_{nearest}, z_{new}$
Output: T

```

1 for  $z_{near} \in Z_{near} \setminus \{z_{min}\}$  do
2    $(x', u', T') \leftarrow \text{Steer}(z_{new}, z_{near})$ 
3   if  $\text{ObstacleFree}(x')$  and  $x'(T') = z_{near}$  and
    $\text{Cost}(z_{new}) + c(x') + \text{SafetyCost}(x') < \text{Cost}(z_{near})$  then
4      $T \leftarrow \text{ReConnect}(z_{new}, z_{near}, T)$ 
5   end
6 end
7 return  $T$ 

```

Şekil 4.5 : Yeni ReWire fonksiyonu.

yöntem önerilmiştir. Bunlardan birincisi bütün yolların engellere yakınlık bakımından maliyetlendirmesi ikinci yöntem ise daha önce bahsedilen MR-RRT* yönteminin kullandığı sadece engellere belirli bir yarıçap içinde yaklaşan yolların cezalandırılması yöntemleridir.

4.3.1 Uzaklık terslerinin toplamı yöntemi

Bu metod kademeli olarak toplam yörünge uzunluğunun kısaltılması aynı zamanda engellere daha uzak yolların seçimi için önerilmiştir. SafetyCost fonksiyonuna giriş olarak uygulanan her yörünge parçasının üzerinden belirli frekansta örneklerin alınması ve her bir örneğin engellere olan minimum uzaklığının hesaplanması gerçekleştirilmiştir. Önerilen bu yöntem ile bulunan bu uzaklıkların tersinin, belirli bir k katsayısı ile çarpıldıktan sonra toplanmasıyla SafetyCost maliyeti elde edilmektedir.

Uzaklıkların tersinin alınması kısa mesafelerde yüksek, uzun mesafelerde düşük değerlerin elde edilmesi için kullanılmaktadır. K katsayısı ise çok yüksek çıkabilecek sonuçları sınırlandırmak amacıyla kullanılır. Safe-RRT* yöntemi için maliyet fonksiyonu Denklem 4.5’de gösterildiği gibi hesaplanmıştır.

$$SafetyCost(x) = \sum_{i=0}^n \frac{k}{\min \|x_i - x_{obs}\|} \quad x_0 \cdots x_n \in x_{trajectory} \quad (4.5)$$

4.3.2 MR-RRT* yöntemi maliyet yaklaşımı

MR-RRT* yaklaşımda ise birinci yöntemle göre farklı bir yaklaşım sergilenir. Bu yöntemle ağaç genişletme işleminde engellere yakın olan yolların cezalandırılması Denklem 4.6 ve 4.7’ de görüldüğü gibidir. Bu yaklaşımda yörünge konum parçalarına ayrılıp her bir konumun engellere en yakın uzaklığı (d_{min}) risk uzaklığından (d_{risk}) küçükse uzaklığın tersi miktarınca yörünge konumunun cezalandırılması gerçekleştirilmektedir. Denklem 4.7’de yörünge konum parçalarına ayrılır ve her bir konum için Denklem 4.6’da gösterilen Risk fonksiyonu sonucu elde edilen değerin toplanıp yörünge konumunun toplam risk maliyetinin hesaplanması gerçekleştirilir.

$$Risk(x) = \begin{cases} 1/d_{min} & \text{kosulunda } 1/d_{min} \geq 1/d_{risk} \\ 0 & \text{kosulunda } 1/d_{min} < 1/d_{risk} \end{cases} \quad (4.6)$$

$$MRCost(x) = \sum_{i=0}^n Risk(x_i) \quad x_0 \cdots x_n \in x_{trajectory} \quad (4.7)$$

4.4 Safe-RRT* ve RRT* Yöntemlerinin Karşılaştırılması

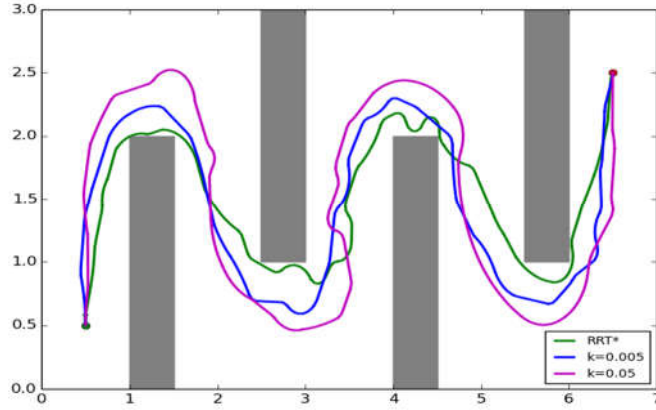
Bu bölümde Safe-RRT* yöntemi farklı katsayılarla incelenmiş ve RRT* yöntemi ile toplam yörünge uzunluğu ve engellere yakınlığı açısından incelenmiştir. Örnek bir harita üzerinde RRT* ve Safe-RRT* yöntemleri sonucunda 1000,2500 ve 5000 düğüm sayısında elde edilen yollar Şekil 4.6 ’da görüldüğü gibidir. Bulunan yollar düğüm sayısı arttıkça kısalmış ve dolambaçlı yollar yerine daha düz yollar elde edilmiştir. Bu durum RRT* yönteminin her iterasyonda daha kısa yolları araması özelliğinden kaynaklanmıştır. Safe-RRT* yönteminin güvenlik katsayısı artırıldığında RRT* yöntemine göre daha uzun fakat daha güvenli yollar bulunduğu gözlemlenmiştir. Bunun dışında düşük k değerinde RRT* yöntemine yaklaştığını, k değerinin sıfır

değerini aldığı durumda ise engellerin etkisi göz ardı edildiğinden RRT* yöntemi ile aynı sonucu vereceğini söyleyebiliriz.

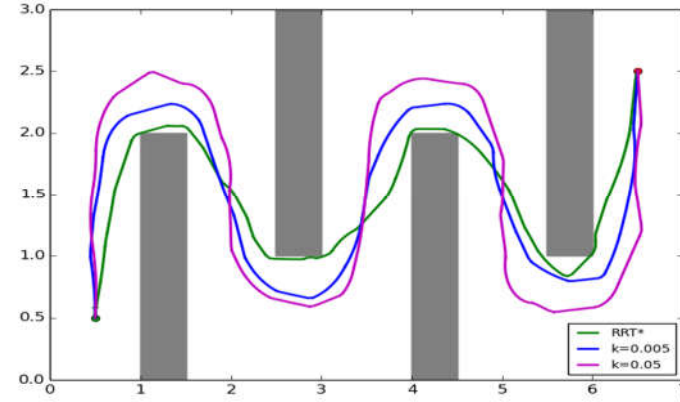
Farklı katsayılar ile 100 adet Safe-RRT* yöntemi simülasyonu sonucunda elde edilen yörünge uzunlukları ve yörüngeler içerisindeki noktaların engellere ortalama uzaklıkları Çizelge 4.1'de görüldüğü gibidir. Bulunan yolları güvenlik ve uzunluk açısından değerlendirmek amacıyla bu parametreler ölçüt olarak alınmıştır. Bu değerler 1000,2500 ve 5000 düğüm sayıları için incelenmiştir.

Çizelgeden görüldüğü üzere Safe-RRT* algoritması sonucu oluşan yörüngelerin k katsayısı arttıkça uzunluğu artmaktadır. Uzun yolların bulunması ne kadar istenmeyen durum olsa da engellere çok yakın noktalardan geçilmemesi amacıyla yolun uzatılması güvenlik kaygılarını gidermektedir. K katsayısı sıfır değerine yaklaştıkça daha güvenli yollar yerine daha kısa güvenli olmayan yolların tercih edilmesi gerçekleşmektedir. Ayrıca düğüm sayısının artırılması bütün durumlarda, RRT* yöntemi altyapısı kullanıldığından bulunan yolların kısalmasına sebep olmaktadır. K, problem odaklı değeri belirlenebilecek bir katsayıdır ve bu yüzden çalışılan uzay metriğine ve engellerden kaçınma derecesine göre farklı değerlerde seçilmesi gerekir.

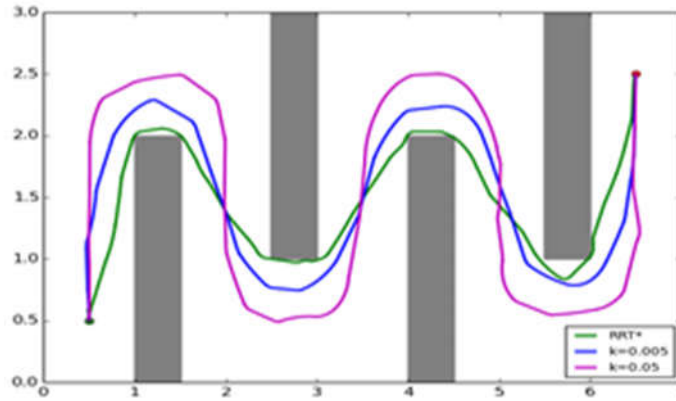
RRT* ve Safe-RRT* yöntemlerinin büyük ölçekli haritadaki ağaç yapısı ve verilen hedef için bulduğu yollar Şekil 4.7'de görüldüğü gibidir. Ortam haritası robot yarıçapı boyutunda genişletilmiş ve robot, nokta robot düzeyine indirilmiştir. Bu genişletme, robotun orta noktasının ulaşamayacağı bölgeleri elemek için kullanılmıştır. Şekil 4.7 'de görüldüğü üzere simülasyon sonuçlarına benzer şekilde Safe-RRT* yöntemi ile engellere daha uzak yollar bulunmuştur. Ağaç yapısı da Safe-RRT* algoritmasında koridorların orta bölümlerinde yoğunlaşırken, klasik RRT* algoritmasında ise boş bölgelere eşit olarak dağılmıştır.



(a) 1000 Düğüm sonucu



(b) 2500 Düğüm sonucu



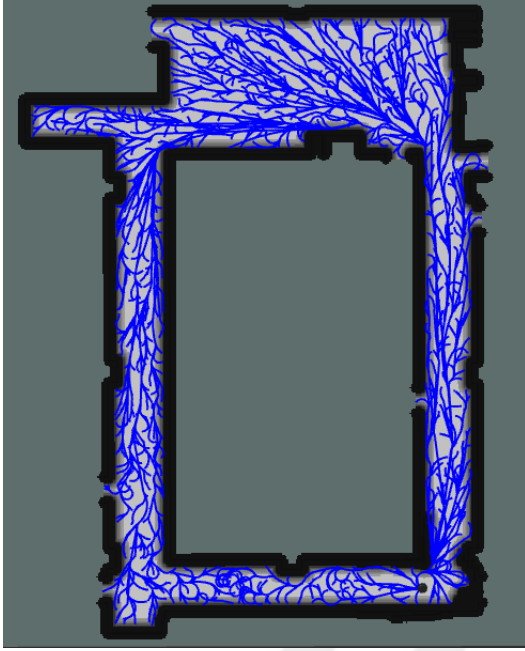
(c) 5000 Düğüm sonucu

Şekil 4.6 : Safe-RRT* yöntemi sonucunda bulunan yollar. Yeşil çizgiler (Klasik RRT*), mavi çizgiler ($k=0.005$) ve magenta çizgiler ($k=0.05$) katsayıları sonucu bulunan yolları göstermektedir.

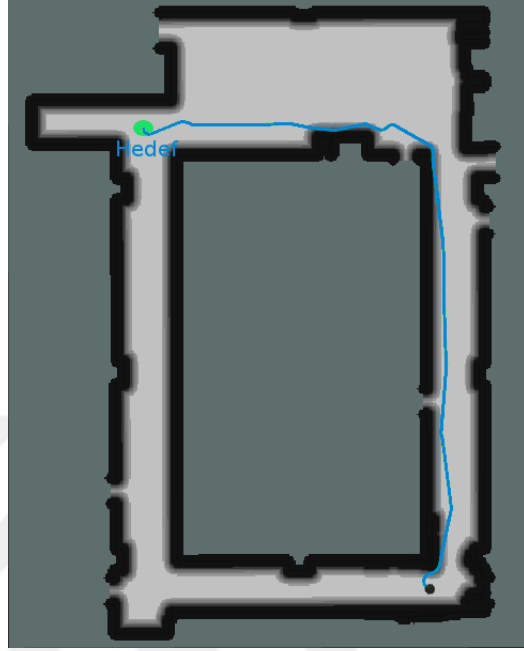


Çizelge 4.1 : RRT* ve Safe-RRT* yöntemleri ile bulunan yol uzunlukları ve engellere mesafe değerleri.

k değeri	1000 Düğüm		2500 Düğüm		5000 Düğüm	
	Yol Uzunluğu	Engellere Mesafe	Yol Uzunluğu	Engellere Mesafe	Yol Uzunluğu	Engellere Mesafe
k=0 (RRT*)	11.32m	0.246m	10.66m	0.225m	10.50m	0.217m
k=0.005	12.37m	0.334m	11.77m	0.331m	11.55m	0.330m
k=0.01	12.97m	0.370m	12.51m	0.374m	12.33m	0.373m
k=0.025	13.95m	0.410m	13.69m	0.430m	13.55m	0.440m
k=0.05	14.35m	0.420m	14.13m	0.450m	14.03m	0.460m



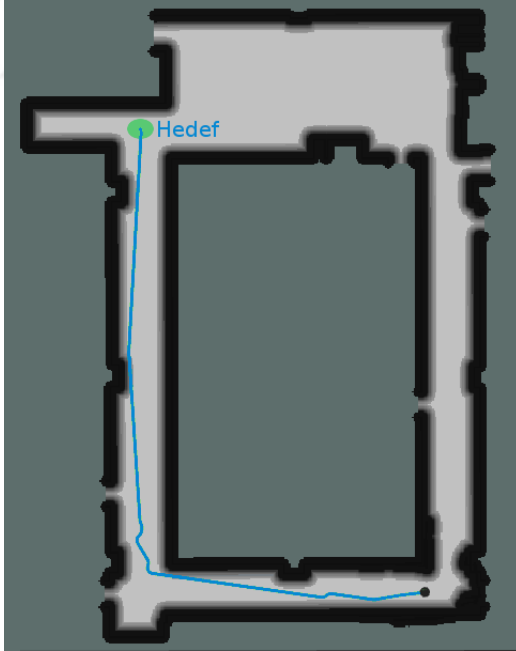
(a) RRT* ağaç yapısı.



(b) RRT* ile bulunan yol.



(c) Safe-RRT* ağaç yapısı



(d) Safe-RRT* ile bulunan yol

Şekil 4.7 : Safe-RRT* ($k=0.05$) ve RRT* yöntemi ile 3000 düğüm sayısında oluşan ağaç yapıları ve bulunan yollar.

5. YOL TAKİBİ, ENGELDEN KAÇMA VE ÖNERİLEN FGM-DW YAKLAŞIMI

RRT* ve Safe-RRT* yöntemleri ile bulunan yolun robot tarafından takip edilmesi için yol takip yöntemlerine ihtiyaç vardır. Bu yöntemler ile elde edilen kontrol sinyallerinin, robot modelinin kusursuz oluşturulmuş ve çevrenin robot hareketine hiçbir etkisi olmadığı durumlarda açık çevrim bir şekilde uygulandığında, robotu hedefe ulaştırabileceğini söyleyebiliriz. Aksi durumda sürekli olarak konumun kontrol edilip en uygun kontrol sinyalinin hesaplandığı kapalı çevrim bir yörünge takip veya engelden kaçınma yöntemine ihtiyaç duyulur.

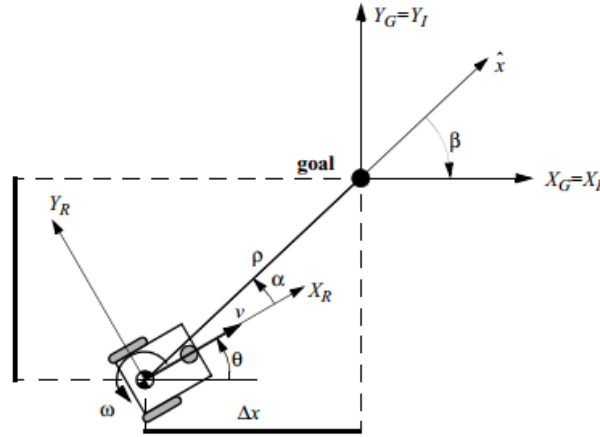
Yörünge takip yöntemleri, robotun yörünge planlama yöntemi sonucu elde ettiği yolları düzgün bir şekilde takip etmesi için tasarlanmıştır. Bu yöntemler çevrede olan biten olayları yorumlamadan sadece robotun iç dinamiklerini kullanarak yörünge takip işlemini gerçekleştirmek üzere tasarlanmıştır. Genel olarak robot modeli, dönme yarıçapları ve tekerlek hızlarını analitik bir şekilde ele alarak yörüngelerin takibini gerçekleştirmektedirler.

Çevre modelinin (örn. hareketli nesnelere, haritadaki hatalar, zemine bağlı tekerleklerdeki kayma vs.) ve robot modelinin içerdiği hata ve belirsizlikler yörünge planlama yöntemlerini dinamik çevrelerde yetersiz kılmaktadır. Engelden kaçınma yöntemlerinin amacı ise sensörler ile algıladığı çevrede robotu hedef konumuna yönlendirirken engellerden kaçınmasını sağlayan en iyi kontrol sinyallerini her zaman adımı için hesaplamaktır. Bu yöntemler harita bilgisine ihtiyaç duymadıklarından yol planlama yöntemlerine göre oldukça hızlı çalışmaktadır. Bu yöntemlerin en önemli dezavantajı ise haritanın geneline ulaşamadığından onu hedefe ulaştıracak ipuçlarına sahip olmamasıdır. Bu yüzden, ancak yol planlama yöntemleri ile birlikte çalıştırıldığında zor navigasyon problemlerini (labirent problemi gibi) çözebilirler.

İlk olarak yol takip yöntemi ve bu yöntemde kullandığımız robot modeli incelenecektir. Daha sonra ise iki adet engelden kaçınma yönteminin kritik bulduğumuz özelliklerini içeren yeni bir engelden kaçınma yöntemi önerilip sonuçları incelenecektir.

5.1 Robot Hareket Modeli

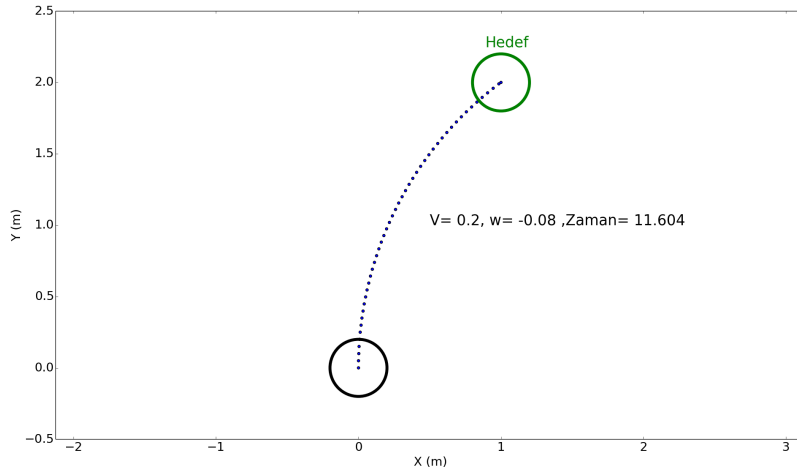
Robot hareket modeli, yörünge planlama, yol takip ve engelden kaçınma yöntemlerinde sıkça kullanılmaktadır. Bu modelin kullanımının asıl amacı robotun hareket etmeden kontrol sinyaline vereceği tepkinin önceden kestirilmesini sağlamaktır. Diferansiyel sürürlü bir robot için bir sonraki konum denklem 5.1'de görüldüğü gibi hesaplanır. Burada robotun orta noktasının Δs kadar yol aldığı, $\Delta\theta$ kadar açı değişimi yaşadığı varsayılarak x, y, θ eski konumundan, sonraki konumu x', y', θ' hesaplanır [30]. Diferansiyel sürürlü robotun koordinat eksenindeki hareketi Şekil 5.1'de görüldüğü gibidir.



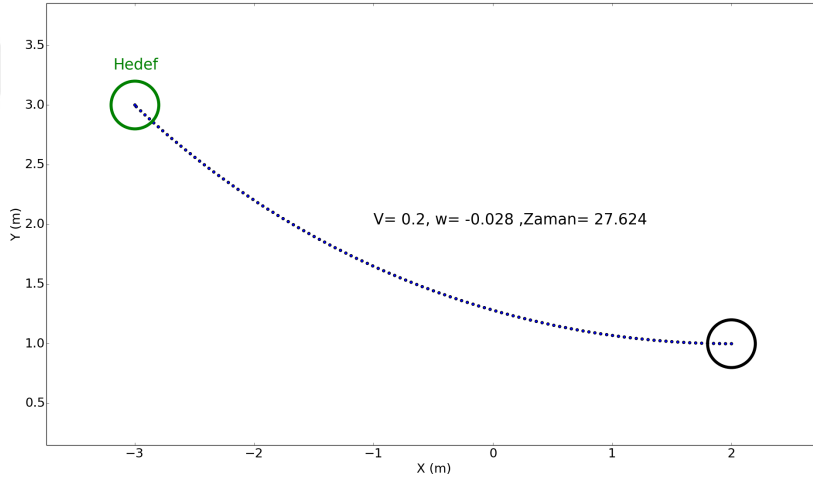
Şekil 5.1 : Robotun koordinat eksenindeki hareketi.

$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} \quad (5.1)$$

Robotun orta noktasının aldığı yol, yol alma süresine bölüldüğünde çizgisel hız (V), açı değişiminin süreye bölünmesiyle açısal hız (w) elde edilebilir. Bu eşitlikleri kullanarak robotun bir konumdan diğer konuma ulaşabilmesi için gerekli açısal ve çizgisel hızları bulmak için dönme yarıçapı teriminin incelenmesi gerekir.



Şekil 5.3 : Hareket modeli ile bulunan yol ve kontrol sinyalleri-1.



Şekil 5.4 : Hareket modeli ile bulunan yol ve kontrol sinyalleri-2.

Oluşturulan hareket modeli ile, robotun yönelimi bilinen bir pozisyondan yönelimsiz bir noktaya ulaşılması için gerekli açısal, çizgisel hız ve süre değişkenleri hesaplanmıştır. Denklem 5.1'deki Δs yerine $V * \Delta t$, $\Delta \theta$ yerine ise $w * \Delta t$ ifadeleri yerleştirilerek Δt aralıklarla üzerinden geçilen yörünge noktaları hesaplanmıştır. Şekil 5.3 ve şekil 5.4'de oluşan yörünge noktaları ve gerekli kontrol sinyalleri gösterilmiştir.

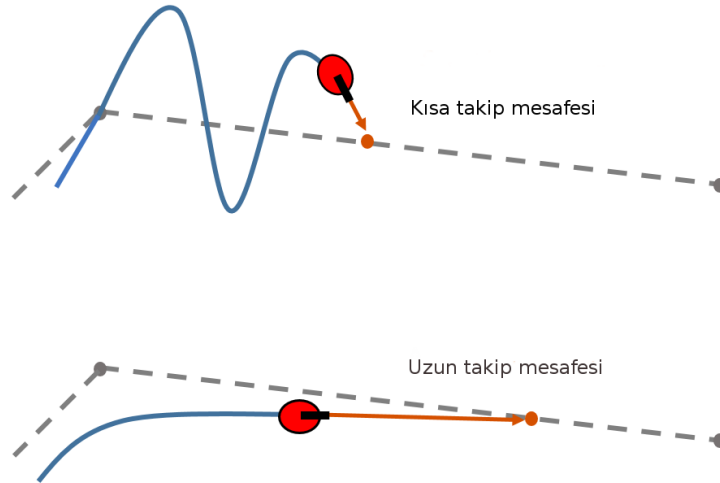
5.2 Pure Pursuit

Pure Pursuit yaklaşımı robotu kendi konumundan yol üzerindeki bir hedef noktaya ulaştırabilecek kontrol sinyallerinin hesaplanması üzerine kurulmuştur. Bu hedef nokta, yörünge üzerinde robota en yakın noktanın belirli bir mesafe ilerisinde seçilir. Bu yöntemle robotun konumunun değişmesiyle birlikte hedef nokta da tekrar hesaplanmaktadır. Robotun hareket modeli kullanılarak bu hedef noktasına

ulaşabilecek kontrol sinyalinin bulunması işlemi gerçekleştirilir. Bu yöntem aşağıdaki adımlardan meydana gelir [31].

- Aracın konumunun belirlenmesi
- Takip edilecek yol üzerinde araca en yakın noktanın hesaplanması.
- Bu noktanın yol üzerinde takip mesafesi kadar ileride bulunan hedef noktasının hesaplanması.
- Hedef noktaya ulaşılması için gerekli kavisli yolun bulunması ve bu eğime göre robotun yönlendirilmesi.
- Ana hedefe ulaşıncaya kadar ilk aşamaya dönülmesi.

Bu yöntem ile, belirlenen takip mesafesinin artırılması yol üzerinde bulunan köşe noktalarından daha kavisli geçilmesine, azaltılması ise daha dalgalı şekilde yolun takip edilmesine sebep olabilir. Bu durum Şekil 5.5'de görülmektedir. Bunun dışında aracı başka bir konuma ulaştıracak kavisinin hesaplanması için kullanılacak hareket modeli ile hesaplanan değerlerin, gerçek değerler ile tutarlı olması gerekmektedir.



Şekil 5.5 : Takip mesafesinin takip kalitesine etkisi [32].

5.3 Engelden Kaçınma

Engelden kaçınma yöntemleri, robot çevresindeki engellerin bilgisini kullanarak hedef noktasına varması için yapması gereken hareketlere karar veren yöntemlerdir. Bu yöntemler kısıtlı bir bilgi yardımıyla robotu hedefe ulaştırabildiklerinden, yol planlama yöntemlerine göre hızlı çalışmaktadırlar.

5.3.1 Literatürde bulunan yöntemler

Engelden kaçınma yöntemleri verdikleri çıkışlara göre iki kategoride incelenebilir. Bunlar yönelim tabanlı yaklaşımlar ve hız uzayı yaklaşımlarıdır.

Yönelim tabanlı yaklaşımlar robotun engelsiz alanda hareketin sağlanması için yönelmesi gereken en uygun açının her zaman adımında tekrar hesaplanması üzerine kurulmuştur. Yapay Potansiyel Alan (APF) [33], Vektör Alan Sıklık Grafiği (VFH,VFH+) [34,35], Yakınlık Şeması (ND) [36], Engel Kısıtlama Metodu (ORM) [37] ve Boşluğu Takip Etme Metodu(FGM) [38] bu yaklaşımlardan en bilinenleridir. APF ve VFH yöntemleri robota engeller tarafından uygulanan itici ve hedef tarafından uygulanan çekici kuvvetlerden çıkarım yaparak en uygun yönelim açısının hesaplanmasını sağlamaktadır. ORM metodu ise engel köşelerinden çıkarım yaparak oluşturduğu yardımcı hedef noktalarını ve engellerin uzaklık ve açı bilgilerini kullanarak en uygun yönelimi hesaplamaktadır. ND ve FGM yöntemleri ise engeller arasındaki boşuklardan yola çıkarak robotun güvenli bir şekilde hedefe ulaşması için yönelmesi gereken açının bulunması için ortaya atılmış yöntemlerdir. Bu yöntemler her ne kadar basit ve efektif bir şekilde uygun yönelimi hesaplasa da robotun dinamiklerini hesaba katmakta yetersiz kalmaktadırlar.

Hız uzayı tabanlı yaklaşımlar robot dinamiklerini hesaba katan özelliklerle donatılmıştır. Bu yöntemler robotun hareketini bir açısal, çizgisel hız çifti (v,w) tarafından oluşturulmuş yay parçaları olarak kabul eder. Eğri Hız Metodu (CVM) [39], Engel Hızları (VO) [40] ve Dinamik Pencere Yaklaşımı (DWA) [41,42] bunlardan en bilinenleridir. CVM yöntemi ilk olarak robotun hareket edebileceği eğri parçalarının kümesini bulur ve bu kümeden engeller ile çarpışma ihtimali olanlarını eler. Kalan eğriler içinden amaç fonksiyonuna göre en yüksek skora sahip eğriyi oluşturabilecek hız çiftini hesaplar. Bu işlem diğer engelden kaçınma yöntemleri gibi her zaman aralığında yeniden hesaplanmaktadır. VO yöntemi ise çevredeki engellerin ve robotun

hız uzayındaki hızını kullanarak çarpışmaya sebep olabilecek hız kümesini belirler. Daha sonra hız uzayında çarpışmaya neden olmayacak hedefe en iyi yaklaşabilen hızın seçilmesi temeli üzerine kurulmuştur. DWA yöntemi ise ilk olarak robotun bir sonraki zaman aralığında hız uzayında ulaşabileceği hız çiftlerini hesaplar. Daha sonra bunlar arasından engellere çarpmadan duramayacak hız çiftlerini eler. Son olarak kalan hız çiftleri amaç fonksiyonunda değerlendirip en uygun hız çiftini hesaplanması gerçekleştirilir. Bu yöntemlerin de bölgesel minimuma takılma, sadece ileri hız çiftlerini kullanma gibi problemleri vardır.

Bu tezde FGM ve DWA'nın özelliklerini kullanarak bir hibrit engelden kaçınma metodu önerilmiştir. Robotu algıladığı bölgede en büyük boşluğa sahip ve hedefe yakın bölgeye yönlendirmeyi garanti etmesinden dolayı FGM metodu kullanılmıştır. DWA ise sahip olduğu amaç fonksiyonu ve robot dinamiklerini göz önüne alan yapısından dolayı karar verici ve kontrolcü olarak kullanılmıştır. İlk olarak FGM yönteminden bahsedilecek, daha sonra DWA'nın çalışma mantığı ve özelliklerine değinilecek, son olarak ise iki yöntemin çeşitli özelliklerini alarak oluşturduğumuz FGM-DW yöntemi incelenecektir.

5.3.2 FGM yöntemi

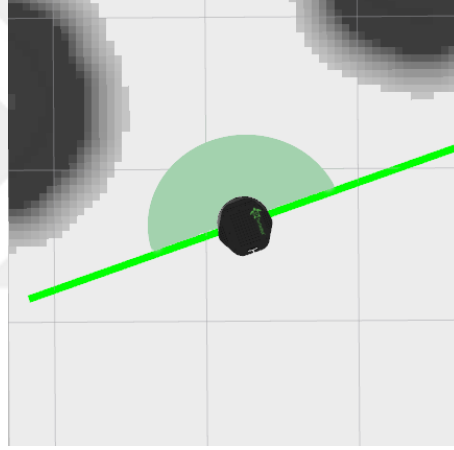
FGM yöntemi her zaman aralığında tekrarlamalı olarak en geniş boşluk merkezinin açısını, robota en yakın engelin uzaklığını ve hedefin robota göre açısını kullanarak robot için yönelim açısı hesaplamaktadır. Bu açı hedef açısı ile boşluk merkezi açısı arasında güvenlik katsayısı kullanılarak hesaplanmaktadır.

Bu yöntemin ilk aşamasında çevredeki engeller robot yarıçapı oranında genişletilir. Bu genişletilme işlemi engeller arasında kalan her boşluğun robot için güvenli olmasını garanti etmektedir. İkinci aşamada ise boşlukların hesaplanma aşaması gerçekleştirilmektedir. Alınan uzaklık ölçümlerinin kutupsal dönüşümü yapıldıktan sonra gözlemlenen uçurum ve ani yükseltiler boşluk bölgelerinin köşeleri olarak hesaplanmaktadır. Robot yarıçapı oranında engellerin karesel ayrılmış harita üzerinde genişletilmesi ve robotun boşluk taraması için kullandığı açısız alan Şekil 5.6'da görüldüğü gibidir. Bu engellerin robot açısına göre kutupsal dönüşümü yapıldığında Şekil 5.7'de görülen grafik elde edilmiştir.

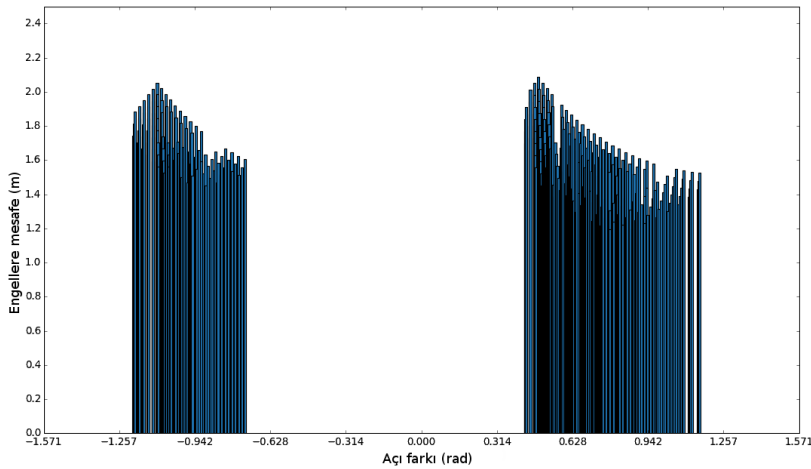
Kutupsal dönüşüm grafiğinden engel köşelerine ulaşmak istendiğinde sert yükseliş ve düşüşlerin olduğu boşluk kenarlar bulunabilmektedir. Bulunan bu boşluk kenarlarının açısı ve uzaklığı bilindiğinden boşluk kenarlarının koordinatları, boşluk genişliği gibi bilgilere de ulaşmak mümkün olmaktadır. En büyük boşluğun orta noktasının ve hedef noktasının açıları ile robot yönelimi arasındaki açı farkları kullanılarak FGM'nin robot için hesapladığı en uygun yönelim açısı Denklem 5.4'deki bağıntıdan bulunmaktadır.

$$\theta_{son} = \frac{\frac{\alpha}{d_{min}} * \theta_{bosluk} + \theta_{hedef}}{\frac{\alpha}{d_{min}} + 1} \quad (5.4)$$

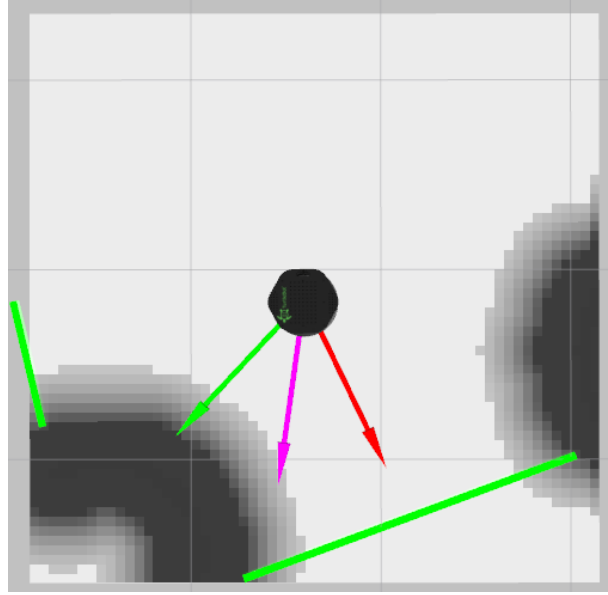
FGM yöntemi en uygun yönelmesi gereken açıyı α parametresine ve d_{min} değişkenine göre hesaplamaktadır. Engellerin robota minimum uzaklığı d_{min} ile, güvenlik katsayısı α ile gösterilmiştir. Şekil 5.8'de örnek FGM sonucu bulunan θ_{son} gösterilmiştir.



Şekil 5.6 : Engellerin robot yarıçapında genişlemesi ve robotun tarama yaptığı bölge.



Şekil 5.7 : Kutupsal dönüştürülmüş engel uzaklıklarının grafiği.



Şekil 5.8 : Hedef açısı (yeşil ok), boşluk açısı (kırmızı ok) ve FGM sonucu hesaplanan hedef açısı (pembe ok) gösterimi.

5.3.3 DWA yöntemi

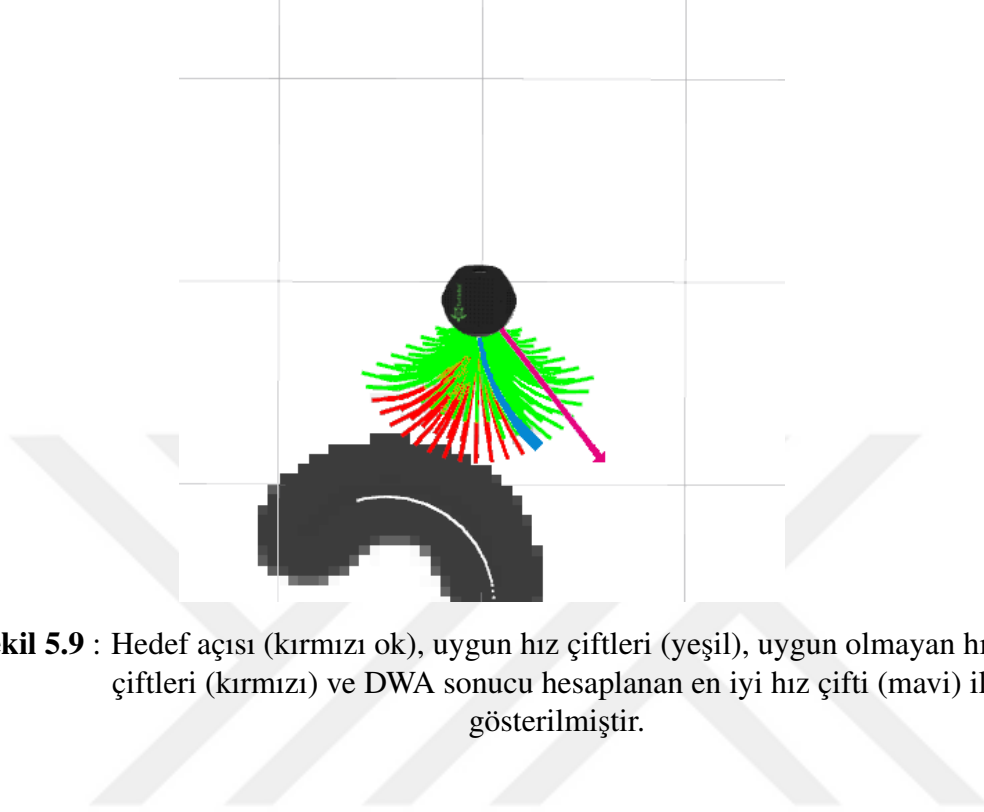
DWA yöntemi robot dinamiklerini hesaba katan hız uzayı yaklaşımlarından bir engelden kaçınma yöntemidir. DWA ilk aşamada robotun o anki açısal ve çizgisel hızından yola çıkarak belirli bir zaman aralığında maksimum açısal ve çizgisel ivmelenme sonucu ulaşabileceği hızları hesaplar. Bu hız penceresi, içerisinde engellere çarpabilecek ve çarpmadan duramayacak hız çiftlerini eler.

İkinci aşamasında ise birinci aşamadan geçen hız çiftlerinin hepsini Denklem 5.5'de görülen amaç fonksiyonuna göre değerlendirir. Bu denklemde yön fonksiyonu hız çiftinin zaman aralığı sonundaki açısının hedef açısına yakınlığını, mesafe fonksiyonu hız çiftinin zaman aralığı boyunca uygulandığında oluşturacağı yörüngenin engellere ne kadar yakın geçeceğini, hız fonksiyonu ise hız çiftinin ne kadar büyük çizgisel hızla sahip olduğunu hesaplar.

$$G(V, w) = \sigma[\alpha * \text{yon}(V, w) + \beta * \text{mesafe}(V, w) + \gamma * \text{hiz}(V, w)] \quad (5.5)$$

Bu yöntemde bu fonksiyonların ağırlıklarına göre en uygun hız çiftinin aranması gerçekleşir. Genel anlamda DWA'nın en hızlı, engellere en uzak geçecek ve hedefe en iyi yönelecek hız çiftini bulma amacı taşıdığını söyleyebiliriz. Bahsedilen fonksiyonlarda farklı metrikler kullanıldığından bu fonksiyonların [0:1] arasında

değerler vermesi için düzleştirme operatörü (σ) kullanılmıştır. DWA sonucu elde edilmiş örnek elenen hız çiftleri, uygun hız çiftleri ve en uygun bulunan hız çifti, Şekil 5.9'da görüldüğü gibidir.

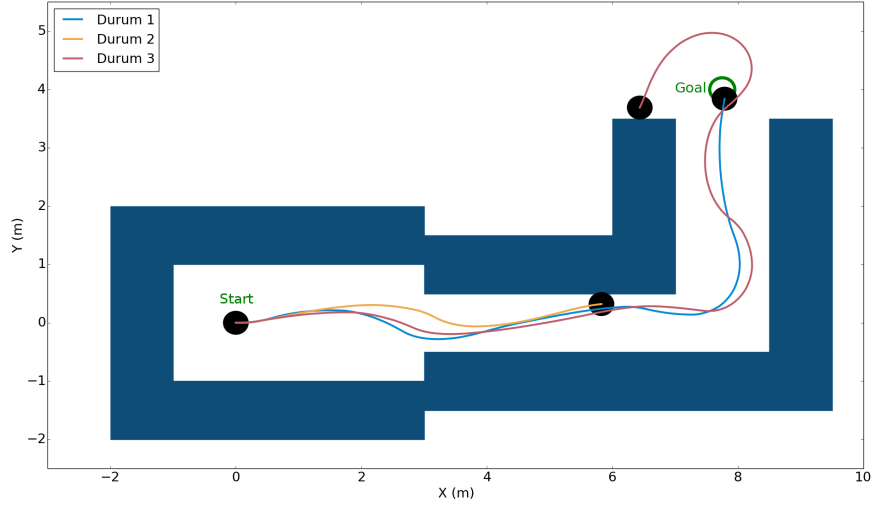


Şekil 5.9 : Hedef açısı (kırmızı ok), uygun hız çiftleri (yeşil), uygun olmayan hız çiftleri (kırmızı) ve DWA sonucu hesaplanan en iyi hız çifti (mavi) ile gösterilmiştir.

5.3.4 FGM-DW yöntemi

FGM yöntemi yönelim tabanlı yaklaşım olduğundan robotu güvenli boşluklara yönlendirse dahi robotun bu yöneline nasıl ulaşacağı ile ilgili bir hesaplama yapmamaktadır. Robotun hareket hızının yüksek olduğu durumlarda aniden gördüğü engeller, FGM sonucu hesaplanan hedef açısının birden değişmesine neden olacağından robota salınımlı hareketler yaptırmaktadır. Bu salınımlar robotların dar bölgelerde kendi iç dinamikleri sebebiyle istediği dönüşü yapamamasına ve engeller ile çarpışmasına neden olmaktadır. Dar koridorlardan oluşan bir alanda Turtlebot için FGM algoritmasınının 0.5m/sn sabit hızda ve P kontrolör ile alınan sonuçlar Şekil 5.10'da görüldüğü gibidir. Robotun hızlı hareket etmesi FGM sonucu bulunan açılarının ani ve fazla değişimine sebep olmasından, engellere yakın bölgelerde yavaş ve salınımsız hareket etmesi istenir. Ayrıca bu salınımlardan doğan hedef bölgenin kaçırılması da robot hızının yavaşlatılmasıyla düzeltilebilmektedir.

DWA yöntemi en hızlı, güvenli ve hedefe yönelen hız çiftini arayan ve robot dinamiklerini kullanan bir hız uzayı yaklaşımıdır. Ayrıca DWA engellere yaklaşıldığında,



Şekil 5.10 : FGM yönteminin koridor ortamında P kontrolör ile oluşan üç farklı durum.

yüksek değere sahip hız çiftlerinin engellere çarpmadan duramayacağını bildiğinden dolayı yavaşlama eğilimindedir. DWA yöntemi engeller robottan uzaklaştığında ulaşabileceği en yüksek hızı seçmeye çalışmaktadır. Ayrıca hız çifti seçimini kendisi yaptığından ayrıca bir kontrolcüye ihtiyaç duymamaktadır.

Yukarıdaki sebeplerden dolayı FGM yönteminin DWA özellikleri ile birleştirilmesi planlanmıştır. FGM genel anlamda robotu engellere yaklaşımadan güvenli bölgelere yönlendirdiğinden, DWA içerisinde bulunan ve yörüngelerin güvenliğinden sorumlu olan mesafe fonksiyonuna ihtiyaç duyulmamıştır. Yeni hibrit yöntemde DWA direkt olarak hedef nokta yerine FGM tarafından hesaplanan uygun yönelim açısına yönlendirilir. Bu sayede DWA dinamik kontrolcü yapısı ve güvenlik sağlayıcı özellikleri ile FGM'yi desteklemektedir.

Yöntem üç aşamadan oluşmaktadır. Birinci aşamada FGM yöntemi ile yönlendirici açının hesaplanması gerçekleştirilir. İkinci aşamada ise DWA yöntemi ile ulaşılabilir ve engeller ile çarpışmayacak hız çiftlerinin bulunması işlemi gerçekleştirilir. Son aşamada ise bu hız çiftlerinin amaç fonksiyonuna göre değerlendirilmesi gerçekleştirilmektedir.

5.3.4.1 FGM ile yönlendirici açının hesaplanması

Yönlendirici açı daha önce FGM yönteminde bahsedildiği gibi Denklem 5.4'de bulunan bağıntıdan hesaplanır. Bu açı, hedef açı ve boşluk açısı arasında α katsayısı ve d_{min} değişkenine bağlı yönelim açısıdır. Bunlardan α 'nın yüksek seçilmesinde

yönlendirici açı en büyük boşluğun merkezine yakın, düşük seçilmesi durumunda ise hedef açısına yakın hesaplanmaktadır. Bu açı engellerin dizilimlerinin ve yakınlığının zamanla değişken olmasından dolayı her zaman aralığında tekrar hesaplanmalıdır.

5.3.4.2 Ulaşılabilir ve çarpışmasız hız çiftlerinin hesaplanması

DWA yöntemi robotun bir sonraki zaman diliminde ulaşılacağı açısal ve çizgisel hız kümesini robota uygulanabilecek maksimum açısal ve çizgisel ivmelenmeler kullanarak hesaplar. Denklem 5.6' da bu hızların nasıl hesaplandığı görülmektedir. Robotun anlık hızı (V_n, w_n) ile, maksimum ivmelenmeler ise (\dot{V}, \dot{w}) ile gösterilmiştir. Zaman aralığı t ile gösterilmiştir. Bu hız kümesi hız uzayında bir pencereyi temsil etmektedir.

$$V_r = \{(V, w) | V \in [V_n - \dot{V} * t, V_n + \dot{V} * t] \wedge w \in [w_n - \dot{w} * t, w_n + \dot{w} * t]\} \quad (5.6)$$

Uygun hız kümesinin hesaplanması için elde edilen bu ulaşılabilir hız kümesi içerisinde engellere çarpmadan duramayacak olanların elenmesi gerekir. Bunu hesaplamak için ise robotun yavaşlama ivmeleri $(\dot{V}_{fren}, \dot{w}_{fren})$ ile oluşacak yörünge ile engeller arasındaki minimum mesafe kullanılır. Uygun hız kümesinin bulunması için bütün hız çiftleri Denklem 5.7'deki koşula göre değerlendirilir. Elde edilen bu küme sadece ulaşılabilir ve çarpışmasız hız çiftlerini içermektedir.

$$V_u = \left\{ (V, w) | V \leq \sqrt{2 * mes(V, w) * \dot{V}_{fren}} \wedge w \leq \sqrt{2 * mes(V, w) * \dot{w}_{fren}} \right\} \quad (5.7)$$

5.3.4.3 Amaç fonksiyonuna göre değerlendirme

DWA yönteminde önerilen amaç fonksiyonunda daha önce bahsedildiği gibi üç adet değerlendirme parametresi vardır. Güvenlik kaygısı FGM tarafından α katsayısı ile sağlandığından önerilen yeni engelden kaçınma yönteminin amaç fonksiyonu sadece en hızlı ve hedef açısına en iyi yaklaşan hız çiftini seçmesi için özelleştirilmiştir.

Hız parametre fonksiyonu, amaç fonksiyonu içerisinde karşılaştırılan hız çiftlerini çizgisel hızına göre derecelendirilmesini sağlamaktadır. Bu fonksiyon, hız çiftlerini [0:1] arası derecelendirmesi için Denklem 5.8'de görüldüğü gibi oluşturulmuştur.

$$hiz(V, w) = \begin{cases} \frac{\|V\|}{V_{max}} & \text{robot hedefe uzaksa} \\ 1 - \frac{\|V\|}{V_{max}} & \text{robot hedefe yakinsa} \end{cases} \quad (5.8)$$

Yönelim parametre fonksiyonu ise hız çiftlerinin zaman aralığı sonunda ulaşacağı açı ile FGM tarafından elde edilen yönlendirici açı arasındaki farkının ($\delta\theta$) derecelendirilmesini sağlar. Bu derecelendirmenin de [0:1] arası yapılması için Denklem 5.9'da görülen hesaplama kullanılmaktadır. Bu yöntem için yeni amaç fonksiyonu ise Denklem 5.10'da görüldüğü gibidir.

$$yon(V, w) = 1 - \frac{|\delta\theta|}{\pi} \quad (5.9)$$

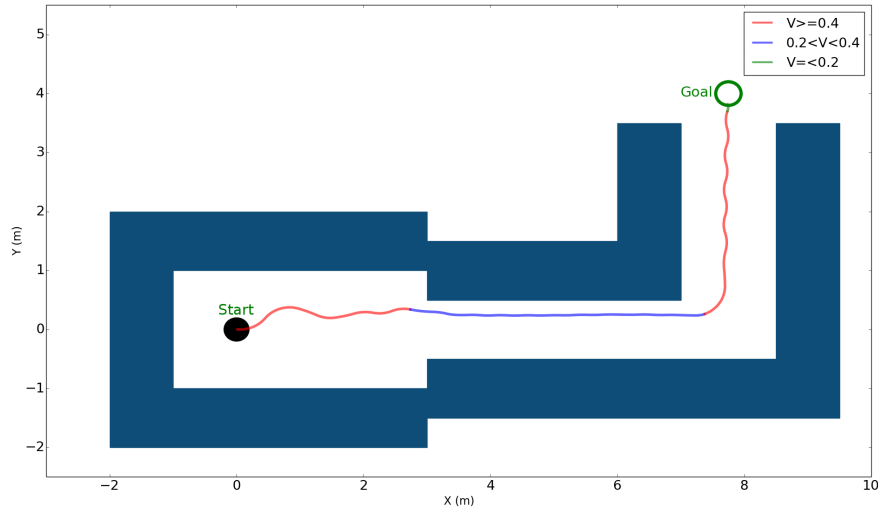
$$G_{yeni}(V, w) = \beta * yon(V, w) + \gamma * hiz(V, w) \quad (5.10)$$

Şekil 5.10'da uygulanan FGM örneği ile önerilen bu metodun farkını göstermek amacıyla aynı problemin önerilen metodla çözülmesi sağlanmıştır. Bu yöntem ile elde edilen yol ve robotun hızının zamana göre değişimi Şekil 5.11 ve 5.12'de görüldüğü gibi gözlemlenmiştir. Bu sonuçlara göre önerilen yöntem robot geniş bölgelerde hareket ettiğinde yüksek hızları, dar bölgelerde ise daha düşük hızları tercih etmektedir. Bu sayede Şekil 5.10'da görülen salınımlar dar bölgelerde azaltılmış, robotun engel köşelerine çarpması engellenmiştir. Ayrıca hız parametre fonksiyonunun hedefe belirli miktar yaklaşıldığında yavaş hızları daha çok ödüllendirdiğinden hedef bölgeyi kaçırmayı önlenmiştir.

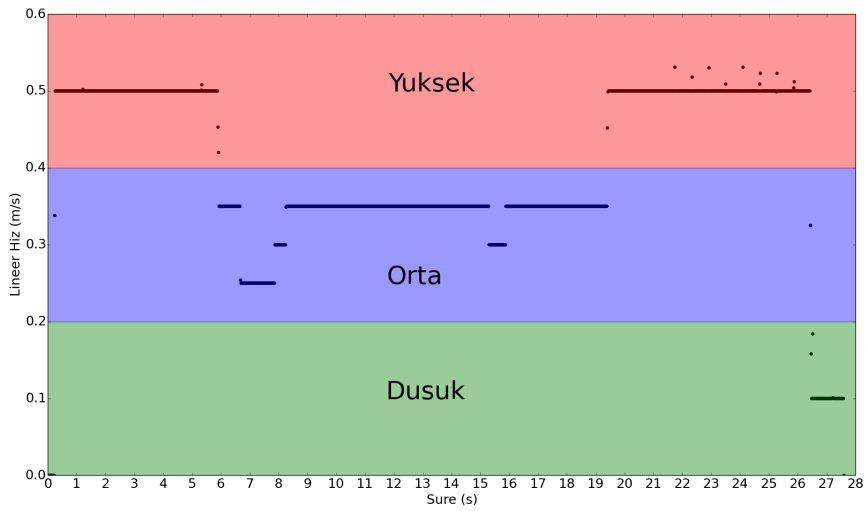
Robot hızının değişim grafiğinde görüldüğü üzere, robot engellere uzak olduğunda hızlı, dar bölgelerden geçerken ise orta hızlarda seyretmiştir. Hedef noktaya belirli bir yarıçapta yaklaştığında ise hızını düşürüp engele yönelmiştir.

Önerilen yöntem ayrıca Gazebo üç boyutlu gerçek dünya simülasyon ortamında Turtlebot platformu ile test edilmiş ve oluşturulan iki farklı dünyada farklı α parametreleri ile takip edilen yollar Şekil 5.13'de gösterilmiştir.

Gerçek dünya değişkenleri ve ölçümleri ile bu yöntemin nasıl sonuçlar ürettiğini gözlemlemek amacıyla laboratuvar ortamında bu yöntemin aynı güvenlik parametreleri ile test edilmesi gerçekleştirilmiştir. Başlangıç ve bitiş noktaları arasında kalan boş alana çapı bilinen silindirik engeller yerleştirilmiş ve yöntemin oluşturduğu hareketi



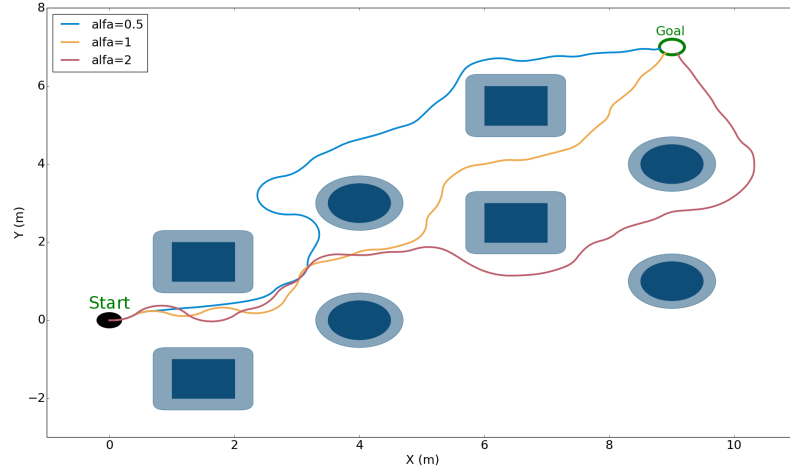
Şekil 5.11 : FGM-DW yöntemi sonucu robotun takip ettiği yol.



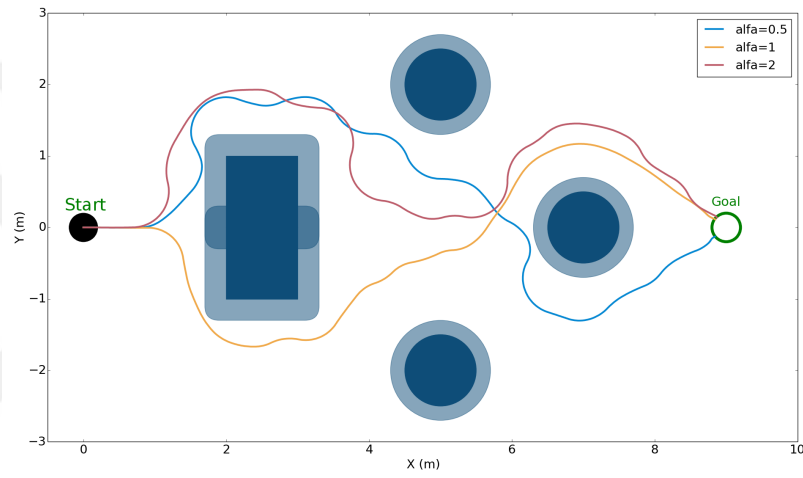
Şekil 5.12 : Robot çizgisel hızının zamana göre değişimi.

gözlemlemek amacıyla konum bilgisi kaydedilmiştir. Çalışılan ortam ve elde edilen sonuçlar sırasıyla Şekil 5.14 ve 5.15’de görüldüğü gibidir.

Şekil 5.15’de görüldüğü üzere robot farklı alfa parametreleri için engellere farklı yakınlıkta yollar takip etmektedir. Önerilen yöntem aynı zamanda robotun hız seçimini yapmakta ve engeller ile çarpışmayı önceden tahmin edebilmektedir. Bu sayede, [43] ve [44]’de önerilen yöntemlerdeki gibi harici bir hız planlayıcısı kullanmaya gerek kalmamıştır. Bu sonuçlardan yola çıkılırsa, alfa parametresinin artması, hedefe ulaşmak için gidilmesi gereken yollun uzamasına sebep olmaktadır. Robotun engellerden uzak seyahat etmesinden dolayı, FGM-DW yöntemi robotu yüksek hızlarda hareket ettirmektedir. Düşük katsayılarında ise tam tersi şekilde robotun engellere yaklaştığı durumlarda FGM-DW tarafından düşük hızlar tercih edilir. Sonuç



(a) FGM-DW ile simülasyon sonucu (ortam-1).



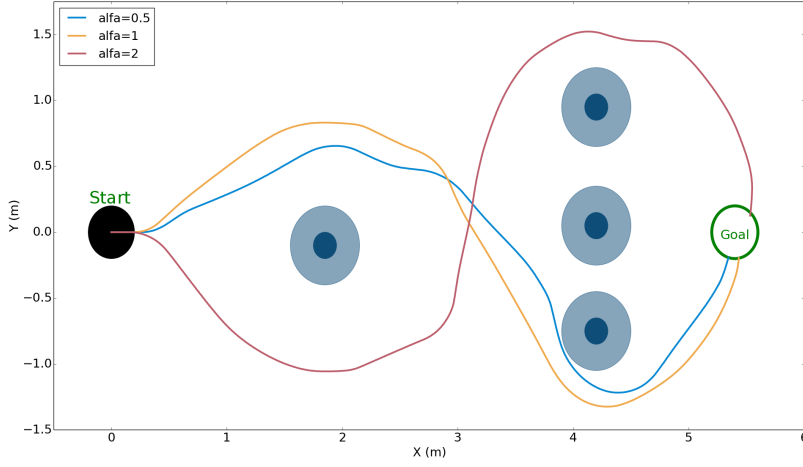
(b) FGM-DW ile simülasyon sonucu (ortam-2).

Şekil 5.13 : FGM-DW ile farklı güvenlik parametreleri ($\alpha=0.5,1,2$) için elde edilen simülasyon sonuçları.

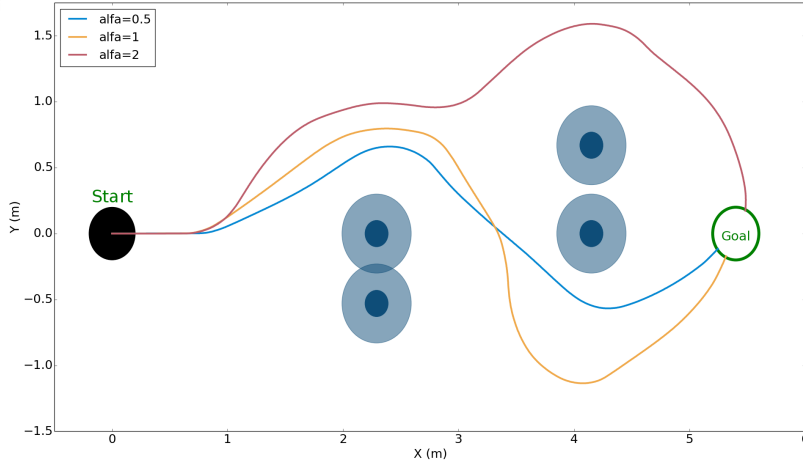


Şekil 5.14 : Gerçek deneylerin yapıldığı test ortamı.

olarak FGM gibi çalışan fakat robot dinamikleri ve modelinin de hesaba katıldığı hızlı ve güvenli bir engelden kaçınma yöntemi elde edilmiştir.



(a) FGM-DW ile gerçek ortam sonuçları (ortam-3).

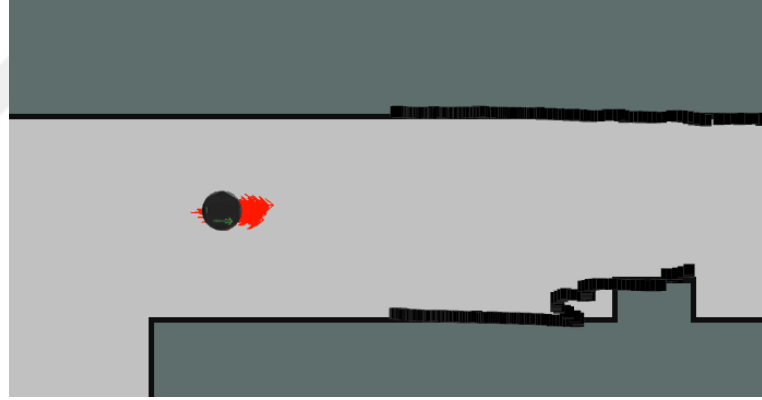


(b) FGM-DW ile gerçek ortam sonuçları (ortam-4).

Şekil 5.15 : FGM-DW ile gerçek ortamda farklı güvenlik parametreleri ($\alpha=0.5,1,2$) için elde edilen sonuçlar.

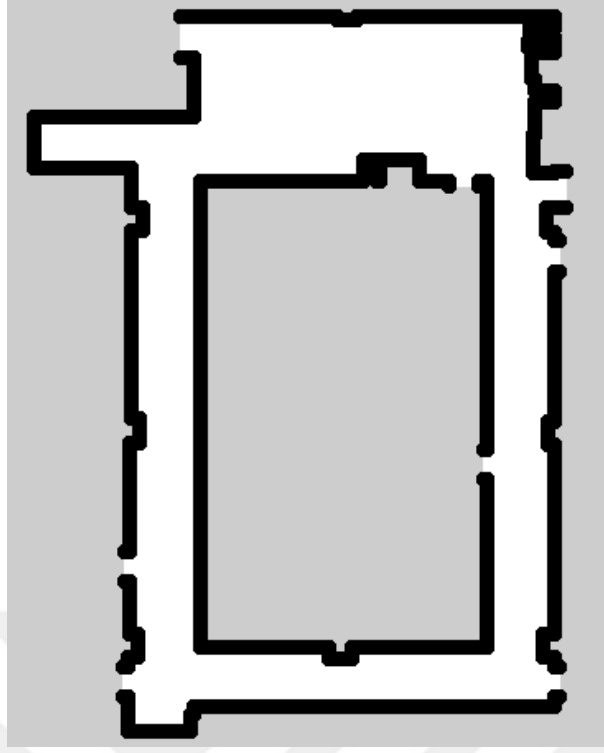
6. DEĞİŞMEZ VE DEĞİŞKEN ORTAMLARDA NAVİGASYON DENEYLERİ

Bu çalışmada, mobil robotlarda hareket planlama problemi çeşitli açılardan ele alınmıştır. Bu problem ilk olarak robotun kendini konumlandırmasını gerektirir. Robotların yürüyen aksamındaki kusurlar, algılayıcı hataları, çevresel zorluklar bu problemin birden çok algılayıcıdan alınan bilgiyle çözülmesini gerektirmektedir. Konumu bilinmeyen bir aracın üzerinden yörünge planlamak ve takibini gerçekleştirmek mümkün olmamaktadır. Bölüm 3'te daha önce bahsedilen konumlandırma uygulaması çalıştırılmış ve robotun kendi konumu hakkında küçük bir hata ile konumunu kestirmesi sağlanmıştır. Turtlebot platformunun konumlandırma örneği yardımıyla elde edilen konum dağılımı Şekil 6.1'de görüldüğü gibi olmaktadır.



Şekil 6.1 : Harita üzerinde konumlandırma yönteminin gerçekleştirilmesi.

Haritanın ve konumun küçük bir hata payı ile bilindiği durumda yörünge planlama yöntemi kullanılarak robotun takip etmesi istenen yörüngeler bulunabilmektedir. İlk olarak bu yöntemlerin çalıştırılması için harita içerisindeki engellerin robot yarıçapı boyutunda genişletilmesi ve harita üzerinde hedef noktasının verilmesi gerekmektedir. Yörünge planlama yöntemi için robotun orta noktasının referans alınmasından dolayı Şekil 6.2'de görüldüğü üzere ortam haritasının robot yarıçapı boyutunda genişletilmesi sağlanmıştır. Ayrıca ROS içerisinde bulunan Rviz görselleştirme aracı içerisinde bulunan yörünge planlama işleri için tasarlanmış hedef verme aracının kullanılarak yörünge planlama algoritmasına hedef bilgisi bildirilmiştir.

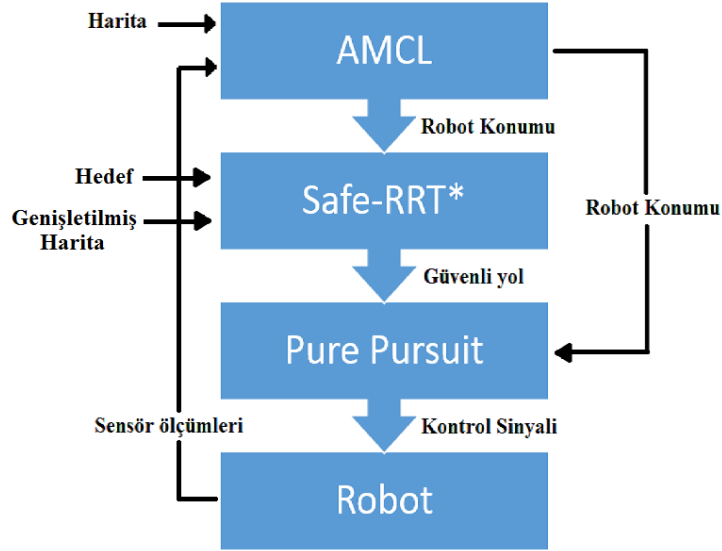


Şekil 6.2 : Ortam haritasının robot yarıçapı oranında genişletilmesi.

6.1 Değişmez Çevrelerde Mobil Robot Navigasyonu

Değişmez çevrelerde mobil robot navigasyonu problemi çevresindeki bütün engellerin koordinatlarını daha önceden bildiğinden engelden kaçınma yöntemlerine ihtiyaç duymamaktadır. Bu tür problemler için tasarlanacak navigasyon sistemi ortamın haritasını, robot konumunu ve hedef noktasını kullanarak robotun hedef noktasına ulaşmasını sağlayabilir. Konumlandırma ve yörünge takibinin eklenmesi ile oluşturulan navigasyon sistemi görev şeması Şekil 6.3’de görüldüğü gibidir.

Değişmez ortamda mobil robot navigasyonu, konumlandırma yöntemi (AMCL) vasıtasıyla robot konumunun kestirilmesi ile başlar. Hedef konumu, robot yarıçapında genişletilmiş harita ve robot konumu Safe-RRT* yönteminde uygun yolun bulunması için kullanılır. Başlangıç noktasından hedef noktasına ulaşan güvenli yol Pure Pursuit takip yöntemine gönderilir. Pure Pursuit yöntemi ise Bölüm 5’de bahsedildiği üzere kendisine gelen yol üzerinde robot konumunu kullanarak kendine hedef noktaları belirlemektedir. Bu hedef nokta yörünge üzerinde robota en yakın noktanın belirli bir mesafe ilerisinde seçilir. Daha sonra robot hareket modeli yardımıyla bu noktaya ulaşabileceği açısız, çizgisel hız çiftini bulur ve robota uygular. Bu sayede yol planlama sonucu elde edilen yörüngeyi takibi gerçekleştirilmiş olur. Geliştirilen



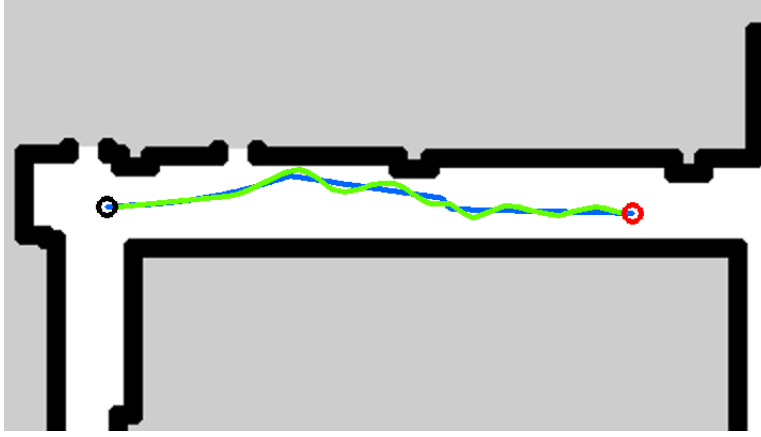
Şekil 6.3 : Değişmez ortamda mobil robot navigasyon sistemi.

navigasyon sistemi için harita üzerinde iki farklı başlangıç ve hedef noktası seçilmiş, yol planlayıcı sonucu ve konumlandırma yardımıyla elde edilen robotun takip ettiği yörüngeler Şekil 6.4’de gösterilmiştir.

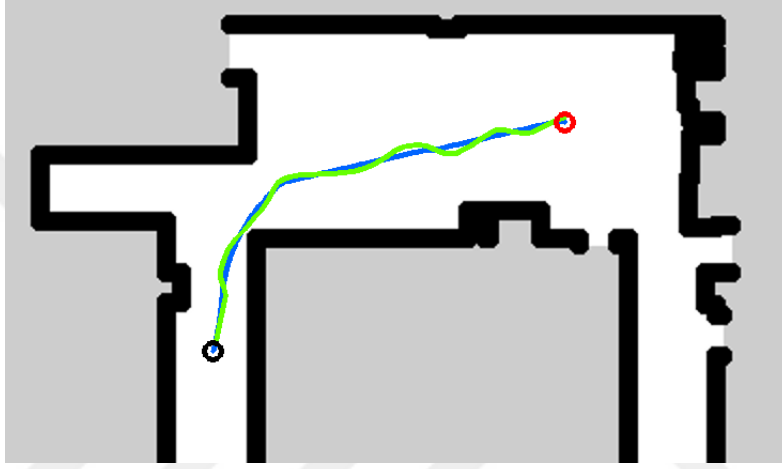
6.2 Değişken Çevrelerde Mobil Robot Navigasyonu

Değişken çevrelerde ise durum farklı bir açıdan ele alınmaktadır. Böyle durumlarda yol planlayıcısı ile elde edilen yol üzerinde engeller bulunabileceğinden dolayı bu yolların dikkatsiz bir şekilde takip edilmesi robotun engellere çarpması gibi sorunlar oluşturmaktadır. Ayrıca yol planlama yöntemi rastgele genişleme ilkesine dayandığından düşük iterasyon sayısında engellere yakın yollar bulabilmektedir. Bu amaçla Şekil 6.5’de görülen değişken çevre navigasyon sistemi geliştirilmiştir.

Bu sistem de konumlandırma yöntemi yardımıyla konum kestirimine gereksinim duyar. Konum bilgisi, hedef bilgisi ve robot yarıçapı oranında genişletilmiş harita kullanılarak Safe-RRT* yöntemi ile başlangıç noktası ile hedef bölgesini birleştiren yol bulunur. Bulunan bu yol üzerinde Pure Pursuit benzeri bir yaklaşım gösterilmiştir. Robotun algıladığı çevre dışında Safe-RRT* tarafından bulunan yol üzerinde noktalar engelden kaçınma yöntemine hedef olarak verilir. Bu noktalar engelden kaçınma yönteminin hedefe ulaşmasında bir ipucu görevi yapmaktadır. İpucu olarak seçilen noktalar FGM-DW yöntemine hedef noktası olarak verilir. Önerdiğimiz bu yeni engelden kaçınma algoritması direkt olarak bu noktaya yönelmek yerine çevrede

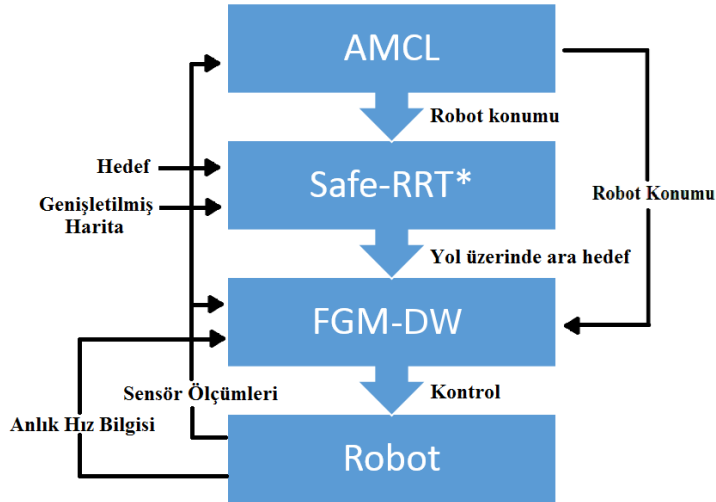


(a) Değişmez çevrede gerçek ortam sonucu (durum-1).



(b) Değişmez çevrede gerçek ortam sonucu (durum-2).

Şekil 6.4 : İki farklı durum için robotun başlangıç (siyah), hedef (kırmızı) noktaları için hesaplanan Safe-RRT* yolu (mavi), Pure Pursuit kullanılarak robotun gittiği gerçek yörünge (yeşil) ile gösterilmiştir.



Şekil 6.5 : Değişken ortamda mobil robot navigasyon sistemi.

gördüğü maksimum boşluk merkezlerini de hesaba katarak robotun yönelmesi gereken doğrultuyu hesaplar. FGM-DW daha önce bahsettiğimiz ikinci ve üçüncü aşaması

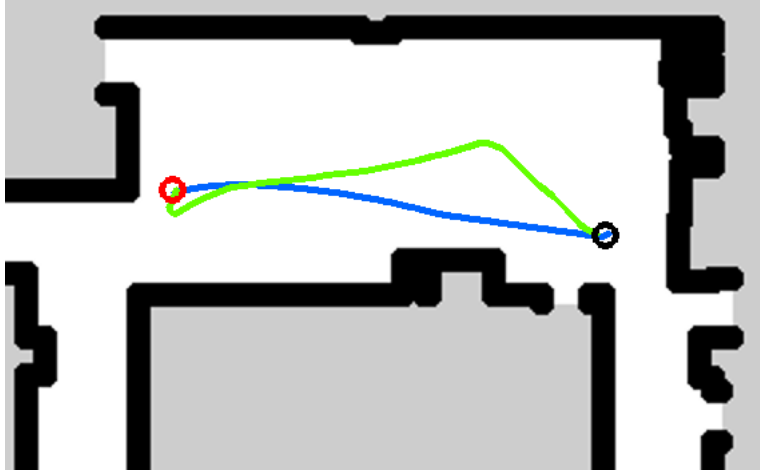
ile bu yöne en hızlı ve güvenli şekilde seyahat etmesi için gerekli kontrol sinyalinin hesaplayıp robota uygulamaktadır.

Bu sayede yol planlama yöntemi ile bulunan yollar dolambaçlı veya riskli olsa da FGM-DW yöntemi ile kısa ve güvenli bir şekilde hedefe ulaşılması sağlanabilmektedir. Yol planlama yöntemiyle bulunan yoldan daha kısa yollardan gitmesinin sebebi yol üzerinde daha ileri noktaları hedef olarak almasından kaynaklanmaktadır. Bu yorum daha önce hesapta olmayan bir engel ile karşılaştığında yolunu uzatma durumu hariç tutulduğunda doğrudur.

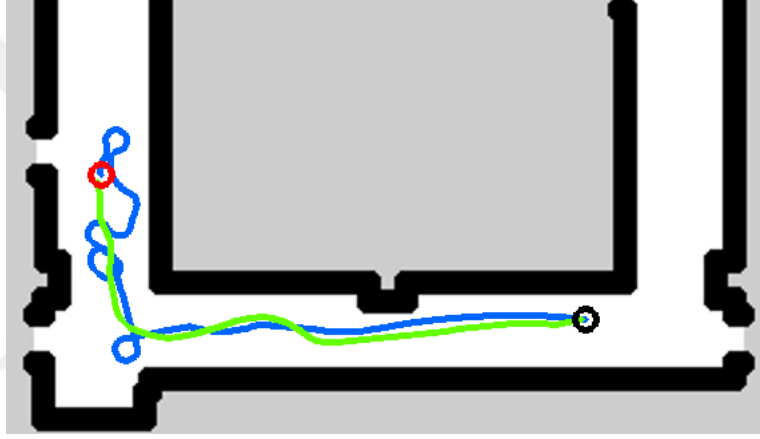
Geliştirilen değişken ortamda navigasyon sistemi için harita üzerinde beş farklı başlangıç ve hedef noktası seçilmiş, yol planlayıcı sonucu ve değişken çevre navigasyonu ile elde edilen robotun takip ettiği yörüngeler Şekil 6.6 ve 6.7'de gösterilmiştir. Robotun dinamik çevre davranışını gözlemlemek amacıyla hedefinden saptırıcı engeller karşısına çıkarılmış ve değişken çevredeki davranışı incelenmiştir. Karışıklık olmaması amacıyla hareketli engeller şekillerde gösterilmemiştir.

Elde edilen sonuçlara göre robotun Safe-RRT* sonucu elde ettiği yollar engel köşelerine yakın geçse dahi geliştirilen FGM-DW yöntemi ile kısa ve daha güvenli yollarda ulaşılabilirdiği görülmektedir. Robotun güvenli bölgelerden geçişi FGM'nin hedef ve boşluk merkezi arasında bölgeleri tercih etmesinden kaynaklanmaktadır. Bu navigasyon sistemi Safe-RRT* sonucu elde ettiği yolun robota en yakın noktasına belirli bir mesafe uzağında (çalışmalarda 3m alınmıştır) hedef nokta seçmesinden kaynaklanmıştır.

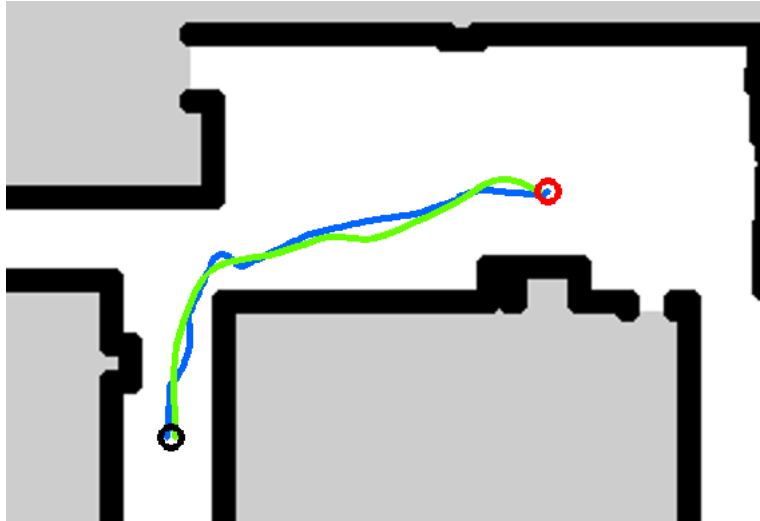
Şekil 6.7'de gösterilen dördüncü durumda robot hareketli engeller ile karşılaşmamış bu yüzden yol takip yöntemi gibi fakat ona göre daha güvenli ve az sınımlı yollardan hedefine ulaşmıştır. Birinci, üçüncü ve beşinci durumlarda robotun değişken çevrelerdeki hareketini gözlemleyebilmek amacıyla karşısına hareket ettiği doğrultu üzerinde engeller çıkartılmış ve bu engellere tepkisi büyük sınımlar şeklinde gözlemlenmiştir. İkinci durumda ise Safe-RRT* yöntemi ile dolambaçlı bir yol elde edilmesine rağmen FGM-DW yöntemi kendisine Pure Pursuit yöntemine göre nispeten uzak hedefler seçtiğinden yolun dolambaçlı kısımlarından etkilenmemiş kısa bir şekilde hedefine ulaşmıştır.



(a) Değişken çevrede gerçek ortam sonucu (durum-1).

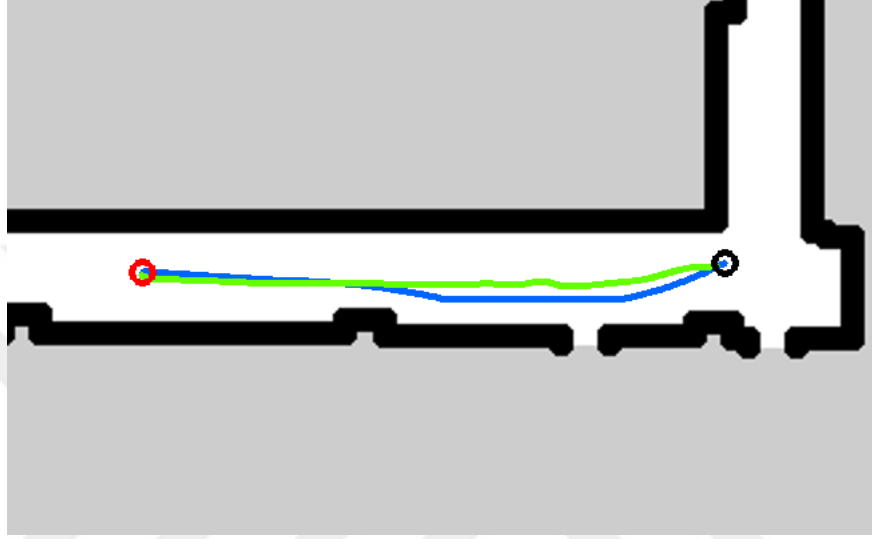


(b) Değişken çevrede gerçek ortam sonucu (durum-2).

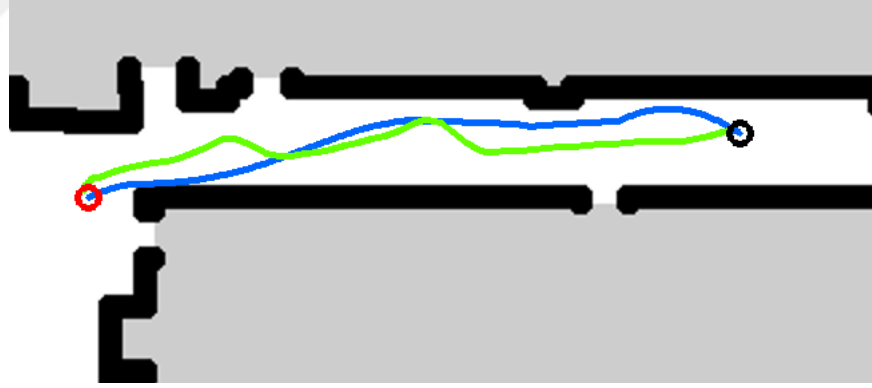


(c) Değişken çevrede gerçek ortam sonucu (durum-3).

Şekil 6.6 : Üç farklı durum için robotun başlangıç (siyah), hedef (kırmızı) noktaları için hesaplanan Safe-RRT* yolu (mavi), robotun FGM-DW kullanılarak gittiği gerçek yörünge (yeşil) ile gösterilmiştir.



(a) Değişken çevrede gerçek ortam sonucu (durum-4).



(b) Değişken çevrede gerçek ortam sonucu (durum-5).

Şekil 6.7 : İki farklı durum için robotun başlangıç (siyah), hedef (kırmızı) noktaları için hesaplanan Safe-RRT* yolu (mavi), robotun FGM-DW kullanılarak gittiği gerçek yörünge (yeşil) ile gösterilmiştir.



7. SONUÇ VE ÖNERİLER

Değişken ve değişmez çevrelerde navigasyon problemi, içerisinde birçok problemi barındırmaktadır. Bunlardan ilki konumlandırma probleminde ortaya çıkabilecek sorunlardır. Robot, konumunu doğru bir şekilde kestirebilmesi için algılayıcı doğruluğuna ve farklı ölçümlerin (enkoder, imu, derinlik tarama gibi) birleştirilmesine ihtiyaç duymaktadır. Algılayıcılardaki hatalar ve tekerlekteki kaymalar robotun istenilen hareketi yapmasına engel olabilmektedir. Bu yüzden robotların engellere yakın bölgelerden geçmesi istenmeyen durumlardır.

Güvenlik kaygısı ile değerlendirme yapan ve hareketi kısıtlı robotlar için tasarlanmış RRT* ve türevi yöntemler ise rasgelelik ilkesine göre genişlediğinden ancak yüksek iterasyon sayılarına ulaştıklarında kendilerinden beklenen yolları üretebilmektedir. Ancak iterasyon sayısındaki artış komşu bulma özelliklerinden dolayı hesaplama maliyetini üstel şekilde artırmaktadır. Bu yöntemleri düşük düğüm sayılarında ise uzun ve daha dolambaçlı yollar elde etmektedir. Yol takip yöntemlerinin de sınımlı hareketi eklendiğinde ortam değişmez olsa dahi bu yolların böyle durumlarda takip edilmesi zorlaşmaktadır.

Bu sebeple yol planlayıcı yöntemlerin yol gösterici olduğu, arka planda engelleri algılayan ve robotu engellerden kaçınmaya yönlendiren engelden kaçınma yöntemlerine ihtiyaç duyulur. Önerilen FGM-DW yönteminin robot dinamiklerini hesaba katması, en hızlı ve güvenli şekilde hedefe ulaşmaya çalışması özellikleri ile bir engelden kaçınma yönteminin istenilen bütün kriterlerine sahip olduğu söylenebilir. Alınan sonuçlar ve kendisini ispatlamış DWA ve FGM yöntemleri göz önüne alındığında FGM-DW yönteminin yönelim tabanlı ve hız uzayı yaklaşımları özelliklerinin birleştirilmesi ile oluşturulmuş başarılı bir engelden kaçınma yöntemi olduğunu söylenebilir.

Eğer navigasyon problemi değişmez çevrelerde çözülecekse yol planlama yöntemi ile engellerden uzak geçen yolların bulunması tercih edilir. Bu yollar, Safe-RRT* yöntemi ile yüksek düğüm sayılarında elde edilebilir. Fakat bu durum yöntemin

çalıřma süresini arttırmaktadır. Bu yüzden daha önceden RRT* üzerine kurulmuş hızlandırılma yöntemleri [23-27] ve Safe-RRT* yönteminin birleřtirilmesiyle daha kısa sürede istenilen güvenli yollar elde edilebilir.

Ayrıca FGM-DW yönteminde kullanılan DWA özellikleri, diđer yönelim tabanlı engelden kaçınma yöntemleri ile de kullanılabilir. Bu yöntemlere örnek olarak [33,37] verilebilir. Elde edilen yönelimlerin, DWA özellikleri ile güvenli bir şekilde takip edilebileceđi FGM-DW yöntemi ile kanıtlanmıştır.



KAYNAKLAR

- [1] **Url-1**, <http://www.gaitech.hk/>, alındığı tarih: 10.04.2015
- [2] **Url-2**, <http://www.generationrobots.com/>, alındığı tarih: 10.04.2015
- [3] **Url-3**, <http://www.ros.org/about-ros>, alındığı tarih: 15.12.2015
- [4] **Url-4**, <http://www.ros.org>, alındığı tarih: 15.12.2015
- [5] **D. Fox, W. Burgard ve S. Thrun** (2005). Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, (7).
- [6] **F. Dellaert, D. Fox, W. Burgard ve S. Thrun** (1999). Monte Carlo localization for mobile robots, Robotics and Automation, Proceedings 1999 IEEE International Conference on, Detroit, MI, 1999, pp. 1322-1328 vol.2.
- [7] **P. Maybeck** (1979). Stochastic Models, Estimation and Control, vol.1, Academic Press, New York.
- [8] **L. P. Kaelbling, A. R. Cassandra, ve J. A. Kurien** (1979). Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation, in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [9] **S. Thrun** (1998). Bayesian landmark learning for mobile robot localization, Machine Learning Journal, Volume 33, Issue 1, pp 41-76
- [10] **Burgard, D. Fox, D. Hennig, ve T. Schmidt** (1996). Estimating the absolute position of a mobile robot using position probability grids, in Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96), pp. 896–901
- [11] **Burgard, A. Derr, D. Fox, ve A. B. Cremers** (1998). Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [12] **Dieter Fox** (2001). KLD–Sampling: Adaptive Particle Filters, In Advances in Neural Information Processing Systems 14
- [13] **Lozano-Perez, T.**(1983). Spatial planning: A configuration space approach. Computers, IEEE Transactions on, 100(2), 108-120.
- [14] **Khatib, O.** (1986). Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, 5(1), 90-98.

- [15] **Lozano-Pérez, T. ve Wesley, M. A.**(1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10), 560-570.
- [16] **Lumelsky, V. J. ve Stepanov, A. A.** (1987). Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2(1-4), 403-430.
- [17] **Hart, P. E., Nilsson, N. J. ve Raphael, B.** (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2), 100-107.
- [18] **Kavraki, L. E., Švestka, P., Latombe, J. C. ve Overmars, M. H.** (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4), 566-580.
- [19] **Kuffner, J. J. ve LaValle, S. M.** (2000). RRT-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 2, pp. 995-1001). IEEE.
- [20] **LaValle, S. M. ve Kuffner Jr, J. J.** (2000). Rapidly-exploring random trees: Progress and prospects.
- [21] **Karaman, S. ve Frazzoli, E.** (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846-894.
- [22] **Karaman, S., Walter, M. R., Perez, A., Frazzoli, E. ve Teller, S.** (2011, Mayıs). Anytime motion planning using the RRT*. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 1478-1483). IEEE.
- [23] **Qureshi, A. H., Iqbal, K. F., Qamar, S. M., Islam, F., Ayaz, Y. ve Muhammad, N.** (2013, Ağustos). Potential guided directional-RRT* for accelerated motion planning in cluttered environments. In *Mechatronics and Automation (ICMA), 2013 IEEE International Conference on* (pp. 519-524). IEEE.
- [24] **Qureshi, A. H., Mumtaz, S., Iqbal, K. F., Ayaz, Y., Muhammad, M. S., Hasan, O. ve Ra, M.** (2014, Mart). Triangular geometry based optimal motion planning using RRT*-motion planner. In *Advanced Motion Control (AMC), 2014 IEEE 13th International Workshop on* (pp. 380-385). IEEE.
- [25] **Kim, D., Lee, J. ve Yoon, S. E.** (2014, Mayıs). Cloud RRT*: Sampling Cloud based RRT*. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (pp. 2519-2526). IEEE.
- [26] **Islam, F., Nasir, J., Malik, U., Ayaz, Y. ve Hasan, O.** (2012, Ağustos). RRT*-smart: Rapid convergence implementation of RRT* towards optimal solution. In *Mechatronics and Automation (ICMA), 2012 International Conference on* (pp. 1651-1656). IEEE.

- [27] **Gammell, J. D., Srinivasa, S. S. ve Barfoot, T. D.** (2014). Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. arXiv preprint arXiv:1404.2334.
- [28] **Luders, B. D., Karaman, S. ve How, J. P.** (2013, Ağustos). Robust sampling-based motion planning with asymptotic optimality guarantees. In AIAA Guidance, Navigation, and Control Conference (GNC), Boston, MA.
- [29] **Shan, T. ve Englot, B.** (2015, Aralık). Sampling-based Minimum Risk path planning in multiobjective configuration spaces. In 2015 54th IEEE Conference on Decision and Control (CDC) (pp. 814-821). IEEE.
- [30] **Siegwart, R., Nourbakhsh, I. R. ve Scaramuzza, D.** (2011). Introduction to autonomous mobile robots. MIT press, (Vol. 2, pp. 271).
- [31] **Coulter, R. C.** (1992). Implementation of the pure pursuit path tracking algorithm (No. CMU-RI-TR-92-01). CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST.
- [32] **Url-5**, <http://www.mathworks.com>, alındığı tarih: 10.04.2016
- [33] **Khatib, O.** (1986). Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, 5(1), 90-98.
- [34] **Borenstein, J. ve Koren, Y.** (1991). The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation, 7(3), 278-288.
- [35] **Ulrich, I. ve Borenstein, J.** (1998, Mayıs). VFH+: Reliable obstacle avoidance for fast mobile robots. In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on (Vol. 2, pp. 1572-1577). IEEE.
- [36] **Minguez, J. ve Montano, L.** (2000). Nearness diagram navigation (nd): A new real time collision avoidance approach. In Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on (Vol. 3, pp. 2094-2100). IEEE.
- [37] **Minguez, J.** (2005, Ağustos). The obstacle-restriction method for robot obstacle avoidance in difficult environments. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2284-2290). IEEE.
- [38] **Sezer, V. ve Gokasan, M.** (2012). A novel obstacle avoidance algorithm: "Follow the Gap Method". Robotics and Autonomous Systems, 60(9), 1123-1134.
- [39] **Simmons, R.** (1996, Nisan). The curvature-velocity method for local obstacle avoidance. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on (Vol. 4, pp. 3375-3382). IEEE.
- [40] **Gal, O., Shiller, Z. ve Rimon, E.** (2009, Mayıs). Efficient and safe on-line motion planning in dynamic environments. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on (pp. 88-93). IEEE.

- [41] **Thrun, D. F. W. B. S., Fox, D. ve Burgard, W.** (1997). The dynamic window approach to collision avoidance. *IEEE Transactions on Robotics and Automation*, 4, 1.
- [42] **Brock, O. ve Khatib, O.** (1999). High-speed navigation using the global dynamic window approach. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* (Vol. 1, pp. 341-346). IEEE.
- [43] **Sezer, V., Ercan, Z., Heceoglu, H., Bogosyan, S. ve Gokasan, M.** (2012, Temmuz). A new fuzzy speed planning method for safe navigation. In *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on* (pp. 381-386). IEEE.
- [44] **Sezer, V.** (2015). Combined fuzzy approach for online speed planning and control with real vehicle implementation. *International Journal of Vehicle Design*, 68(4), 329-345.



ÖZGEÇMİŞ

Ad-Soyad : Aykut ÖZDEMİR
Doğum Tarihi ve Yeri : ÇORUM / 1990
E-Posta : ozdemirayk@itu.edu.tr



ÖĞRENİM DURUMU:

- **Lisans:** Kocaeli Üniversitesi Mekatronik Mühendisliği
- **Y. Lisans:** İstanbul Teknik Üniversitesi Mekatronik Mühendisliği (halen)

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- Sezer V., **Ozdemir A.**, 2016: A Hybrid Obstacle Avoidance Method: Follow the Gap with Dynamic Window Approach (*incelenme aşamasında*),